



UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Fundamentos de NodeJS

QXD0020 - Desenvolvimento de Software para Web

Prof. Bruno Góis Mateus (brunomateus@ufc.br)

Agenda

- Introdução
- Criando um projeto Node
- Criando um servidor web com Node

Introdução



Introdução

Node

- É uma cross-platform runtime de código aberto que permite que desenvolvedores criem aplicações server-side em JS
- Executada “diretamente” no sistema operacional, fora do contexto do navegador
- Prover suporte a API mais tradicionais dos sistemas operacionais
 - Ex: HTTP, FileSystem

Introdução

História

- Enquanto a web tem 30 anos, JavaScript 26, Node tem apenas 12 anos
- Antes do sucesso do Node, a Netscape havia investido no LiveWire, o ambiente capaz de criar páginas web dinâmicas usando JavaScript no server-side, no entanto não obteve sucesso
- Aplicações server-side com JavaScript se popularizam a partir da introdução do Node.js
 - Fator decisivo: Timing
 - JavaScript passou ser utilizado em aplicações de maior porte graças a Web 2.0. Ex: Flickr, Gmail, etc.
 - Engine JavaScript melhoraram consideravelmente devido a competição entre navegadores
- Node usa a V8 ou Chrome V8, uma engine open-source JavaScript do Projeto Chromium que evoluiu bastante devido a essa competição

Introdução

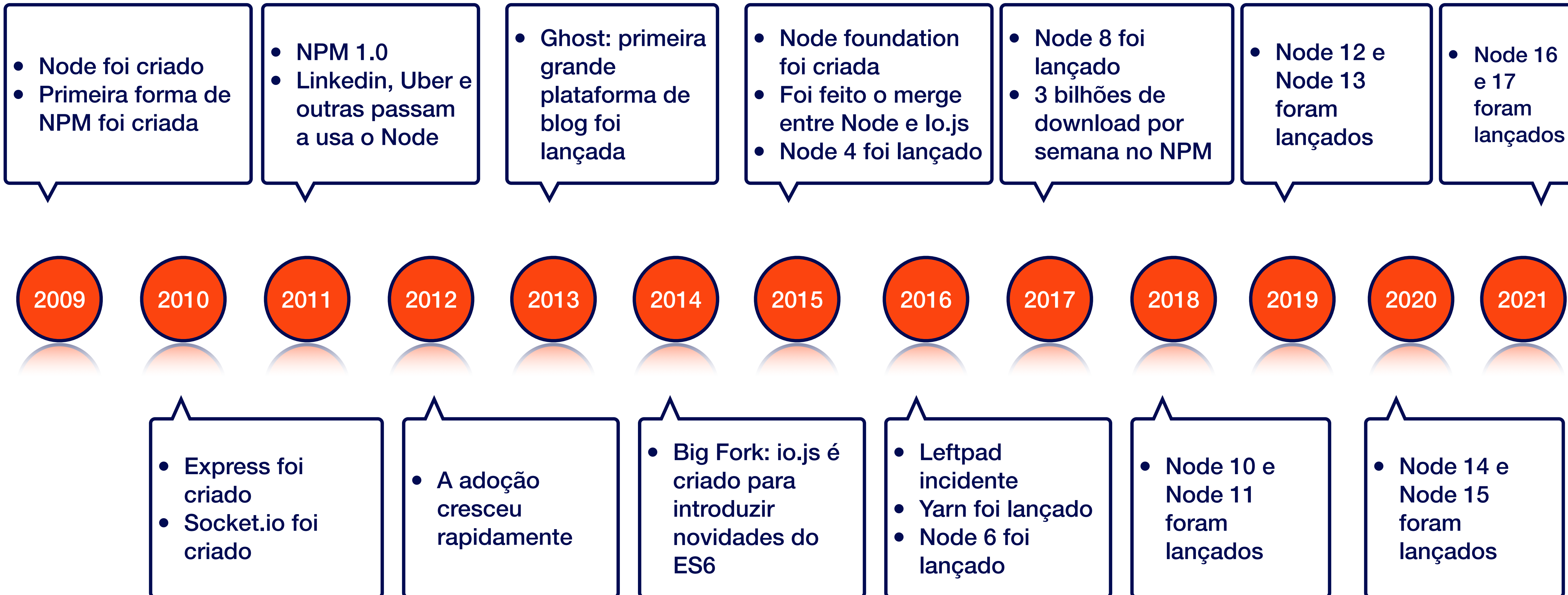
História - v8

- Engine de alto desempenho JavaScript e WebAssembly
- Escrita em C++
- Usada no Chrome e no Node entre outros projetos
- Compila e executa código JS, gerencia a alocação de memória e realiza a desalocação de objetos não necessário (*garbage collector*)



