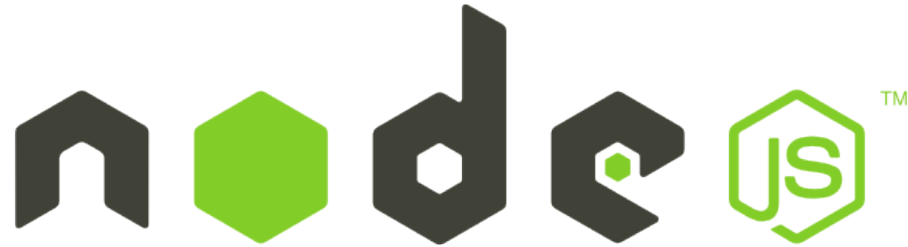


Desenvolvendo em





Sobre mim

- Raul Sena Ferreira
- Ciência da Computação (UFRRJ)
 - Centro Acadêmico de CComp
 - Representante Discente
- Desenvolvedor Web (COPPE-UFRRJ)
- Freelancer nas horas vagas
- I.A. aplicada, Data Science, Big Data, IoT, Web Development, Web Security



Sobre mim

- Menção Honrosa -> XIX Maratona Programação - ACM/SBC
- Entre os vencedores -> 1º Desafio em Big Data - EMC²/UFRJ
- Entre os 40 selecionados -> II Hackathon de Dados Educacionais - INEP (Brasília-DF)
- 3º Lugar -> primeira etapa Launching People - Samsung



Introdução

- Criado por Ryan Dahl, no início de 2009
- Mantido posteriormente pela Joyent
- Intuito inicial era o de se criar aplicações baseados em features do gmail (push capabilities)



Quem usa ?



