



FINDBOOKS: PLATAFORMA DIGITAL PARA AUTOMAÇÃO DE EMPRÉSTIMO E ACESSO AMPLIADO A ACERVOS BIBLIOTECÁRIOS

Guilherme De Oliveira Lobo¹
Igor Matheus De Oliveira Batista²
Luccas Henrique Cardum Bismara³
Matheus Antunes Dos Santos⁴
Vinicius Wenzel Bulik Moreira⁵
Anderson Reis De Campos⁶
Alessandro Augusto Jordão⁷

Resumo: O presente artigo aborda a criação da plataforma FindBooks, que visa solucionar a dificuldade de acesso a livros em bibliotecas, organizando acervos e automatizando processos de empréstimo de forma prática e acessível. A metodologia incluiu o desenvolvimento da aplicação em Visual Studio, utilizando C# para o backend e HTML, CSS e JavaScript no front-end, além do SQLite como banco de dados. Como conclusão, o projeto propõe-se a ser um facilitador no acesso ao conhecimento, promovendo a leitura e ampliando a visibilidade das bibliotecas, com potencial de expansão para regiões com pouco acesso a recursos culturais.

Palavras-Chaves: Bibliotecas, Tecnologia, Automação, Empréstimo de Livros, Acesso à Informação.

Abstract: This article presents the development of the FindBooks platform, designed to address the challenge of accessing books in libraries by organizing collections and automating loan processes in a practical and accessible way. The methodology involved application development in Visual Studio, using C# for the backend and HTML, CSS, and JavaScript for the front-end, with SQLite as the database. In conclusion, the project aims

to facilitate access to knowledge, promote reading, and enhance the visibility of libraries, with potential for expansion to regions with limited access to cultural resources.

Keywords: Libraries, Technology, Automation, Book Loans, Information Access.

1. INTRODUÇÃO

O desenvolvimento e aplicação de tecnologias digitais têm transformado diversos setores, incluindo o sistema de bibliotecas, que, historicamente, desempenha um papel essencial no acesso à informação e na promoção da leitura. Neste contexto, a tecnologia se apresenta como uma ferramenta estratégica para modernizar e automatizar processos tradicionais, como o empréstimo de livros, tornando-o mais acessível e conveniente para um público cada vez mais conectado. Diante desse cenário, surge o FindBooks, uma proposta de plataforma digital concebida para facilitar a interação entre bibliotecas e leitores, proporcionando um sistema centralizado de busca e empréstimo de livros em um ambiente online.

O objetivo do FindBooks é conectar bibliotecas de diferentes tipos e regiões, permitindo que leitores acessem acervos variados de forma prática e eficiente. A plataforma possibilita que usuários verifiquem a disponibilidade de livros em tempo real, ampliando as oportunidades de acesso e promovendo uma experiência mais organizada e atrativa. Ao mesmo tempo, as bibliotecas encontram uma oportunidade para divulgar seus acervos e atrair novos leitores, apresentando-se como um ponto de encontro entre a tecnologia e o incentivo à leitura.

A relevância deste projeto está diretamente associada à necessidade de incentivar a leitura e facilitar o acesso à informação, especialmente em uma era digital onde, paradoxalmente, o contato com livros físicos e o hábito de leitura têm se tornado cada vez menos frequentes. Além disso, o FindBooks atende a uma demanda social significativa ao promover o acesso a livros em comunidades onde há limitações de recursos e de infraestrutura para bibliotecas físicas. Ao combinar uma interface intuitiva com um sistema de busca eficiente, o projeto visa fortalecer o papel social das bibliotecas e contribuir para a formação de uma comunidade leitora inclusiva, acessível e conectada.

2. OBJETIVO

Com o objetivo de solucionar dificuldades enfrentadas por leitores ao buscarem livros específicos, oferecendo uma ferramenta centralizada que permita a verificação da disponibilidade de obras em tempo real, incentivando a leitura e promovendo o acesso ao conhecimento, buscou-se como objetivo central, criar uma plataforma digital acessível e eficiente que facilite o acesso de leitores a acervos de diferentes bibliotecas, sejam elas públicas ou privadas.

