



UNIVERSIDADE FEDERAL DO OESTE DO PARÁ
INSTITUTO DE ENGENHARIA E GEOCIÊNCIAS
PROGRAMAÇÃO WEB

LAURAH DIAS VASCONCELOS

BOOKLOVER

DOCUMENTO DE VISÃO

VERSÃO 1.0

SANTARÉM – PA

2025

Histórico de Revisões			
Versão	Data	Descrição	Autor
1.0	06/10/2025	Criação do documento.	Laurah.

Sumário

1. Introdução	4
2. Justificativa	4
3. Objetivos do Sistema	5
3.1 Objetivo Geral	5
3.2 Objetivos Específicos	5
4. Escopo do Sistema	5
4.1 Funcionalidades Principais	5
4.2 Funcionalidades Futuras	6
5. Descrição dos Envolvidos	6
6. Requisitos Funcionais	7
7. Requisitos Não Funcionais	8
8. Regras de Negócio	9
9. Considerações Finais	10
10. Diagrama de componentes	10
11. Modelo de dados	11

1. Introdução

O projeto **BookLover** tem como objetivo o desenvolvimento de uma plataforma web moderna e responsiva para gerenciamento pessoal de leituras. Inspirada em plataformas como o Skoob, a aplicação busca oferecer um ambiente intuitivo e colorido para que usuários possam registrar livros lidos, atribuir notas, escrever resenhas, acompanhar seu progresso de leitura e organizar suas listas (lidos, lendo, quero ler).

A aplicação será desenvolvida com **HTML, CSS, JavaScript, PHP e MySQL**, dividida em **front-end e back-end**, garantindo tanto uma interface atrativa para o usuário quanto uma estrutura robusta de autenticação e persistência de dados. O sistema adotará a metodologia **mobile first**, priorizando o uso em dispositivos móveis sem comprometer a experiência em telas maiores.

O projeto visa demonstrar a integração entre front-end e back-end em uma aplicação completa, enfatizando **usabilidade, responsividade e segurança**. Sua arquitetura modular possibilita futuras expansões, como recomendações automáticas, integração com APIs literárias e funcionalidades de socialização entre leitores.

2. Justificativa

A leitura é uma prática fundamental para o desenvolvimento pessoal e intelectual, mas muitos leitores ainda carecem de ferramentas simples e acessíveis para registrar e acompanhar seus hábitos de leitura. Plataformas existentes, embora completas, muitas vezes são complexas ou limitadas em suas versões gratuitas.

O sistema **BookLover** surge como uma solução leve, escalável e acessível, voltada a usuários que desejam **organizar seus livros, notas e resenhas** em um ambiente seguro, personalizável e fácil de usar. Com foco em acessibilidade, design colorido e responsividade, o projeto se propõe a unir **simplicidade e funcionalidade**, tornando o gerenciamento de leituras algo prazeroso e prático.

3. Objetivos do Sistema

3.1 Objetivo Geral

Desenvolver uma aplicação web de **gerenciamento de biblioteca pessoal**, com front-end e back-end integrados, que permita cadastrar livros, avaliar leituras, adicionar resenhas, acompanhar progresso e realizar autenticação de usuários.

3.2 Objetivos Específicos

- Criar páginas dinâmicas e interativas com design responsivo (mobile first).
- Permitir o **cadastro, edição e exclusão (CRUD)** de livros pelo usuário.
- Implementar **registro e login de usuários** com autenticação segura.
- Exibir **informações e estatísticas pessoais** (quantidade de livros lidos, média de notas, etc.).
- Permitir **avaliação por nota e registro de resenhas** para cada livro.
- Implementar **validação de formulários** tanto no front-end quanto no back-end.
- Utilizar **banco de dados MySQL** para persistência de dados.
- Garantir a **segurança e privacidade** das informações do usuário.

4. Escopo do Sistema

4.1 Funcionalidades Principais

- Cadastro e autenticação de usuários (login, logout, recuperação de senha).
- Adição, edição e exclusão de livros (CRUD completo).
- Registro de notas, status de leitura e resenhas pessoais.
- Filtros e busca por título, autor, status ou nota.
- Visualização de estatísticas pessoais no perfil.
- Layout responsivo e interface intuitiva, com paleta colorida.

