

## 7. DOCUMENTAÇÃO

### 7.1. Levantamento de Requisitos

#### Requisitos Funcionais

##### 1. Cadastro e Gestão de Usuários

- **Cadastro de Usuários:** Permitir que usuários individuais e instituições se registrem na plataforma fornecendo informações como nome, e-mail, senha e tipo de usuário (doador ou instituição).
- **Autenticação e Autorização:** Implementar um sistema de login e logout seguro, com autenticação baseada em JSON Web Tokens (JWT) para sessões de usuário e autorização de acesso a funcionalidades específicas.
- **Gerenciamento de Perfis:** Permitir que usuários atualizem suas informações pessoais e preferências, e que instituições possam gerenciar seus dados de contato e informações de doação.

##### 2. Gestão de Doações

- **Cadastro de Livros:** Permitir que os doadores registrem livros para doação, fornecendo informações como título, autor, categoria, estado de conservação e descrição adicional.
- **Solicitações de Doação:** Permitir que instituições façam solicitações de livros, especificando suas necessidades e categorias desejadas.
- **Busca e Filtros:** Implementar uma funcionalidade de busca que permita aos usuários encontrar livros com base em critérios como título, autor, categoria e localização. Fornecer filtros para refinar os resultados.

##### 3. Administração do Sistema

- **Painel de Administração:** Fornecer um painel de administração para gerenciar usuários, livros e solicitações de doação. Os administradores devem poder revisar e aprovar ou rejeitar doações e solicitações.

## Requisitos Não Funcionais

### 1. Desempenho

- **Tempo de Resposta:** Garantir que as páginas e funcionalidades do site carreguem rapidamente, com tempos de resposta aceitáveis para garantir uma experiência de usuário fluida.

### 2. Usabilidade

- **Interface Amigável:** Desenvolver uma interface intuitiva e fácil de usar, com navegação clara e acessível para todos os tipos de usuários, incluindo aqueles com pouca familiaridade com tecnologia.
- **Acessibilidade:** Garantir que o site seja acessível a pessoas com deficiência, seguindo diretrizes de acessibilidade web, como WCAG (Web Content Accessibility Guidelines).

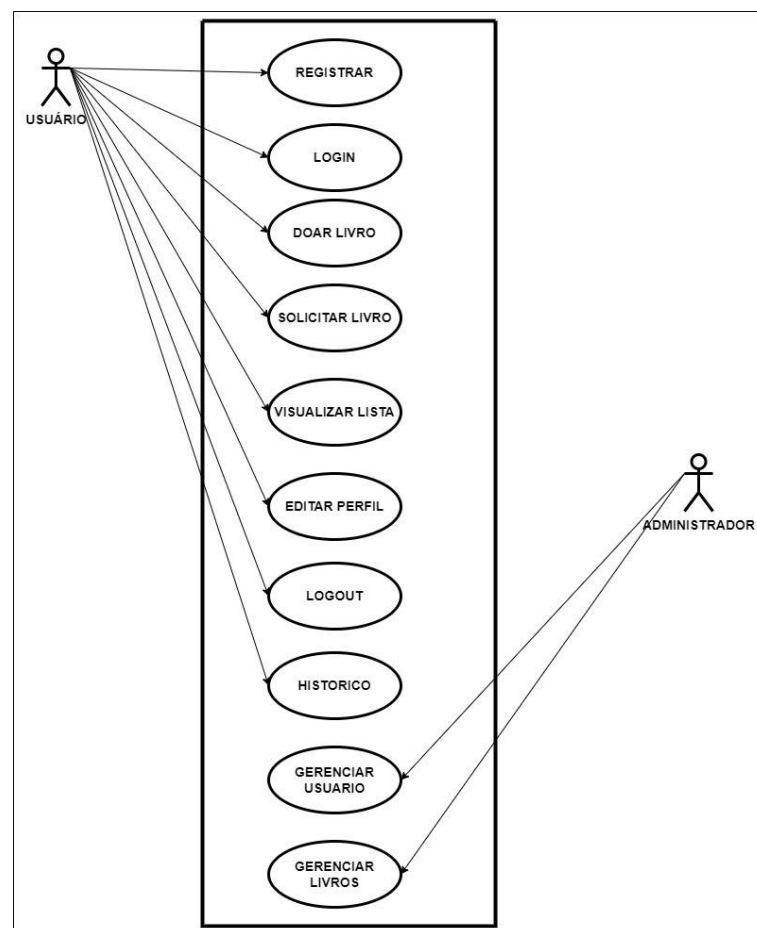
### 3. Manutenibilidade

- **Código Bem Documentado:** O código fonte deve ser bem documentado e organizado para facilitar a manutenção e futuras atualizações.
- **Facilidade de Atualizações:** O sistema deve permitir a implementação de novas funcionalidades e melhorias com o mínimo de impacto nas operações existentes.

### 4. Compatibilidade

- **Browsers e Dispositivos:** Garantir que o site funcione corretamente em diferentes navegadores (Chrome, Firefox, Safari, etc.) e em diversos dispositivos, incluindo desktops, tablets e smartphones.

