

# BookList: Sistema Web para Organização e Controle de Leituras

Victor H. Muniz<sup>1</sup>, Jayme Rebula Klein P. Silva<sup>1</sup>,  
Leonardo Bravo Estácio<sup>1</sup>

<sup>1</sup>Instituto Federal de Santa Catarina (IFSC)  
R. Heitor Villa Lobos, 225 - São Francisco, Lages - SC, Brazil

**Abstract.** *In a world where reading plays a crucial role in personal and professional development, expanding knowledge, creativity and critical sense, it is observed that many individuals struggle to maintain an effective reading routine, caused by difficulties such as lack of time, motivation or organization, together with the vast availability of works in different formats, which highlight a significant gap in monitoring and managing progress and reading history. This article proposes the development of an application to organize and control the reading of literary works, meeting the need for a tool that helps readers manage their reading in an efficient and enjoyable way, where the methodological approach included exploratory research to determine the essential requirements for developing an application, which will be implemented with a user-friendly interface and a management system for past and future readings. The results indicate that the developed application has the potential to mitigate the challenges faced by readers, facilitating the management of their reading through a structured and interactive system.*

**Resumo.** *Em um mundo onde a leitura desempenha um papel crucial no desenvolvimento pessoal e profissional, ampliando conhecimento, criatividade e senso crítico, observa-se que muitos indivíduos lutam para manter uma rotina de leitura eficaz, devido à dificuldades como a falta de tempo, motivação ou organização, juntamente com a vasta disponibilidade de obras em diversos formatos, que evidenciam um gap significativo no acompanhamento e gestão do progresso e histórico de leituras, logo, propõe-se o desenvolvimento de um aplicativo para organizar e controlar a leitura de obras literárias, atendendo à necessidade de uma ferramenta que auxilie os leitores a gerenciar suas leituras de forma eficiente e prazerosa, onde a metodologia adotada envolveu um estudo exploratório para identificar os requisitos fundamentais para criar o sistema, que foi desenvolvido com uma interface intuitiva e recursos que facilitam a interação do usuário, além de ser de fácil compreensão para o manuseio. Os resultados indicam que o aplicativo desenvolvido possui o potencial de mitigar os desafios enfrentados pelos leitores, facilitando o gerenciamento de suas leituras através de um sistema estruturado e interativo.*

**Keywords** - Organization, Reading, Gamification, Application development

## 1. Introdução

A leitura é uma atividade essencial para o desenvolvimento pessoal e profissional, no entanto, muitas vezes, pode ser difícil manter o controle e a organização das leituras realizadas. Segundo Locke and

Latham (2002), Zimmerman et al. (1992) e SCHOEP and WOOD (2015), quando os leitores estabelecem metas para si mesmos, eles tendem a mudar positivamente seu comportamento e, quando atingem suas metas de leitura, sua confiança e sensação de realização aumentam.

Durante séculos o ensino da leitura tem sido um dos objetivos fundamentais do ensino básico e, uma vez que uma boa parte das aprendizagens posteriores têm como veículos os textos escritos, este ensino tem sido alargado para níveis mais avançados de escolarização da Cruz (2005). Segundo Shen et al. (2006), 83,9% dos alunos leem informações on-line frequentemente todos os dias e 69,3% leem e-mails todos os dias. Em contraste, apenas 31,4% leem jornais, e 33,1% deles leem revistas frequentemente todos os dias (Shen et al., 2006).

A falta de motivação para a leitura tem sido apontada como a causa de vários contratempos, pois muitos que têm dificuldades para ler evitam situações em que seja preciso fazê-lo. Ademais, como esses sujeitos não conseguem desenvolver a prática da leitura, ocorre a diminuição da motivação e a instalação lenta e gradual da relutância para realizar essa atividade. (Tourinho, 2011)

A facilidade e a praticidade que a tecnologia oferece para as pessoas é algo que não pode ser ignorado. Com o advento de sistemas web e aplicativos de celulares, as pessoas agora têm acesso a uma variedade de serviços que antes eram impossíveis ou muito complicados de se obter. Através dessas tecnologias, é possível fazer compras, pedir comida, reservar hotéis, agendar compromissos e muito mais, tudo com apenas alguns toques na tela do celular ou no computador, segundo Rocha (2021), inovações tecnológicas vieram para nos trazer facilidades, praticidade e comodidades. Além disso, esses sistemas permitem uma maior interação e comunicação entre as pessoas, o que é fundamental para a construção de um mundo mais conectado e colaborativo.

Em um mundo cada vez mais digital, armazenar informações sobre atividades realizadas se tornou uma prática comum e eficaz. Um sistema de gerenciamento de leitura pode contribuir como uma ferramenta valiosa para garantir que as informações importantes sejam armazenadas e acessíveis de maneira organizada e segura. A capacidade de armazenar informações em um sistema permite que se tenha uma visão geral do que foi lido e a capacidade de revisar essas informações posteriormente de maneira fácil e rápida. Dessa forma, é possível organizar e categorizar as informações de forma personalizada, tornando mais fácil para o usuário encontrar e acessar as informações quando necessário. Em resumo, armazenar informações do que se está sendo lido pode ser uma estratégia eficaz para maximizar a aprendizagem, a organização e a produtividade em diversos contextos pessoais e profissionais.

A gamificação é uma técnica que consiste em utilizar elementos de competição em situações do cotidiano, com o objetivo de motivar e engajar as pessoas. Essa abordagem tem sido cada vez mais utilizada em diferentes áreas, incluindo a educação. Uma forma de aplicar a gamificação para manter o hábito da leitura é através de aplicativos e plataformas que transformam a leitura em uma atividade mais interativa e desafiadora. Por exemplo, é possível criar um sistema de recompensas por meio de pontos, que incentivam o usuário a ler mais livros ou ainda incluir elementos de competição, como classificação de leitores mais ativos, para estimular a participação. Dessa forma, a gamificação pode ser uma ferramenta poderosa para incentivar a leitura, tornando-a mais divertida e envolvente. O uso de tecnologia para melhorar a educação tem sido bastante explorado. Por exemplo, o uso de videojogos na aprendizagem tem despertado bastante a curiosidade da comunidade científica, devido à sua capacidade de motivar os jogadores e os fazer persistir. (Barata et al., 2013)

O objetivo deste trabalho é desenvolver um sistema web que forneça um controle sobre leituras planejadas, em andamento e já finalizadas, junto com suas respectivas classificações e opiniões, porém, focada na competição entre usuários da plataforma utilizando elementos de gamificação para gerar

competitividade e sociabilidade entre os usuários. Além disso, notificar usuários sobre o lançamento de continuações para leituras que já realizaram. Para atingir esse objetivo, serão definidos os seguintes objetivos específicos:

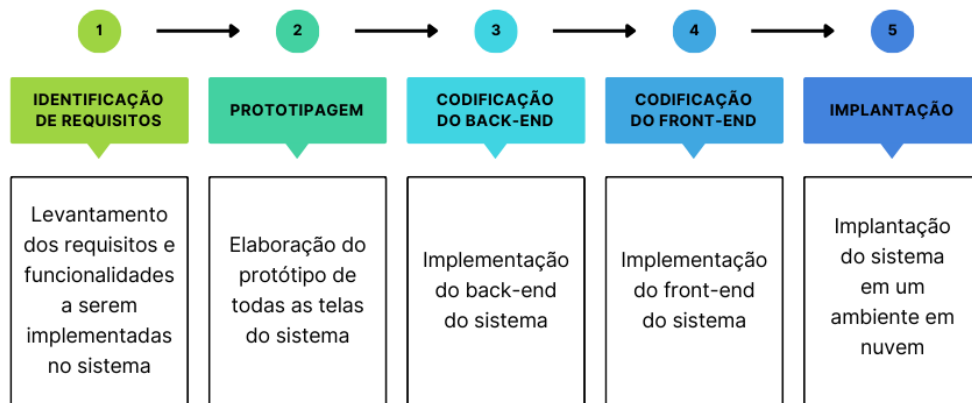
- Identificar os requisitos necessários para o desenvolvimento da aplicação;
- Compreender as técnicas de gamificação que serão utilizadas;
- Desenvolver um protótipo de interface para o usuário;
- Compreender a interação entre front-end e back-end
- Implementar o sistema, de maneira responsiva, para utilização via computador ou dispositivos móveis;
- Realizar a implantação do sistema em uma plataforma em nuvem.

