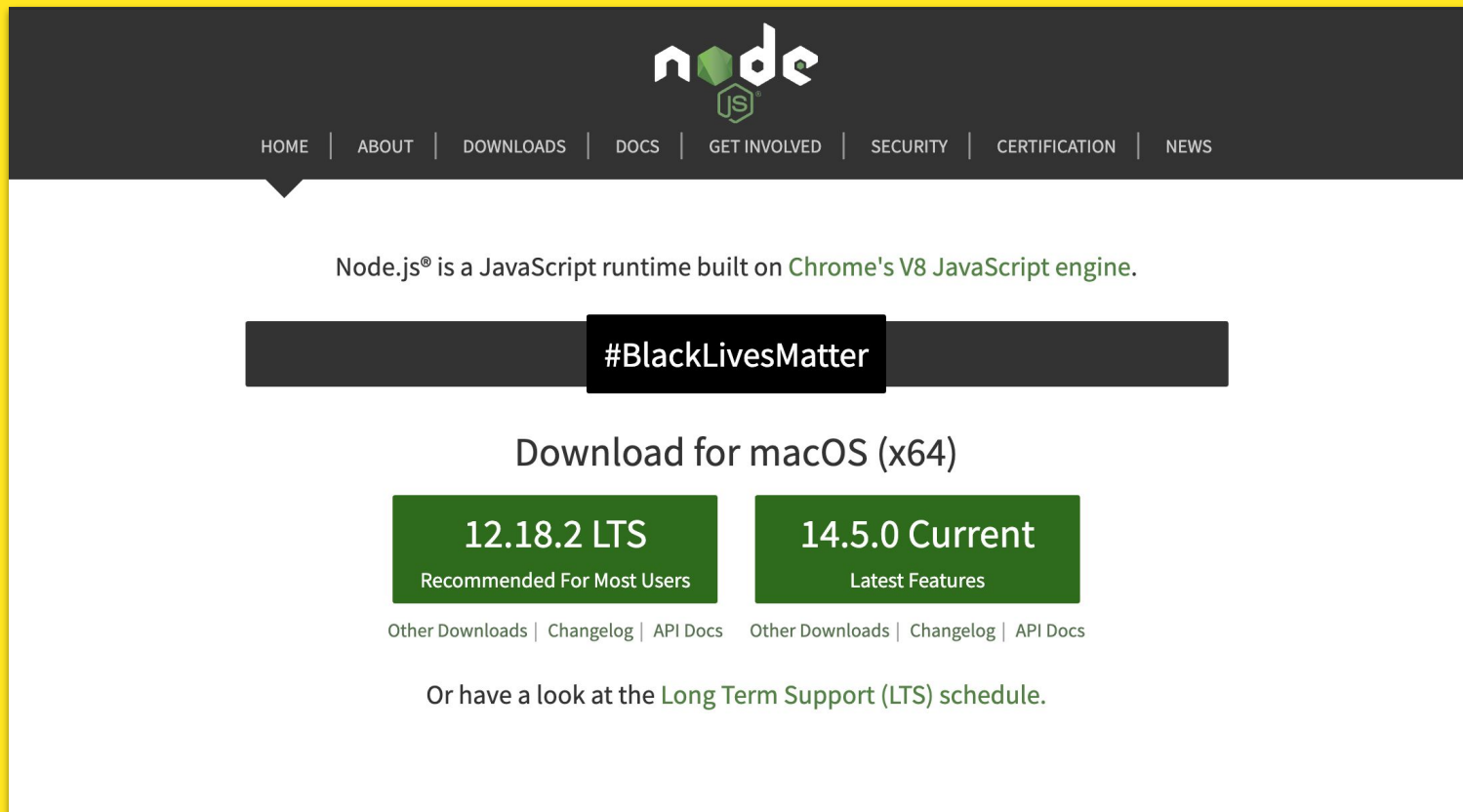




# Node, NPM e JSDOM

O que é **Node**?

