

Software **NODE.JS**

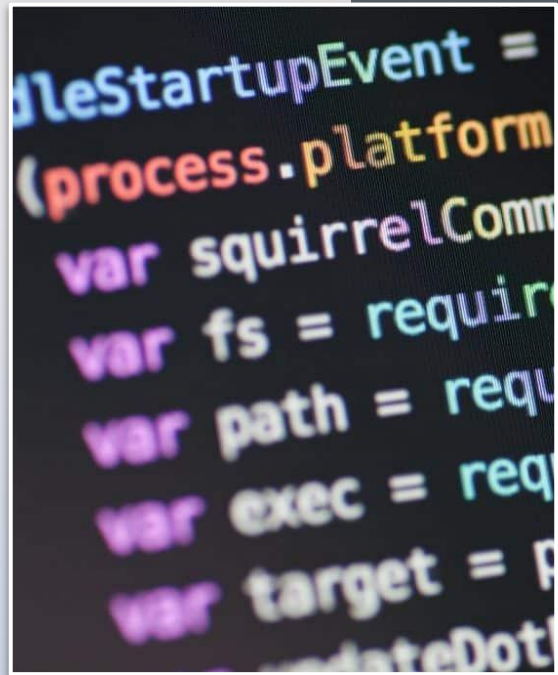
Heitor, César, Vinicius Lopes, João V Costa

Desenvolvimento de sistemas



O que é o Node.js

No coração do desenvolvimento moderno, o Node.js se destaca como um ambiente de execução JavaScript de última geração. Com uma abordagem assíncrona orientada a eventos, e focada em criar servidores http, ele também oferece uma base sólida para a criação de aplicativos de rede escalonáveis. Ao unir a flexibilidade do JavaScript com a potência de um tempo de execução de alto desempenho, o Node.js abre as portas para a construção de soluções inovadoras e eficientes.



Historia:

O Node.js nasceu em 2009 como uma resposta às tentativas de rodar códigos JavaScript em modo server side (são linguagens que o SERVIDOR entende, ou que rodam no servidor), uma vez que a linguagem tinha como meta principal a manipulação do DOM (Document Object Model) e deixar as aplicações web mais interativas e dinâmicas.





Arquitetura do Node.js:

A arquitetura do Node.js é conhecida por ser orientada a eventos e assíncrona

• Loop de Eventos:

É o coração da arquitetura, sendo um ciclo contínuo que aguarda eventos e os processa quando eles ocorrem, permitindo que o Node.js seja altamente responsivo e eficiente, registrando callbacks (uma função executada após outra.)

• Event Emitters:

Ele lida com eventos, onde possuem métodos para registrar funções de callback para eventos específicos, notificando todos os ouvintes (callbacks) registrados para esse evento.

• Assincronismo:

Ele faz com que em vez de esperar que uma operação seja concluída antes de prosseguir, ele continua a executar outras tarefas enquanto aguarda a operação assíncrona ser concluída. Permitindo que o Node.js lide com várias operações de entrada/saída simultaneamente.

• Callbacks, Promises e async/await:

O Node.js utiliza callbacks (funções de retorno) como uma abordagem inicial. Promises e a sintaxe async/await, que tornam o código assíncrono mais legível e gerenciável.

• Thread Única

Permite que o Node.js delegue tarefas assíncronas para sistemas subjacentes (como o sistema operacional) e retome a execução quando as tarefas forem concluídas.


• Módulos e Escopo:

A modularidade é uma parte fundamental da arquitetura, na qual os módulos são unidades independentes que ajudam a manter o código organizado e a reutilizar o código facilmente.





Node.js **X** Outras linguagens.

- Venus has a beautiful name and is the second planet from the Sun
- 



Exemplos de uso do Node.Js



Machinery

Mercury is the closest planet to the Sun

Hardware

Saturn is composed of hydrogen and helium

Robotics

Jupiter is the biggest planet in the Solar System

Building

Despite being red, Mars is actually a cold place



Exemplo de um código Node.js:

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

```
const hostname = '127.0.0.1';
```

```
const port = 3000;
```

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

```
  res.statusCode = 200;
```

```
  res.setHeader('Content-Type', 'text/plain');
```

```
  res.end('Hello World');
```

```
});
```

```
server.listen(port, hostname, () => {
```

```
  console.log(`Server running at http://${hostname}:${port}/`);
```

```
});
```

Thanks!