Como objetivos específicos, buscou-se: (a) a centralização de acervos, (b) automatização do empréstimo, (c) melhoria na organização e visibilidade dos acervos, (d) usabilidade e experiência do usuário: Desenvolver uma interface intuitiva e organizada que facilite o uso da plataforma tanto para leitores quanto para as instituições, garantindo uma experiência agradável e eficiente e (d) expansão do acesso à informação, propondo uma solução escalável, com potencial para ser replicada em diferentes localidades. Além disso, busca-se a promoção da leitura e inclusão cultural e do uso de tecnologia em bibliotecas.

Esses objetivos convergem para a criação de uma solução inovadora e sem fins lucrativos que atua como mediadora entre as bibliotecas e os leitores, promovendo o acesso ao conhecimento, organizando acervos e incentivando o engajamento com a leitura em uma plataforma inclusiva e acessível.

3. METODOLOGIA

Para o desenvolvimento da plataforma FindBooks, adotou-se uma metodologia estruturada em etapas que compreendem o planejamento, a codificação e a implementação da aplicação. Abaixo, detalham-se as ferramentas e tecnologias empregadas em cada fase para garantir o funcionamento adequado da plataforma, a integração entre as interfaces de usuário e a base de dados, além da organização e controle das versões do projeto.

Primeiramente, foram definidos os requisitos funcionais e não funcionais da plataforma, com base no objetivo central de facilitar o acesso dos leitores a acervos de diferentes bibliotecas e automatizar o processo de empréstimo de livros. Nesse estágio,

identificaram-se as funcionalidades principais, como a consulta de acervos, o sistema de reservas e a gestão de disponibilidade dos livros. Também foi especificada a estrutura do banco de dados, projetando-se a organização de dados como informações de usuários, acervos das bibliotecas e status dos empréstimos.

O ambiente de desenvolvimento escolhido foi o Visual Studio, uma IDE que suporta as tecnologias necessárias para o projeto, fornecendo recursos avançados para a edição de código, depuração e gerenciamento de bibliotecas e dependências. O Visual Studio permite uma integração fluida entre o *front-end* e o *back-end*, oferecendo suporte nativo à linguagem C# e tecnologias de design para a interface do usuário.

Destaca-se que o desenvolvimento do *back-end* foi feito em C#, uma linguagem robusta e segura para aplicações de médio e grande porte, que permite a criação de APIs e serviços que garantem a comunicação entre a interface do usuário e o banco de dados. Além disso, utilizou-se C# para implementar as funcionalidades principais do sistema, como o gerenciamento de usuários e de bibliotecas cadastradas, a consulta de livros, o controle de disponibilidade dos exemplares e o sistema de empréstimo e devolução de livros. Ademais, C# oferece recursos para tratar a segurança dos dados, de forma que informações pessoais dos usuários e dados das bibliotecas são protegidos e gerenciados de acordo com práticas recomendadas de segurança.

Destaca-se também que o *front-end* foi desenvolvido utilizando HTML, CSS e JavaScript, permitindo uma interface intuitiva e responsiva para os usuários da plataforma. HTML foi utilizado para a estruturação das páginas; CSS para o design, estilização e responsividade; e JavaScript para funcionalidades dinâmicas e interatividade. Além disso, o design foi projetado para ser intuitivo, com menus organizados e layout amigável. A interface foi otimizada para garantir uma experiência fluida em diferentes dispositivos, como desktops, tablets e smartphones, permitindo aos usuários realizar buscas e reservas com facilidade. Ademais, a escolha do SQLite como banco de dados foi motivada pela simplicidade e eficiência que ele oferece para aplicações de pequeno e médio porte. O banco de dados armazena informações cruciais, como o catálogo de livros de cada biblioteca, dados de usuários, registros de reservas e histórico de empréstimos.

Adicionalmente, o banco de dados foi projetado com tabelas e relacionamentos que suportam as funcionalidades do sistema, garantindo que as consultas e operações de

atualização, como reservas e devoluções de livros, sejam rápidas e eficientes. A estrutura foi otimizada para suportar o crescimento da plataforma sem comprometer o desempenho.