Introdução

História - Timeline



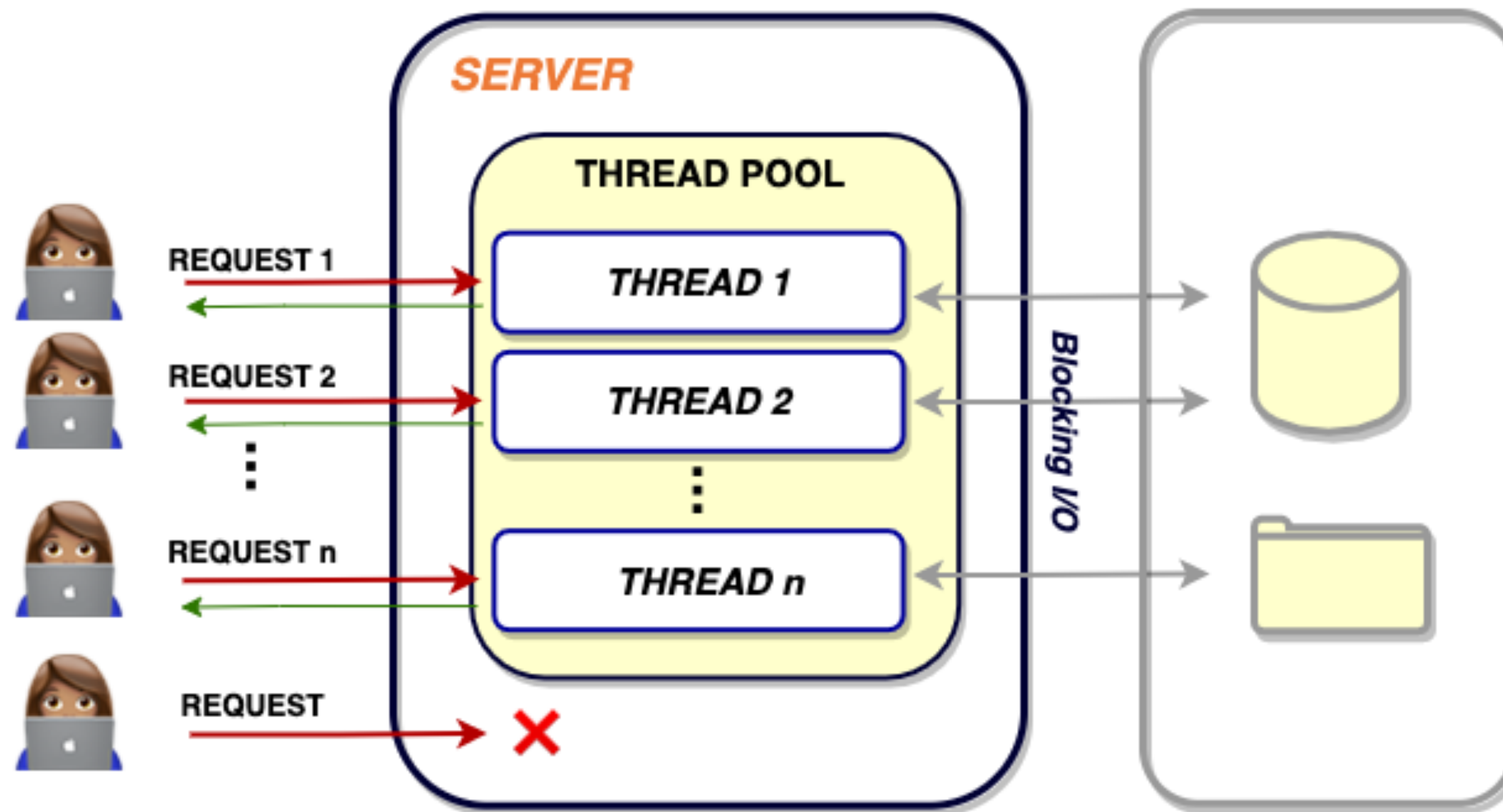
Introdução

Características

- Node.js app são executadas em um único processo
 - Não é necessária a criação de uma thread para cada requisição
- Fornece um conjunto de operações primitivas de I/O assíncronas
 - Evita que códigos de maneira geral códigos bloqueantes
- Escalável e mais simples de debugar, não há concorrência entre threads
- Novidades do ESMA Script podem ser usadas sem problemas já que o usuário possui o controle do ambiente de execução
 - No front-end sem node dependemos dos navegadores