A estrutura restante deste documento se compõe em seis seções organizadas da seguinte forma: A Seção 2 apresenta as definições e os elementos teóricos referenciais que são necessários para um melhor entendimento da proposta. A Seção 3, apresenta a abordagem proposta e detalha a arquitetura de sistema projetada. Além disso, apresenta os trabalhos correlatos, onde foram buscadas referências para o desenvolvimento do trabalho. A Seção 4 mostra os aspectos de implementação da aplicação proposta destacando como os componentes da arquitetura de sistema projetada foram implementados, além de discussões gerais sobre o desenvolvimento do sistema e imagens das páginas do sistema e de trechos de código. A seção 5 resume o trabalho desenvolvido para implantar o sistema em uma plataforma em nuvem. A seção 6 discorre sobre a possibilidade de trabalhos futuros. Por fim, a Seção 7 apresenta as considerações finais sobre o trabalho.

## 2. Metodologia

A metodologia adotada para o desenvolvimento da aplicação será baseada em um estudo exploratório, que inclui a revisão bibliográfica, pesquisa de mercado e testes com usuários. Com os requisitos definidos, realizamos a prototipagem da aplicação utilizando a plataforma *Figma* e fluxogramas via *Canva*, e em seguida a implementação do sistema utilizando os *frameworks .NET* e *Angular*. Com a versão inicial do sistema finalizada, realizamos a implantação do sistema em um ambiente em nuvem da plataforma *Fly.io*.

Figura 1. Metodologia



Fonte: Autor

A Figura 1 apresenta em ordem as etapas seguidas para o desenvolvimento e implementação do sistema, desde a identificação dos requisitos e funcionalidades, à implantação do sistema após a finalização da implementação.

### 3. Referencial Teórico

Esta seção está dividida em quatro partes. A primeira parte contextualiza a utilização de técnicas de gamificação como forma de auxiliar a leitura. A segunda parte aborda técnicas de gamificação e seus benefícios. A terceira parte discorre sobre trabalhos relacionados. A quarta parte explica as tecnologias usadas no desenvolvimento da aplicação.

#### 3.1. Gamificação

A gamificação tem sido amplamente utilizada como uma técnica para aumentar o engajamento dos usuários em diversas áreas, incluindo educação e entretenimento. No contexto da leitura, a gamificação pode ser aplicada através de estratégias que buscam aumentar o interesse e a motivação dos leitores, bem como melhorar a compreensão e retenção de informações.

O livro “Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps” Zichermann and Cunningham (2011) apresenta diversos exemplos concretos de como técnicas de

gamificação podem ser aplicadas na leitura. De acordo com os autores, a criação de desafios de leitura, a oferta de recompensas e o uso de feedback imediato são algumas das técnicas de gamificação que têm sido utilizadas para melhorar a experiência de leitura dos usuários.

Um exemplo de sucesso apresentado pelos autores é a plataforma de leitura “Goodreads”, que utiliza a gamificação para incentivar a participação e interação dos usuários. Ao oferecer pontos por atividades como avaliar livros, participar de grupos de discussão e completar desafios de leitura, a plataforma busca aumentar a motivação e o engajamento dos usuários. Além disso, a plataforma oferece medalhas e certificados para os usuários que completam metas de leitura específicas, como ler uma quantidade determinada de livros em um período de tempo específico.

Outra técnica de gamificação utilizada na leitura é a progressão de níveis. Ao oferecer níveis de leitura que indicam o nível de habilidade ou conhecimento do leitor, as plataformas podem tornar a experiência de leitura mais gratificante e motivadora. O uso de enquetes de imediato também pode ser aplicado na leitura, através da disponibilização de resumos e questionários para verificar a compreensão dos leitores.

Os autores destacam que a gamificação tem sido um fator chave para o sucesso da plataforma “Goodreads”, que, durante a elaboração deste artigo, possui mais de 55 milhões de membros ativos em todo o mundo. Através da gamificação, a plataforma tem sido capaz de incentivar a leitura frequente e melhorar a compreensão e retenção de informações dos usuários.

As técnicas de gamificação apresentadas por Zichermann e Cunningham (2011) têm mostrado resultados significativos na promoção da leitura e do engajamento dos usuários e podem ser utilizadas como uma ferramenta importante para melhorar a experiência de leitura dos usuários.

O sistema incorpora a entrega de insígnias como uma técnica de gamificação para incentivar a finalização de leituras. Essas insígnias, conferidas aos usuários após a conclusão de uma obra, funcionam como um reconhecimento visual e tangível do seu progresso e dedicação. Esse sistema de recompensas cria um sentimento de realização e pertencimento, estimulando não apenas a continuidade da leitura, mas também a participação ativa na comunidade da plataforma. Ao receberem essas insígnias, os usuários são motivados a definir e atingir metas pessoais de leitura, transformando o ato isolado de ler em uma experiência compartilhada e socialmente enriquecedora. Este elemento de gamificação é um poderoso meio de transformar o hábito da leitura em uma jornada lúdica e gratificante.

### **3.2. Artigos Relacionados**

Como dito por Thelwall and Kousha (2017) em “Goodreads: Uma Rede Social para Leitores”, este artigo apresenta uma análise descritiva do *Goodreads* enquanto uma rede social para leitores de livros. O autor enfatiza as principais funcionalidades do Goodreads, como a descoberta de livros, avaliações e revisões, bem como a interação entre os usuários. O artigo também discute a utilidade do Goodreads como uma ferramenta de marketing para autores e editores. Em geral, o artigo demonstra que o Goodreads é uma plataforma valiosa para os amantes de livros, fornecendo recursos para ajudá-los a descobrir novas leituras e se conectar com outros leitores.

Em Burlamaque and Barth (2015), é realizada uma análise sobre as funcionalidades do Skoob, uma rede social dedicada a leitores, e seus impactos na educação. Segundo o artigo, os usuários podem compartilhar suas experiências de leitura, trocar impressões sobre livros, divulgar resenhas e atribuir notas às obras lidas. Com sua meta de leitura, o site incentiva os usuários a aumentarem seu envolvimento com os livros. Cada usuário tem uma estante virtual em seu perfil, onde pode catalogar e classificar as obras lidas ou desejadas. O Skoob se destaca pela interação e o compartilhamento de informações sobre livros e leitura.

Além das duas plataformas citadas, outras plataformas semelhantes são o Cabeceira, Bookly e Beanstack. O Cabeceira é uma plataforma brasileira com uma proposta semelhante ao Goodreads e Skoob, com um design mais moderno. O Bookly, dentre todas as plataformas pesquisadas, é o mais completo em termos de gamificação, porém, apenas na versão *premium* da plataforma, que exige uma assinatura mensal. Por fim, o Beanstack tem como público-alvo instituições escolares, com implementações voltadas para salas de aula.

No artigo intitulado “*Motivating students to read extensively*” de SCHOEP and WOOD (2015), é conduzida uma pesquisa abrangente sobre os benefícios da leitura extensiva. O estudo é realizado em um ambiente de sala de aula e destaca uma lacuna preocupante em termos de engajamento, participação e disposição dos alunos em realizar atividades de leitura fora do contexto escolar, apesar do interesse evidente em aprimorar suas habilidades de leitura. A pesquisa estabelece e comprova os benefícios da leitura extensiva no desenvolvimento da segunda língua dos alunos. Esses benefícios incluem melhorias na compreensão e fluência de leitura, desenvolvimento de vocabulário e ortografia, impactos positivos na motivação e, conseqüentemente, melhorias nas habilidades de escrita, fala e audição.

