

Descrição das escolhas arquiteturais do trabalho final

Back-end

Descrição resumida:

Para o back-end, escolhemos Node JS pela sua praticidade, O NPM (Node Package Manager) é o gerenciador de pacotes do Node.js e também é o maior repositório de softwares do mundo. Isso faz do Node.js uma plataforma com potencial para ser utilizada em qualquer situação. Para a persistência dos dados, escolhemos o Mongoose, uma biblioteca para utilizarmos o MongoDB, um banco de dados não relacional, orientado a documentos, o que traz mais praticidade para o projeto. Por fim, para expor as funcionalidades como um serviço, utilizamos o padrão REST.

- **Node JS**

Node.js é um interpretador de JavaScript assíncrono com código aberto orientado a eventos, criado por Ryan Dahl em 2009, focado em migrar a programação do Javascript do cliente para os servidores, criando aplicações de alta escalabilidade (como um servidor web), manipulando milhares de conexões/eventos simultâneas em tempo real numa única máquina física.

- **ODM Mongoose**

Mongoose é uma biblioteca do Nodejs que proporciona uma solução baseada em esquemas para modelar os dados da sua aplicação. Ele possui sistema de conversão de tipos, validação, criação de consultas e hooks para lógica de negócios.

Mongoose fornece um mapeamento de objetos do **MongoDB** similar ao ORM (Object Relational Mapping), ou ODM (Object Data Mapping) no caso do Mongoose. Isso significa que o Mongoose traduz os dados do banco de dados para objetos JavaScript para que possam ser utilizados por sua aplicação.

- **API Rest**

A **Representational State Transfer (REST)**, em português Transferência de Estado Representacional, é uma abstração da arquitetura da World Wide Web, mais precisamente, é um estilo arquitetural que consiste de um conjunto coordenado de restrições arquiteturais

aplicadas a componentes, conectores e elementos de dados dentro de um sistema de hipermídia distribuído.

O **REST** ignora os detalhes da implementação de componente e a sintaxe de protocolo com o objetivo de focar nos papéis dos componentes, nas restrições sobre sua interação com outros componentes e na sua interpretação de elementos de dados significantes.

Front-end

Descrição resumida:

Para o front-end, utilizar o React, devido o conhecimento com o framework e sua praticidade e leveza.

- **React**

O **React** (também denominado React.js ou ReactJS) é uma biblioteca JavaScript de código aberto com foco em criar interfaces de usuário em páginas web.