4.2 Funcionalidades Futuras

- Compartilhamento público de resenhas e perfis.
- Recomendação de livros com base no histórico do usuário.
- Integração com APIs literárias (Google Books, Open Library).
- Modo escuro e personalização de tema.

5. Descrição dos Envolvidos

O sistema **BookLover** foi projetado para atender diferentes perfis de usuários e papéis técnicos:

Usuário / Perfil	Responsabilidade
Leitor (usuário final)	Criar conta, adicionar livros, avaliar e registrar resenhas.

Administrador (futuro)	Gerenciar usuários, revisar cadastros e manter integridade do sistema.
Equipe técnica	Desenvolver, testar, manter e evoluir o sistema.

6. Requisitos Funcionais

ID	Descrição
RF01	O sistema deve permitir o cadastro de usuários com nome, e-mail e senha.
RF02	O sistema deve permitir o login e logout de usuários registrados.
RF03	O sistema deve permitir cadastrar livros com título, autor, ano, editora e capa.
RF04	O sistema deve permitir editar e excluir livros cadastrados.
RF05	O sistema deve permitir atribuir uma nota (1 a 5) e adicionar resenha a cada livro.
RF06	O sistema deve permitir filtrar e pesquisar livros por título, autor, status e nota.
RF07	O sistema deve armazenar os dados em um banco MySQL.
RF08	O sistema deve validar formulários no front-end e no back-end.

RF09	O sistema deve exibir estatísticas básicas no perfil do usuário.
RF10 (futuro)	O sistema poderá exportar os dados da biblioteca em CSV.

7. Requisitos Não Funcionais

ID	Descrição
R	O sistema deve seguir o princípio mobile-first , garantindo usabilidade em qualquer dispositivo.
R	O sistema deve utilizar HTML5, CSS3, JavaScript e PHP .
R	O sistema deve persistir dados em MySQL .
R	O sistema deve apresentar tempo de resposta inferior a 3 segundos em conexões padrão.

R	As senhas devem ser armazenadas de forma criptografada (bcrypt).
R	O layout deve ser acessível, com contraste adequado e design intuitivo.
R	O sistema deve proteger-se contra ataques de injeção SQL e XSS.

8. Regras de Negócio

ID	Descrição
RNG001	Apenas usuários autenticados podem adicionar, editar ou excluir livros.
RNG002	Cada livro deve possuir título e autor obrigatoriamente.

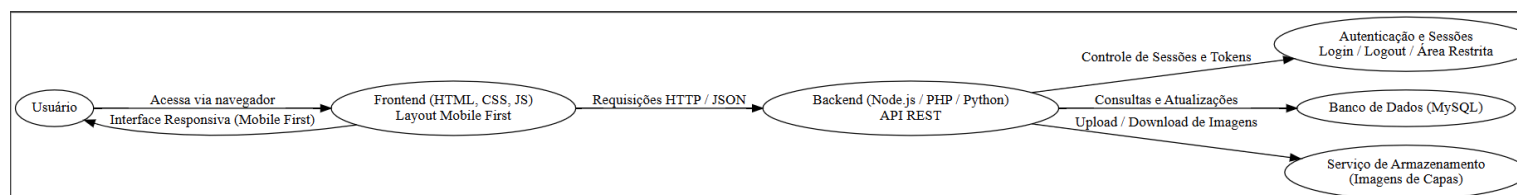
RNG003	As informações dos livros devem ser exibidas em ordem alfabética ou por data de adição.
RNG004	A sessão do usuário deve ser encerrada após o logout.
RNG005 (futuro)	O sistema poderá permitir interação social (curtir, comentar, seguir).
RNG006 (futuro)	O sistema poderá recomendar livros com base em leituras anteriores.

