

Backend com Node.js

Por Wanderson Guimarães

Sobre o Professor

- Desenvolvedor Fullstack - 15 anos (mais ou menos)
- Trabalhou nos principais bancos do país
- Leciona no Senac desde 2010

O que vamos aprender hoje:

- Conceitos fundamentais do Node.js
- Configuração do ambiente
- Módulos nativos e CommonJS
- NPM e gerenciamento de pacotes
- Criando um servidor HTTP básico
- Manipulação de arquivos com FS

Ferramentas que vamos usar

- VS Code com extensões úteis (ESLint, Prettier)
- Postman ou navegador para testar requisições
- Terminal (bash, zsh, PowerShell).

O que é Node.js

- Node.js é um runtime JavaScript.
- Construído sobre o engine V8 do Chrome.
- Permite rodar JS no backend.
- Ideal para aplicações escaláveis e assíncronas.

O que é um runtime ???

- Um Runtime JavaScript é o ambiente onde o código JavaScript é executado fora do navegador.
- Em outras palavras: É um conjunto de ferramentas (engine, APIs, bibliotecas e gerenciadores de tarefas) que permite rodar JavaScript em um computador, servidor ou dispositivo — sem precisar de um navegador.

Vantagens do Node.js

- Alta performance (V8).
- Assíncrono e não bloqueante
- Grande ecossistema (NPM)
- Um só idioma: JavaScript
- Comunidade ativa

Onde o Node.js é usado

- APIs REST
- Chat em tempo real (ex: socket.io)
- Ferramentas CLI
- Aplicações em tempo real
- Microserviços

Instalando o Node.js

- <https://nodejs.org>
- Escolher a versão LTS
- Verificar com:
 - `node -v`
 - `npm -v`

Primeiro script em Node.js

- `console.log('Olá, mundo do Node!');`
- Vamos rodar nosso código usando :
- `node app.js`

Módulos em Node.js

- Tipos de módulos:

- Nativos: fs, path, http
- Externos (via NPM)
- Criados por você
- Exemplo de chamada : `const fs = require('fs');`

NPM = Node Package Manager

- Gerenciador de Pacotes : permite instalar ,compartilhar e gerenciar bibliotecas (ou pacotes) .
- Instala pacotes prontos que facilitam o desenvolvimento .
- Gerencia dependências de forma automática
- Permite scripts personalizados
- Tem um vasto repositório online com diversos pacotes : <https://www.npmjs.com/>

Exemplos de uso do npm

// Inicializar um projeto

```
npm init -y
```

// Instalar um pacote (ex: axios)

```
npm install axios
```

// Rodar um script definido em package.json

```
npm run start
```


Mais exemplos de uso de NPM

- `npm install chalk`
- `const chalk = require('chalk');`
- `console.log(chalk.green("Texto colorido!"));`

Exportando Módulos

```
//app.js
```

```
const calc = require('./calc');
```

```
console.log(calc.soma(2, 3));
```

```
// calc.js
```

```
function soma(a, b) {
```

```
  return a + b;
```

```
}
```

```
module.exports = { soma };
```


Manipulando arquivos

- `const fs = require('fs');`
- `fs.readFile('arquivo.html', 'utf8', (err, data) => {`
- `if (err) throw err;`
- `console.log(data);`
- `});`

Exercício

- Crie um arquivo chamado **mensagem.txt** e coloque conteúdo
- Crie um arquivo chamado **lerMensagem.js** para ler o arquivo.

Criando um servidor http

```
const http = require('http');
```

```
const server = http.createServer((req, res) => {
```

```
  res.write('olá do servidor!');
```

```
  res.end();
```

```
});
```

```
server.listen(3001, () => {
```

```
  console.log('Servidor rodando em http://localhost:3001');
```

```
});
```