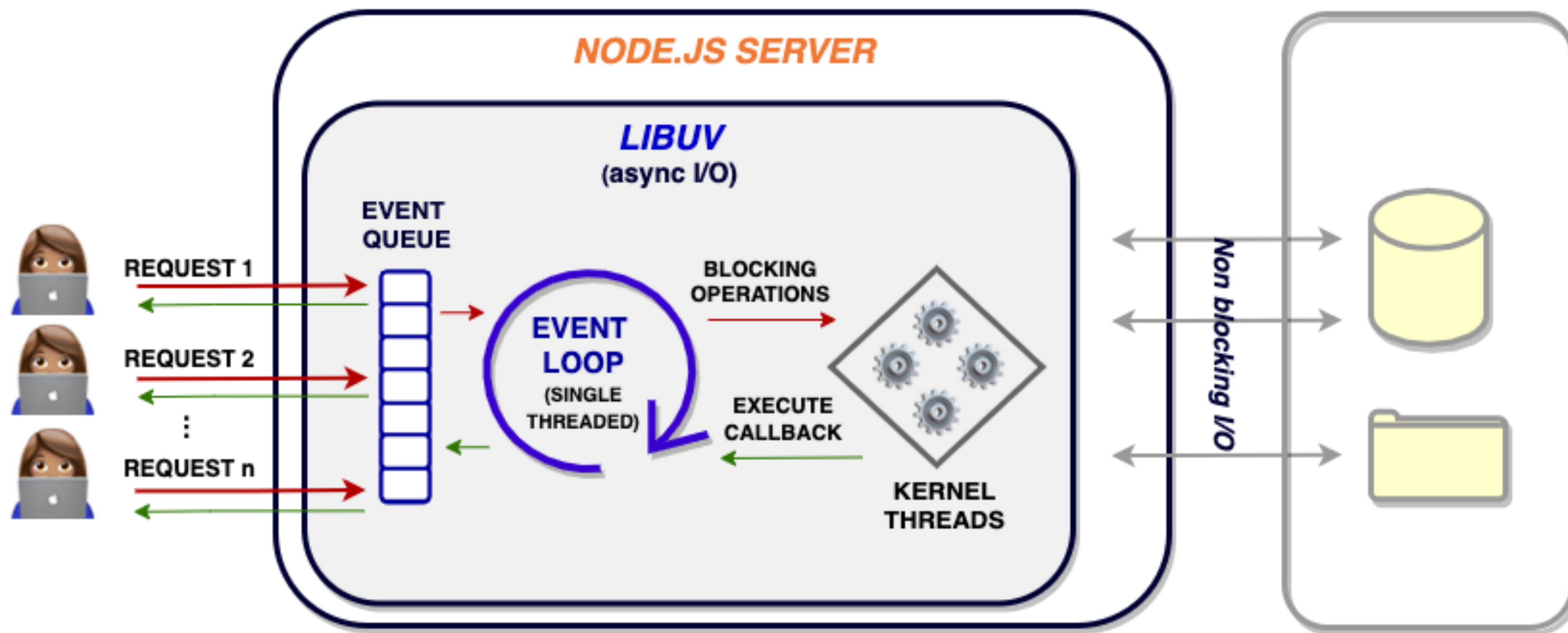
Introdução

Características



Introdução

Características



Introdução

Vantagens

- Excelente desempenho e escalável
- Escrito em Js, familiar para desenvolvedores Web
- Grande comunidade de usuários e desenvolvedores
- O gerenciador de pacote do Node, NPM, prover acesso a diversas bibliotecas reusáveis
 - Gerenciamento de dependências
- Portável, disponível para Windows, macOS, Linux, Solaris, FreeBSD, OpenBSD, WebOS, and NonStop OS

Criando um projeto Node



Criando um projeto Node

Hello World

- O seu primeiro programa em Node

```
console.log("Olá mundo");
```

```
node app.js
```

