

## **Minimundo**

O projeto consiste na criação de um site destinado à troca de livros entre usuários. A plataforma oferece um ambiente onde leitores podem cadastrar seus livros, buscar títulos desejados, solicitar trocas e avaliar outros usuários após as trocas serem concluídas.

## **Cenário Geral**

A plataforma de trocas de livros é composta por diversas abas que permitem a interação entre os usuários e a gestão de livros e trocas. Os principais atores são os usuários cadastrados, que podem ser tanto solicitantes quanto solicitados em uma troca.

## **Usuários**

Cadastro e Login:

- Os usuários devem se cadastrar na plataforma fornecendo informações como nome, email, senha (que será criptografada para armazenamento seguro), endereço (rua, número, bairro, cidade, estado, CEP), descrição pessoal e localização.

- Após o cadastro, os usuários podem efetuar login utilizando seu email e senha.

- Caso esqueçam a senha, os usuários podem recuperá-la informando o email cadastrado.

Perfil:

- Usuários podem atualizar seus dados pessoais a qualquer momento, incluindo nome, senha, descrição e localização.

- Cada usuário possui uma reputação associada, que é avaliada com base nas trocas realizadas.

## **Livros**

### **Cadastro de Livros:**

- Usuários podem cadastrar livros disponíveis para troca, informando detalhes como título, autor, gênero, estado de conservação, descrição, data de publicação, número da edição e disponibilidade.

- Os livros cadastrados são listados na página principal do usuário e podem ser visualizados por outros usuários.

### **Busca de Livros:**

- Os usuários podem buscar livros disponíveis para troca utilizando filtros como título, autor, gênero e localização.

- Os resultados da busca exibem detalhes básicos do livro e informações sobre o usuário que cadastrou o livro.

## **Trocas**

### **Solicitação de Troca:**

- Um usuário interessado em um livro pode solicitar a troca diretamente na plataforma.

- O dono do livro recebe uma notificação com o nome do usuário solicitante e pode visualizar os livros que o solicitante possui.

- O dono do livro pode aceitar ou recusar a solicitação de troca. Se a troca for aceita, ambos os usuários podem se comunicar para combinar os detalhes da troca física.

### **-Gestão de Trocas:**

- Após aceitar a troca, os livros são listados na página "Livros Trocados" de ambos os usuários como "em processo".

- Quando a troca física é realizada, o status pode ser atualizado para "concluído". Para isso, o usuário que recebeu a solicitação deve mudar o status, e o outro usuário deve confirmar.

- Caso a troca não ocorra, o status pode ser alterado para "não concluído".

## **Avaliação**

### **Avaliação de Usuários:**

- Após uma troca ser concluída, os usuários podem avaliar uns aos outros, dando uma nota de 0 a 5 e adicionando uma descrição da experiência de troca.
- Essas avaliações contribuem para a reputação dos usuários, ajudando outros a decidir com quem realizar trocas no futuro.

### **Funcionamento da Avaliação**

1. Realização da Troca: Após a troca de livros ser concluída, os usuários envolvidos na troca podem avaliar a experiência.
2. Inserção da Avaliação: Cada usuário que participou da troca pode criar uma avaliação, atribuindo uma nota e, opcionalmente, escrevendo um comentário sobre a troca.

## **Reputação e Classificação**

### **Reputação:**

- A reputação de um usuário é um indicador de sua confiabilidade na plataforma e é construída com base nas avaliações recebidas de outras trocas.

### **Funcionamento da Reputação**

1. Inicialização: Cada usuário tem uma reputação inicial que pode ser configurada ao criar o usuário.
2. Atualização da Reputação: Com base nas avaliações recebidas após cada troca, a reputação do usuário pode ser recalculada. Isso pode ser feito periodicamente ou após cada nova avaliação.

- Média entre 0 e 2: Reputação "Ruim"
- Média entre 3 e 4: Reputação "Regular"
- Média entre 4 e 5: Reputação "Excelente"

### **Classificação de Livros:**

- Os livros podem ser classificados quanto ao seu estado de conservação, o que ajuda a definir expectativas sobre a qualidade do livro recebido em uma troca.