Assinala-se também que o GitHub foi utilizado para armazenar o código-fonte e gerenciar o controle de versão do projeto. A ferramenta permitiu a organização e o monitoramento do progresso, facilitando a colaboração entre desenvolvedores e a documentação das mudanças. Cada etapa de desenvolvimento foi visionada, permitindo um histórico de alterações que facilita o rastreamento e correção de eventuais erros.

Por fim, destaca-se que, após a implementação das funcionalidades principais, serão realizados testes rigorosos de funcionalidade, usabilidade e desempenho para garantir que a plataforma opera de acordo com os requisitos definidos. Testes de integração serão conduzidos para verificar a comunicação entre o *back-end* e o *front-end*, enquanto testes de carga foram executados para avaliar o desempenho do sistema sob diferentes níveis de uso.

Com a conclusão dos testes, a plataforma será preparada para implantação em um ambiente de produção. Estabelecer-se-á um plano de monitoramento contínuo para identificar e corrigir possíveis falhas, garantindo a disponibilidade e segurança da aplicação para os usuários.

Essa metodologia visa criar uma plataforma robusta e acessível, que se destaca pela organização dos acervos e pela facilidade de uso, possibilitando que os usuários encontrem e reservem livros com eficiência e segurança.

4. DESENVOLVIMENTO

Sobre o desenvolvimento da plataforma FindBooks foi planejado para garantir uma aplicação robusta, segura e amigável, focada em facilitar o acesso a acervos de diferentes bibliotecas e simplificar o processo de empréstimo de livros. A equipe adotou uma abordagem detalhista, com atenção especial à coerência, responsividade e usabilidade da aplicação, buscando empregar ferramentas modernas e amplamente testadas no mercado.

Para o ambiente de desenvolvimento integrado (IDE), foi escolhido o Visual Studio 2022, uma versão atualizada e reconhecida pela eficiência e suporte a tecnologias complexas como C#. Esse ambiente é ideal para o desenvolvimento do código *back-end*,

possibilitando que a lógica de programação necessária para o sistema de empréstimos seja desenvolvida de forma segura e otimizada. A linguagem C# foi selecionada para o desenvolvimento do *back-end* devido à sua robustez, segurança e capacidade de integrar-se bem com bancos de dados e aplicações web, garantindo que a plataforma funcione de maneira estável mesmo em condições de alto uso.

No *front-end*, foram adotadas tecnologias conhecidas e amplamente utilizadas na indústria: HTML, CSS e JavaScript. O HTML é responsável pela estrutura das páginas, enquanto o CSS define o layout e a estética visual, oferecendo uma interface agradável e moderna para os usuários. O JavaScript, por sua vez, é empregado para adicionar interatividade, permitindo uma experiência dinâmica e responsiva que melhora a usabilidade do sistema, garantindo que os usuários possam navegar e realizar ações com facilidade e eficiência.

Para o gerenciamento e organização de dados, foi adotado o banco de dados SQLite. Este banco de dados é leve e seguro, o que facilita o armazenamento de um grande volume de informações, como bibliotecas, livros, palavras-chave e registros de usuários, sem a necessidade de um servidor dedicado. Essa escolha proporciona uma solução prática para o volume de dados previsto e permite uma escalabilidade gradual conforme o sistema cresce e incorpora novas bibliotecas.

O GitHub foi utilizado como plataforma para o controle de versões e administração do projeto. Ele possibilita que a equipe acompanhe as mudanças no código de forma colaborativa, registrando cada modificação e facilitando o monitoramento do progresso do projeto. O GitHub também atua como um repositório centralizado onde o código é armazenado, oferecendo uma visão detalhada do desenvolvimento e permitindo a colaboração entre todos os membros da equipe.

(a) Requisitos Funcionais

Os requisitos funcionais definidos para o sistema estabelecem as funcionalidades essenciais da plataforma, que incluem:

- Login (RF 01): Interface onde usuários e bibliotecas inserem seus dados de acesso para entrar no sistema, garantindo uma experiência segura e personalizada.