# O que é Node?



The image is a screenshot of the Node.js website homepage. At the top, there is a dark grey navigation bar with the Node.js logo in the center. Below the logo, a horizontal menu contains links: HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, CERTIFICATION, and NEWS. The main content area is white. It starts with the text "Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine." followed by a dark grey banner with the text "#BlackLivesMatter" in white. Below this, the text "Download for macOS (x64)" is centered. Underneath, there are two green buttons: the left one says "12.18.2 LTS" and "Recommended For Most Users", and the right one says "14.5.0 Current" and "Latest Features". Below these buttons, there are two sets of links: "Other Downloads | Changelog | API Docs" for the LTS version and "Other Downloads | Changelog | API Docs" for the current version. At the bottom, it says "Or have a look at the Long Term Support (LTS) schedule." In the bottom right corner of the entire image, outside the website screenshot, is a large black icon resembling a code editor's "Run" button, consisting of a less-than sign, a capital L, and a greater-than sign.

node

HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | CERTIFICATION | NEWS

Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

#BlackLivesMatter

Download for macOS (x64)

**12.18.2 LTS**  
Recommended For Most Users

**14.5.0 Current**  
Latest Features

Other Downloads | Changelog | API Docs    Other Downloads | Changelog | API Docs

Or have a look at the Long Term Support (LTS) schedule.

<L>

Alguns arquivos são necessários para um projeto  
que utiliza **Node**

# Já sabemos como funciona o **package.json**

```
{  
  "name": "jsdom",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "dependencies": {},  
  "devDependencies": {},  
  "scripts": {},  
  "keywords": [],  
  "author": "Gabriel Ramos",  
  "license": "ISC"  
}
```

# Mas vamos revisar

- Nome
- Versão
- Descrição
- Arquivo principal
- Dependências
- Dependências de desenvolvimento
- Scripts
- Palavras-chave
- Autores
- Licença

```
{  
  "name": "jsdom",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "dependencies": {},  
  "devDependencies": {},  
  "scripts": {},  
  "keywords": [],  
  "author": "Gabriel Ramos",  
  "license": "ISC"  
}
```

Que comando utilizamos para **instalar** um novo pacote?

Que comando utilizamos para **instalar** um novo pacote?

```
npm install [nome-do-pacote]
```



O que é o **NPM**?

# O que é o NPM?

The screenshot displays the NPM website interface. At the top, there's a navigation bar with links for Products, Pricing, Documentation, and Community. Below this is a search bar with the NPM logo and a search button. The main content area is divided into three columns. The left column, titled 'Popular libraries', lists several popular packages: lodash, react, chalk, request, commander, moment, express, react-dom, tslib, and prop-types. The middle column, titled 'Discover packages', features a grid of categories: Front-end, Back-end, CLI, Documentation, CSS, Testing, IoT, Coverage, Mobile, Frameworks, Robotics, and Math. The right column, titled 'By the numbers', provides statistics on the number of packages (1,333,551), downloads in the last week (19,624,648,457), and downloads in the last month (81,441,561,111).

♥ Name Picker Malfunction

Products Pricing Documentation Community

**npm** Search packages Search

### 🔥 Popular libraries

- lodash
- react
- chalk
- request
- commander
- moment
- express
- react-dom
- tslib
- prop-types

### 🔍 Discover packages

- Front-end
- Back-end
- CLI
- Documentation
- CSS
- Testing
- IoT
- Coverage
- Mobile
- Frameworks
- Robotics
- Math

### 📊 By the numbers

Packages  
**1,333,551**

Downloads · Last Week  
**19,624,648,457**

Downloads · Last Month  
**81,441,561,111**

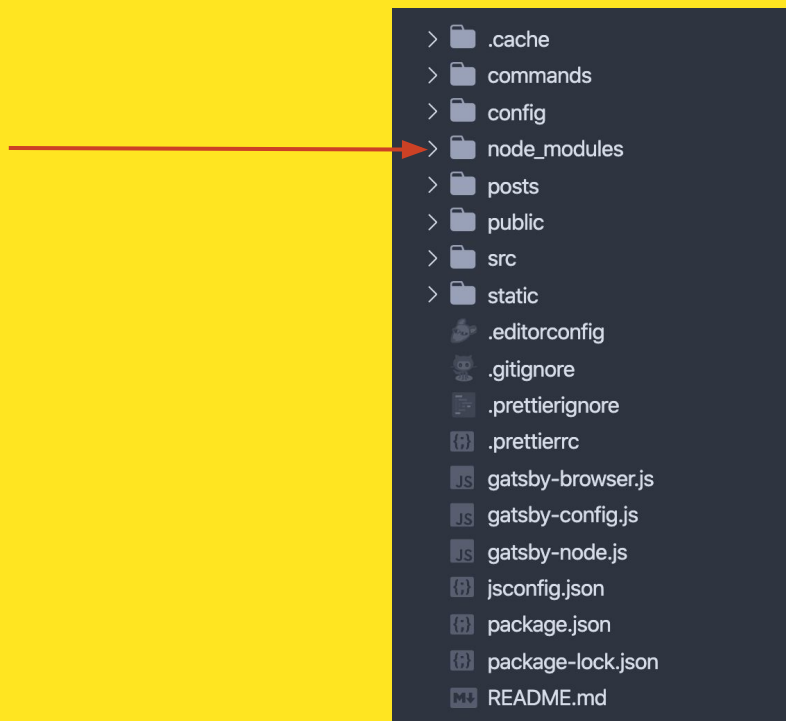


Então quando executamos comandos como esse

```
npm install [nome-do-pacote]
```

Estamos informando ao gerenciador de pacotes do node (**N**ode **P**ackage **M**anager) o que ele deve fazer