## **Exemplo de Uso**

1. Cadastro: Fulano se cadastra na plataforma fornecendo seus dados pessoais.
2. Cadastro: Ciclano se cadastra na plataforma fornecendo seus dados pessoais.
3. Cadastro de Livro: Fulano cadastra um livro chamado "Orgulho e Preconceito" de Jane Austen, informando que o livro está em bom estado.
4. Cadastro de Livro: Ciclano cadastra um livro chamado "Hamlet" de William Shakespeare, informando que o livro está em bom ótimo estado.
5. Busca e Solicitação: Ciclano, interessado no livro de Fulano, realiza uma busca e encontra "Orgulho e Preconceito". Ele solicita a troca oferecendo um livro de sua coleção.
4. Aceitação de Troca: Fulano recebe a solicitação de Ciclano, verifica os livros oferecidos e decide aceitar a troca.
5. Troca Física: Fulano e Ciclano combinam os detalhes e realizam a troca fisicamente.
6. Atualização de Status: Após a troca física, Fulano muda o status para "concluído" e Ciclano confirma a atualização.
7. Avaliação: Ambos avaliam a experiência de troca, contribuindo para a reputação mútua na plataforma.

## **Requisitos Funcionais**

Identificador: RF001

Descrição: Criar Cadastro

Detalhes: O usuário fará seu cadastro no sistema registrando nome, email, senha, endereço (rua, número, bairro, cidade, estado, CEP), descrição, e localização.

Dependência: N/A

Identificador: RF002

Descrição: Recuperar Senha

Detalhes: O usuário poderá recuperar a senha informando o email utilizado no cadastro.

Dependência: RF001

Identificador: RF003

Descrição: Efetuar Login

Detalhes: O usuário efetua o login utilizando email e senha cadastrados.

Dependência: RF001

Identificador: RF004

Descrição: Consultar Página Principal

Detalhes: O usuário irá entrar automaticamente na página principal, onde aparecerão os livros que foram disponibilizados pelo autor e o histórico de trocas realizadas.

Dependência: RF003

Identificador: RF005

Descrição: Alterar Dados Pessoais

Detalhes: O usuário poderá alterar e incluir os dados pessoais nas configurações (nome, senha, descrição, localização e endereço).

Dependência: RF003

Identificador: RF006

Descrição: Cadastro de Livros

Detalhes: O usuário irá cadastrar livros disponíveis para troca, adicionando os detalhes do livro (título, autor, gênero, estado de conservação, descrição, data de publicação, número da edição, e disponibilidade).

Dependência: RF003

Identificador: RF007

Descrição: Buscar Livros

Detalhes: O usuário irá buscar livros disponíveis para troca, podendo ser por título, autor, gênero, localização; serão exibidos detalhes básicos do livro e o usuário dono do livro.

Dependência: RF003

Identificador: RF008

Descrição: Solicitar a Troca de Livro

Detalhes: O usuário irá solicitar a troca de livro. O dono do livro será notificado e aparecerá o nome do usuário que solicitou a troca. O dono do livro poderá ver os livros que o solicitante possui e deverá aceitar ou recusar a troca. Se a troca for aceita, será habilitada a comunicação entre os dois e os livros trocados serão apresentados para os dois usuários na página "Livros Trocados" como "em processo".

Dependência: RF007

Identificador: RF009

Descrição: Livros Trocados

Detalhes: O usuário poderá visualizar a troca dos livros (o livro que ele ofereceu na troca, o livro que ele pegou na troca, e a data que foi realizada a troca no sistema). Somente o usuário que recebeu a solicitação da troca poderá mudar o status de "em processo" para "concluído". O outro usuário terá que aceitar para a troca ser definitiva. Caso não ocorra a troca, deverá ser marcado a opção de "não concluído".

Dependência: RF008

Identificador: RF010

Descrição: Avaliação dos Usuários

Detalhes: Após uma troca de livro ser concluída, será habilitada a opção de avaliação de usuário, onde cada usuário poderá avaliar o outro dando uma nota de 0 a 5, e uma pequena descrição da troca.

Dependência: RF009

## Requisitos Não Funcionais

Identificador: RNF001

Descrição: Segurança

Detalhes: A senha do usuário deve ser criptografada antes de ser armazenada no banco de dados.

