**1. Visão Geral**

O projeto consiste em um e-commerce de livros desenvolvido com uma arquitetura baseada em frontend, backend e um painel de administração. O objetivo do sistema é fornecer uma plataforma onde os usuários possam comprar livros de maneira eficiente, com recursos como personalização, recomendação de livros e sistema de alerta de preços.

**2. Tecnologias Utilizadas**

* **Node.js**: Plataforma de execução JavaScript no servidor.
* **React.js**: Biblioteca para construção da interface do usuário.
* **Express.js**: Framework para construção da API no backend.
* **MongoDB**: Banco de dados NoSQL utilizado para armazenamento de informações dos usuários, livros e pedidos.
* **JWT (JSON Web Token)**: Utilizado para autenticação e autorização dos usuários.
* **Redux**: Utilizado no frontend para gerenciamento do estado global da aplicação.

**3. Estrutura do Projeto**

* **admin/**: Diretório que contém o painel de administração, onde é possível gerenciar produtos, pedidos e usuários.
* **backend/**: Diretório contendo a API do servidor responsável por intermediar as requisições entre o frontend e o banco de dados. Aqui estão definidos:
  + As rotas para manipulação de livros, usuários e pedidos.
  + A autenticação de usuários (login e registro).
  + Conexão com o MongoDB.
* **frontend/**: Diretório contendo o código da interface do usuário. Utiliza o React.js e Redux para exibir páginas dinâmicas, realizar requisições à API e manipular o estado global da aplicação.
* **node\_modules/**: Diretório onde as dependências do Node.js estão armazenadas.
* **package.json**: Arquivo de manifesto que define as dependências, scripts e configurações do projeto.
* **package-lock.json**: Arquivo de bloqueio de versões das dependências instaladas, garantindo que todos os ambientes usem as mesmas versões.

**4. Configurações do Projeto**

**4.1. Instalação do Projeto**

Para rodar o projeto localmente, siga os seguintes passos:

1. Clone o repositório do projeto:

bash

Copiar código

git clone <URL-do-repositório>

1. Navegue até o diretório do projeto e instale as dependências:

bash

Copiar código

cd Projeto-integrador

npm install

**4.2. Configuração do Backend**

O backend utiliza **Express.js** e se comunica com um banco de dados MongoDB. Para configurar o banco de dados, siga os passos abaixo:

1. Crie um arquivo .env no diretório backend/ com as seguintes variáveis:

env

Copiar código

MONGO\_URI=<sua-string-de-conexão-com-o-MongoDB>

JWT\_SECRET=<seu-segredo-para-assinatura-JWT>

1. Inicie o servidor do backend:

bash

Copiar código

npm run server

**4.3. Configuração do Frontend**

Para rodar o frontend, vá até o diretório frontend/ e rode o comando:

bash

Copiar código

npm start

Isso irá iniciar a aplicação em modo de desenvolvimento. A interface estará disponível no navegador na porta padrão.

**5. Principais Funcionalidades**

* **Cadastro e Login de Usuários**: Os usuários podem se registrar no sistema e fazer login com autenticação via JWT.
* **Catálogo de Livros**: Exibição de uma lista de livros disponíveis com opções de filtro e busca.
* **Carrinho de Compras**: Funcionalidade para adicionar e remover itens do carrinho, com cálculo automático do total.
* **Sistema de Recomendação**: Baseado no histórico de compras e preferências dos usuários.
* **Painel de Administração**: Onde os administradores podem gerenciar produtos, visualizar pedidos e acompanhar os usuários.

**6. API Endpoints**

**6.1. Autenticação**

* **POST** /api/users/login: Autenticação do usuário e retorno do token JWT.
* **POST** /api/users/register: Registro de um novo usuário.

**6.2. Livros**

* **GET** /api/books: Retorna todos os livros disponíveis.
* **GET** /api/books/:id: Retorna detalhes de um livro específico.

**6.3. Pedidos**

* **POST** /api/orders: Cria um novo pedido de compra.
* **GET** /api/orders/user/:id: Retorna os pedidos de um usuário específico.

**7. Segurança**

* **Autenticação JWT**: Todas as rotas que exigem autenticação utilizam JWT para verificar se o usuário está logado.
* **Criptografia de Senhas**: As senhas dos usuários são armazenadas de forma segura no banco de dados utilizando hashing (bcrypt).
* **Proteção de Rotas**: Rotas administrativas são protegidas para garantir que apenas usuários com permissões adequadas possam acessá-las.

**8. Próximos Passos e Melhorias Futuras**

* **Implementação de um sistema de avaliações e comentários** para cada livro.
* **Otimização do sistema de recomendações**, utilizando machine learning para sugestões mais precisas.
* **Integração com serviços de pagamento** para processar transações diretamente pela plataforma.