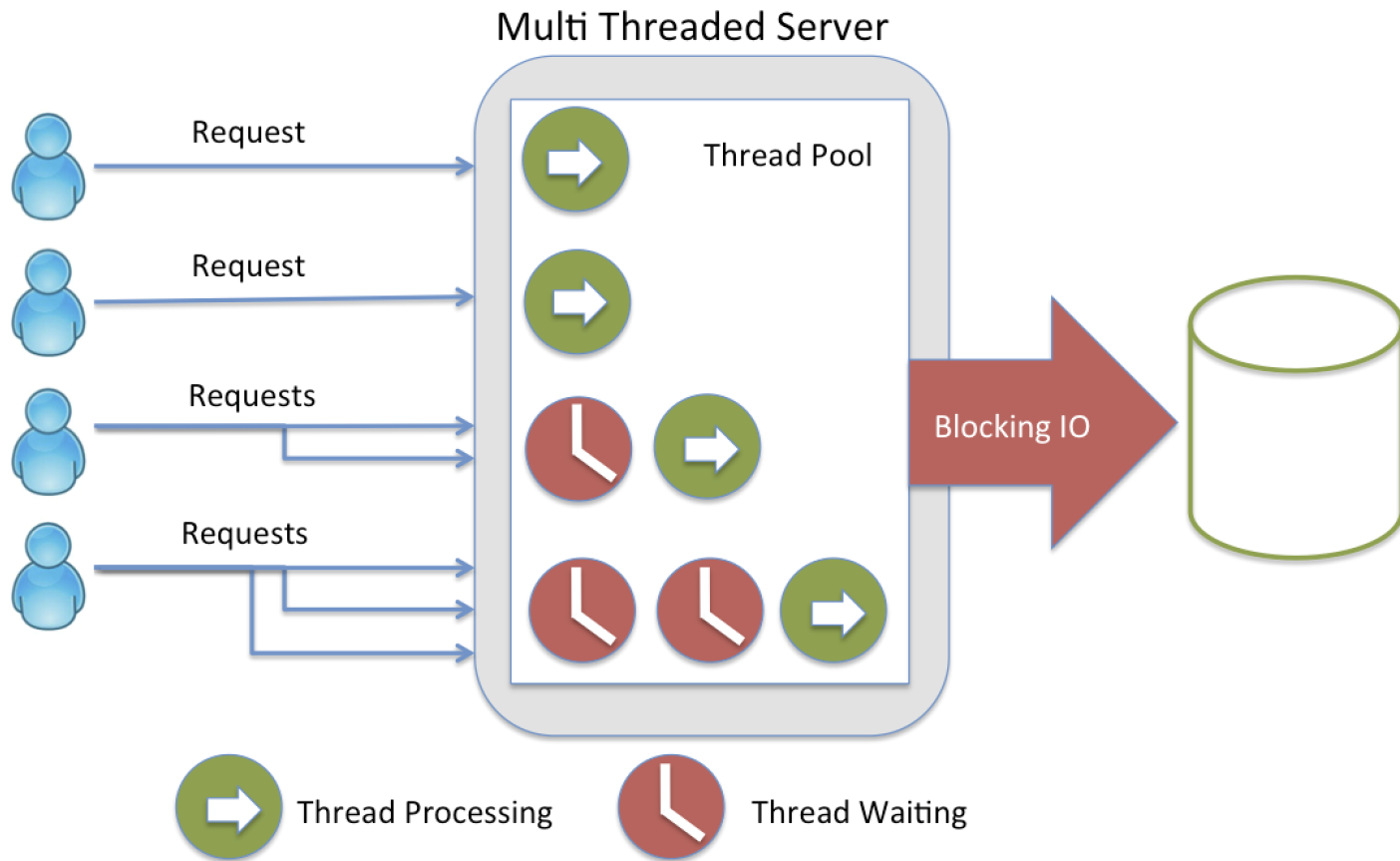


O que é ?

- Framework Web server-side assíncrono
- Escrito em C++ e javascript V8 (Engine)
- Orientado a evento
- Single thread
- Suporta milhões de conexões simultâneas
- Feito para aplicações em tempo real com alta concorrência e uso intensivo de dados



Como funciona ?