Dependência: RF001

## Diagramas de banco de datos (ERDs)



<https://dbdiagram.io/e/66966f018b4bb5230e7b114b/669aaf5e8b4bb5230ed56a9>



## Criação das tabelas

```
CREATE TABLE reputacao (  
    id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY NOT NULL,  
    nivel VARCHAR(20) NOT NULL  
);
```

```
CREATE TABLE classificacaolivro (  
    id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY NOT NULL,  
    nivel VARCHAR(20) NOT NULL  
);
```

```
CREATE TABLE usuario (  
    id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY NOT NULL,  
    nome VARCHAR(50) NOT NULL,  
    email VARCHAR(50) UNIQUE NOT NULL,  
    senha VARCHAR(255) NOT NULL,  
    rua VARCHAR(50),  
    numero CHAR(5),  
    bairro VARCHAR(30),  
    cidade VARCHAR(30) NOT NULL,  
    estado CHAR(2) NOT NULL,  
    cep VARCHAR(10) NOT NULL,  
    data_cadastro DATE DEFAULT CURRENT_DATE,  
    descricao VARCHAR(100),  
    reputacao_id INT,  
    CONSTRAINT fk_reputacao FOREIGN KEY (reputacao_id) REFERENCES reputacao (id),  
    CONSTRAINT chk_estado CHECK (length(estado) = 2)  
);
```

```

CREATE TABLE livro (
    id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY NOT NULL,
    titulo VARCHAR(100) NOT NULL,
    autor VARCHAR(100) NOT NULL,
    genero VARCHAR(40) NOT NULL,
    estado_de_conservacao VARCHAR(40) NOT NULL,
    descricao VARCHAR(255),
    data_publicacao DATE,
    num_edicao CHAR(5),
    disponivel BOOLEAN DEFAULT TRUE,
    usuario_id INT NOT NULL,
    CONSTRAINT fk_usuario FOREIGN KEY (usuario_id) REFERENCES usuario (id),
    CONSTRAINT chk_num_edicao CHECK (length(num_edicao) <= 5)
);

```

```

CREATE TABLE troca (
    id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY NOT NULL,
    data DATE NOT NULL DEFAULT CURRENT_DATE,
    solicitante_id INT NOT NULL,
    solicitado_id INT NOT NULL,
    livro_solicitado_id INT NOT NULL,
    livro_oferecido_id INT NOT NULL,
    estado VARCHAR(40) NOT NULL,
    avaliacao_livro TEXT,
    troca_autorizada BOOLEAN DEFAULT FALSE,
    troca_solicitada BOOLEAN DEFAULT TRUE,
    classificacao_livro_id INT,
    CONSTRAINT fk_solicitante FOREIGN KEY (solicitante_id) REFERENCES usuario (id),
    CONSTRAINT fk_solicitado FOREIGN KEY (solicitado_id) REFERENCES usuario (id),
    CONSTRAINT fk_livro_solicitado FOREIGN KEY (livro_solicitado_id) REFERENCES livro (id),
    CONSTRAINT fk_livro_oferecido FOREIGN KEY (livro_oferecido_id) REFERENCES livro (id),
    CONSTRAINT fk_classificacao_livro FOREIGN KEY (classificacao_livro_id) REFERENCES classificacaolivro (id)
);

```

-- Criação da tabela avaliacao

```
CREATE TABLE avaliacao (  
  id INT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY NOT NULL,  
  nota INT NOT NULL,  
  comentario VARCHAR(255),  
  data DATE NOT NULL DEFAULT CURRENT_DATE,  
  avaliador_id INT NOT NULL,  
  troca_id INT NOT NULL,  
  avaliado_id INT NOT NULL,  
  CONSTRAINT fk_avaliador FOREIGN KEY (avaliador_id) REFERENCES usuario (id),  
  CONSTRAINT fk_troca FOREIGN KEY (troca_id) REFERENCES troca (id),  
  CONSTRAINT fk_avaliado FOREIGN KEY (avaliado_id) REFERENCES usuario (id),  
  CONSTRAINT chk_nota CHECK (nota BETWEEN 1 AND 5)  
);
```

-- Índices adicionais para melhorar o desempenho

```
CREATE INDEX idx_usuario_cidade ON usuario (cidade);  
CREATE INDEX idx_livro_titulo ON livro (titulo);  
CREATE INDEX idx_troca_data ON troca (data);  
CREATE INDEX idx_avaliacao_data ON avaliacao (data);
```

