



# Node.js

Backend ↔ Frontend

Prof. Fernando Raposo  
[ferfox@gmail.com](mailto:ferfox@gmail.com) - <https://ferfox1981.github.io/>

# Roteiro

- Apresentação
- Browsers
- Javascript
- Node
- Parte Prática
- Referências

# Quem sou eu?

- Formado pelo Centro de Informática - UFPE
- Mestrado em Engenharia de Software - CIn/UFPE
- Analista SERPRO (Serviço Federal de Processamento de Dados)
- Ex-professor UNIAESO

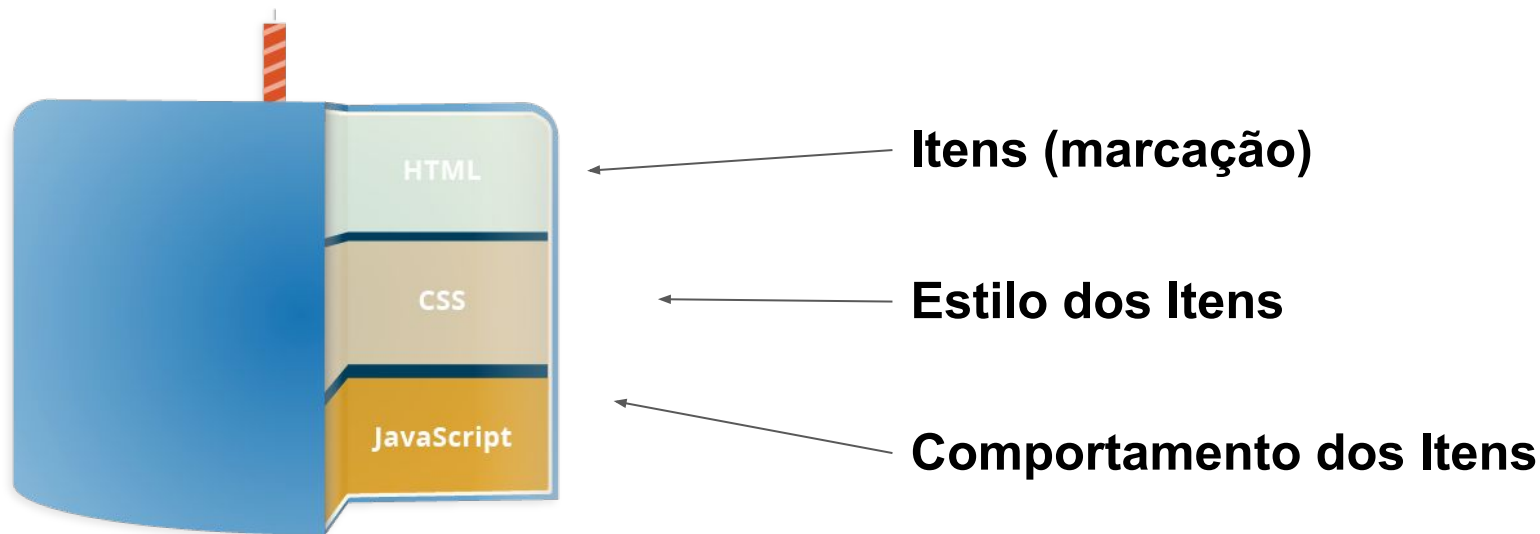
# Antes... uma retrospectiva acerca de Browsers

- Antigamente como o padrão eram sites estáticos, a web era “**paradona**”



# Javascript

- Javascript era utilizado para deixar esses Browsers “**paradões**” um pouco mais dinâmicos;
- Assim, havia a seguinte pilha de desenvolvimento:



