

Gerenciador de Livros

Humberto Alves Mesquita

Isabella Barbosa de Andrade Mesquita

1. Explicação Técnica

1.1. Visão Geral

O projeto consiste em uma aplicação web *Full-Stack* desenvolvida para o gerenciamento de um catálogo de livros, a nossa arquitetura segue o modelo cliente-servidor, com uma interface de usuário (frontend) desacoplada de uma *API RESTful* (backend), que por sua vez se comunica com um banco de dados relacional. O objetivo é permitir que o usuário realize operações de CRUD (Criar, Ler, Atualizar e Deletar) de forma dinâmica e interativa.

1.2. Arquitetura

A aplicação está dividida em três camadas principais:

a) Frontend:

A interface do usuário foi construída com HTML, CSS e JavaScript, com a biblioteca jQuery sendo o principal motor para a interatividade, todas as operações que alteram ou buscam dados, como cadastrar um novo livro ou atualizar a lista, são realizadas através de chamadas AJAX. Isso proporciona uma experiência de usuário fluida, sem a necessidade de recarregar a página, a formatação e o envio de dados para o backend são feitos no formato JSON, para a organização e otimização do código JavaScript do frontend, foi utilizado o Webpack, que empacota os módulos em um único arquivo (bundle.js) para ser carregado pelo navegador.

b) Backend:

O servidor foi desenvolvido em Node.js utilizando o framework Express.js para a criação de uma API RESTful, as principais responsabilidades do backend são:

- Gerenciamento de Rotas: Definir os endpoints da API (/livros, /livro/:id, etc.) para cada operação CRUD.
- Processamento de Requisições: Utilizar middlewares essenciais como o CORS, para permitir requisições do frontend, e o express.json(), para interpretar o corpo das requisições enviadas em JSON.
- Lógica de Negócio: Validar os dados recebidos.
- Comunicação com o Banco de Dados: Executar as queries SQL para persistir e consultar os dados.

A estrutura do código foi modularizada, separando as rotas da lógica de conexão com o banco de dados, o que promove maior organização e manutenibilidade.

c) Banco de Dados:

Foi escolhido o PostgreSQL, um sistema de gerenciamento de banco de dados relacional de código aberto, a comunicação entre a aplicação Node.js e o PostgreSQL é feita através do driver node-postgres (pg), uma única tabela, chamada **livro** foi criada para armazenar os dados, contendo os campos id, titulo, autor e npags. Por segurança, todas as operações de inserção e atualização de dados utilizam parâmetros preparados (parameterized queries), uma técnica essencial para prevenir ataques de SQL Injection.

1.3. Fluxo de Dados

1. O usuário interage com a página HTML no navegador.
2. Uma ação (ex: clicar em "Salvar") dispara um evento JavaScript, gerenciado pelo jQuery.
3. O jQuery monta um objeto JSON e envia uma requisição AJAX para o endpoint apropriado no backend Express.
4. O servidor Express recebe a requisição, utiliza o middleware `express.json()` para processar os dados e direciona para a rota correspondente.
5. A rota executa uma função que utiliza o driver pg para se comunicar com o PostgreSQL, enviando uma query parametrizada.
6. O banco de dados processa a query e retorna o resultado para o backend.
7. O backend formata a resposta como JSON e a envia de volta para o frontend.
8. O jQuery, na função de success da chamada AJAX, recebe a resposta e atualiza o HTML da página (o DOM) dinamicamente para refletir a mudança.

2. Módulos e Tecnologias Utilizadas

Frontend

- HTML5: Estruturação da página.
- CSS3: Estilização visual.
- JavaScript (ES6+): Lógica do lado do cliente.
- jQuery: Manipulação do DOM e requisições AJAX.
- Webpack: Empacotador de módulos JavaScript.

Backend

- Node.js: Ambiente de execução do lado do servidor.
- Express.js: Framework para construção da API RESTful.
- CORS: Middleware para habilitar o compartilhamento de recursos entre origens diferentes.
- node-postgres (pg): Driver para a conexão com o banco de dados PostgreSQL.

Banco de Dados

- PostgreSQL: Sistema de Gerenciamento de Banco de Dados Relacional.

Obs: O passo a passo para rodar localmente pode ser lido no repositório no GitHub.

4. Referência

Link do vídeo demonstrativo: <https://www.youtube.com/watch?v=vtr875hXReA>

Repositório no GitHub: https://github.com/HumbertoMesquita/Gerenciador_de_Livros