## Simulação de Troca com Avaliação e Atualização de Reputação

```
INSERT INTO reputacao (nivel) VALUES ('Excelente'), ('Regular'), ('Ruim');
```

```
INSERT INTO classificacaolivro (nivel) VALUES ('Novo'), ('Usado - Ótimo Estado'), ('Usado - Bom Estado'), ('Usado - Regular'), ('Usado - Ruim');
```

```
INSERT INTO usuario (nome, email, senha, cidade, estado, cep, data_cadastro)
```

```
VALUES
```

```
('Fulano', 'fulano@example.com', 'senha123', 'São Paulo', 'SP', '01000-000', '2023-07-01'),
```

```
('Ciclano', 'ciclano@example.com', 'senha123', 'Rio de Janeiro', 'RJ', '20000-000', '2023-07-02');
```

```
SELECT * FROM usuario;
```

```
-- Inserir dados na tabela Livro
```

```
INSERT INTO livro (titulo, autor, genero, estado_de_conservacao, usuario_id)
```

```
VALUES
```

```
('Orgulho e Preconceito', 'Jane Austen', 'Romance', 'Usado - Bom Estado', 1),
```

```
('1984', 'George Orwell', 'Distopia', 'Novo', 2);
```

```
SELECT * FROM livro;
```

```
-- Sistema de troca dos livros
```

```
-- Fulano solicita a troca do livro com ciclano
```

```
INSERT INTO troca (data, solicitante_id, solicitado_id, livro_solicitado_id, livro_oferecido_id, estado, troca_solicitada, troca_autorizada)
```

```
VALUES ('2024-07-16', 1, 2, 2, 1, 'Solicitado', TRUE, FALSE);
```

```
SELECT * FROM troca;
```

```
-- Ciclano verifica e aceita trocar os livros
```

```
UPDATE troca
```

```
SET troca_autorizada = TRUE, estado = 'Autorizado'
```

WHERE id = 1;

-- A troca só será realizada se a troca autorizada for TRUE

SELECT \* FROM troca WHERE troca\_solicitada = TRUE AND troca\_autorizada = TRUE;

-- Os dois se encontram e trocam os livros pessoalmente

-- Ciclano marca no sistema a troca como 'Concluida', para assim os livros serem trocados de donos no sistema

UPDATE troca

SET estado = 'Concluido'

WHERE id = 1;

-- Troca de Livros no Sistema

-- Livro 2 (De ciclano) agora é de fulano

UPDATE livro

SET usuario\_id = 1

WHERE id = 2;

-- Livro 1 (De fulano) agora é de ciclano

UPDATE livro

SET usuario\_id = 2

WHERE id = 1;

SELECT \* FROM livro WHERE id IN (1, 2);

-- Avaliacao e Reputacao

SELECT \* FROM avaliacao

-- Fulano avalia Ciclano

```
INSERT INTO avaliacao (nota, comentario, avaliador_id, troca_id, avaliado_id)
VALUES (5, 'Ótima troca!', 1, 1, 2);
```

```
-- Ciclano avalia Fulano
```

```
INSERT INTO avaliacao (nota, comentario, avaliador_id, troca_id, avaliado_id)
VALUES (4, 'Tudo certo!', 2, 1, 1);
```

```
-- avaliacoes realizadas
```

```
SELECT
    a.id,
    a.nota,
    a.comentario,
    a.data,
    a.troca_id,
    a.avaliado_id,
    u.nome AS nome_avaliado
FROM
    avaliacao a
JOIN
    usuario u ON a.avaliado_id = u.id
WHERE
    a.avaliador_id = 2; -- alterar o id da pessoa que realizou a avaliacao
```

```
-- avaliacoes recebidas
```

```
SELECT
    a.id,
    a.nota,
    a.comentario,
    a.data,
    a.troca_id,
    a.avaliador_id,
    u.nome AS nome_avaliador
FROM
    avaliacao a
```

JOIN

usuario u ON a.avaliador\_id = u.id

WHERE

a.avaliado\_id = 2; -- colocar o id da pessoa que recebeu a avaliacao