

Plataforma Node.js

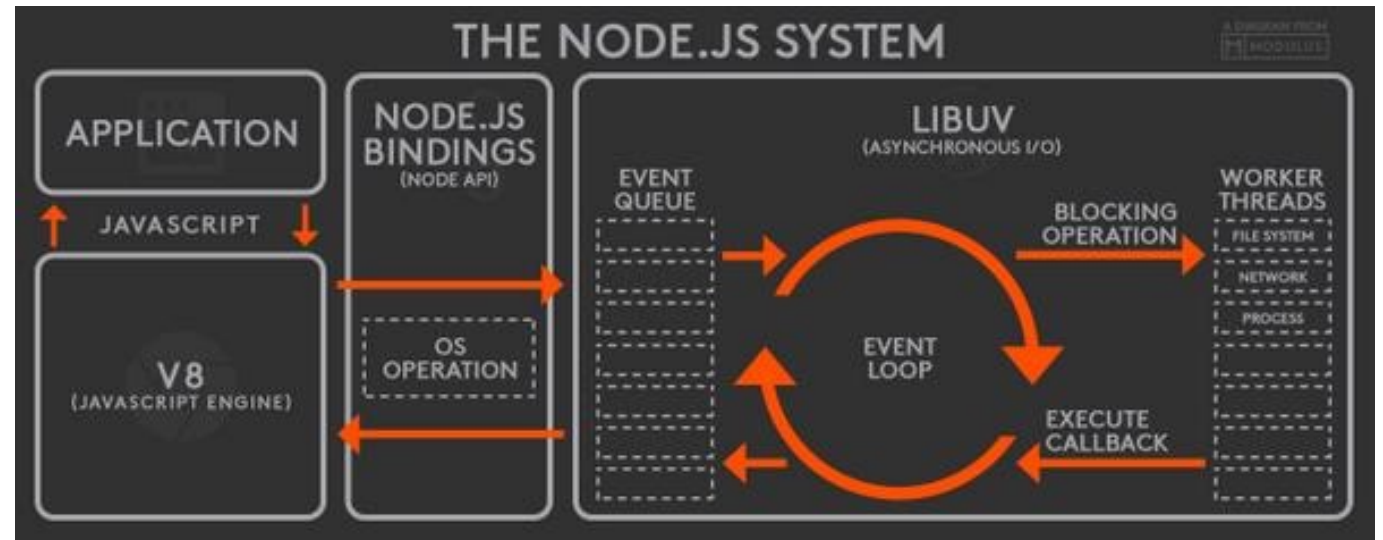
Arquitetura da plataforma Node.js

Arquitetura Node.js

A plataforma Node.js é orientada a eventos com modelo não bloqueante de I/O

Temas relacionados

- [JavaScript single Thread Programming](#)
- [Event-Driven Architecture \(EDA\)](#)
- [Blocking vs Non-Blocking](#)
- [Problema C10K](#)
- [Reverse proxy](#) e [Load balancing](#)



Outras Fontes:

- [Node.js Architecture – Idealwebtutor](#)
- [EDA em uma Arquitetura de Microsserviços \(Marcelo M. Gonçalves\)](#)
- [What is a Reverse Proxy vs. Load Balancer? \(NGINX\)](#)

Arquitetura - Blocking vs Non-Blocking

Abordagem Blocking

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

// blocks here until file is read
const data = fs.readFileSync('/file.md');

console.log(data);

// will run after console.log
moreWork();
```

Abordagem Non-Blocking

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

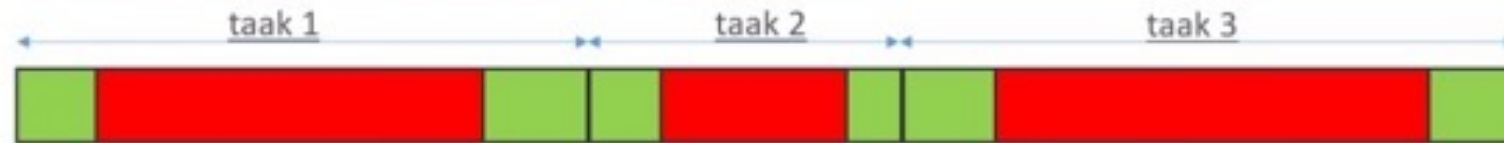
fs.readFile('/file.md', (err, data) => {
  if (err) throw err;
  console.log(data);
});

// will run before console.log
moreWork();
```

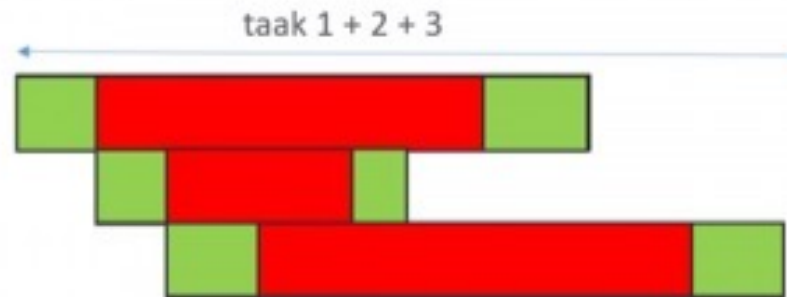
Fonte: [Overview of Blocking vs Non-Blocking | Node.js \(nodejs.org\)](https://nodejs.org/en/docs/guides/blocking-vs-non-blocking/)