O artigo “*Exploring the role of gamification in motivating students to learn*” de Baah et al. (2023) analisa o papel da gamificação na motivação dos estudantes em sala de aula, a gamificação é vista como uma abordagem educacional emergente que envolve o uso de elementos de jogos para incentivar e motivar os alunos a aprenderem. Os autores discutem como a gamificação pode melhorar o engajamento do aluno, o desempenho acadêmico e a retenção de informações. Além disso, o artigo destaca a importância do design de jogos eficaz na gamificação e como isso pode impactar a motivação do aluno. Os resultados do estudo sugerem que a gamificação pode ser uma estratégia promissora para motivar os alunos a aprender, mas que o design adequado do jogo é fundamental para o sucesso dessa abordagem.

### 3.3. Comparação com plataformas similares

O Quadro 1 apresenta outras plataformas com funcionalidades parecidas com as desenvolvidas no atual sistema abordado neste artigo, diferenciando-as por funcionalidades que são implementadas, ou que são pagas.

Quadro 1. Comparação entre plataformas semelhantes.

	Estante virtual	Progresso pessoal	Conexões entre usuários	Gamificação
Skoob	Possui	Possui	Possui	Limitado
Goodreads	Possui	Limitado	Limitado	Não possui
Cabeceira	Possui	Possui	Não possui	Possui
Bookly	Possui	Possui	Não possui	Possui
Beanstack	Possui	Possui	Possui	Possui
Este projeto	Possui	Possui	Possui	Possui

### 3.4. Tecnologias Utilizadas

O C# é uma linguagem de programação moderna, orientada a objetos e fortemente tipada, desenvolvida pela Microsoft como parte de sua plataforma *.NET*, que oferece um conjunto de ferramentas e bibliotecas, que será utilizada para elaboração do back-end da aplicação.

Para o desenvolvimento do front-end será utilizado Angular e TypeScript, *framework* e linguagem que trabalham em conjunto entregando *SPAs (Single-Page Applications)* responsivos e escaláveis.

Além disso, para facilitar a implementação dos elementos de responsividade e fluidez, serão utilizados alguns componentes do Bootstrap, um *framework* CSS.

Para o armazenamento de dados da aplicação, o banco de dados escolhido foi o MongoDB, banco não-relacional que armazena os dados em formato JSON.

A documentação da API foi gerada utilizando Swagger, ferramenta que traz a lista de recursos disponíveis na API REST, como *endpoints*, classes, exemplos de chamadas, entre outros.

A prototipagem e organização de fluxogramas foi realizada, respectivamente, no Figma, uma ferramenta de design gráfico colaborativa, e no Canva, uma ferramenta online para edição de imagens.

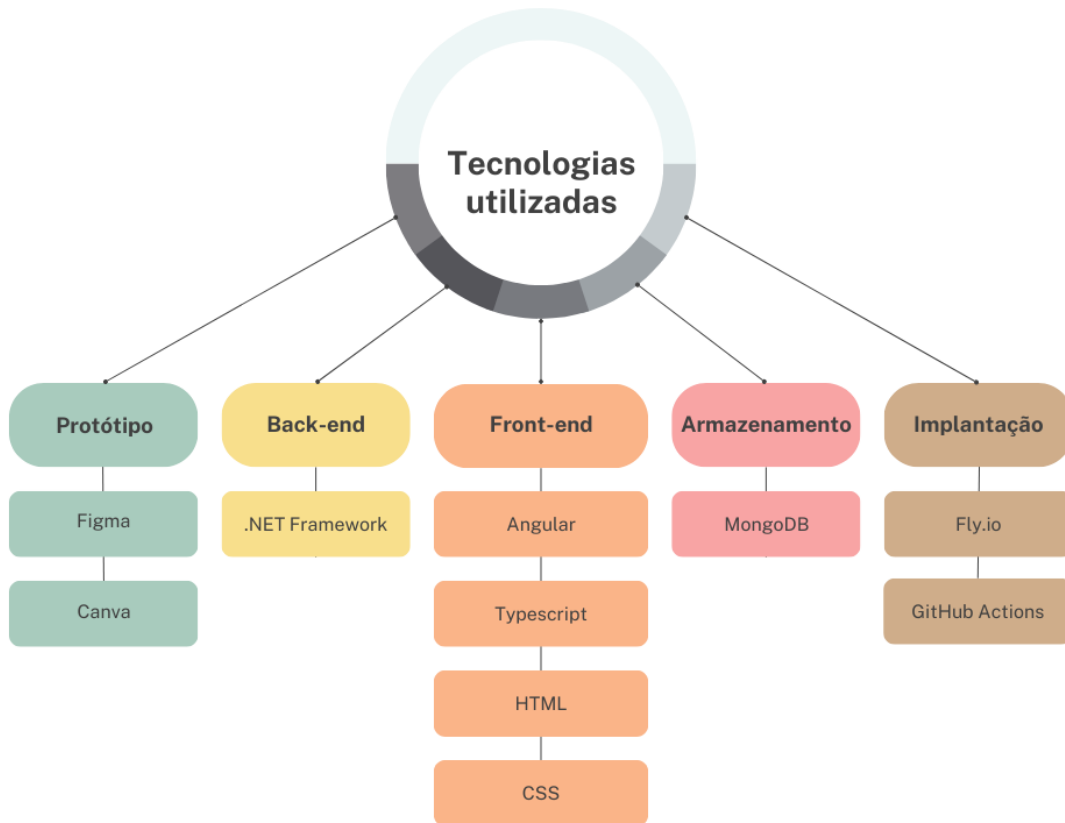
Para agilizar o processo de desenvolvimento do front-end, optamos por utilizar alguns elementos do Bootstrap, uma ferramenta que fornece um conjunto de estruturas CSS prontas para estilização de elementos. Com o Bootstrap, é possível criar uma interface limpa e consistente sem a necessidade de escrever CSS personalizado.

A implantação do sistema foi realizada pela plataforma de computação em nuvem Fly.io, que se destaca pela simplicidade de configuração quando comparado à outras alternativas. O armazenamento dos dados do sistema é realizado no MongoDB Atlas. Além disso, foi configurado um *workflow* no GitHub para implementar as mudanças no sistema toda vez que seja aprovado um *pull request* na ramificação principal do projeto do Git.

Documentações detalhadas de cada uma das tecnologias utilizadas pode ser encontrada nos *links* abaixo.

- C#
- Angular e Typescript
- MongoDB
- Swagger
- Bootstrap
- Fly.io e Github Actions

Figura 2. Tecnologias utilizadas



Fonte: Autor

A Figura 1 apresenta as tecnologias utilizadas na prototipagem das telas e funcionalidades do sistema, no back-end e front-end da aplicação, no armazenamento dos dados do sistema e na implantação.

## 4. Desenvolvimento

Nesta seção serão abordados os principais aspectos relacionados ao desenvolvimento do sistema. Será apresentado o diagrama de casos de uso, destacando entidades e funcionalidades relacionadas. Além disso, serão discutidos os requisitos funcionais e não funcionais do sistema. Por fim, o protótipo de cada uma das telas, elaborados para *desktop* e dispositivos móveis. O objetivo é fornecer um panorama abrangente do desenvolvimento do sistema, demonstrando a abordagem adotada e as funcionalidades implementadas.

### 4.1. Diagrama de Casos de Uso

A Figura 3 descreve as interações dos diferentes usuários com o sistema, assim como as funcionalidades que cada um pode exercer.



Figura 3. Diagrama de casos de uso



Fonte: Autor

Alguns usuários, os moderadores, terão a possibilidade de realizar alterações no sistema, como a adição de novas obras ou modificar informações em obras já cadastradas.