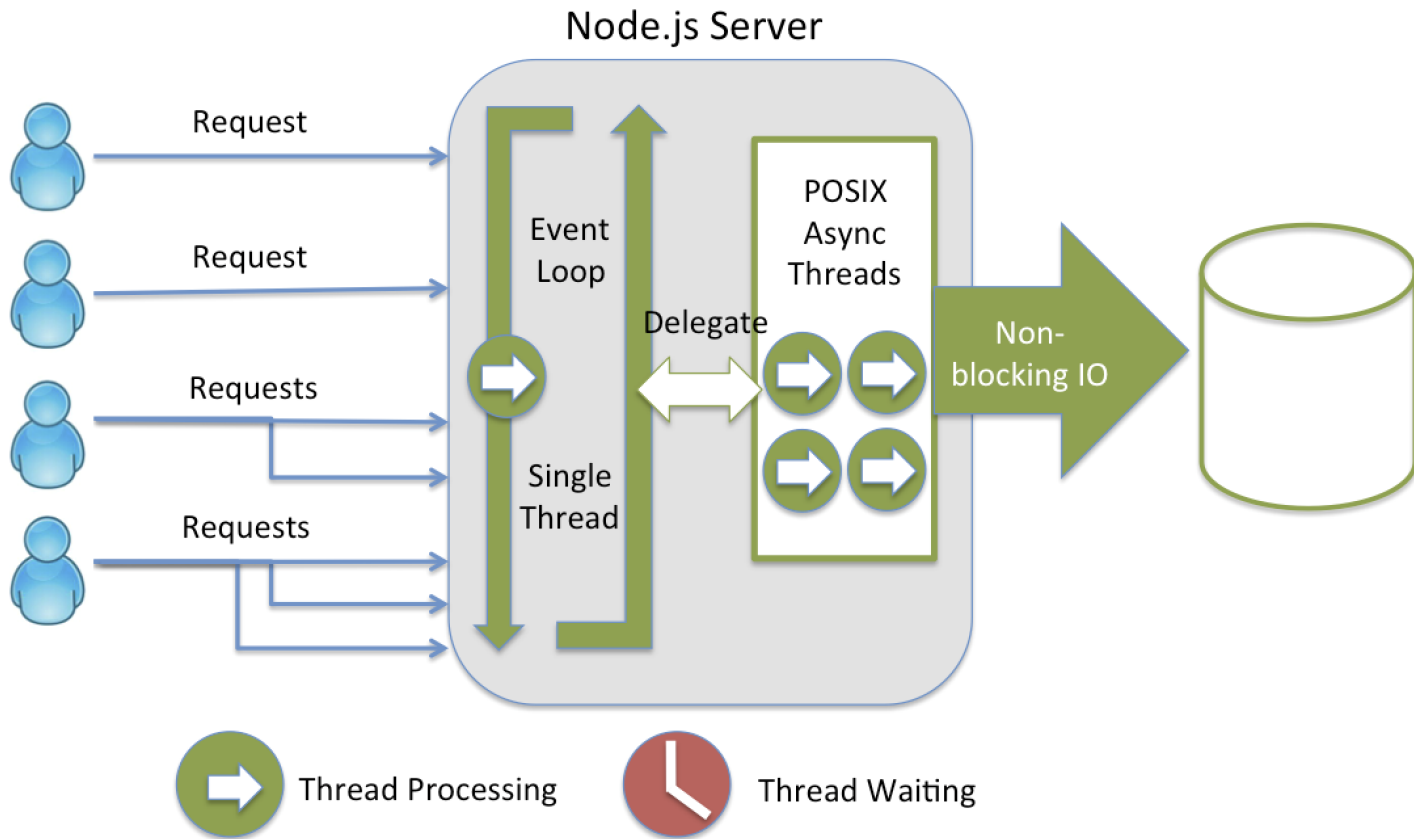


Como funciona ?

- E se tivermos 10 ou até 50 mil usuários conectados ?
- Em sistemas de alta concorrência deve-se evitar muitas threads



Como funciona ?



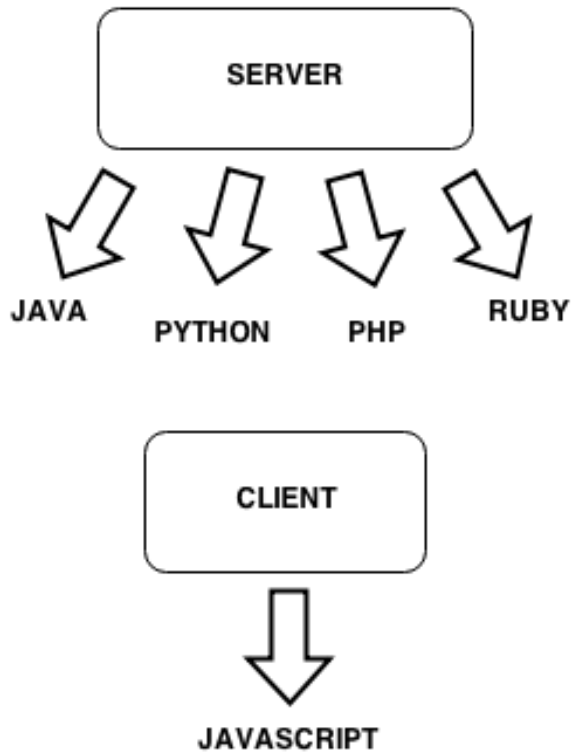


Vantagens ?

- Javascript everywhere

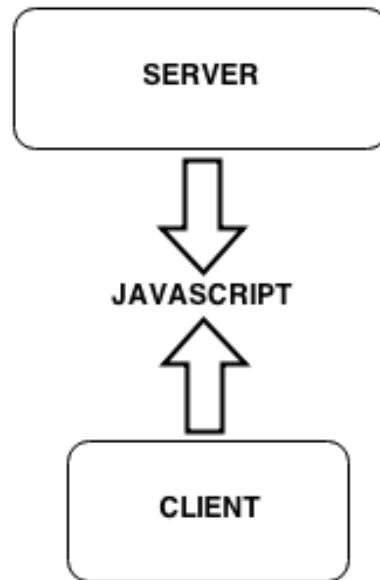


MODELO CLÁSSICO



Vantagens ?