- Cadastro da Instituição (RF 02): Tela de cadastro destinada às bibliotecas, onde preenchem informações institucionais, como CNPJ e nome fantasia, para integrarem seus acervos à plataforma.
- Cadastro do Livro (RF 03): Permite que as bibliotecas registrem informações detalhadas sobre os livros em seus acervos, tornando-os visíveis para os usuários.
- Consulta de Disponibilidade (RF 04): Ferramenta de pesquisa onde os usuários podem verificar a disponibilidade de títulos específicos, facilitando a busca por livros de interesse.
- Controle de Estoque (RF 05): Interface para que as bibliotecas monitorem seu estoque e atualizem a disponibilidade dos livros, essencial para um sistema de empréstimos eficiente.
- Relatório de Empréstimos (RF 06): Gera relatórios de entrada e saída de livros, registrando datas e horários para facilitar o controle e a análise de uso.

(b) Requisitos Não Funcionais

Além das funcionalidades principais, foram definidos requisitos não funcionais para garantir o desempenho e a usabilidade do sistema:

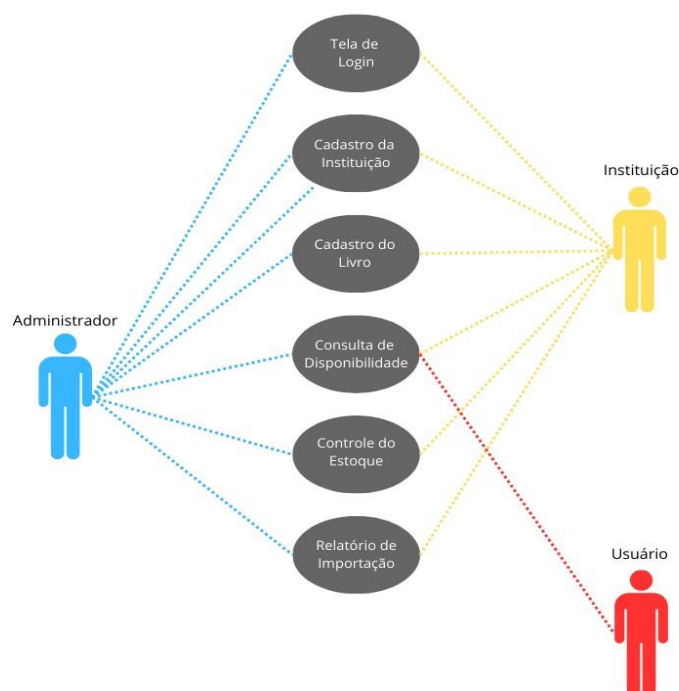
- Autenticação Multi-Fator (RNF 01): Para maior segurança, especialmente para administradores, foi implementada a autenticação em duas etapas.
- Suporte a 500 Usuários Simultâneos (RNF 02): O sistema deve operar sem degradação perceptível do desempenho, garantindo uma experiência estável.
- Tempo de Resposta (RNF 03): As consultas de pesquisa devem ser processadas em até 2 segundos, visando a eficiência e a satisfação do usuário.
- Armazenamento de Dados (RNF 04): A capacidade de armazenar até 1 milhão de registros de livros, com fácil expansão, garante a viabilidade do sistema a longo prazo.
- Interface Intuitiva (RNF 05): Com foco em usabilidade, a plataforma deve ter uma curva de aprendizado inferior a 20 minutos para novos usuários.
- Modularidade do Sistema (RNF 06): A arquitetura modular facilita a atualização de componentes sem impactar o sistema como um todo.

- Backup Diário Automatizado (RNF 07): O sistema conta com backups diários, garantindo que, em caso de falha, a recuperação seja possível em até 1 hora. empresas de locação de equipamentos. Com flexibilidade, segurança e escalabilidade, essa integração não só aprimora a experiência do cliente, mas também assegura que a empresa esteja preparada para responder às demandas do mercado com agilidade e precisão.

(c) Tabelas de Caso de Uso

As tabelas de caso de uso detalham cada funcionalidade que o sistema FindBooks precisa oferecer.

