



**Bootcamp Atlântico Avanti**  
**Curso de Desenvolvimento Full Stack Básico**

**Equipe PageTurners**

Kendriks da Paixão  
Leticia de Miranda Puga  
Maria Alexsandra Tomaz  
Matheus Harlen Saraiva de Nojoza  
Rafael Euclides dos Santos  
Rayssa Silva Santana

**“Conectando Leitores para Troca de Livros”**

**Requisitos**

**Fortaleza-CE**  
**2024**

# 1. Introdução

Este documento descreve os requisitos para o desenvolvimento de uma plataforma digital destinada a facilitar a troca de livros entre leitores. Os requisitos funcionais incluem o cadastro de usuários, gestão de livros, realização de trocas e empréstimos, busca por livros, sistema de avaliação e segurança dos dados. Os requisitos não funcionais garantem a compatibilidade com diversas plataformas, uso de tecnologias atuais, manutenção regular, otimização para qualquer computador e boa acessibilidade na interface. A plataforma visa promover um ambiente confiável e engajado para os amantes da leitura.

## 2. Requisitos Funcionais

**RF 1:** O sistema deve permitir o cadastro de usuários, incluindo os seguintes campos:

- Nome;
- Email;
- Endereço;
- CPF;
- Senha;
- Avaliação do usuário por parte de outros usuários.

**RF 2:** O sistema deve permitir a gestão de livros, incluindo:

- Cadastro de livros com os campos: nome, autor, data de publicação, descrição, estado de conservação, disponibilidade de troca e disponibilidade de empréstimo.

**RF 3:** O sistema deve possibilitar a realização de trocas e empréstimos de livros, incluindo:

- Solicitação de trocas;
- Solicitação de empréstimo;
- Sistema de aceitação ou recusa de trocas;
- Histórico de trocas do usuário;
- Formas de envio.

**RF 4:** O sistema deve oferecer uma funcionalidade de busca por livros, permitindo:

- Busca por título, autor e gênero;
- Filtragem por estado do livro, disponibilidade para troca e disponibilidade para empréstimo.

**RF 5:** O sistema deve incluir um sistema de avaliação de usuários e empréstimos, incluindo:

- Avaliação de usuários após a troca;
- Sistema de pontos para incentivar trocas e melhorar a reputação do perfil dentro da plataforma;
- Avaliação de empréstimos por ambas as partes (condições de entrega e devolução, logística).

**RF 6:** O sistema deve garantir a segurança dos dados, incluindo:

- Autenticação de usuários;
- Verificação de segurança para confirmar dados do usuário;
- Recuperação de email e senha;
- Sistema de nome de usuários.

### 3. Requisitos Não Funcionais

**RNF 1:** O sistema deverá ser compatível com o maior número de plataformas de sistemas operacionais.

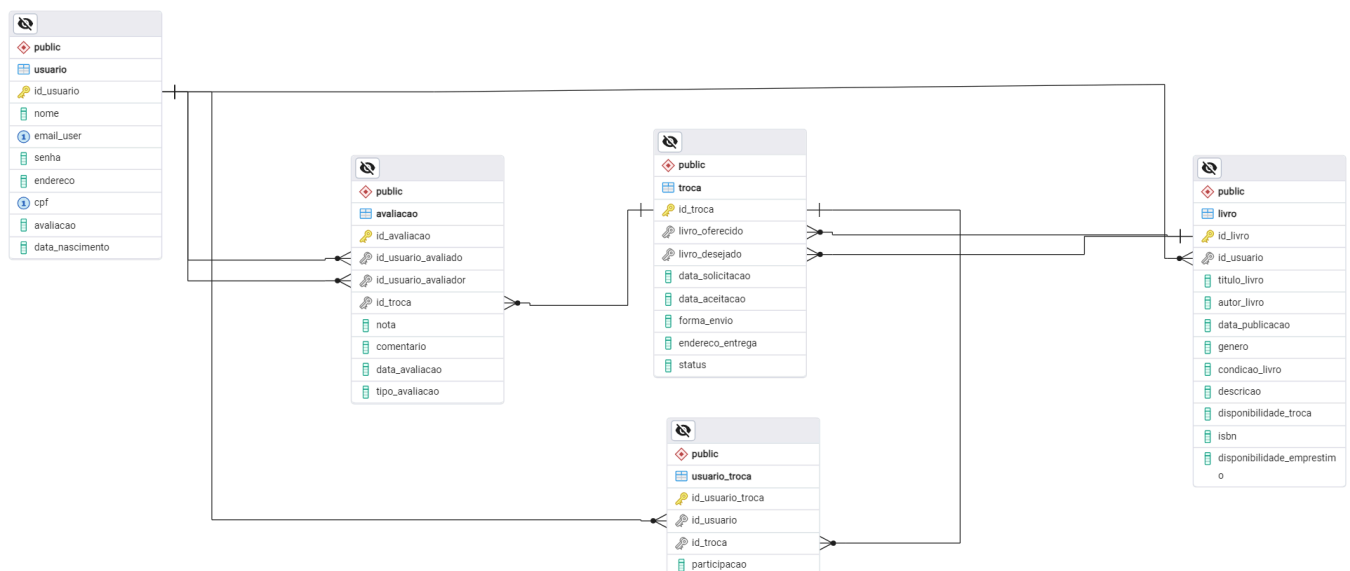
**RNF 2:** O sistema deverá utilizar tecnologias compatíveis com os padrões atuais da computação.