## Diagrama de caso de uso



**Figura 1:** Diagrama de caso de uso.

Esses requisitos fornecerão uma base sólida para o desenvolvimento do site de doação de livros, ajudando a garantir que o sistema atenda às expectativas dos usuários e às necessidades da comunidade, enquanto mantém padrões elevados de segurança e desempenho.

### 7.2. Arquitetura do sistema

A arquitetura do sistema para o site de doação de livros será projetada para garantir um desempenho eficiente, escalabilidade e segurança. A arquitetura será baseada em uma estrutura moderna de desenvolvimento de software, utilizando uma abordagem cliente-servidor e seguindo os princípios de design de sistemas distribuídos.

Diagrama de arquitetura

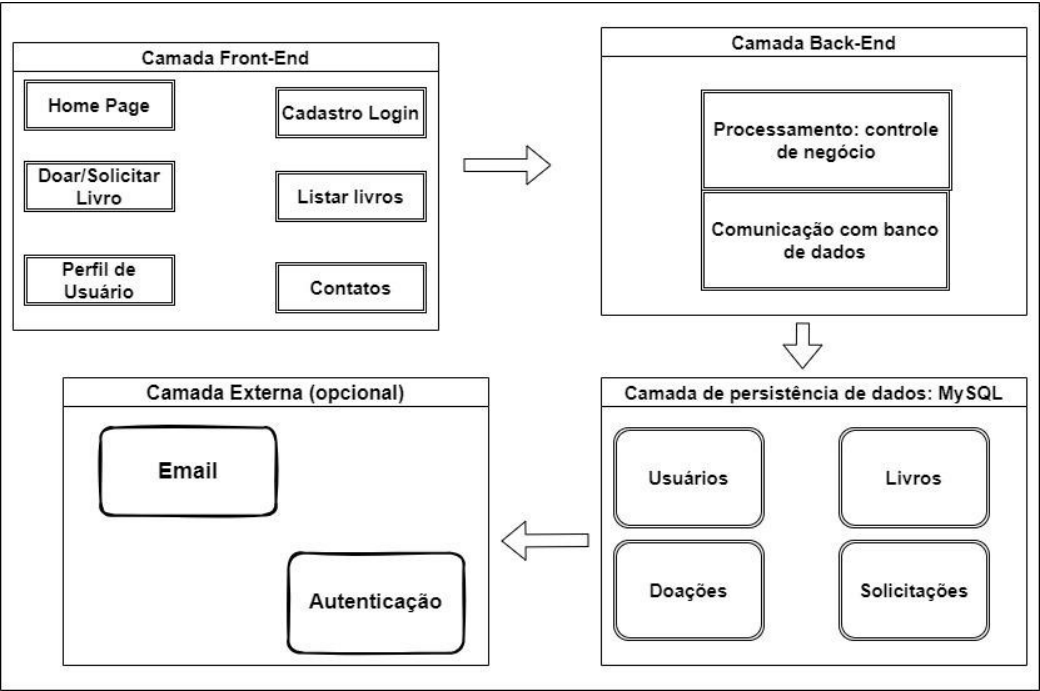


Figura 2: Diagrama de arquitetura do sistema.