Figura 1: Diagrama de Caso de Uso Fonte: Elaboração Própria



Fonte: Elaboração Própria

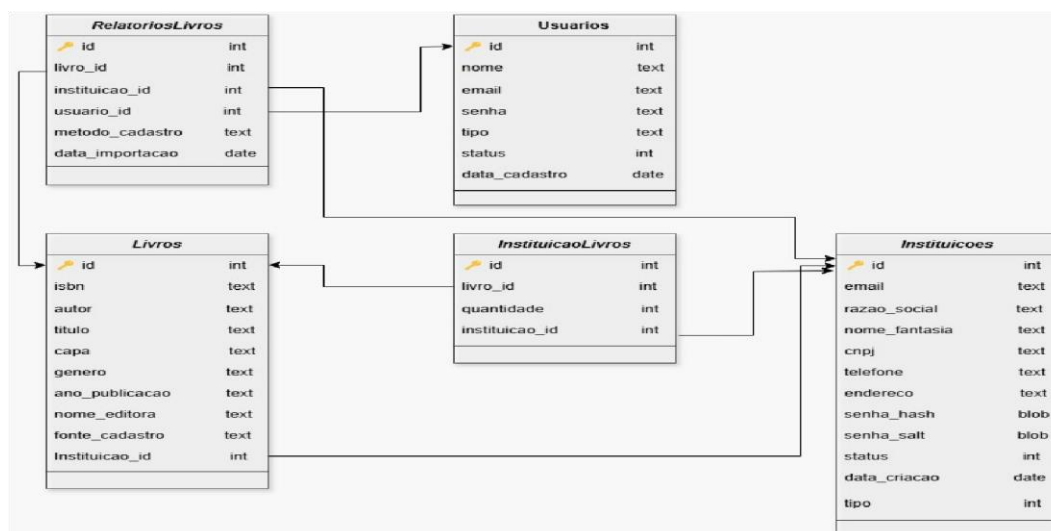
Cada requisito funcional (RF) é descrito com informações sobre a pré-condição, atores envolvidos, fluxo principal de ações e fluxos alternativos, detalhando o que acontece caso algum passo não seja seguido corretamente.

1. RF01 - Login (Administrador/Instituição) o Descrição: Tela onde instituições inserem seus dados de acesso para entrar no sistema.

- o Ator: Administrador, Instituição. o Pré-Condição: A interface de login deve estar online.
 - o Fluxo Principal: O sistema abre a tela de login, a instituição preenche as credenciais, valida as informações e concede acesso se os dados estiverem corretos.
 - o Fluxo Alternativo: Se as credenciais forem inválidas, o sistema exibe uma mensagem de erro.
2. RF02 - Cadastro da Instituição o Descrição: Interface para cadastro das bibliotecas, onde preenchem dados institucionais. o Ator: Instituição. o Pré-Condição: A interface de cadastro deve estar online.
- o Fluxo Principal: A tela de cadastro é aberta, a instituição escolhe a categoria e insere os dados exigidos, concluindo o cadastro após validação.
 - o Fluxo Alternativo: Em caso de erro nos dados ou CNPJ não encontrado, o sistema informa o erro e orienta para correção.
3. RF03 - Cadastro de Livros o Descrição: Tela para que o representante da instituição registre novos livros na plataforma. o Ator: Instituição.
- o Pré-Condição: A interface de cadastro de livros deve estar online e o ator logado no sistema.
 - o Fluxo Principal: A instituição acessa a tela, preenche informações sobre o livro, como título, gênero e quantidade, e confirma o cadastro, que é armazenado no banco de dados.
 - o Fluxo Alternativo: Se algum dado obrigatório estiver ausente, o sistema exibe uma mensagem de erro.
4. RF04 - Consulta de Disponibilidade de Livros o Descrição: Tela onde usuários e instituições consultam a disponibilidade de livros. o Ator: Usuário/Instituição.
- o Pré-Condição: O sistema deve estar conectado e sincronizado com o banco de dados.
 - o Fluxo Principal: O usuário acessa a tela de consulta, insere informações para busca e visualiza a disponibilidade do livro.
 - o Fluxo Alternativo: Se os dados são inválidos ou o livro não está cadastrado, o sistema exibe uma mensagem de erro.