## 4.2. Levantamento de requisitos

Aqui estão descritos os requisitos funcionais de todo o sistema.

Quadro 2. Requisitos do sistema

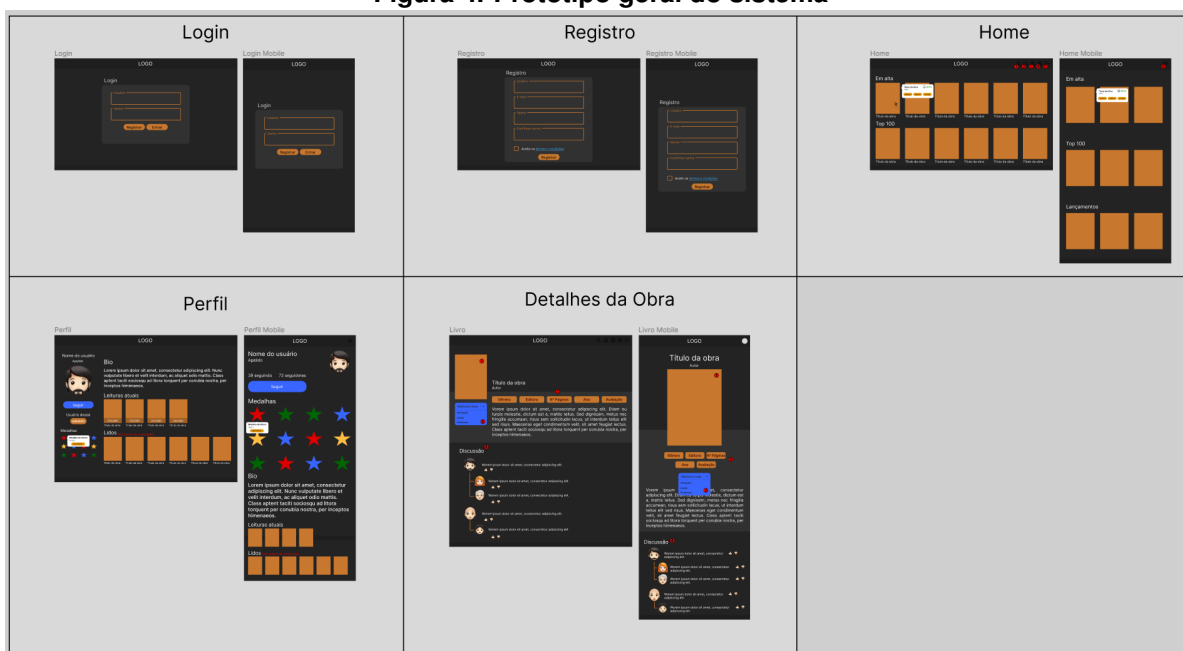
Requisito		Prioridade
RNF1	O sistema deve criptografar dados de usuário	Alta
RNF2	O sistema deve possuir design responsivo	Alta
RF1	O sistema deve permitir o login do usuário	Alta
RF2	O sistema deve permitir acessar a tela de cadastro	Alta
RF3	O sistema deve permitir recuperar uma conta	Alta
RF4	O sistema deve permitir o registro de um usuário	Alta
RF5	A tela de registro deve redirecionar o usuário para a página de login	Média
RF6	O sistema deve apresentar as obras em diferentes ordenações	Alta
RF7	O sistema deve permitir expandir a visualização das obras	Alta
RF8	O sistema deve apresentar todas as informações referentes à obra	Alta
RF9	O sistema deve permitir adicionar a obra à listas pessoais	Alta
RF10	O sistema deve permitir avaliar a obra	Alta
RF11	O sistema deve permitir realizar comentários sobre a obra	Média
RF12	O sistema deve apresentar todas as informações do usuário	Alta
RF13	O sistema deve permitir um usuário “seguir” outro usuário	Média

### 4.3. Protótipo

A Figura 4 apresenta um protótipo completo do sistema, desenvolvido para fornecer uma visão geral e facilitar o desenvolvimento do mesmo. Este protótipo foi criado na plataforma Figma, que oferece diversas funcionalidades de prototipagem, como animações, transições, conversão de protótipos para código, entre outras, porém como o objetivo do protótipo era apenas definir o layout das páginas do sistema, utilizamos apenas protótipos básicos, sem implementar animações ou utilizar a ferramenta de conversão.

Com esse objetivo em mente, a maioria das funcionalidades previstas foram inseridas no protótipo, a fim de visualizar a melhor maneira de posicioná-las no sistema real, considerando conceitos de experiência de usuário. As telas desenvolvidas de acordo com o protótipo construído serão apresentadas na seção seguinte.

**Figura 4. Protótipo geral do sistema**



Fonte: Autor

#### 4.4. Telas do Sistema

Esta subseção apresenta o modelo final das telas do sistema, que foi desenvolvido para ser acessado por meio de um navegador *web*, seja em um dispositivo móvel ou *desktop*. Após o usuário acessar o *link* do sistema ([booklist.fly.dev](http://booklist.fly.dev)), este deverá realizar o login com seu usuário e senha para obter acesso.

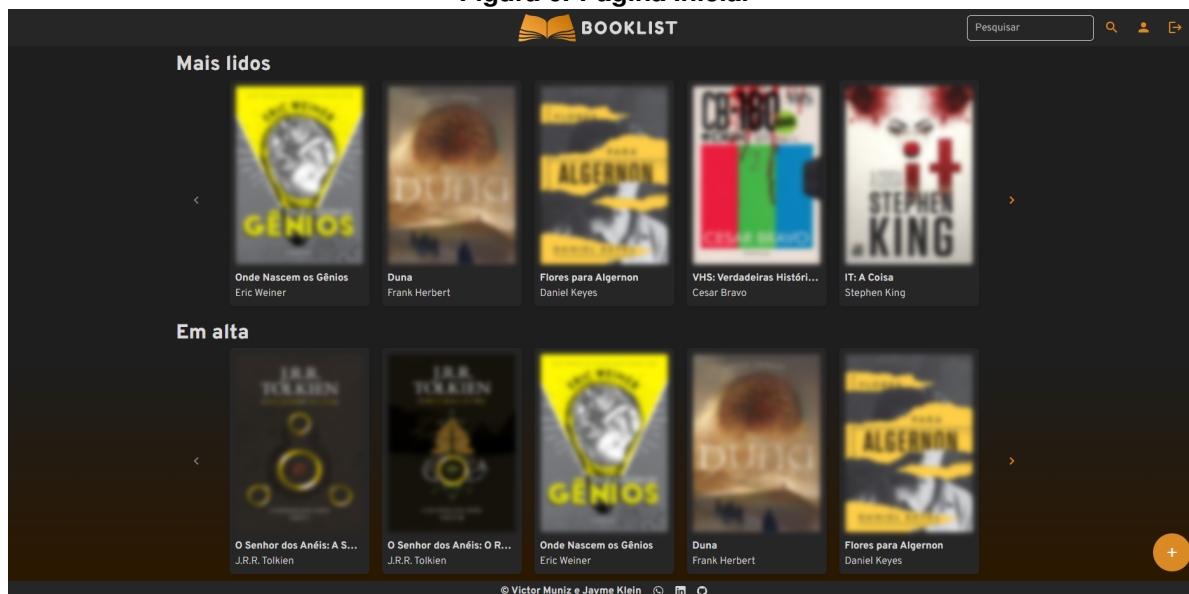
**Figura 5. Landing Page**



Fonte: Autor

A Figura 5 apresenta a *landing page* do sistema, que possui os componentes de login e registro. O login só é permitido mediante cadastro prévio, que pode ser acessado a partir desta tela clicando em “Registrar”. Ao realizar o registro, os dados do usuário são criptografados e armazenados em um banco de dados, e o usuário é redirecionado para a página de login, onde pode inserir suas credenciais para acessar o sistema.