Diagrama de classes

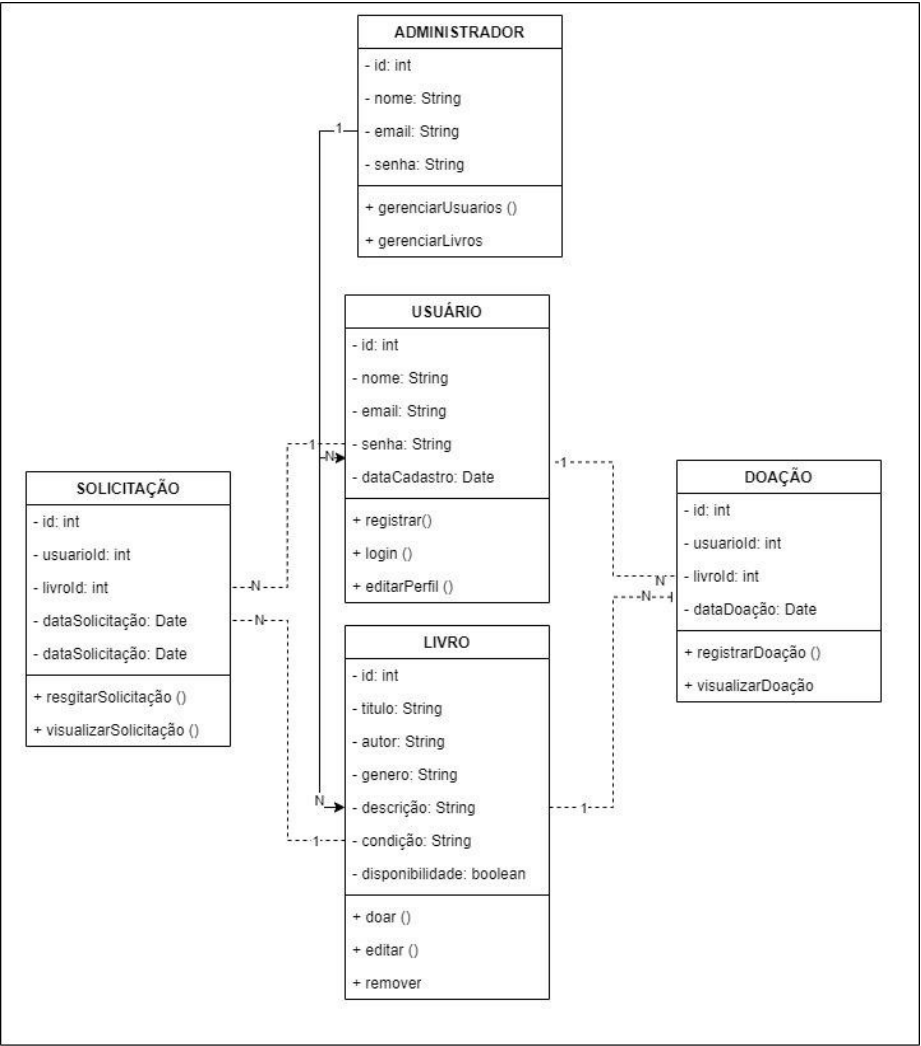


Figura 3: Diagrama de classes.

Camada de Apresentação (Frontend)

A camada de apresentação é responsável pela interface do usuário e pela interação com o sistema. Será desenvolvida utilizando **HTML**, **CSS** e **JavaScript**, com **Bootstrap** para estilização e **React** para a construção de uma interface de usuário dinâmica e interativa.

- **HTML/CSS:** Definirá a estrutura e o estilo das páginas web, garantindo que o site seja visualmente atraente e responsivo.
- **JavaScript:** Adicionará interatividade à interface, permitindo que os usuários interajam com os componentes do site de forma dinâmica.
- **Bootstrap:** Fornecerá um framework CSS para design responsivo, facilitando a construção de uma interface que se adapte a diferentes dispositivos e tamanhos de tela.
- **React:** Será utilizado para criar componentes de UI reutilizáveis e gerenciar o estado da aplicação de forma eficiente, oferecendo uma experiência de usuário fluida e responsiva.

### **Camada de Aplicação (Backend)**

A camada de aplicação é responsável pelo processamento das lógicas de negócios e pela gestão das interações entre o frontend e o banco de dados. Será desenvolvida utilizando **PHP** e o framework **Laravel**.

- **PHP:** Linguagem de programação do lado do servidor que será utilizada para processar requisições e gerar respostas.
- **Laravel:** Framework PHP que fornecerá uma estrutura organizada para o desenvolvimento backend, incluindo roteamento, controle de requisições, autenticação, e gestão de banco de dados. Laravel também facilitará a implementação de APIs RESTful para comunicação com o frontend.

### **Camada de Persistência (Banco de Dados)**

A camada de persistência é responsável pelo armazenamento e recuperação de dados. Utilizaremos **MySQL** como sistema de gerenciamento de banco de dados.

- **MySQL:** Sistema de banco de dados relacional que armazenará informações sobre usuários, livros, doações e solicitações. Será estruturado para suportar consultas eficientes e garantir a integridade dos dados.

## Comunicação entre Frontend e Backend

A comunicação entre o frontend e o backend será realizada através de **APIs RESTful**. As APIs fornecerão endpoints para operações como cadastro de usuários, registro de livros, e gestão de doações.

- **RESTful APIs:** Interfaces que permitirão a troca de dados entre o frontend e o backend, utilizando formatos como JSON para facilitar a comunicação e a interoperabilidade.

## 6. Infraestrutura e Deploy

A infraestrutura do site será projetada para garantir a disponibilidade e a escalabilidade do sistema.

- **Servidor Web:** Será configurado um servidor web para hospedar o backend e o frontend. **Apache** pode ser utilizado para gerenciar as requisições HTTP e servir o conteúdo do site.
- **Ambiente de Hospedagem:** O site será hospedado na plataforma do Github utilizando o serviço do Github pages para disponibilizar o site.

Esta arquitetura proporcionará uma base sólida para o desenvolvimento do site de doação de livros, garantindo uma operação eficiente, segura e escalável. Ela permitirá a criação de uma plataforma que atenda às necessidades dos usuários e se adapte ao crescimento futuro.

## 7.3. Design

### Design do Frontend

- ➔ **Frontend** é a camada visível do site e será construída usando HTML, CSS, JavaScript, Bootstrap e React.
- ➔ **Estrutura e Layout:** A estrutura básica do site será criada com HTML, definindo o esqueleto das páginas, como a página inicial, páginas de perfil, páginas de listagem de livros e formulários de doação. A estrutura será organizada em um layout responsivo que se adapta a diferentes tamanhos de tela.
- ➔ **Estilização:** CSS será utilizado para aplicar estilos às páginas, garantindo uma aparência atraente e coesa. **Bootstrap** será usado para estilização e design responsivo,

fornecendo uma base sólida para componentes de UI como botões, formulários e modais.

➔ **Interatividade: JavaScript e React** serão usados para adicionar interatividade. React permitirá a criação de componentes reutilizáveis e a gestão eficiente do estado da aplicação, enquanto JavaScript fornecerá a lógica para interação com a API e manipulação dinâmica do DOM.