Arquitetura – Programação Síncrona vs Assíncrona

- Synchronon (sequentieel)

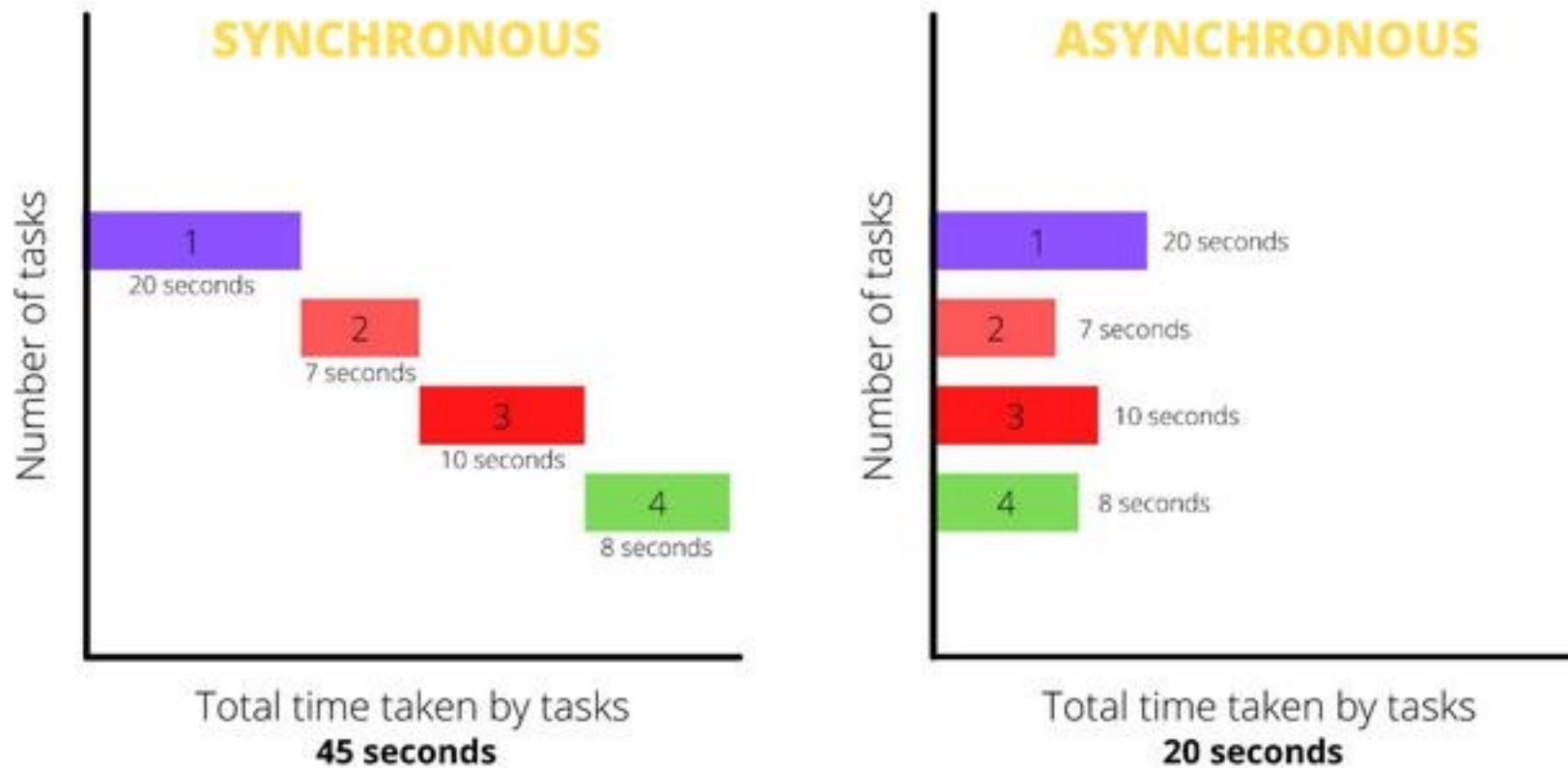


- Asynchroon (callbacks):



Fonte: [Introduction to Node.js \(SynTouch\)](#)

