TREINAMENTO NINJA PARA MESTRES DO JAVASCRIPT

VITOR LUCAS

01

O que é NodeJS?

Introdução ao Node.js

Princípios Essenciais

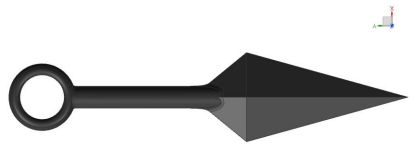
Node.js é uma poderosa plataforma que permite executar JavaScript no lado do servidor. Para dominar Node.js, é importante entender tanto os princípios básicos quanto os avançados. Este ebook cobre esses princípios com explicações claras e exemplos de código práticos.

02

Concetto di Single-Thread e Non-Blocking I/O

Single-Thread e Non-Blocking I/O

Node.js opera em um único thread usando um modelo de I/O não-bloqueante. Isso significa que, ao realizar operações de I/O (como ler um arquivo), o Node.js pode continuar executando outras tarefas em vez de esperar pela conclusão da operação.



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

// Operação não-bloqueante
fs.readFile('file.txt', 'utf8', (err, data) => {
  if (err) throw err;
  console.log(data);
});

console.log('Este log aparece primeiro');
 put your code here
```

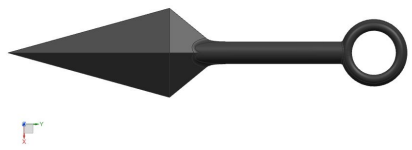
Vitor Lucas de Sena Lima

03

Módulos e NPM

Módulos e NPM

Node.js usa um sistema de módulos para organizar o código. O NPM (Node Package Manager) é usado para gerenciar dependências externas.



```
// Utilizando um módulo próprio
const myModule = require('./myModule');

// Utilizando um módulo de terceiros instalado via NPM
const express = require('express');
const app = express();

app.get('/', (req, res) => {
  res.send('Hello World');
});

app.listen(3000, () => {
  console.log('Servidor rodando na porta 3000');
});
```

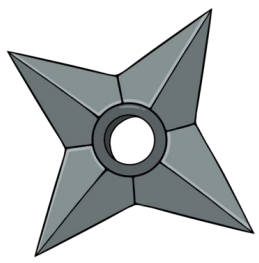
Vitor Lucas de Sena Lima

04

Callbacks, Promises e Async/Await

Callbacks, Promises e Async/Await

Node.js utiliza callbacks para operações assíncronas, mas Promises e async/await oferecem uma forma mais elegante de lidar com essas operações.



Callbacks

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

fs.readFile('file.txt', 'utf8', (err, data) => {
  if (err) throw err;
  console.log(data);
});
```

Vitor Lucas de Sena Lima

Callbacks, Promises e Async/Await

Promisses

```
const fs = require('fs').promises;

fs.readFile('file.txt', 'utf8')
  .then(data ⇒ console.log(data))
  .catch(err ⇒ console.error(err));
```

Vitor Lucas de Sena Lima

AsyncAwait

```
const fs = require('fs').promises;

(async () ⇒ {
  try {
    const data = await fs.readFile('file.txt', 'utf8');
    console.log(data);
  } catch (err) {
    console.error(err);
  }
})();
```

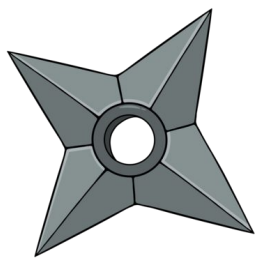
Vitor Lucas de Sena Lima

05

Criação de um Servidor HTTP

Criação de um Servidor HTTP

Node.js facilita a criação de servidores HTTP, permitindo a construção de aplicações web.



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

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World\n');
});

server.listen(3000, () => {
  console.log('Servidor rodando na porta 3000');
});
```

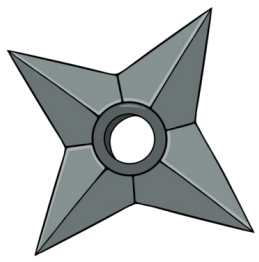
Vitor Lucas de Sena Lima

06

Utilização do Express.Js

Utilização do Express.js

Express.js é um framework minimalista para Node.js, que simplifica a criação de aplicações web e APIs.



```
const express = require('express');
const app = express();

app.get('/', (req, res) => {
  res.send('Hello World');
});

app.listen(3000, () => {
  console.log('Servidor rodando na porta 3000');
});
```

Vitor Lucas de Sena Lima

07

Trabalhando com Banco de Dados

Trabalhando com Banco de Dados

Node.js pode se conectar a vários tipos de bancos de dados, facilitando operações CRUD (Create, Read, Update, Delete).

```
const { MongoClient } = require('mongodb');

async function main() {
  const uri = 'mongodb://localhost:27017';
  const client = new MongoClient(uri);

  try {
    await client.connect();
    const database = client.db('mydatabase');
    const collection = database.collection('mycollection');

    const doc = { name: 'John', age: 30 };
    await collection.insertOne(doc);

    const result = await collection.findOne({ name: 'John' });
    console.log(result);
  } finally {
    await client.close();
  }
}

main().catch(console.error);
```

Vitor Lucas de Sena Lima

Agradecimentos

Obrigado por ler até aqui

Esse Ebook foi gerado por IA, e diagramado por humano. O passo a passo se encontra no meu Github.

Esse conteúdo foi gerado com fins didáticos de construção, não foi realizado uma validação cuidadosa humana no conteúdo e pode conter erros gerados por uma IA.

https://github.com/SenaVitor

Criado por: Vitor Lucas