**RNF 3:** O sistema deverá ser atualizado e mantido regularmente.

**RNF 4:** O sistema deverá ser otimizado para funcionar em qualquer computador.

**RNF 5:** O sistema deverá oferecer uma interface com boa acessibilidade.

### Diagrama Entidade-Relacionamento (ER):



## Queries SQL:

```
CREATE DATABASE PageTurners;
```

```
CREATE TABLE Usuario (  
    id_usuario SERIAL PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    email_user VARCHAR(100) NOT NULL UNIQUE,  
    senha VARCHAR(255) NOT NULL,  
    endereco VARCHAR(255),  
    cpf VARCHAR(11) NOT NULL UNIQUE,  
    avaliacao DECIMAL(3, 2) DEFAULT 0,  
    data_nascimento DATE CHECK (data_nascimento <= CURRENT_DATE -  
INTERVAL '16 years')  
);
```

```
CREATE TABLE Livro (  
    id_livro SERIAL PRIMARY KEY,  
    id_usuario INT,  
    titulo_livro VARCHAR(255) NOT NULL,  
    autor_livro VARCHAR(255) NOT NULL,  
    data_publicacao DATE,  
    genero VARCHAR(100),  
    condicao_livro VARCHAR(50),  
    descricao TEXT,  
    disponibilidade_troca BOOLEAN DEFAULT TRUE,  
    isbn VARCHAR(20),  
    disponibilidade_emprestimo BOOLEAN DEFAULT FALSE,  
    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario)  
);
```

```
CREATE TABLE Troca (  
    id_troca SERIAL PRIMARY KEY,  
    livro_oferecido INT,  
    livro_desejado INT,  
    data_solicitacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    data_aceitacao TIMESTAMP,  
    forma_envio VARCHAR(100),  
    endereco_entrega VARCHAR(255),  
    status VARCHAR(50) DEFAULT 'solicitada' CHECK (status IN  
('solicitada', 'aceita', 'separacao', 'enviada', 'concluida')),  
    FOREIGN KEY (livro_oferecido) REFERENCES Livro(id_livro),  
    FOREIGN KEY (livro_desejado) REFERENCES Livro(id_livro)  
);
```

```

CREATE TABLE Usuario_Troca (
    id_usuario_troca SERIAL PRIMARY KEY,
    id_usuario INT,
    id_troca INT,
    participacao VARCHAR(50) NOT NULL CHECK (participacao IN
('solicitante', 'aceitador')),
    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario),
    FOREIGN KEY (id_troca) REFERENCES Troca(id_troca)
);

CREATE TABLE Avaliacao (
    id_avaliacao SERIAL PRIMARY KEY,
    id_usuario_avaliado INT,
    id_usuario_avaliador INT,
    id_troca INT,
    nota INT CHECK (nota >= 0 AND nota <= 5),
    comentario TEXT,
    data_avaliacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    tipo_avaliacao VARCHAR(50),
    FOREIGN KEY (id_usuario_avaliado) REFERENCES Usuario(id_usuario),
    FOREIGN KEY (id_usuario_avaliador) REFERENCES Usuario(id_usuario),
    FOREIGN KEY (id_troca) REFERENCES Troca(id_troca)
);

-- Alguns indices de indexação para melhoria na busca
CREATE INDEX idx_titulo_livro ON Livro(titulo_livro);
CREATE INDEX idx_autor_livro ON Livro(autor_livro);
CREATE INDEX idx_isbn ON Livro(isbn);
CREATE INDEX idx_genero ON Livro(genero);
CREATE INDEX idx_email_user ON Usuario(email_user);
CREATE INDEX idx_livro_oferecido ON Troca(livro_oferecido);
CREATE INDEX idx_livro_desejado ON Troca(livro_desejado);
CREATE INDEX idx_usuario_avaliado ON Avaliacao(id_usuario_avaliado);
CREATE INDEX idx_usuario_avaliador ON Avaliacao(id_usuario_avaliador);

```