➔ **Componentes:** O design do frontend incluirá componentes como:

◆ **Barra de Navegação:** Um menu para navegação entre páginas.

◆ **Formulários de Cadastro e Login:** Para permitir que usuários se registrem e façam login.

◆ **Listagem de Livros:** Componentes para exibir livros disponíveis para doação e solicitações.

◆ **Formulários de Doação:** Para permitir que doadores registrem livros.

## **Paleta de cores**

### **Cor Primária (Azul Suave): #4A90E2**

➔ Transmite confiança e tranquilidade, sendo agradável à vista e ideal para botões e cabeçalhos.

### **Cor Secundária (Verde Claro): #A4D65E**

➔ Representa renovação e sustentabilidade, adequada para destacar informações importantes como chamadas de ação (botões de "Doar" ou "Receber").

### **Cor de Fundo (Cinza Claro): #F5F5F5**

➔ Um fundo claro e neutro para deixar o site limpo e não sobrecarregar a interface visual.

### **Cor de Destaque (Amarelo Suave): #FDD835**



- ➔ Pode ser usada para destacar detalhes ou pequenos elementos importantes, como ícones ou links de destaque.
- ➔ **Cor para Texto (Cinza Escuro): #4F4F4F**
- ➔ Facilita a leitura, com um contraste suave, ideal para parágrafos e textos de descrição.



Figura 4: Paleta de cores.

### Tipografia:

- ➔ **Fonte Principal (Títulos e Cabeçalhos): Montserrat (Sans-serif)**
- ➔ **Fonte Secundária (Texto e Parágrafos): Roboto (Sans-serif)**
- ➔ **Fonte para Destaques (Botões, Chamadas de Ação): Poppins (Sans-serif)**

### Hierarquia de texto:

- ➔ **H1 (Título principal):** Montserrat, 36px, bold.
- ➔ **H2 (Subtítulos):** Montserrat, 28px, semi-bold.
- ➔ **Parágrafos:** Roboto, 16px, regular.
- ➔ **Botões:** Poppins, 18px, semi-bold.

### Design do Backend

**Backend** é responsável pelo processamento de dados e lógica do servidor, e será desenvolvido com PHP e o framework Laravel.

- **Arquitetura de Aplicação:** A aplicação backend será estruturada utilizando a arquitetura MVC (Model-View-Controller) proporcionada pelo Laravel. Isso permitirá uma separação clara entre a lógica de negócios (Controller), a estrutura de dados (Model) e a apresentação (View).
- **Endpoints da API:** O backend fornecerá APIs RESTful para permitir a comunicação entre o frontend e o banco de dados. Exemplos de endpoints incluem:
- **Gerenciamento de Requisições:** Laravel gerenciará as requisições, processando dados enviados pelo frontend e retornando respostas apropriadas. Utilizará middleware para autenticação e segurança.
- **Segurança:** Implementação de autenticação JWT para proteger endpoints sensíveis e garantir que apenas usuários autorizados possam acessar certas funcionalidades. Laravel fornecerá funcionalidades de criptografia e proteção contra ataques comuns.