JAVASCRIPT EVERYWHERE





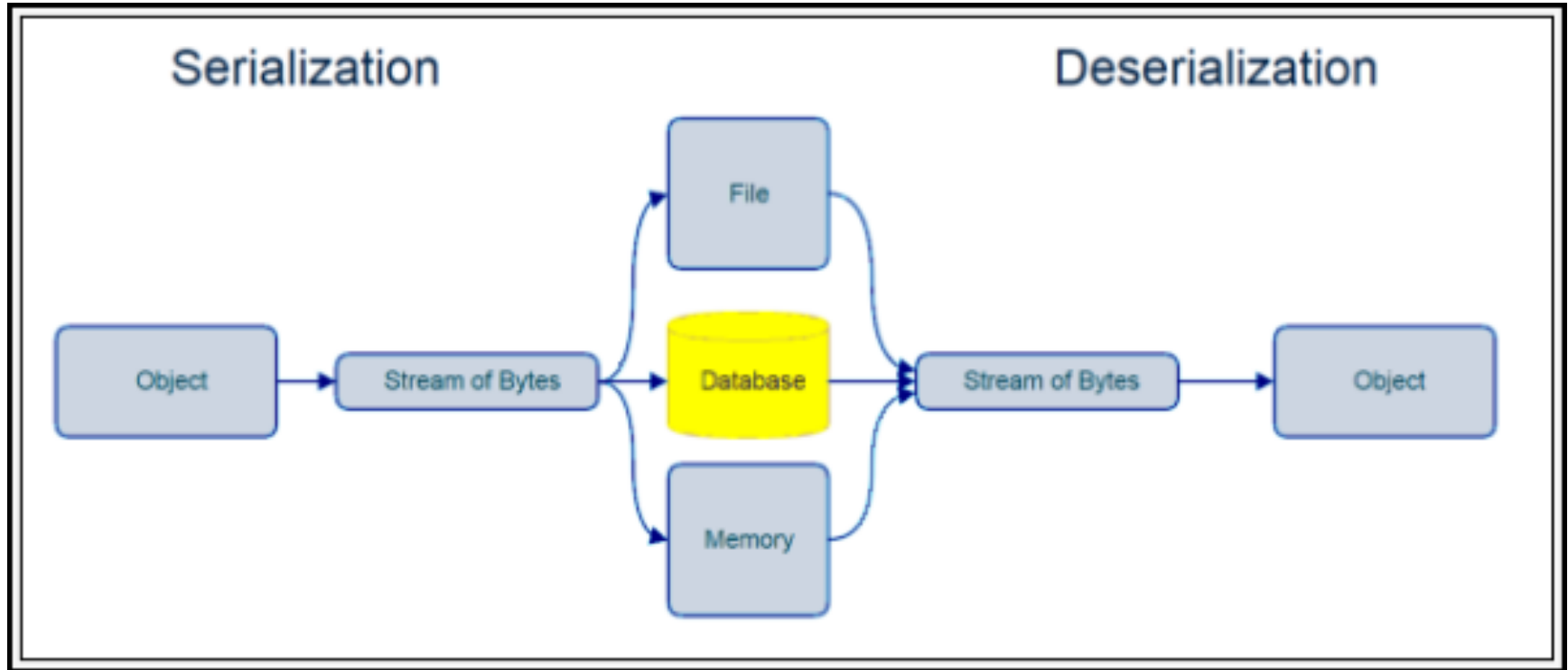
Vantagens ?

- Javascript everywhere
- Comunicação com JSON sem serialização



Vantagens ?

Modelo Convencional





Vantagens ?

Node.js

MONGODB (Database)



SEND BACK OBJECT



GET INFORMATION
OR POST TO DB



NODE (API)



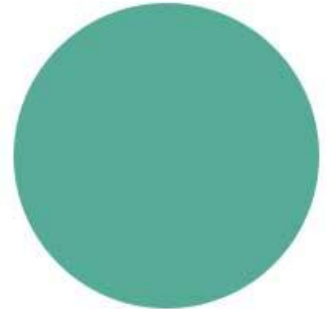
RETURN JSON OBJECT



AJAX CALL TO NODE API



ANGULAR (Frontend)



RETURNS JSON TO ANGULAR

ALL AJAX CALLS
NO PAGE REFRESH



Vantagens ?