# Javascript nos Browsers



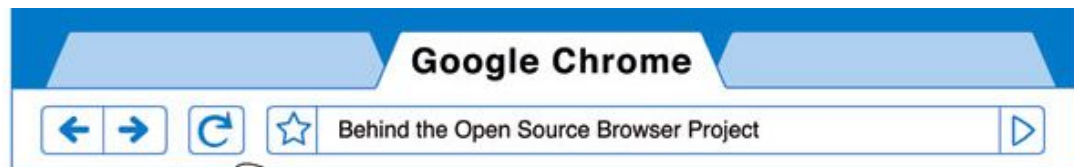
- Javascript foi feito pela **Netscape** (um browser que não existe mais);
- Conforme os sites foram evoluindo, coisas como abaixo ficaram muito comuns, o browser **congelava!!**



# Google Chrome

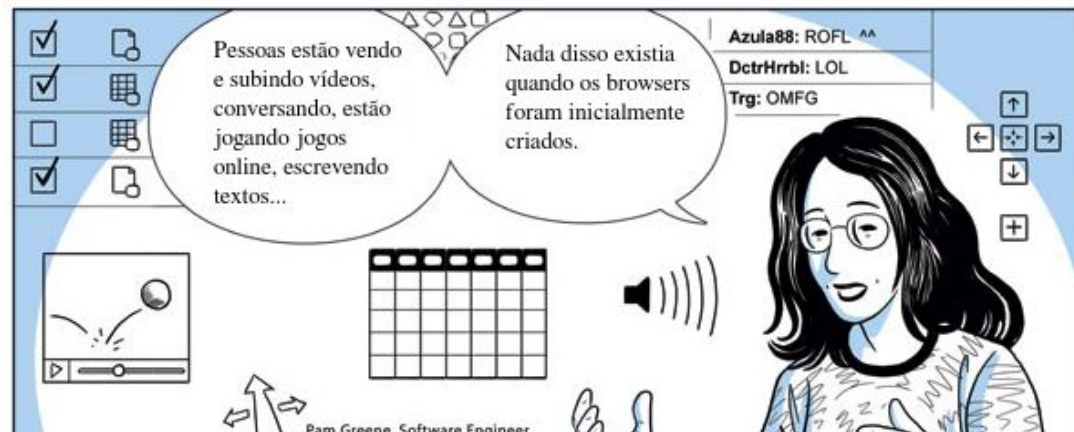
- Tudo começou a mudar quando o Google notou que os browsers não proviam mais **Webpages**, eles proviam **APLICAÇÕES**
- Dadas estas necessidades foi desenvolvido o **Google Chrome**
- O Google escreveu sua implementação do interpretador Javascript para o Chrome e chamou-a de **V8**;
- Há até um [quadrinho](#) que explica a criação do Chrome





Brian Rakowski,  
Product Manager

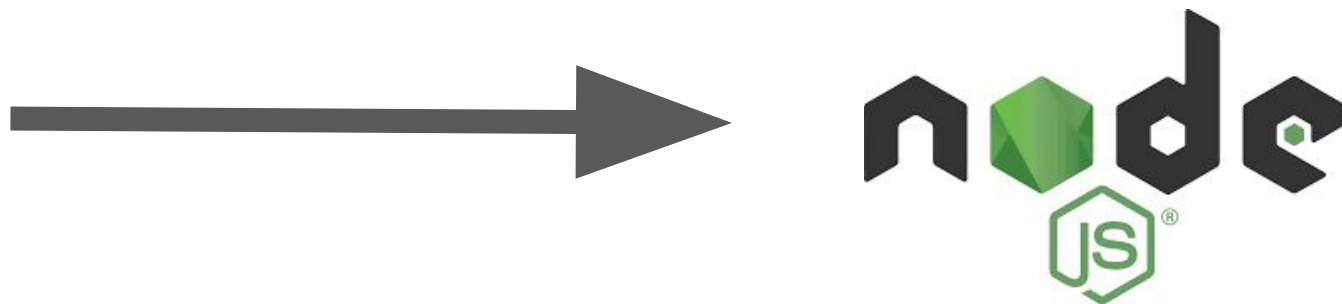
Hoje em dia o que usamos  
na web não são apenas  
páginas, mas sim aplicações





# Node

- De início o Chrome se mostrou **bem melhor** que seus concorrentes Internet Explorer (Chakra), Netscape... e o interpretador V8 era o seu diferencial;
- Então... E se extrairmos o V8 do Chrome e o utilizássemos como um Framework de desenvolvimento????????