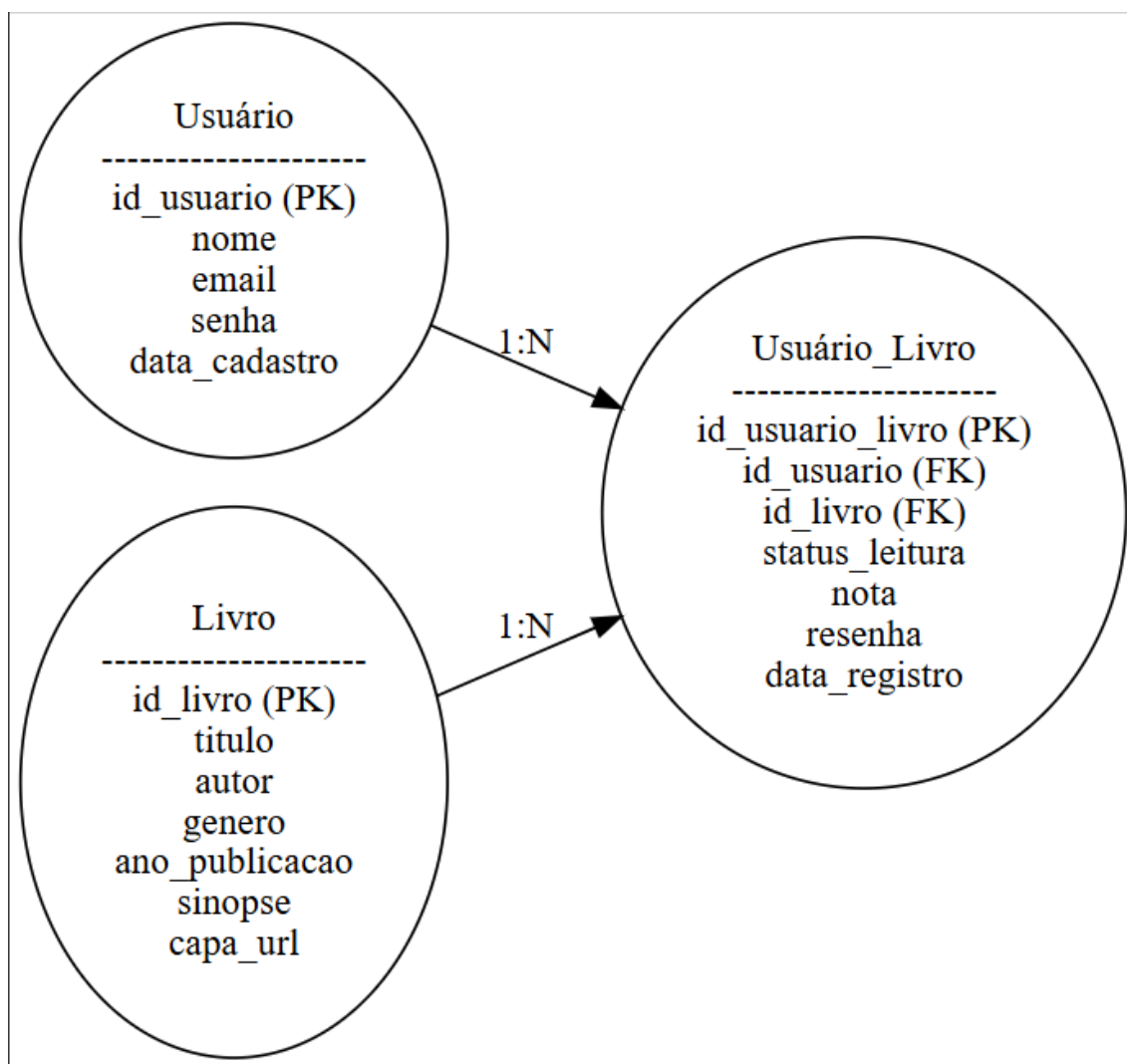
9. Considerações Finais

O projeto **BookLover** visa demonstrar na prática o desenvolvimento de um sistema web completo — com páginas dinâmicas, responsivas, banco de dados, autenticação e CRUD. Sua proposta combina **design moderno, usabilidade, e segurança**, atendendo tanto aos requisitos acadêmicos quanto às boas práticas de engenharia de software.

Esta documentação servirá como base para o **diagrama de componentes** e o **modelo de dados (ER)**, que detalham a estrutura técnica e o relacionamento entre as entidades do sistema.

10. Diagrama de componentes





11. Modelo de dados

O modelo de dados foi desenvolvido para garantir o armazenamento seguro, consistente e escalável das informações dos usuários e dos livros cadastrados no sistema.