- Javascript everywhere
- Comunicação com JSON sem serialização
- Fácil instalação



Vantagens ?

```
sudo apt-get update
```

```
sudo apt-get install nodejs
```

```
sudo apt-get install npm
```



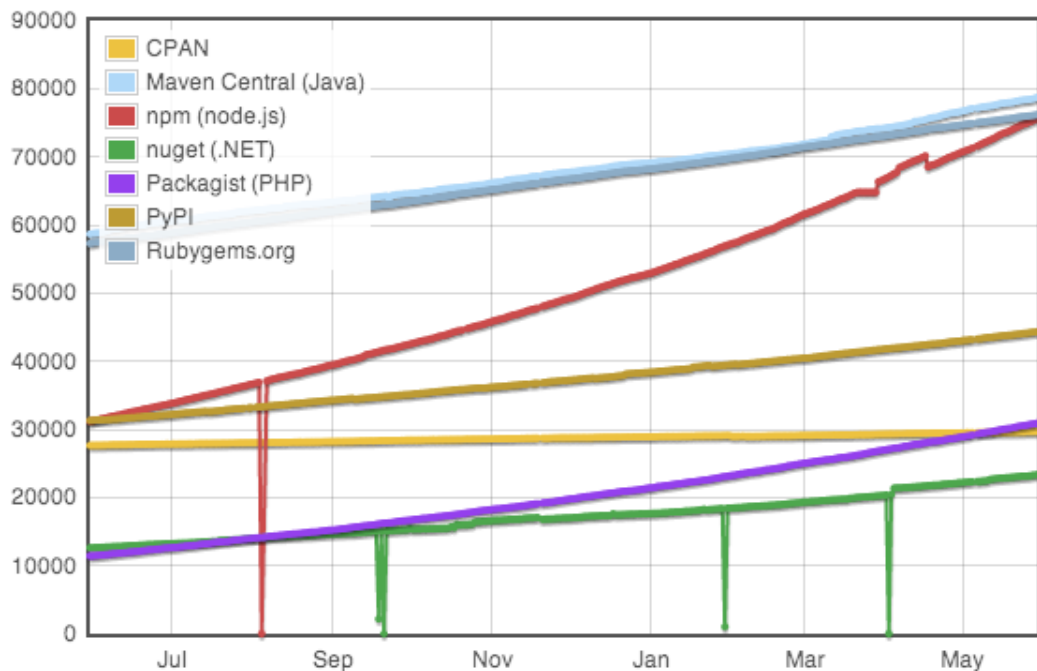

Vantagens ?

- Javascript everywhere
- Comunicação com JSON sem serialização
- Fácil instalação
- Gerenciador de pacotes eficiente (npm)



Vantagens ?

Module Counts



Include

- ☐ Clojars (Clojure)
- ☒ CPAN
- ☐ CPAN (search)
- ☐ CRAN (R)
- ☐ Hackage (Haskell)
- ☒ Maven Central (Java)
- ☐ MELPA (Emacs)
- ☒ npm (node.js)
- ☒ nuget (.NET)
- ☒ Packagist (PHP)
- ☐ Pear (PHP)
- ☒ PyPI
- ☒ Rubygems.org

time period ☐ all time ☒ last year ☐ last 90 days ☐ last 30 days ☐ last 7 days



Vantagens ?

- Javascript everywhere
- Comunicação com JSON sem serialização
- Fácil instalação
- Gerenciador de pacotes eficiente (npm)
- Template Engine (Jade)



Vantagens ?

Fork me on GitHub

jade
Node Template Engine

[Home](#) [Tutorial](#) [Language Reference](#) [API](#) [Command Line](#) [Online Demo](#)

```
doctype html
html(lang="en")
  head
    title= pageTitle
    script(type='text/javascript').
      if (foo) {
        bar(1 + 5)
      }
  body
    h1 Jade - node template engine
    #container.col
      if youAreUsingJade
        p You are amazing
      else
        p Get on it!
      p
        Jade is a terse and simple
        templating language with a
        strong focus on performance
        and powerful features.
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Jade</title>
    <script type="text/javascript">
      if (foo) {
        bar(1 + 5)
      }
    </script>
  </head>
  <body>
    <h1>Jade - node template engine</h1>
    <div id="container" class="col">
      <p>You are amazing</p>
      <p>
        Jade is a terse and simple
        templating language with a
        strong focus on performance
        and powerful features.
      </p>
    </div>
  </body>
</html>
```



Vantagens ?

- Javascript everywhere
- Comunicação com JSON sem serialização
- Fácil instalação
- Gerenciador de pacotes eficiente (npm)
- Template Engine (Jade)
- Integração com MongoDB, Redis, Underscore.js e outros



Desvantagens ?