5. RF05 - Controle de Estoque o Descrição: Interface para que a instituição monitore e atualize seu estoque de livros. o Ator: Instituição.
 - o Pré-Condição: A interface de estoque deve estar conectada ao banco de dados.
 - o Fluxo Principal: A instituição acessa o estoque, verifica e atualiza a quantidade de livros e confirma as alterações.
 - o Fluxo Alternativo: Se o banco de dados estiver offline ou não sincronizado, o sistema exibe uma mensagem de erro.
6. RF06 - Relatório de Importação o Descrição: Tela onde a instituição pode monitorar e gerenciar os livros importados.
 - o Ator: Instituição.
 - o Pré-Condição: A interface de estoque deve estar conectada e sincronizada com o banco de dados.
 - o Fluxo Principal: A instituição abre a tela de importação, visualiza os livros importados.
 - o Fluxo Alternativo: Se o banco de dados estiver offline ou não sincronizado, o sistema exibe uma mensagem de erro.

Figura 2: Diagrama E.R. (Entidade Relacionamentos)



Fonte: Elaboração Própria

(d) Estrutura das Tabelas de Banco de Dados

Para armazenar e organizar as informações, foram definidas duas tabelas principais:

Livros e Instituições.

1. Tabela 'Livros' o Id: Identificador único da instituição (tipo: Int, não nulo), funciona como chave primária.

- o Capa: Imagem ilustrativa da capa do livro (tipo: LongText, não nulo). o Título: Título do livro (tipo: LongText, não nulo).
- o ISBN: Código único de identificação do livro (tipo: LongText, não nulo), que auxilia no controle e pesquisa de registros específicos.
- o Gênero: Categoria ou gênero literário do livro (tipo: LongText, não nulo). o Autor: Nome do autor do livro (tipo: LongText, não nulo).
- o AnoPublicação: Ano de publicação do livro (tipo: DateTime(6), não nulo), facilitando filtros por data e análises temporais do acervo.
- o NomeEditora: Nome da editora para identificação (tipo: text, não nulo).

FonteCadastro: Identificador da origem do cadastro (tipo: Text, não nulo), para cadastro do tipo manual ou em lote. o Instituição_id: Identificador da instituição associada (tipo: Int, não nulo).

2. Tabela 'Instituições' o Id: Identificador único da identificação (tipo: Int, não nulo), utilizado como chave primária.

- o Email: Endereço de e-mail para contato da instituição (tipo: LongText, não nulo), utilizado para comunicação e recuperação de dados.
- o RazãoSocial: Nome jurídico da instituição, (tipo: Text, não nulo), utilizado para consulta.
- o Senha: Senha de cadastro da instituição (tipo: Blob, não nulo), para segurança e identificação.
- o NomeFantasia: Nome fantasia da instituição (tipo: LongText, não nulo), para identificação pública.
- o CNPJ: Registro nacional da instituição (tipo: Int, não nulo), assegurando a validade legal das instituições cadastradas.
- o Telefone: Telefone de contato (tipo: Int, não nulo), que facilita o suporte e a comunicação direta.
- o Endereço: Localização física da instituição (tipo: LongText, não nulo), essencial para consultas regionais e orientações ao usuário.

- o Status: Indica quando o usuário esta ativo ou inativo (tipo: Int, não nulo), para controlar o acesso sem a necessidade de exclusão.
 - o DataDeCriação: Data de cadastro da instituição no sistema (tipo: DateTime, não nulo), para controle histórico.
 - o Tipo: Define o tipo do usuário (Administrador/Instituição), (tipo: Int, não nulo), utilizado para gerenciar níveis de acesso.
3. Tabela 'Instituição Livros' o Id: Identificador único de associação (tipo: Int, não nulo), utilizado como chave primária.
- Livro_id: Identificador único de livro (tipo: Int, não nulo). para identificação de livros.
- o Quantidade: Consulta de quantidade de livros cadastrados (tipo: Int, não nulo).
 - o Instituição_id: identificador de usuário por registro (tipo: Int, não nulo), assegurando a validade legal das instituições cadastradas.
4. Tabela 'Usuários' o Id: Identificador único de associação (tipo: Int, não nulo), utilizado como chave primária.
- o Nome: Nome do Usuário (tipo: Text, não nulo), para identificação.
 - o Email: Endereço de e-mail para contato do Usuário (tipo: LongText, não nulo), utilizado para comunicação e recuperação de dados.
 - o Senha: Senha de cadastro do usuário para identificação (tipo: LongText, não nulo). Utilizado para autentificação.
 - o Tipo: Define o tipo do usuário (Administrador/Instituição), (tipo: Int, não nulo), utilizado para gerenciar níveis de acesso.
 - o Status: Indica quando o usuário está ativo ou inativo (tipo: Int, não nulo), para controlar o acesso sem a necessidade de exclusão.
 - o DataCadastro: Data e hora de criação de registro (tipo: DateTime, não nulo), assegura controle de estoque.
5. Tabela 'Relatório Livros' o Id: Identificador único de associação (tipo: Int, não nulo), utilizado como chave primária.
- o Livro_id: Identificador único de livro (tipo: Int, não nulo), para identificação de livros.