# Node

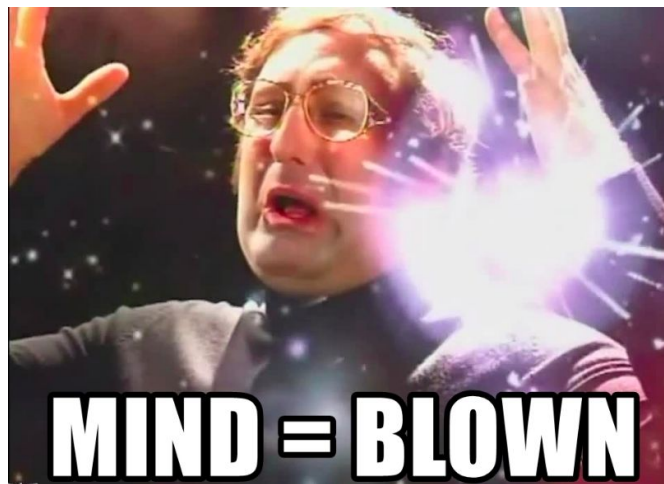
- De início o Chrome se mostrou **bem melhor** que seus concorrentes Internet Explorer (Chakra), Netscape... e o interpretador V8 era o seu diferencial;
- Então... E se extrairmos o V8 do Chrome e o utilizássemos como um Framework de desenvolvimento?????????



# Node



- Antigamente era muito comum a **separação** clara entre Frontend e Backend
  - No Frontend era utilizado: HTML + CSS + Javascript
  - No Backend usava-se JAVA, Python, C++, C#...
- Até aí os devs Front e Backend não se misturavam...
- Mas com o **Node** a mesma linguagem pode ser usada no Backend e no Frontend!



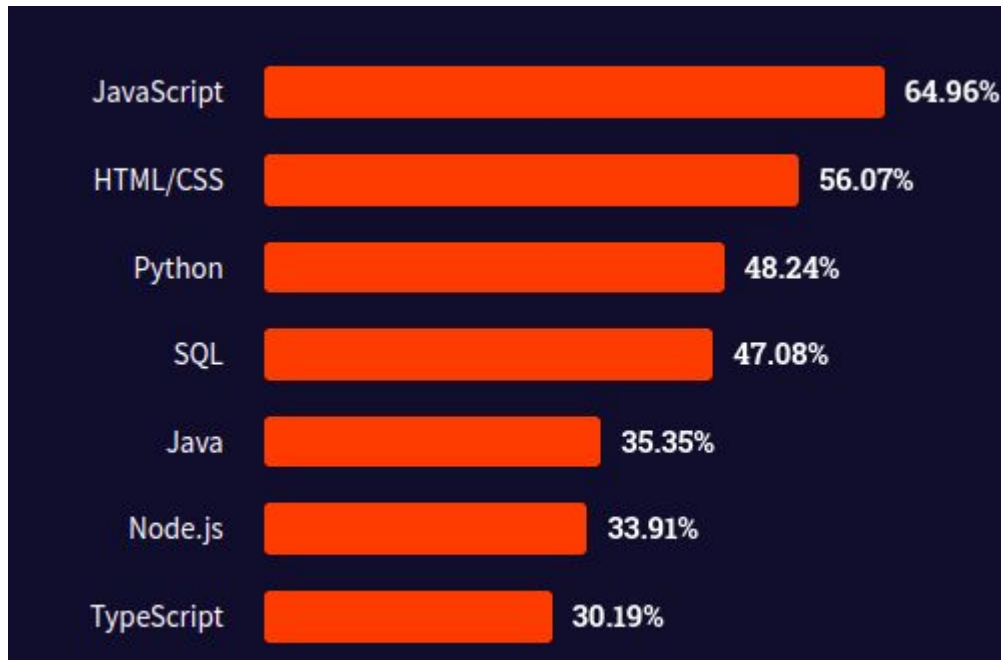
# Node

## Vantagens

- Poder usar Javascript no Backend e Frontend;
- Curva de aprendizado rápida;
- Comunidade muito [ativa](#);

## Desvantagens

- Programação assíncrona;
- Fracamente tipada;



# Node do Frontend

- Node no Frontend na verdade trata-se de embarcar Javascript numa página HTML;
- Vamos ver um exemplo dinâmico

Prática!