#### 4. Design do Banco de Dados

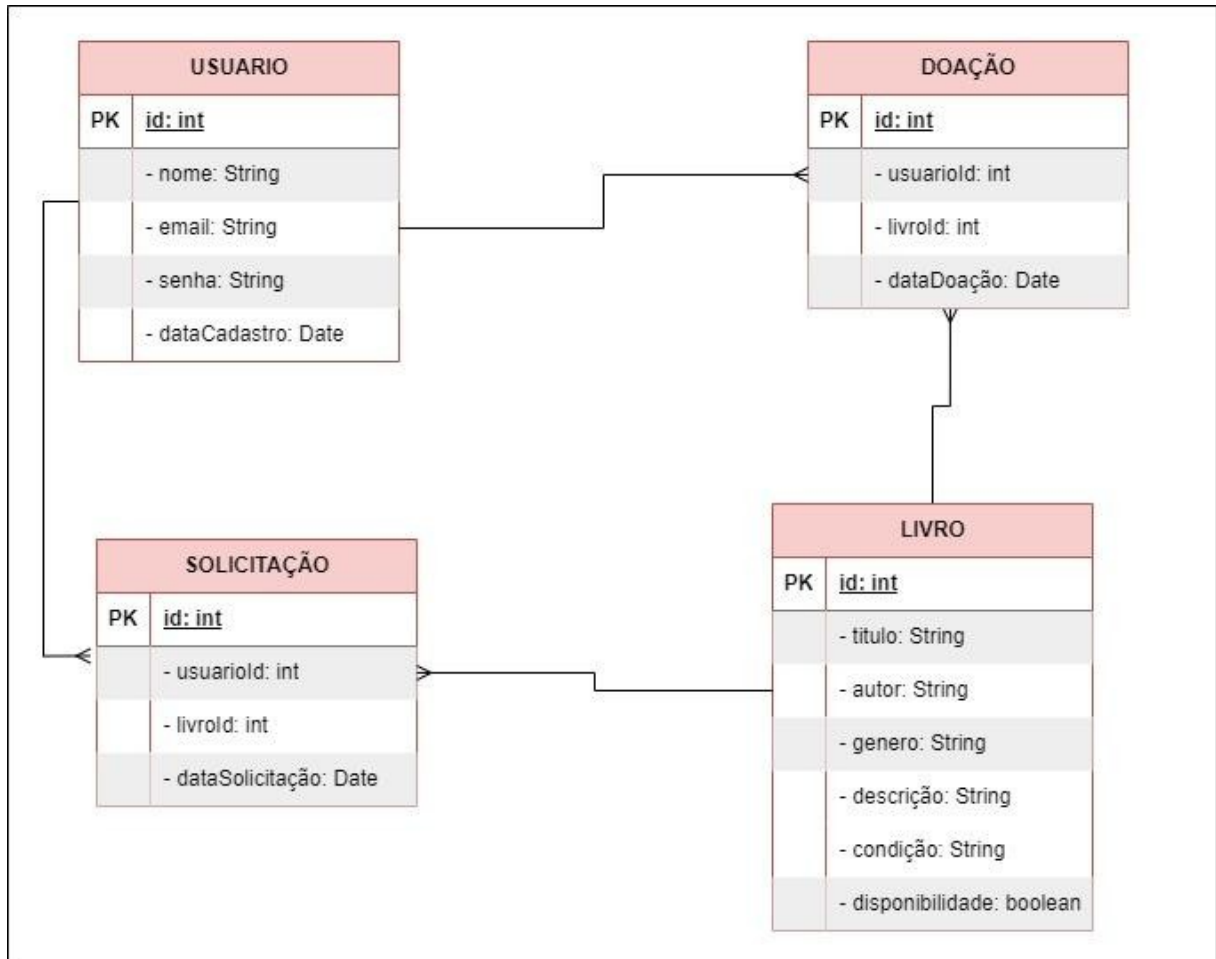
**Banco de Dados** será implementado usando MySQL, e a estrutura de dados será projetada para suportar as operações necessárias para o site.

➔ **Esquema do Banco de Dados:** O esquema incluirá tabelas para armazenar informações sobre usuários, livros, doações e solicitações. A seguir, estão alguns exemplos de tabelas:

- ◆ **Usuário:** Armazena informações de usuários, como ID, nome, e-mail, senha (hash) e tipo de usuário.
- ◆ **Livros:** Armazena informações sobre livros, como ID, título, autor, categoria, estado e descrição.
- ◆ **Doação:** Registra as doações de livros, incluindo informações sobre o doador, o livro e a instituição receptora.
- ◆ **Solicitação:** Armazena solicitações feitas por instituições, incluindo detalhes sobre os livros solicitados e as necessidades da instituição.

➔ **Relacionamentos:** As tabelas serão inter-relacionadas para garantir a integridade dos dados.

### Diagrama de Entidade-Relacionamento (ER)



**Figura 5:** Diagrama ER.

### Relacionamentos entre Tabelas

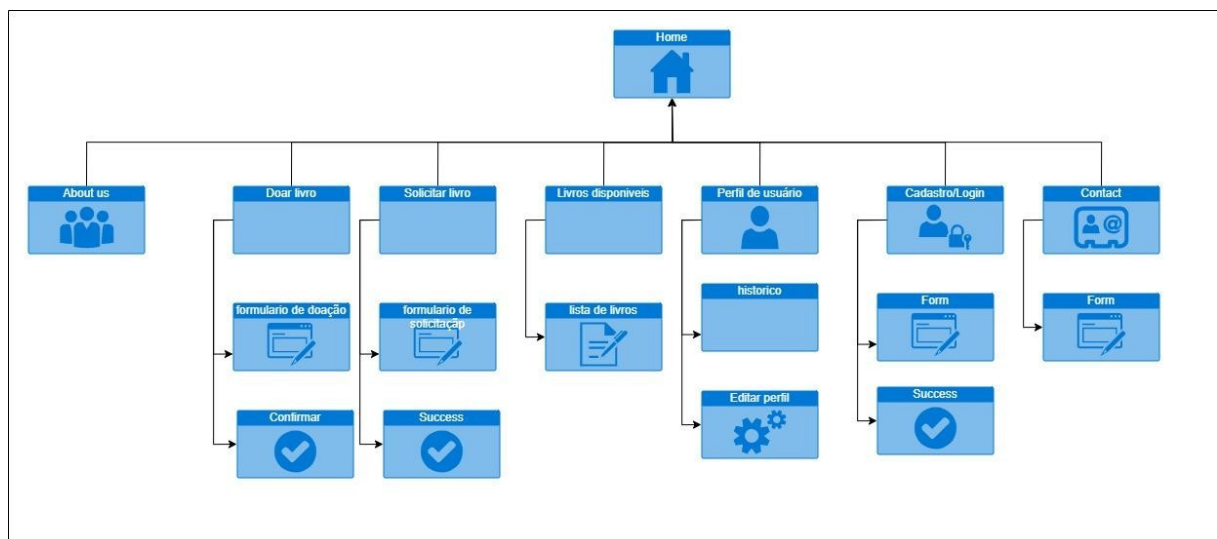
Os relacionamentos entre as tabelas são definidos por chaves estrangeiras que garantem a integridade referencial:

- **Usuários e Doações:** Um usuário pode fazer várias doações, mas cada doação é associada a um único usuário. Relacionamento 1 entre users e donations.