Criando um projeto Node

Módulos nativos

- [assert](#)
- [buffer](#)
- [child_process](#)
- [console](#)
- [cluster](#)
- [crypto](#)
- [dgram](#)
- [dns](#)
- [events](#)
- [fs](#)
- [http](#)
- [http2](#)
- [https](#)
- [net](#)
- [os](#)
- [path](#)
- [perf_hooks](#)
- [process](#)
- [querystring](#)
- [readline](#)
- [repl](#)
- [stream](#)
- [string_decoder](#)
- [timers](#)
- [tls](#)
- [tty](#)
- [url](#)
- [util](#)
- [v8](#)
- [vm](#)
- [wasi](#)
- [worker](#)
- [zlib](#)

Criando um projeto Node

Módulos nativos

Nome	Descrição
<u>console</u>	Prover um console para debug
<u>events</u>	Prover uma API para o gerenciamento de eventos
<u>fs</u>	Prover uma API para interagir com o sistema de arquivos
<u>http</u>	Prover uma implementação HTTP cliente/servidor
<u>os</u>	Prover propriedades e métodos utilitários relacionados ao sistema operacional
<u>path</u>	Prover utilitários para trabalhar com path e diretórios
<u>querystring</u>	Prover utilitários para “ <i>parsear</i> ” e formatar URL de string de consulta (querystring)

Criando um projeto Node

Módulos nativos

Nome	Descrição
<u>repl</u>	Prover uma implementação Read-Eval-Print-Loop (REPL) disponível como uma versão standalone, mas que também pode ser adicionada a outras aplicações
<u>timers</u>	Prover funções para agendar execuções de funções em um período futuro
<u>url</u>	Prover utilitários para resolução e “parseamento” de URL

Criando um projeto Node

NPM

- Node Package Manager - Gerenciador de pacotes do Node
- Inicialmente era uma maneira de fazer download e gerenciar as dependências
- Atualmente é também utilizado em projetos front-end
- Possui mais de 1.3 milhões de pacotes disponíveis
 - Maior repositório de software do mundo



Criando um projeto Node

- Para iniciar um projeto **node**, é necessário criar um arquivo chamado **package.json**
 - Lista todas as dependências do projeto e suas versões
 - Torna o processo de build reproduzível e portanto mais fácil de compartilhar com outros desenvolvedores
 - Deve conter pelo menos o atributo **name** e **version**
- A maneira mais simples de criar esse arquivo é usando o comando:
 - **\$ npm init --yes**

Criando um projeto Node

```
{
  "name": "my package",
  "description": "",
  "version": "1.0.0",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "repository": {
    "type": "git",
    "url": "https://github.com/monatheoctocat/my_package.git"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/monatheoctocat/my_package/issues"
  },
  "homepage": "https://github.com/monatheoctocat/my_package"
}
```

Nome do diretório

Informação contida no README ou string vazia

Sempre 1.0.0

Script de test vazio

No caso de se um repositório git

Sempre vazio

Sempre vazio

Por padrão ISC

Caso hospedado no GitHub

Caso hospedado no GitHub

Criando um projeto Node

NPM e suas funções

- Instalar e atualizar dependências
 - `$ npm install` ou `$ npm install <package-name>`
 - `$ npm update` ou `$ npm update <package-name>`
- Versionamento
- Execução de tarefas
 - Ex: Executar em produção, testar ...