# index.html

```
<html>

  <body>

    <input type="button" onClick="teste()" value="teste">

  </body>

  <script>

    function teste(){

      console.log('teste');

    }

  </script>

</html>
```

# Javascript do Backend (Instalação do Node)

- Linux (a melhor opção): <https://nodejs.org/en/download/package-manager/>
- Windows: <https://nodejs.org/en/download/>



<https://nodejs.org/en/download/>



Download the Node.js source code or a pre-built installer for your platform, and start developing today.

**LTS**

Recommended For Most Users

**Current**

Latest Features



Windows Installer

node-v14.15.5-x64.msi



macOS Installer

node-v14.15.5.pkg



Source Code

node-v14.15.5.tar.gz

Windows Installer (.msi)

32-bit

64-bit

# Node (Testando)

- Abra agora um arquivo e escreva:

```
console.log(Oi Node)
```

- Salve-o como inicio.js
- E vá ao prompt de comando na pasta onde salvou e digite:

```
C:\Users\User\Desktop>node inicio.js  
Oi Node  
  
C:\Users\User\Desktop>_
```

- Observe que ao apertar F12 no browser e digitar o mesmo comando o resultado sai igual. É tudo Javascript! No Browser ou no servidor.



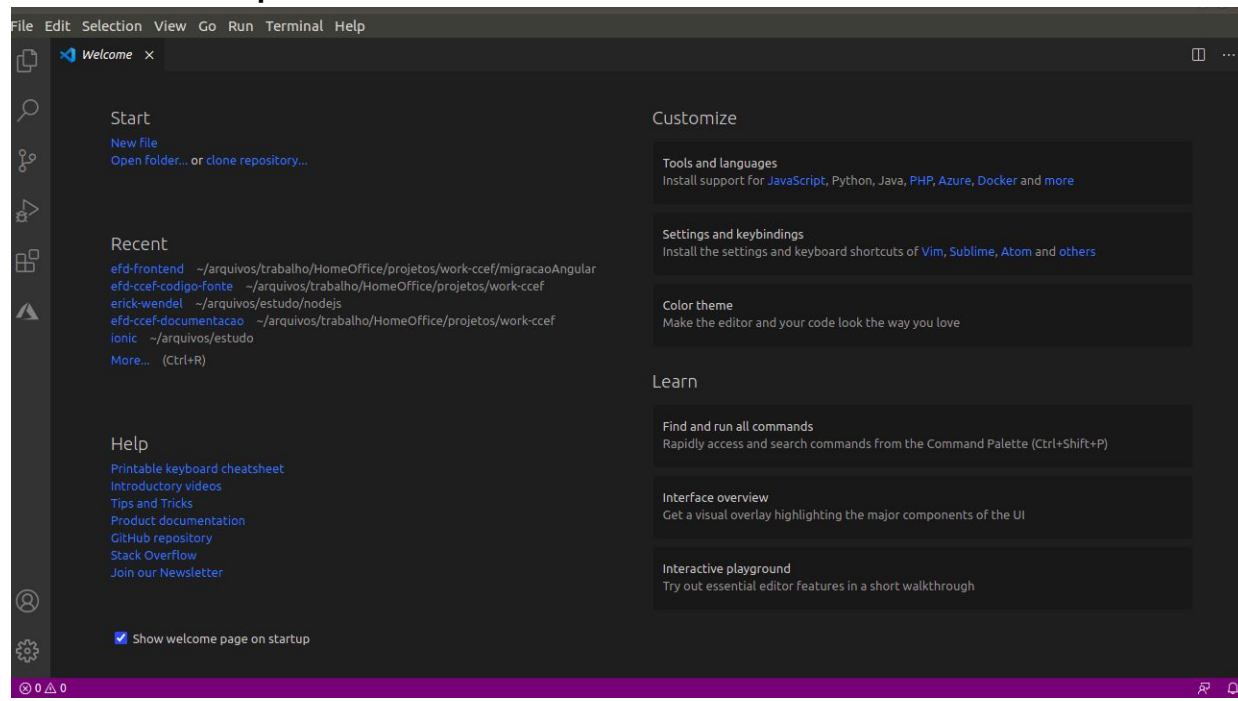
# Node

The screenshot shows a web browser's developer console with the following content:

- Elements** | **Console** | Sources | Network | Performance | Memory | Application | Security
- top | Filter | Default levels
- Some messages have been moved to the Issues panel. [View issues](#)
- Mixed Content:** The page at 'https://www.google.com/search?q=tabela+brasileiro&oq=tabela+brasil /search?q=tabela+bra.../0fnk7q;mt;fp;1;:1 eiro&ags=\_\_t=ubuntu&sourceid=chrome&ie=UTF-8#sie=lg;/g/1lfmzksb3y;2;/m/0fnk7q;mt;fp;1;:' was loaded over HTTPS, but requested an insecure element 'http://t2.gstatic.com/images?q=tbn:ANd9GcTapUIVZchwbZ6zEEpsVA1Yfeqn8sdU3v8tZ8MpNAKAQmsGZLtlDcVvfLB4BA'. This request was automatically upgraded to HTTPS, For more information see <https://blog.chromium.org/2019/10/no-more-mixed-messages-about-https.html>