Figura 6. Página Inicial



Fonte: Autor

A Figura 6 apresenta a tela inicial do sistema, onde estarão dispostas as principais obras

- Na seção “Em Alta”, estão destacados os livros que estão no topo das listas de leitura dos usuários, considerando apenas as listas “planejado” e “em andamento”, já que esse filtro busca elencar as principais obras no cenário atual.
- Na seção “Top 100”, são apresentados os livros mais lidos pela comunidade na plataforma. Para este filtro, considera-se apenas leituras concluídas.

No cabeçalho da tela principal, assim como em todas as outras telas do sistema, encontra-se a barra de navegação, com os itens:

1. Pesquisar
2. Perfil do usuário
3. Logout

No canto inferior direito, os usuários moderadores podem acessar a tela de inserção de obra, apresentada na Figura 7.

Figura 7. Tela de inserção de obra

**Inserir Obra**

Título  
Digite o título

Autor  
Digite o autor

Páginas  
Digite o número de páginas

Gênero  
Digite o gênero da obra

Editora  
Digite a editora

Ano  
Digite o ano de publicação

Síntese  
Escreva a sinopse

Capa  Nenhum arquivo escolhido

Fonte: Autor

Figura 8. Detalhes da Obra

**Onde Nascem os Gênios**  
Eric Weiner

Romance | Darkside | 352 páginas | 2015 | 4.3 ★

**Síntese**  
Vamos embarcar em uma viagem única e sem bilhete de volta para conhecer a origem e as raízes que alimentaram a mente e o espírito dos grande gênios da humanidade, conduzidos pelo aclamado autor Eric Weiner, um inveterado andarilho, amante da estrada e apaixonado pelo mundo em que vivemos. Um livro diferente: ora um diário de viagem, ora um romance repleto de informações históricas e costurado com [...ver mais](#)

**Resenhas**

Arkhandyr (Victor Muniz) - 14/11/2023 21:43 ★★★★★  
Ótimo livro!

Jaymeklein (Jayme Klein) - 14/11/2023 21:43 ★★★★★  
Não gostei muito, mas é envolvente.

**Publicar resenha**  
Escreva sua resenha aqui

★★★★★

Fonte: Autor

A Figura 8 apresenta a tela de detalhes da obra selecionada, onde além das informações principais da obra, traz outras classificações e informações (item 3), como editora, ano de publicação e a média da avaliação dos usuários. Além disso, também possui um botão para que o usuário possa

adicionar a obra em suas listas pessoais. Por fim, uma seção de interação onde os usuários do sistema podem escrever resenhas sobre a obra e curtir as resenhas de outros usuários.

Figura 9. Perfil de Usuário

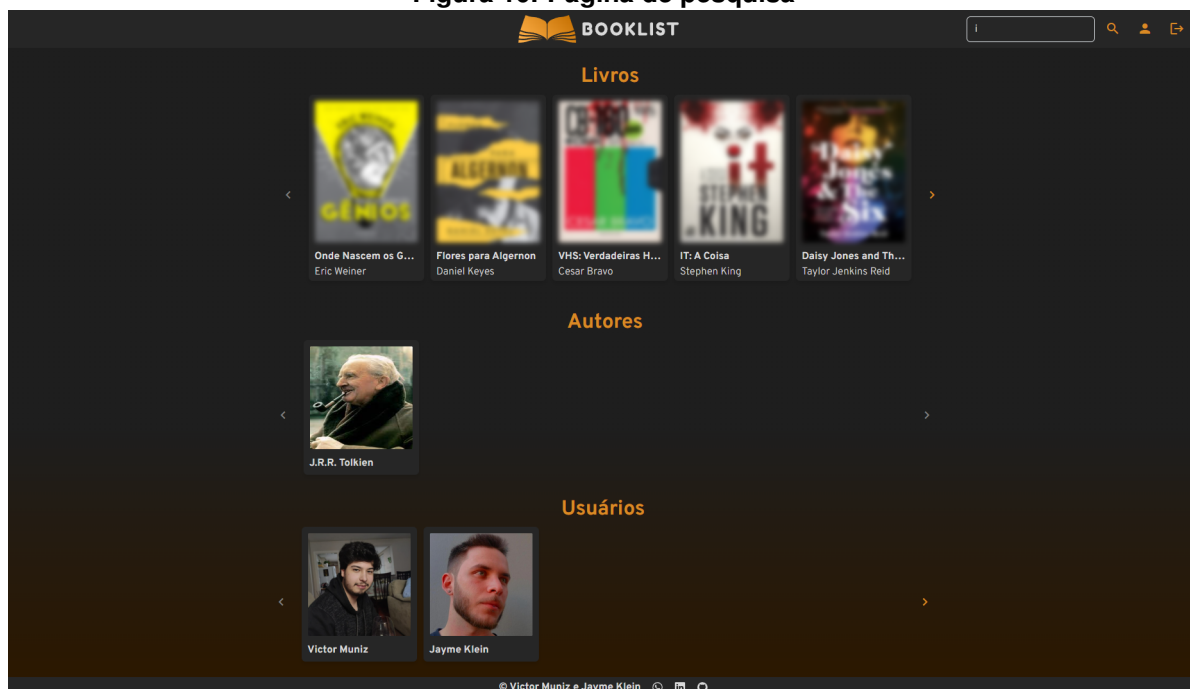


Fonte: Autor

Na Figura 9 é apresentada a tela de perfil de usuário. Essa tela é responsável por trazer todas as informações públicas de um usuário do sistema (nome, usuário, imagem, biografia), e apresentar as funcionalidades abaixo.

- Opção de seguir o perfil;
- As medalhas que o perfil possui;
- A biografia do perfil, editado pelo dono da conta;
- As listas de leitura;

Figura 10. Página de pesquisa



Fonte: Autor

Na Figura 10, é mostrada uma imagem da página de pesquisa do sistema. Nessa página, estão integradas as pesquisas de livros, autores e usuários, sem precisar especificar pelo que o usuário está procurando. Todos os resultados redirecionam para a página do item específico.

#### 4.5. Gamificação

A gamificação do sistema foi implementada por meio de reforços positivos, onde o usuário recebe insígnias por utilizar a plataforma. Estas insígnias estão listadas abaixo.

Quadro 3. Insígnias

Nome	Requisito
Leitor I	Adicionar uma obra à lista de obras finalizadas
Leitor II	Adicionar cinco obras à lista de obras finalizadas
Leitor III	Adicionar dez obras à lista de obras finalizadas
Reviews I	Publicar uma resenha
Reviews II	Publicar cinco resenhas
Reviews III	Publicar dez resenhas
BookLister I	Utilizar o sistema por um mês
BookLister II	Utilizar o sistema por seis meses
BookLister III	Utilizar o sistema por doze meses

#### 4.6. API

O código responsável por interligar o front-end com o banco de dados é, no escopo deste projeto, uma API (*Application Programming Interface*). Esta API foi elaborada em .NET no padrão Minimal API,

que consiste em um modelo mais simples e moderno de implementação de APIs, disponível na versão 6.0 do .NET Core.

Como descrito na Seção 3.4, a documentação da API foi realizada via Swagger, e uma descrição detalhada de todos os métodos e seus respectivos parâmetros, bem como os *schemas* das classes implementadas pela aplicação encontra-se nos apêndices deste artigo.

## 4.7. Filtros

Na página inicial, as obras possuem duas ordenações, “Em Alta” e “Top 100”. Para o devido funcionamento, foram criadas consultas no back-end utilizando o pacote MongoDB Driver para .NET.