Criando um projeto Node

Pacotes populares

ex

Express



Socket.io

asynC

UNDERSCORE.JS

AXIOS



Fastify



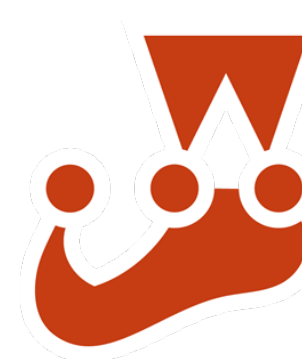
Rxjs



Lodash



Ramda



Jest



ESLint



Nodemon



Dotenv



Yargs



Passport



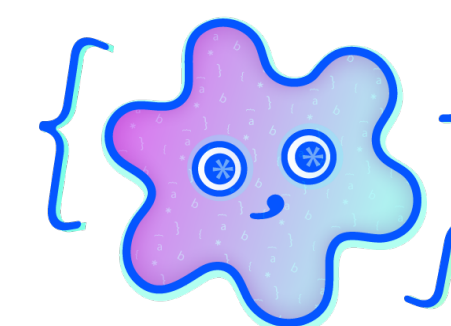
Nodemailer



Mongoose



date-fns

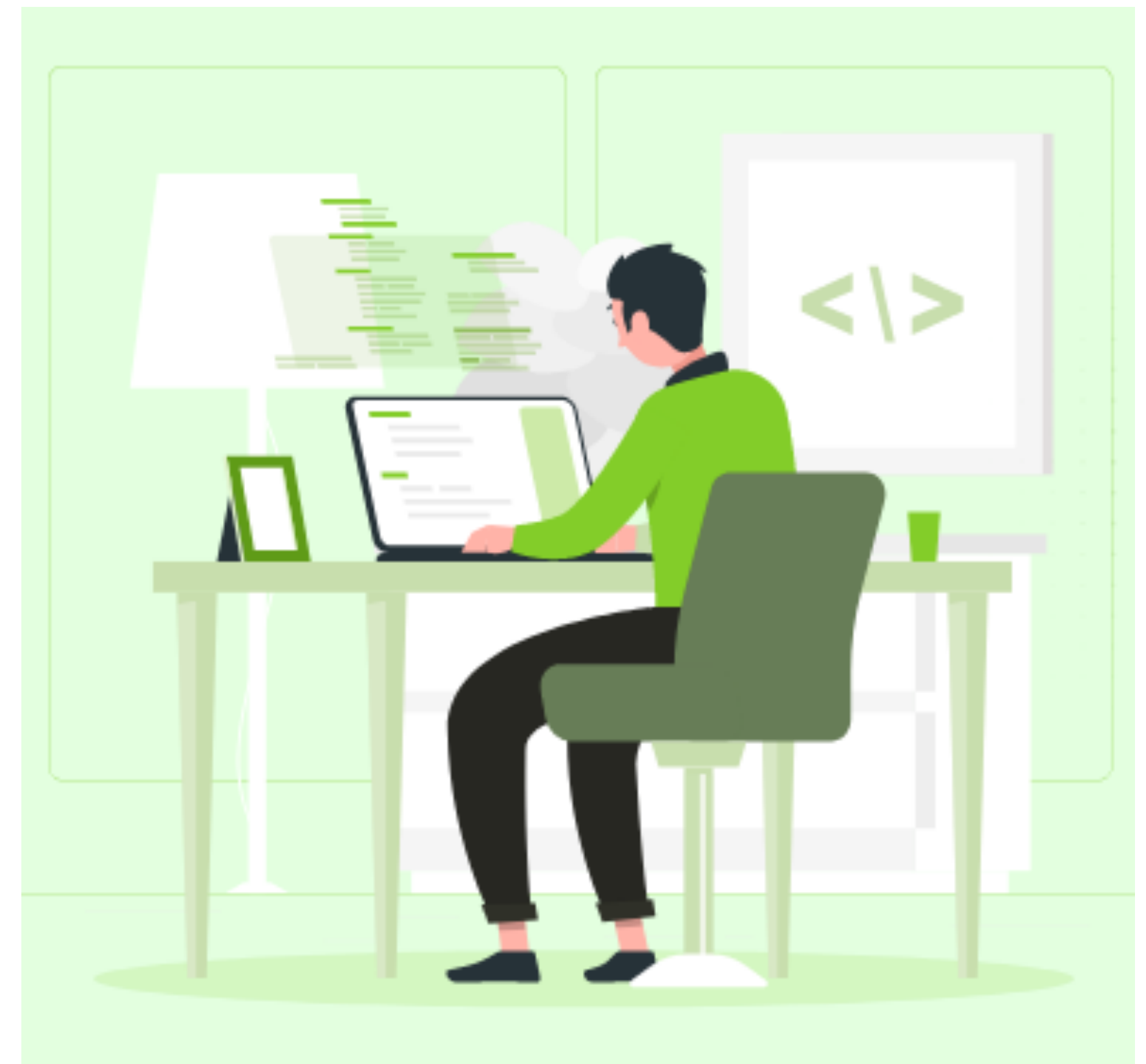


glob



[Lista completa contendo 40 pacotes](#)

Criando um servidor web com Node



Referências

- <https://nodejs.dev/learn>
- <https://leanylabs.com/blog/npm-packages-for-nodejs/>
- <https://flaviocopes.com/node-core-modules/>
- <https://docs.npmjs.com/>
- <https://scoutapm.com/blog/nodejs-architecture-and-12-best-practices-for-nodejs-development>
- <https://leanylabs.com/blog/npm-packages-for-nodejs/>
- <https://ctrly.blog/nodejs-layered-architecture/>

Por hoje é só