- Mixed Content:** The page at 'https://www.google.com/search?q=tabela+brasileiro&oq=tabela+brasil /search?q=tabela+bra.../0fnk7q;mt;fp;1;:1 eiro&ags=\_\_t=ubuntu&sourceid=chrome&ie=UTF-8#sie=lg;/g/1lfmzksb3y;2;/m/0fnk7q;mt;fp;1;:' was loaded over HTTPS, but requested an insecure element 'http://t3.gstatic.com/images?q=tbn:ANd9GcQyMs7V4PbM4Sx1UctXMmujFILULUru\_gjwzP\_5DNbPyClxGNiWtNAeWb0Rkw'. This request was automatically upgraded to HTTPS, For more information see <https://blog.chromium.org/2019/10/no-more-mixed-messages-about-https.html>
- Mixed Content:** The page at 'https://www.google.com/search?q=tabela+brasileiro&oq=tabela+brasil /search?q=tabela+bra.../0fnk7q;mt;fp;1;:1 eiro&ags=\_\_t=ubuntu&sourceid=chrome&ie=UTF-8#sie=lg;/g/1lfmzksb3y;2;/m/0fnk7q;mt;fp;1;:' was loaded over HTTPS, but requested an insecure element 'http://t2.gstatic.com/images?q=tbn:ANd9GcR5N8LpmZRvdaizTeS4NsouenUJFRngWf2\_0Gyped825GGv8jtJEBI\_ERTgDw'. This request was automatically upgraded to HTTPS, For more information see <https://blog.chromium.org/2019/10/no-more-mixed-messages-about-https.html>
- Mixed Content:** The page at 'https://www.google.com/search?q=tabela+brasileiro&oq=tabela+brasil /search?q=tabela+bra.../0fnk7q;mt;fp;1;:1 eiro&ags=\_\_t=ubuntu&sourceid=chrome&ie=UTF-8#sie=lg;/g/1lfmzksb3y;2;/m/0fnk7q;mt;fp;1;:' was loaded over HTTPS, but requested an insecure element 'http://t2.gstatic.com/images?q=tbn:ANd9GcTIMIcaWkezpUYagWb1Vfd\_419MB4LkCi5NJ2tEnuFDGcnmMUp-tTfATibLYA'. This request was automatically upgraded to HTTPS, For more information see <https://blog.chromium.org/2019/10/no-more-mixed-messages-about-https.html>
- Failed to load resource: net::ERR\_CONNECTION\_CLOSED** [qqgNxmlBsBoBpQN8\\_192x108.jpg:1](#)
- > console.log('Oi Node')
- Oi Node
- < undefined

