

1. Introdução ao Node.js e SQLite

O que é Node.js? Node.js é um ambiente de execução JavaScript no lado do servidor que permite criar aplicações de rede escaláveis. Utiliza um modelo baseado em eventos e uma arquitetura não-bloqueante, ideal para aplicações que precisam lidar com muitas conexões simultâneas.

O que é SQLite? SQLite é um banco de dados relacional leve e autocontido, ideal para aplicações que não requerem um servidor de banco de dados completo. É frequentemente usado em aplicações móveis e pequenas aplicações web.

Por que usar Node.js com SQLite?

- **Simplicidade:** SQLite é fácil de configurar e usar.
- **Desempenho:** Ideal para projetos pequenos e médios.
- **Integração:** Node.js e SQLite são bem integrados através de pacotes como sqlite3.

2. Preparação do Ambiente

Instalação do Node.js

1. Baixe e instale o Node.js a partir do site oficial:

<https://nodejs.org/pt/download/prebuilt-installer>

Verifique a instalação com os comandos:

```
node -v
```

```
npm -v
```

Instalação do SQLite

1. Baixe e instale o SQLite a partir do site oficial.
2. Verifique a instalação com o comando:

```
sqlite3 --version
```

Configuração do Projeto

1. Crie um diretório para seu projeto e navegue até ele:

```
mkdir meu-projeto
```

```
cd meu-projeto
```

3. Instalação e Configuração do Projeto Node.js

Criar um novo projeto Node.js

1. Inicialize um novo projeto Node.js:

```
npm init -y
```

Instalar dependências essenciais

1. Instale o Express e o sqlite3:

`npm install express sqlite3`

Configuração inicial

Arquivo app.js no raiz do projeto.

Instalação de front-end na subpasta “public” (exemplo: arquivo index.html).

https://github.com/almeida-cma/MB_TEMA2/tree/main/NODE_USERS

Comando para iniciar aplicação

Sempre que precisar iniciar uma aplicação em Node.JS é só digitar na pasta do arquivo principal o comando `node` seguido `do nome do arquivo principal`. No nosso exemplo seria:

`node app.js`