Figura 11. Filtro “Em Alta”

```
2 references
public async Task<IEnumerable<Book>> GetTrendingBooks(int page)
{
    var pipeline = context.Books.Aggregate()
        .Lookup("Users_Books", "_id", "Book_id", "Users_Books")
        .Unwind("Users_Books")
        .Match(Builders<BsonDocument>.Filter.In("Users_Books.List", new BsonArray { "planning", "reading" }))
        .Group(new BsonDocument { { "_id", "$_id" }, { "BookId", new BsonDocument("$first", "$_id" ) }, { "ReadingCount", new BsonDocument("$sum", 1) }
        .Project(new BsonDocument("Id", "$BookId"))
        .Sort(Builders<BsonDocument>.Sort.Descending("ReadingCount"));

    var result = await pipeline.ToListAsync();

    var bookIds = result.Select(document => document["Id"].AsObjectId);

    var filter = Builders<Book>.Filter.In("_id", bookIds);
    var books = await context.Books.Find(filter).ToListAsync();
    var unreadBooks = await context.Books.Find(Builders<Book>.Filter.Not(filter)).ToListAsync();

    return books.Concat(unreadBooks).Skip((page - 1) * paginationSize).Take(paginationSize);
}
```

Fonte: Autor

Figura 12. Filtro “Top 100”

```
2 references
public async Task<IEnumerable<Book>> GetTopBooks(int page)
{
    var pipeline = context.Books.Aggregate()
        .Lookup("Users_Books", "_id", "Book_id", "Users_Books")
        .Unwind("Users_Books")
        .Match(Builders<BsonDocument>.Filter.In("Users_Books.List", new BsonArray { "done" }))
        .Group(new BsonDocument { { "_id", "$_id" }, { "BookId", new BsonDocument("$first", "$_id" ) }, { "ReadingCount", new BsonDocument("$sum", 1) }
        .Project(new BsonDocument("Id", "$BookId"))
        .Sort(Builders<BsonDocument>.Sort.Descending("ReadingCount"));

    var result = await pipeline.ToListAsync();

    var bookIds = result.Select(document => document["Id"].AsObjectId);

    var filter = Builders<Book>.Filter.In("_id", bookIds);
    var books = await context.Books.Find(filter).ToListAsync();
    var unreadBooks = await context.Books.Find(Builders<Book>.Filter.Not(filter)).ToListAsync();

    return books.Concat(unreadBooks).Skip((page - 1) * paginationSize).Take(paginationSize);
}
```

Fonte: Autor

O código de ambas as consultas criadas estão dispostas nas figuras 11 e 12.

## 5. Implantação

Após a implementação das funcionalidades básicas, com o sistema já estável para testes, foi realizada a implantação em uma plataforma em nuvem, a fim de disponibilizar o acesso para o público-alvo. Para isso, foram realizados estudos sobre containerização, para criação dos arquivos de configuração do back-end e front-end da aplicação.



Figura 13. Arquivo Dockerfile (back-end)

```
Dockerfile  ▹ ×
1  FROM mcr.microsoft.com/dotnet/aspnet:7.0 AS base
2  WORKDIR /app
3  EXPOSE 80
4
5  FROM mcr.microsoft.com/dotnet/sdk:7.0 AS build
6  WORKDIR /src
7  COPY ["BookList/BookList.csproj", "BookList/"]
8  RUN dotnet restore "BookList/BookList.csproj"
9  COPY . .
10 WORKDIR "/src/BookList"
11 RUN dotnet build "BookList.csproj" -c Release -o /app/build
12
13 FROM build AS publish
14 RUN dotnet publish "BookList.csproj" -c Release -o /app/publish /p:UseAppHost=false
15
16 FROM base AS final
17 WORKDIR /app
18 COPY --from=publish /app/publish .
19 ENTRYPOINT ["dotnet", "BookList.dll"]
20
```

Fonte: Autor

Figura 14. Arquivo Dockerfile (front-end)

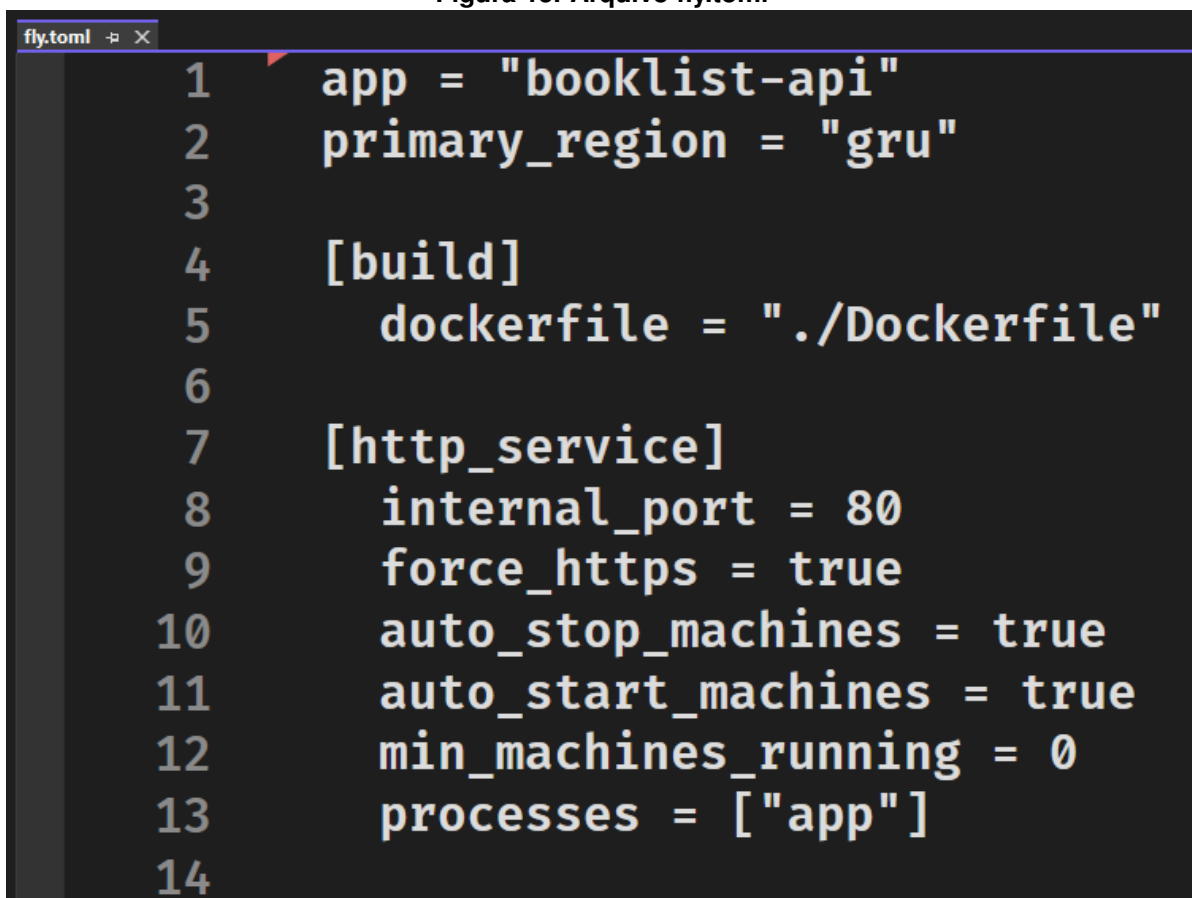
```
Dockerfile  ▹ ×
1  FROM node:20-alpine AS build
2  WORKDIR /booklist
3
4  COPY src ./src/
5  COPY package*.json .
6  COPY angular.json .
7  COPY tsconfig*.json .
8
9  RUN npm cache clean --force
10 RUN npm install
11 RUN npm run build
12
13 FROM nginx:alpine
14
15 COPY --from=build /booklist/dist/booklist /usr/share/nginx/html
16
17 COPY ./nginx.conf /etc/nginx/conf.d/default.conf
18
19 EXPOSE 4200
20
```

Fonte: Autor

Nas Figuras 13 e 14, estão descritos os Dockerfiles do back-end e front-end, respectivamente. Esses arquivos são usados para construir as imagens dos contêineres, estabelecendo uma ordem de execução de passos para criação das mesmas.