O sistema é composto por cinco principais entidades: **Usuário**, **Livro**, **Usuário_Livro**, **Favoritos** e **Notificações**. Cada uma possui papéis distintos dentro da aplicação e se relaciona conforme as ações realizadas pelos usuários no sistema (como adicionar livros, avaliar leituras ou favoritar obras).

Entidade	Descrição	Campos Principais

Usuário	Representa os leitores cadastrados na plataforma. Contém informações básicas de identificação e autenticação.	id_usuario, nome, email, senha, data_cadastro
Livro	Contém os dados de cada livro adicionado à biblioteca, incluindo título, autor, gênero, sinopse e capa.	id_livro, titulo, autor, genero, ano_publicacao, sinopse, capa_url
Usuário_Livro	Relação entre usuários e livros, permitindo registrar o status de leitura, notas e resenhas de cada obra.	id_usuario_livro, id_usuario, id_livro, status_leitura, nota, resenha, data_registro
Favoritos	(Futura expansão) Permite ao usuário marcar livros favoritos.	id_favorito, id_usuario, id_livro, data_favorito
Notificações	(Futura expansão) Permite ao sistema enviar mensagens e alertas personalizados para os usuários.	id_notificacao, id_usuario, mensagem, lida, data_envio

O modelo relacional segue a seguinte estrutura de dependência:

- **Usuário** possui relacionamento **1:N** com **Usuário_Livro**, pois um mesmo usuário pode ter vários livros associados.
- **Livro** também possui relacionamento **1:N** com **Usuário_Livro**, já que um mesmo livro pode estar vinculado a diferentes usuários.
- A tabela **Usuário_Livro** atua como uma **tabela associativa**, registrando as ações e avaliações do usuário sobre os livros.

- As entidades **Favoritos** e **Notificações** estão relacionadas diretamente à tabela **Usuário**, mantendo o vínculo lógico com o comportamento e as interações do leitor.

script do modelo de dados

Criação do Banco de Dados

```
CREATE DATABASE biblioteca_virtual;
```

```
USE biblioteca_virtual;
```

Tabela: usuarios (informações dos leitores cadastrados)

```
CREATE TABLE usuarios (
    id_usuario INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    email VARCHAR(120) NOT NULL UNIQUE,
    senha VARCHAR(255) NOT NULL,
    data_cadastro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Tabela: livros (informações dos livros adicionados)

```
CREATE TABLE livros (
    id_livro INT AUTO_INCREMENT PRIMARY KEY,
    titulo VARCHAR(200) NOT NULL,
    autor VARCHAR(150) NOT NULL,
    genero VARCHAR(100),
    ano_publicacao INT,
    sinopse TEXT,
    capa_url VARCHAR(255),
    data_cadastro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

);

Tabela: usuarios_livros (relaciona os usuários com os livros, registrando status e notas)

```
CREATE TABLE usuarios_livros (

    id_usuario_livro INT AUTO_INCREMENT PRIMARY KEY,

    id_usuario INT NOT NULL,

    id_livro INT NOT NULL,

    status_leitura ENUM('Lendo', 'Concluído', 'Pausado', 'Abandonado')
    DEFAULT 'Lendo',

    nota DECIMAL(3,1) CHECK (nota >= 0 AND nota <= 10),

    resenha TEXT,

    data_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)

        ON DELETE CASCADE

        ON UPDATE CASCADE,

    FOREIGN KEY (id_livro) REFERENCES livros(id_livro)

        ON DELETE CASCADE

        ON UPDATE CASCADE

);
```

Tabela: favoritos (registra livros marcados como favoritos pelos usuários)

```
CREATE TABLE favoritos (

    id_favorito INT AUTO_INCREMENT PRIMARY KEY,

    id_usuario INT NOT NULL,

    id_livro INT NOT NULL,

    data_favorito TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
```

```
FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY (id_livro) REFERENCES livros(id_livro)

ON DELETE CASCADE

ON UPDATE CASCADE

);

Tabela: notificacoes (armazena mensagens e alertas personalizados para
os usuários)

CREATE TABLE notificacoes (

    id_notificacao INT AUTO_INCREMENT PRIMARY KEY,

    id_usuario INT NOT NULL,

    mensagem TEXT NOT NULL,

    lida BOOLEAN DEFAULT FALSE,

    data_envio TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)

    ON DELETE CASCADE

    ON UPDATE CASCADE

);
```