- **Livros e Doações:** Um livro pode ser doado várias vezes, mas cada doação refere-se a um único livro. Relacionamento 1 entre books e donations.
- **Usuários e Solicitações:** Uma instituição pode fazer várias solicitações, mas cada solicitação é associada a uma única instituição. Relacionamento 1 entre users e requests.

### 7.3. Estrutura do Projeto

#### Mapa do site



**Figura 6:** Mapa do site.

#### 1. Página Inicial

**Banner Rotativo:** Imagens destacando a importância da doação de livros.

**Breve Descrição:** O que é o projeto e como funciona.

**Botões de Ação:** Links para "Doar Livros", "Solicitar Livros" e "Fazer Parte da Comunidade".

## **2. Sobre Nós**

**Missão e Visão:** Explicação dos objetivos do projeto.

**Equipe:** Informações sobre os membros da equipe, se aplicável.

**História:** Breve histórico da iniciativa.

## **3. Doar Livros**

**Formulário de Doação:** Campos para preenchimento (nome, e-mail, título do livro, autor, condição do livro, etc.).

**Instruções de Doação:** Como e onde entregar os livros.

**Benefícios da Doação:** Informações sobre como a doação impacta a comunidade.

## **4. Solicitar Livros**

**Formulário de Solicitação:** Campos para preenchimento (nome, e-mail, título do livro desejado, categoria, etc.).

**Critérios de Solicitação:** Requisitos para solicitar livros (ex.: pertencente a uma instituição educacional, etc.).

## **5. Contato**

**Formulário de Contato:** Para dúvidas e sugestões.

**Informações de Contato:** Endereço de e-mail, telefone, redes sociais.

## **6. Área do Usuário**

**Cadastro/Login:** Para doadores e receptores.

**Perfil do Usuário:** Gerenciamento de informações pessoais e histórico de doações/solicitações.

**Notificações:** Atualizações sobre novas doações ou solicitações.

## **7. Rodapé**

**Links Rápidos:** Acesso rápido a páginas importantes.

**Redes Sociais:** Ícones para seguir o projeto em plataformas sociais.

**Copyright:** Informação sobre direitos autorais.

### **Considerações Técnicas**

**Design Responsivo:** Garantir que o site funcione bem em dispositivos móveis e desktops.

**Acessibilidade:** Implementar práticas de acessibilidade para usuários com deficiência.

**SEO:** Estruturar o site com boas práticas de SEO para melhor posicionamento nos motores de busca.

### **Documentação do Código**

- ➔ **Comentários:** Incluir comentários claros e concisos no código para explicar a lógica e a função de partes importantes do código.
- ➔ **README.md:** O arquivo README deve fornecer uma visão geral do projeto, instruções de configuração, e orientações sobre como contribuir para o desenvolvimento do frontend.

# 7.4. Protótipo

## Wireframes

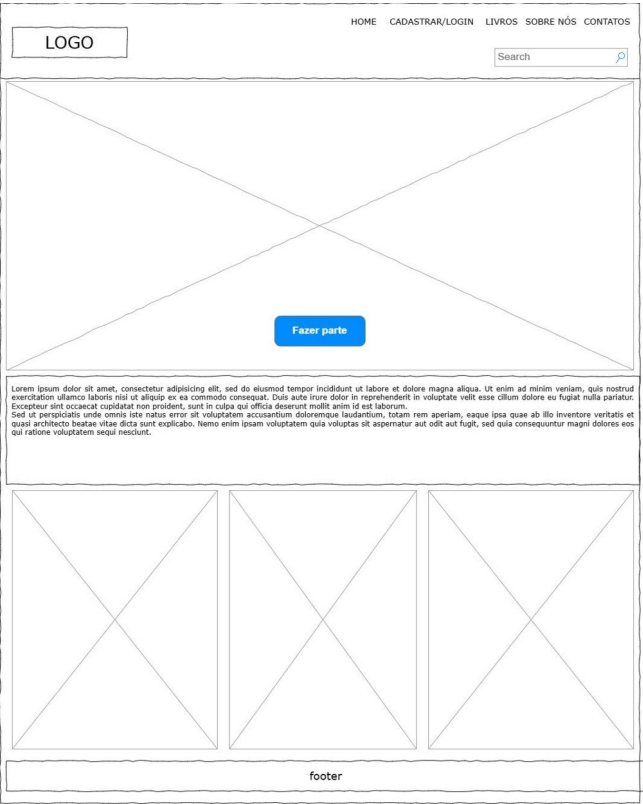


Figura 7: Wireframe Home.