Após criados os Dockerfiles necessários, foi definida a plataforma responsável por hospedar o sistema. Por motivos de precificação e complexidade, optou-se por utilizar a plataforma Fly.io para este objetivo. Definida a plataforma, outro estudo foi realizado para configurar corretamente o ambiente, já que essas configurações são diferentes para cada plataforma. No caso da plataforma escolhida, a configuração é realizada por meio de um arquivo fly.toml

Figura 15. Arquivo fly.toml

A screenshot of a code editor showing the contents of a fly.toml file. The editor has a dark background with light-colored text. The file name 'fly.toml' is visible in the top-left corner of the editor window. The code is as follows:

```
1  app = "booklist-api"
2  primary_region = "gru"
3
4  [build]
5    dockerfile = "./Dockerfile"
6
7  [http_service]
8    internal_port = 80
9    force_https = true
10   auto_stop_machines = true
11   auto_start_machines = true
12   min_machines_running = 0
13   processes = ["app"]
14
```

Fonte: Autor

Na Figura 15, o arquivo fly.toml, responsável por definir o nome e a região do aplicativo, configurações de rede e segurança, além de configurações internas da plataforma.

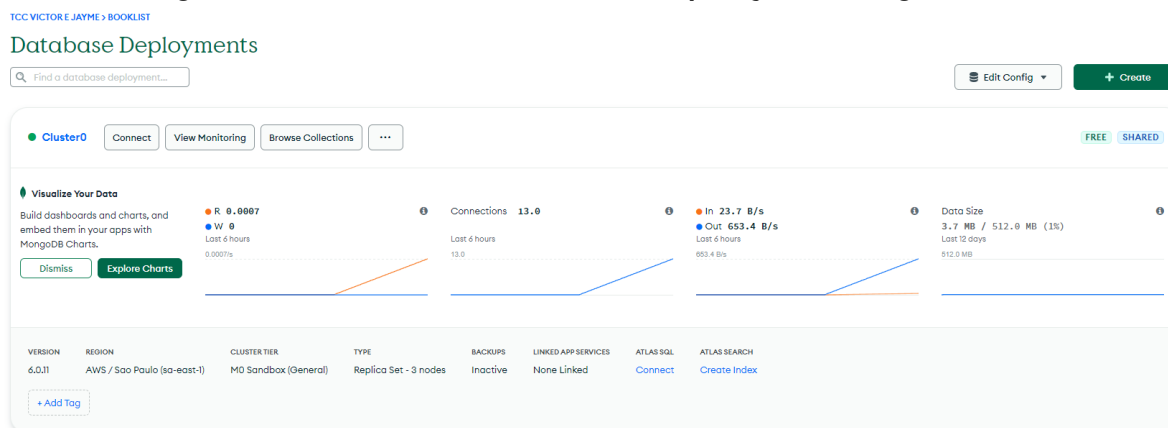
Figura 16. Arquivo fly.yaml

```
1  name: Fly Deploy
2
3  on:
4    push:
5      branches:
6        - master
7
8  env:
9    FLY_API_TOKEN: ${ secrets.FLY_API_TOKEN }
10
11 jobs:
12   deploy-front:
13     name: Deploy booklist-front
14     runs-on: ubuntu-latest
15     steps:
16       - uses: actions/checkout@v3
17
18       - uses: superfly/flyctl-actions/setup-flyctl@master
19
20       - run: flyctl deploy ./client --remote-only
21
22   deploy-back:
23     name: Deploy booklist-back
24     runs-on: ubuntu-latest
25     steps:
26       - uses: actions/checkout@v3
27
28       - uses: superfly/flyctl-actions/setup-flyctl@master
29
30       - run: flyctl deploy ./server --config ./BookList/fly.toml --remote-only
```

Fonte: Autor

A atualização do sistema implantado foi automatizada utilizando um *workflow*, que é uma sequência de passos executada quando determinada ação ocorre, criado no GitHub Actions, ferramenta de CI/CD (integração e entrega contínuas) do GitHub. Desta forma, não é necessário atualizar o sistema de forma manual, já que sempre que o código da ramificação principal do projeto é alterado, o *workflow* é executado. Na Figura 16, está descrito o workflow configurado.

**Figura 17. Cluster do banco de dados da aplicação no MongoDB Atlas**



Fonte: Autor

Por fim, um *dashboard* do *cluster*, mostrando o status e as estatísticas de conexões e consultas de dados realizados ao banco de dados da aplicação pode ser observado na Figura 17.

Esta foi a etapa mais enriquecedora deste projeto, pois engloba desde uma análise detalhada das plataformas disponíveis, compreensão de seu funcionamento, criação dos arquivos de configuração e *scripts* de execução. até a correção de bugs encontrados após a conclusão dessas etapas, o que destaca a complexidade do processo de implementação.

## 6. Trabalhos Futuros

Como forma de aprimorar o sistema, sugerimos as seguintes implementações que não foram realizadas nesta versão do projeto.

- Desenvolver um algoritmo de recomendação personalizado com base nas preferências e histórico de leitura do usuário. Isso inclui considerar o gênero da obra, autor, avaliações e outras informações relevantes.
- Garantir que o sistema seja acessível para pessoas com deficiência. Isso envolve simplificar as telas, fornecer suporte para leitores de tela e incluir descrições alternativas para imagens.
- Expandir as abordagens de gamificação para eventos periódicos, como desafios diários ou missões mensais.
- Mostrar uma barra de progresso das leituras em andamento na página de perfil dos usuários.

## 7. Conclusão

Este trabalho alcançou seu objetivo de desenvolver e implementar um sistema web com design responsivo, enfatizando uma interface de fácil utilização para a organização e controle de leituras, integrado com técnicas de gamificação para engajar os usuários. Além de permitir sugestões de inclusão e modificações de obras por parte dos usuários, este projeto foi uma plataforma valiosa para os alunos consolidarem seus conhecimentos práticos em front-end e back-end.

Através deste processo, foi possível uma imersão realista nas tecnologias atuais e nas demandas do mercado de trabalho na área de desenvolvimento web. Os alunos envolvidos no projeto tiveram a oportunidade de aplicar conceitos teóricos em um ambiente prático, enfrentando e superando desafios de engenharia de software que refletem o dinamismo do campo de Tecnologia da Informação.

## Referências

- Baah, C., Govender, I., and Subramaniam, P. R. (2023). Exploring the role of gamification in motivating students to learn. *Cogent Education*, 10(1):2210045.
- Barata, G., Gama, S., Jorge, J., and Gonçalves, D. (2013). Melhorando o ensino universitário com a gamificação. In *PORTUGUESE CONFERENCE ON HUMAN-MACHINE INTERACTION*, volume 5.
- Burlamaque, F. V. and Barth, P. A. (2015). Redes sociais e o ensino: o skoob como ferramenta para o letramento digital e literário. *Nuances: estudos sobre Educação*, 26(3):53–73.
- da Cruz, V. M. L. (2005). *Uma abordagem cognitiva às dificuldades na leitura: avaliação e intervenção*. Universidade Técnica de Lisboa (Portugal).
- Locke, E. A. and Latham, G. P. (2002). Building a practically useful theory of goal setting and task motivation: A 35-year odyssey. *American psychologist*, 57(9):705.
- Rocha, M. S. A. (2021). A influência da tecnologia na educação. *Tecnologias e mídias digitais na educação: conceitos práticos e teóricos*, page 7.
- SCHOEP, M. and WOOD, E. (2015). Motivating students to read extensively. *ISSN 1756-509X*, 1:16.
- Shen, L.-b. et al. (2006). Computer technology and college students' reading habits. *Chia-nan annual bulletin*, 32:559–572.
- Thelwall, M. and Kousha, K. (2017). Goodreads: A social network site for book readers. *Journal of the Association for Information Science and Technology*, 68(4):972–983.
- Tourinho, C. (2011). Refletindo sobre a dificuldade de leitura em alunos do ensino superior: “deficiência” ou simples falta de hábito. *Revista Lugares de Educação*, 1(2):325–346.
- Zichermann, G. and Cunningham, C. (2011). *Gamification by design: Implementing game mechanics in web and mobile apps*. ”O’Reilly Media, Inc.”.
- Zimmerman, B. J., Bandura, A., and Martinez-Pons, M. (1992). Self-motivation for academic attainment: The role of self-efficacy beliefs and personal goal setting. *American educational research journal*, 29(3):663–676.