# VS Code

- Ambiente de desenvolvimento [Multi-plataforma](#)
- Desenvolvido pela Microsoft



# Primeiro Projeto Node

- Abrir o **VS Code** numa pasta, abrir o terminal integrado e digitar

```
npm init -y
```

- O projeto será criado no modo expresso
- Observe que um arquivo `package.json` é criado.
- Este arquivo é um descritor do seu projeto
  - Dependências
  - Atalhos
  - Etc...
- OBS: A criação do projeto em passo-a-passo se dá com:

```
npm init
```

# Primeiro Programa Node

- Criar um arquivo `teste.js` na mesma pasta
- Escrever:

```
console.log('Olá Javascript!')
```

- Para executar, vá no terminal e digite

```
node teste.js
```

- OBS: O ponto e vírgula é opcional!

# Variáveis

- Antes de 2015 variáveis eram declaradas como **var**

```
var a = 10  
var b = 'Teste'
```

- O ECMAScript 2015 propôs dois novos tipos **let** e **const**

```
let a = 10  
const b = 50
```

- Obviamente, redefinir uma variável `const` dará erro de compilação.

# Variáveis

- Podemos ter coisas mais complexas, como por exemplo um objeto **JSON**.

```
let a = { nome: 'Fernando', idade: 29}
```

- O acesso é exatamente como o de um objeto:

```
console.log(a[ 'nome' ])
```

- Mas nada nos impede de criar um objeto de fato!

```
class Aluno {
```

```
}
```

```
new Aluno()
```



# Servidor

- O Node é uma linguagem extremamente simples e **produtiva**;
- Com algumas linhas de código podemos criar um servidor Web operacional;
- Existem diversas bibliotecas que podem evitar que se reinvente a roda;
- Bibliotecas podem ser acessadas pelo [NPM](#).
- Uma biblioteca interessante é a [Express](#).

**npm i express**



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

app.get('/', function (req, res) {
  res.send('Hello World')
})

app.listen(3000)
```

# Servidor

- Crie um arquivo index.js e cole exatamente o código que está na página do Express.

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

const app = express()

app.get('/', function (req, res) {

  res.send('Hello World')

})

app.listen(3000)
```

- Execute o código abaixo para levantar o servidor na porta 3000

**node index.js**

- Vá no browser no endereço <http://localhost:3000>



# HTML

Nome:

Enviar sua mensagem

```
<!DOCTYPE html>
<html>
<body>
  <form action="http://localhost:3000/uniaeso/" method="get">
    <div>
      <label for="nome">Nome:</label>
      <input type="text" id="nome" name="nome" />
    </div>

    <div class="button">
      <button type="submit">Enviar sua mensagem</button>
    </div>
  </form>
</body>
</html>
```

- Agora crie um arquivo teste.html com o conteúdo acima

# Node

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

app.get('/uniaeso', function (req, res) {
  let nome = req.query['nome']
  console.log(nome)
  res.send('Servidor Recebeu: '+nome)
})

app.listen(3000)
```

- Altere o index.js para este código e veja o que acontece ao apertar o botão.

# Conclusão

- Com o Node podemos usar a mesma linguagem no Backend e no Frontend;
- Ecossistema rico;
- É uma linguagem fácil de aprender, MAS...
  - Promises (programação assíncrona)
  - Arrow functions
  - Callbacks
- Pode ser usado em associação com diversos outros Frameworks Web:
  - Angular
  - React
  - Ionic
  - Vue
  - ...

# Indo Além

- [Programando na nuvem \(Replit\)](#)
- [Curso Node.js Microsoft \(gratuito\)](#)
- [Imersão em desenvolvimento de APIs com Node.js \(Erick Wendel\)](#)