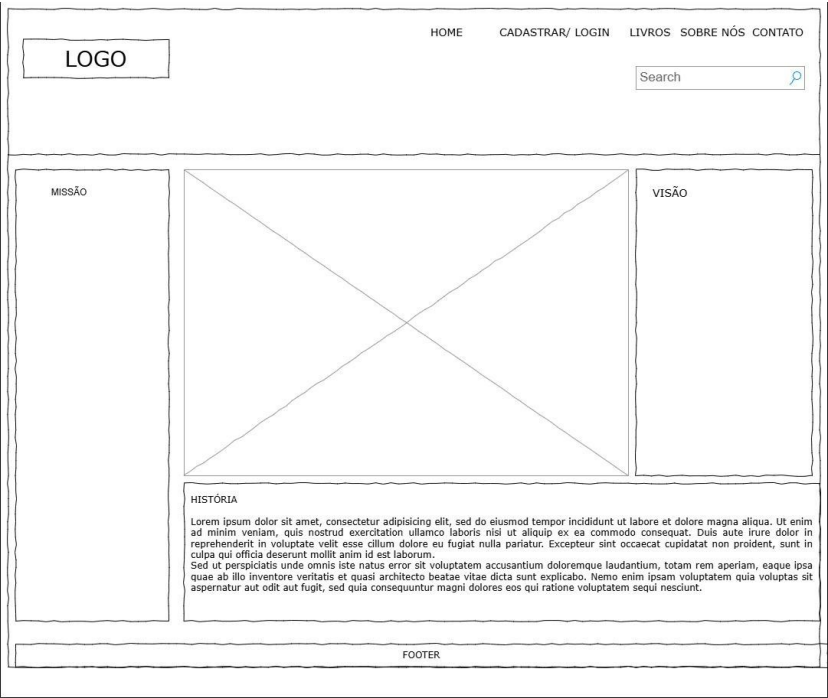


Figura 8: Wireframe da pagina sobre nós.

LOGO

HOME CADASTRAR/LOGIN LIVROS SOBRE NÓS CONTATOS

Search

Login

Nome

Email

Senha

Tipo

SIGN IN

Esqueceu a senha?

Cadastrar

SIGN UP

footer

**Figura 9:** Wireframe do formulário.

LOGO

HOME CADASTRAR/ LOGIN LIVROS SOBRE NÓS CONTATO

Search

SEU NOME

SEU EMAIL

ASSUNTO

MENSAGEM

ENVIAR

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute inure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