## 8. Apêndices

### 8.1. Apêndice I - Documentação da API

#### ***GET* /catalog/trending/{page} Catálogo 'em alta' da página inicial**

Endpoint responsável por trazer o catálogo 'em alta' para a página inicial

##### Parameters

Type	Name	Description	Schema
path	page <i>required</i>		integer (int32)

##### Responses

Code	Description	Links
200	OK  <i>Content</i> application/json	No Links

#### ***GET* /catalog/top/{page} Catálogo 'top 100' da página inicial**

Endpoint responsável por trazer o catálogo 'top 100' para a página inicial

##### Parameters

Type	Name	Description	Schema
path	page <i>required</i>		integer (int32)

##### Responses

Code	Description	Links
200	OK  <i>Content</i> application/json	No Links

## **GET /booksByAuthor/{filter}** Filtro da página inicial

Endpoint responsável por filtrar os livros pelo autor

### Parameters

Type	Name	Description	Schema
path	<b>filter</b> <i>required</i>		string

### Responses

Code	Description	Links
200	OK  <i>Content</i> application/json	No Links

## **GET /book/{id}** Seleciona livro

Endpoint responsável por trazer as informações do livro selecionado para a página de obra

### Parameters

Type	Name	Description	Schema
path	<b>id</b> <i>required</i>		string

### Responses

Code	Description	Links
200	OK  <i>Content</i> application/json	No Links

## **POST /add** Adiciona livro

Endpoint responsável por adicionar um novo livro no banco de dados

### Responses

Code	Description	Links
200	OK	No Links

## **POST /addToList** Adiciona livro à lista

Endpoint responsável por adicionar um livro à uma lista de leituras

### Responses

Code	Description	Links
200	OK	No Links

## **POST /removeFromList** Remove livro da lista

Endpoint responsável por remover um livro da lista de leituras

### Responses

Code	Description	Links
200	OK	No Links

## **GET /books/{bookId}/{username}** Status de leitura do livro

Endpoint responsável por informar se o usuário já possui o livro em sua estante

### Parameters

Type	Name	Description	Schema
path	<b>bookId</b> <i>required</i>		string
path	<b>username</b> <i>required</i>		string

### Responses

Code	Description	Links
200	OK	No Links

## **POST /books/addReview** Adiciona resenha a um livro

Endpoint responsável por adicionar uma resenha a um livro

### Responses

Code	Description	Links
200	OK	No Links



## **POST /books/deleteReview** Remove resenha de um livro

Endpoint responsável por remover uma resenha de um livro

### Responses

Code	Description	Links
200	OK	No Links

## **GET /reviews/{bookId}** Resenhas da obra

Endpoint responsável por trazer todas as resenhas da obra na página da obra

### Parameters

Type	Name	Description	Schema
path	<b>bookId</b> <i>required</i>		string

### Responses

Code	Description	Links
200	OK	No Links

## **POST /reviews/likeReview** Curte uma resenha de livro

Endpoint responsável por marcar uma resenha como 'curtida'

### Responses

Code	Description	Links
200	OK	No Links

## **POST /reviews/dislikeReview** Remove curtida de uma resenha de livro

Endpoint responsável por desmarcar uma resenha como 'curtida'

### Responses

Code	Description	Links
200	OK	No Links

## **POST /register** Registro

Endpoint responsável por adicionar um novo usuário no banco de dados

### Responses

Code	Description	Links
200	OK	No Links

## **POST /login** Login

Endpoint responsável por gerar um token para o usuário realizando login

### Responses

Code	Description	Links
200	OK	No Links

## **GET /user** Usuário

Endpoint responsável por trazer as informações de um usuário logado

### Responses

Code	Description	Links
200	OK	No Links

## **POST /logout** Logout

Endpoint responsável por deslogar o usuário da sessão

### Responses

Code	Description	Links
200	OK	No Links

## **POST /token GetToken**

Endpoint responsável por retornar o token de autenticação

### *Responses*

Code	Description	Links
200	OK	No Links

## **GET /profile/{username} Seleciona usuário**

Endpoint responsável por trazer as informações do usuário selecionado para a página de perfil de usuário

### *Parameters*

Type	Name	Description	Schema
path	<b>username</b> <i>required</i>		string

### *Responses*

Code	Description	Links
200	OK	No Links

## **GET /{list}/{username}/{page} Estante virtual do usuário**

Endpoint responsável por trazer as listas do usuário selecionado para a página de perfil

### *Parameters*

Type	Name	Description	Schema
path	<b>page</b> <i>required</i>		integer (int32)
path	<b>list</b> <i>required</i>		string
path	<b>username</b> <i>required</i>		string

### *Responses*

Code	Description	Links
200	OK	No Links

## **GET /badges/{username}** Conquistas do usuário

Endpoint responsável por trazer as conquistas do usuário selecionado para a página de perfil de usuário

### Parameters

Type	Name	Description	Schema
path	username <i>required</i>		string

### Responses

Code	Description	Links
200	OK	No Links

## **GET /bookCount/{username}** Número de livros em cada lista

Endpoint responsável por trazer o número de livros que o usuário possui em cada lista

### Parameters

Type	Name	Description	Schema
path	username <i>required</i>		string

### Responses

Code	Description	Links
200	OK	No Links

## **GET /author/{name}** Seleciona autor

Endpoint responsável por trazer as informações do autor selecionado para a página de autor

### Parameters

Type	Name	Description	Schema
path	name <i>required</i>		string

### Responses

Code	Description	Links
200	OK	No Links

## **GET /searchBooks/{filter}/{page} Filtra livros**

Endpoint responsável por filtrar os livros pelo nome

### Parameters

Type	Name	Description	Schema
path	<b>filter</b> <i>required</i>		string
path	<b>page</b> <i>required</i>		integer (int32)

### Responses

Code	Description	Links
200	OK  <i>Content</i> <b>application/json</b>	No Links

## **GET /searchAuthors/{filter}/{page} Filtra autores**

Endpoint responsável por filtrar os autores pelo nome

### Parameters

Type	Name	Description	Schema
path	<b>filter</b> <i>required</i>		string
path	<b>page</b> <i>required</i>		integer (int32)

### Responses

Code	Description	Links
200	OK  <i>Content</i> <b>application/json</b>	No Links

## **GET /searchUsers/{filter}/{page} Filtra usuários**

Endpoint responsável por filtrar os usuários pelo nome

### Parameters

Type	Name	Description	Schema
path	<b>filter</b> <i>required</i>		string
path	<b>page</b> <i>required</i>		integer (int32)

### Responses

Code	Description	Links
200	OK  <i>Content</i> <b>application/json</b>	No Links

## **POST /follow Segue um usuário**

Endpoint responsável por seguir um usuário

### Responses

Code	Description	Links
200	OK	No Links

## **POST /unfollow Deixa de seguir um usuário**

Endpoint responsável por deixar de seguir um usuário

### Responses

Code	Description	Links
200	OK	No Links

## **GET /followStatus/{user}/{user2}** Verifica se um usuário segue outro usuário

Endpoint responsável por verificar o status de interação entre os usuários

### Parameters

Type	Name	Description	Schema
path	<b>user2</b> <i>required</i>		string
path	<b>user</b> <i>required</i>		string

### Responses

Code	Description	Links
200	OK	No Links