- o Método: Definir para qual o livro foi cadastro sendo manual para registros individuais ou lote para registros em massa (tipo: Int, não nulo), assegurando eficiência da plataforma.
- o Usuário_id: Indica quando o usuário está ativo ou inativo (tipo: Int, não nulo), para controlar o acesso sem a necessidade de exclusão.

Instituição_id: Identificador da instituição associada (tipo: Int, não nulo). o

DataImportação: Data de realização de importação em lote (tipo: DateTime, não nulo), para constar a data do livro adicionado no sistema.

Essas tabelas são fundamentais para garantir a consistência e organização dos dados da plataforma FindBooks, permitindo uma interação estruturada entre usuários, bibliotecas e o sistema de consulta e empréstimo de livros. A organização de dados por meio dessas tabelas possibilita que o sistema execute as consultas e operações com eficiência, assegurando que a experiência do usuário seja rápida e satisfatória.

5. CONSIDERAÇÕES FINAIS

Em conclusão, a plataforma FindBooks surge como uma solução inovadora para otimizar o acesso a acervos bibliotecários, conectando leitores e instituições em um ambiente digital intuitivo e acessível. Com uma interface moderna e a utilização de tecnologias confiáveis para armazenamento, controle de versões e segurança, o sistema facilita o processo de consulta e empréstimo de livros, promovendo a organização dos acervos e incentivando a leitura. O desenvolvimento da plataforma atende a demandas tecnológicas atuais, permitindo que as bibliotecas aumentem sua visibilidade e alcancem novos públicos. Além disso, o sistema FindBooks é uma ferramenta escalável, com potencial para ser replicada em diversas regiões, especialmente naquelas onde o acesso a recursos culturais é restrito. O projeto facilita o empréstimo de livros, mas também desempenha um papel social significativo ao aproximar comunidades de diversos gêneros literários, fortalecendo a inclusão cultural e o desenvolvimento educacional.

6. REFERÊNCIAS

- GITHUB, Inc. *Documentação do GitHub: controle de versão e colaboração*. Disponível em: <https://docs.github.com>. Acesso em: 1 nov. 2024.
- MICROSOFT. *ASP.NET Core MVC overview*. Disponível em: <https://learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-8.0>. Acesso em: 1 nov. 2024.
- MICROSOFT CORPORATION. *Documentação do Visual Studio 2022*. Disponível em: <https://docs.microsoft.com/visualstudio>. Acesso em: 1 nov. 2024.
- MOZILLA DEVELOPER NETWORK. *CSS: cascading style sheets reference*. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>. Acesso em: 1 nov. 2024.
- MOZILLA DEVELOPER NETWORK. *JavaScript reference*. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference>. Acesso em: 1 nov. 2024.
- SQLITE. *SQLite documentation*. Disponível em: <https://www.sqlite.org/docs.html>. Acesso em: 1 nov. 2024.

^{1,2,3,4,5} **Graduando em Análise e Desenvolvimento de Sistemas, Athon – Ensino Superior Sorocaba, Brasil.**

⁶ **Pós-graduado em Engenharia de Software, Professor, Athon – Ensino Superior, Sorocaba, Brasil**

⁷ **Doutorando em Tecnologia Nuclear (IPEN/USP), Professor, Athon – Ensino Superior, Sorocaba, Brasil, jordao@usp.br.**