Quando instalamos um projeto, todas suas dependências ficarão na pasta **node\_modules**



Não devemos **versionar** (colocar no nosso repositório) o conteúdo dessa pasta pois ela contém muitos **arquivos específicos ao sistema operacional**.

A ideia é que os arquivos **package.json** sejam como uma "receita". Assim, sempre que alguém clonar nosso repositório e quiser instalar nosso projeto, basta executar **npm install** e instalar tudo localmente.

querySelector

getElementById

querySelectorAll

Em um projeto com **Node**, por não estarmos trabalhando com um navegador, não conseguimos utilizar alguns métodos que já conhecemos

getElementByName

getElementsByTagName

SERÁ?

$$y = ax^2 + bx + c$$

$$(x_1, x_2) = \frac{-b \pm \Delta}{2a}$$



	30°	45°	60°
sin	$\frac{1}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{3}}{2}$
cos	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$
tan	$\frac{\sqrt{3}}{3}$	1	$\sqrt{3}$



# Ih, ó lá... O tal do JSDOM



**jsdom**

jsdom is a pure-JavaScript implementation of many web standards, notably the WHATWG [DOM](#) and [HTML](#) Standards, for use with Node.js. In general, the goal of the project is to emulate enough of a subset of a web browser to be useful for testing and scraping real-world web applications.

The latest versions of jsdom require Node.js v10 or newer. (Versions of jsdom below v16 still work with previous Node.js versions, but are unsupported.)

## Basic usage





"**jsdom** is a pure-JavaScript implementation of many **web standards**"

Boa parte da web é feita através de **especificações**

Dessa forma, cada **navegador** (ou plataforma)  
decide como implementar essas **especificações**

Vamo ver se funciona mesmo...

Vamos iniciar **um novo projeto** para testar e instalar o **jsdom**

```
npm init -y && npm install jsdom
```

Vamos criar um arquivo **index.js** e importar o **JSDOM**

```
const { JSDOM } = require('jsdom');
```

# Agora vamos criar uma string como se fosse nosso HTML

```
const { JSDOM } = require('jsdom');

const html = `
  <!DOCTYPE html>
  <html>
    <body>
      <p>meu paragrafo</p>
    </body>
  </html>
`;
```

Agora vamos criar uma instância do **JSDOM** usando o **HTML** que montamos

```
const { JSDOM } = require('jsdom');

const html = `
  <!DOCTYPE html>
  <html>
    <body>
      <p>meu paragrafo</p>
    </body>
  </html>
`;

const DOM = new JSDOM(html);
```

Isso nos retornará um elemento igual ao **DOM** do navegador.



Agora é só realizar as nossas consultas no **document**, como já conhecemos

```
const { JSDOM } = require('jsdom');

const html = `
  <!DOCTYPE html>
  <html>
    <body>
      <p>meu paragrafo</p>
    </body>
  </html>
`;

const DOM = new JSDOM(html);

const meuParagrafo = DOM.window.document.querySelector('p');
console.log(`Texto do parágrafo é ${meuParagrafo.textContent}`);
```



Agora execute esse arquivo pelo **terminal** e me diz o que apareceu aí pra vocês

```
node index.js
```

**JSDOM** nos permite trabalhar e simular um ambiente de **DOM** mesmo fora do navegador

OK... Mas o que eu faço com isso?

Você pode até não utilizar **JSDOM** nesse desafio...

Não é obrigatório e a ideia não é fazer vocês mudarem nada do que já desenvolveram. **Sério!**

Mas com certeza ainda vai esbarrar nele muitas  
vezes por aí no **mercado**

Principalmente quando falamos de **testes**

Então é um assunto que vale a pena ser comentado

:)

A light-colored dog is captured mid-jump, with its front paws raised high and its head tilted back. The entire image is overlaid with a semi-transparent yellow filter. The word "DÚVIDAS" is centered in the middle of the image in a bold, black, sans-serif font.

# DÚVIDAS



Issaê

# Links úteis:

- [JSDOM](#)
- [Entendendo o package.json](#)