Arquitetura – Programação Síncrona vs Assíncrona



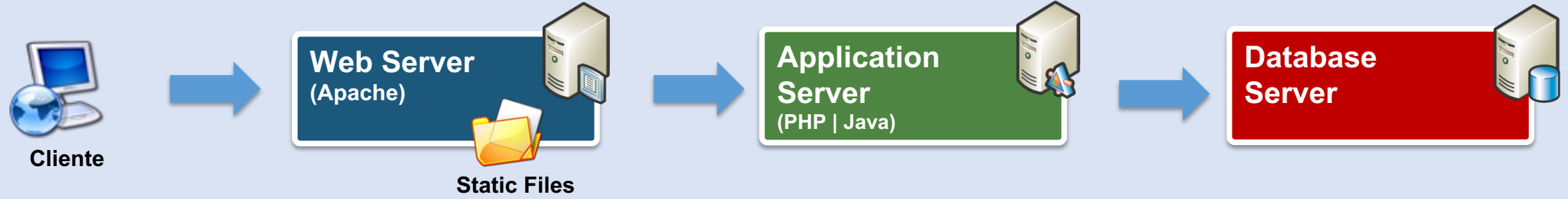
Fonte: [How Node.js overcome the problem of blocking of I/O operations?](#) (GeeksforGeeks)

Plataformas Web Tradicionais



Plataformas Tradicionais x Plataforma Node.js

Plataformas Web Tradicionais



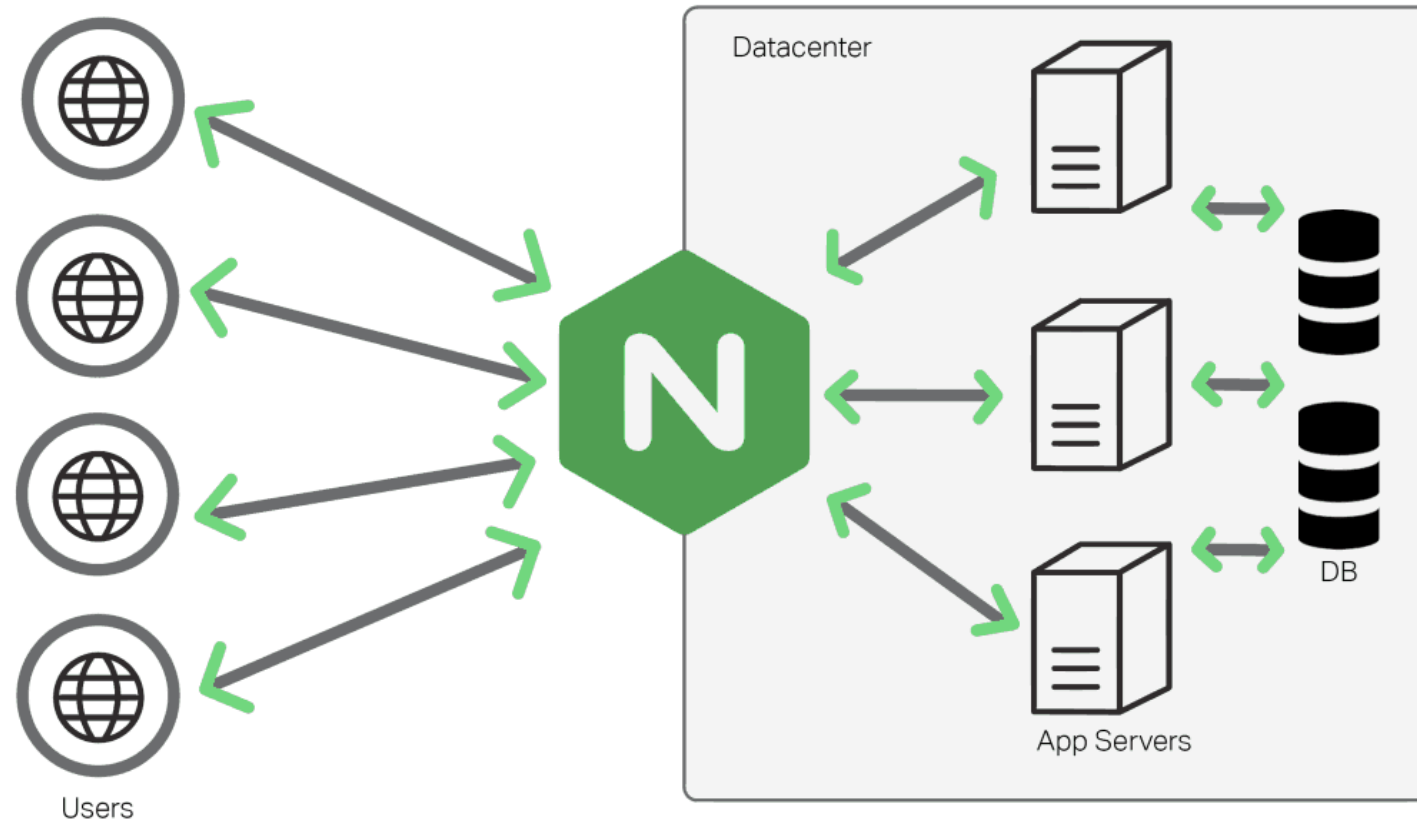
Plataforma Node.js



Plataforma Node.js Otimizada (NGINX + Proxy Reverso + Cluster Servers)



Arquitetura – Load Balancing e Proxy Reverso



Fonte:

- [What is a Reverse Proxy vs. Load Balancer? \(NGINX\)](#)
- [Nginx Reverse Proxy for Scalability](#)