FOOTER

**Figura 10:** Wireframe da pagina de contato.



Prototipo de Telas

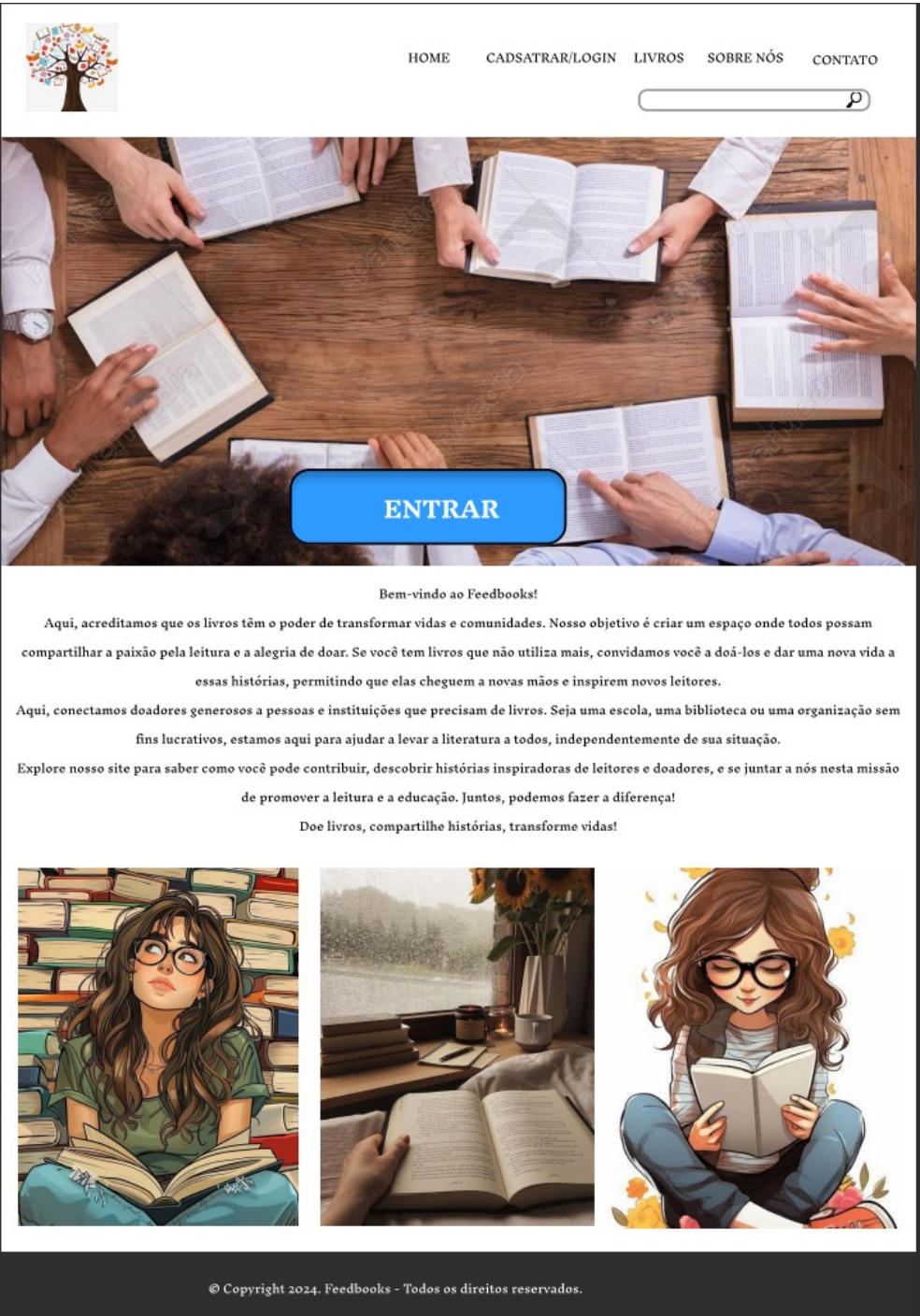



Figura 11: Página inicial.



HOME   CADSATRAR/LOGIN   LIVROS   SOBRE NÓS   CONTATO


Nome

Email

Assunto

Mensagem

Enviar



Endereço

Perini Business Park


Rua Dona Francisca, 8300. Ágora Tech Park | Sala 307.


Bairro: Zona industrial | Joinville/SC | CEP: 89.219-600


contato@omunga.com


Telefone


+55 (47) 3305-6716












© Copyright 2024. Feedbooks - Todos os direitos reservados.

**Figura 12:** Página de contato.



HOME   CADSATRAR/LOGIN   LIVROS   SOBRE NÓS   CONTATO

LOGIN

NOME

EMAIL

SENHA

TIPO

Cadastrar


SIGN IN

SIGN UP

[Esqueceu a senha?](#)

"Cada livro doado é uma oportunidade para alguém descobrir novos mundos."

"Transforme sua estante em uma ponte para o conhecimento!"



© Copyright 2024. Feedbooks - Todos os direitos reservados.

Figura 13: Página de cadastro e login.

## 7.5 Configuração do Ambiente de Desenvolvimento

### Hardware

- **Computador de Desenvolvimento:** O ambiente será configurado em um computador com especificações adequadas para suportar o desenvolvimento de software. Idealmente, deve ter um processador moderno (Intel i5 ou equivalente), 8 GB de RAM (mínimo recomendado), e pelo menos 100 GB de espaço livre em disco para armazenar o código, bibliotecas e ferramentas.
- **Dispositivos de Teste:** Testar a aplicação em diferentes dispositivos, como smartphones e tablets, para garantir que o site seja responsivo e funcione corretamente em várias plataformas.

### Software

- **Sistema Operacional:** O desenvolvimento pode ser realizado em diferentes sistemas operacionais, como Windows, macOS ou Linux. A escolha do sistema operacional pode influenciar as ferramentas e configurações utilizadas.
- **IDE e Editores de Código:** Utilizar um ambiente de desenvolvimento integrado (IDE) ou editores de código que suportem PHP, Laravel, HTML, CSS, JavaScript e outras tecnologias. Exemplos incluem:
  - **Visual Studio Code:** Popular editor de código com suporte para diversas linguagens e extensões.
  - **PHPStorm:** IDE específica para PHP e Laravel, oferecendo ferramentas avançadas para desenvolvimento e depuração.
- **Servidores Locais:** Utilizar servidores locais para desenvolvimento e teste. Ferramentas como:
  - **XAMPP:** Pacotes que fornecem um ambiente de servidor Apache, MySQL e PHP para desenvolvimento local.

- **Navegadores Web:** Testar o site em vários navegadores, como Google Chrome, Mozilla Firefox, Safari e Microsoft Edge, para garantir compatibilidade e consistência na exibição.

## **Prototipagem e Modelagem de Dados**

**Draw.io:** ferramenta de diagramas online gratuita.

O **Figma:** ferramenta para design de interfaces (UI) e prototipagem de experiências de usuário (UX). Ele oferece recursos colaborativos em tempo real e é amplamente usado para o design de sites e aplicativos.

## **Banco de Dados:**

- ➔ **MySQL:** Para gerenciar as informações sobre os livros, usuários e doações.
  - ◆ Usar o **phpMyAdmin** para uma interface gráfica amigável de gerenciamento do banco de dados.
- ➔ **Microsoft Workbench:** excelente ferramenta gráfica para trabalhar com bancos de dados MySQL. Ele permite criar e gerenciar esquemas, realizar consultas SQL, visualizar dados e gerar diagramas ER (Entity-Relationship). É útil tanto no desenvolvimento quanto na manutenção do banco de dados.

## **Controle de Versão:**

- ➔ **Git:** Para controle de versão e colaboração.
  - ◆ Usar o GitHub ou GitLab para armazenar o código e facilitar o versionamento e backup.
- ➔ **GitHub Desktop:** Para o trabalho colaborativo entre os membros da equipe.