- Debugar em javascript não é tarefa fácil
- Ainda não é uma ferramenta 100% estável
- As bibliotecas “third-part” carecem de testes mais robustos de segurança e performance
- Alguns módulos podem gerar erros de dependência dificultando a integração



Onde usar ?

Sistemas “Data-intensive”

Múltiplos usuários “real-time” (chat)

Sistemas que usam dados em formato JSON

Fila de inputs, e.g.: “like do Facebook”

Sistema de streaming de dados (áudio / vídeo)

Sistemas de monitoramento em tempo real



Onde não usar ?

Sistemas transacionais

Sistemas “CPU-intensive”

Sistemas com o formato “request-response”

Sistemas que necessitem usar bancos de dados relacionais

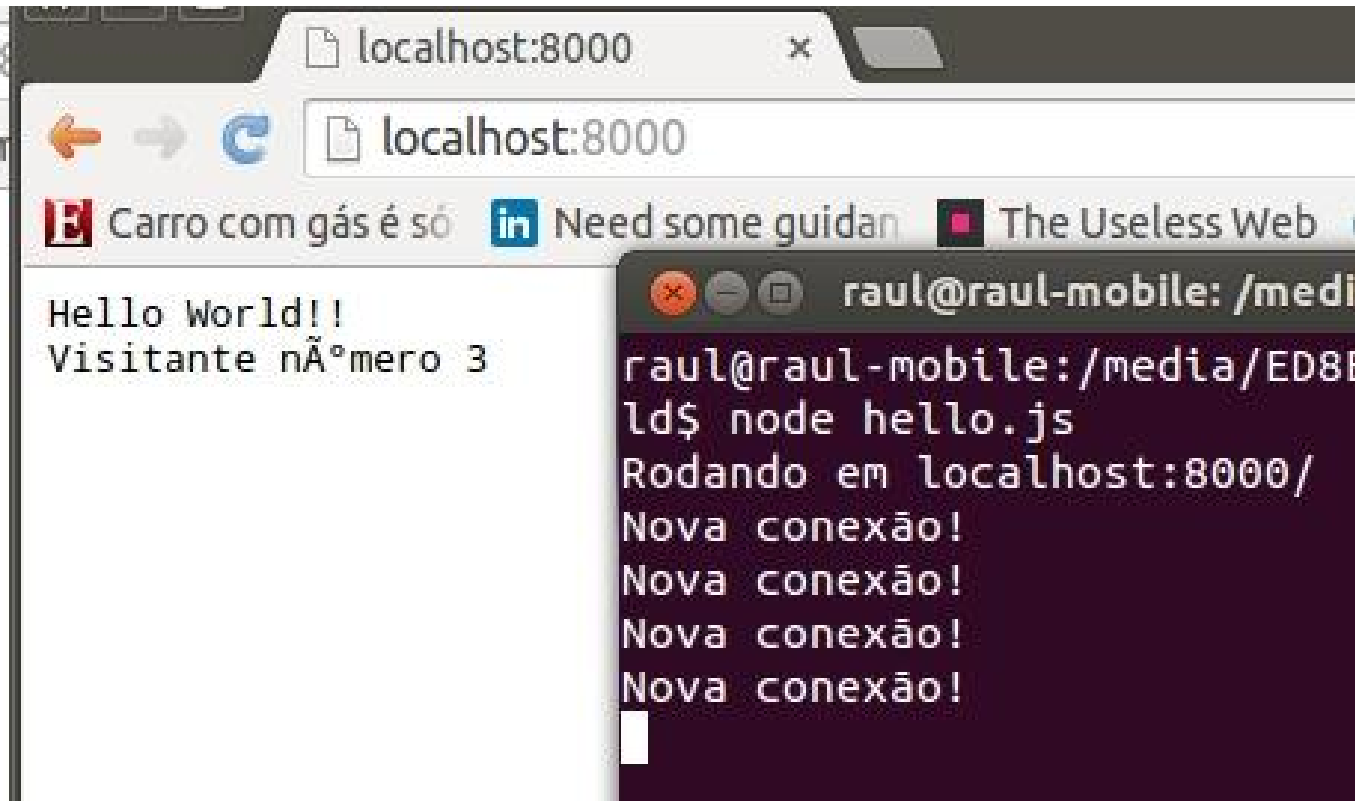


Hello World

```
//carrega o módulo http, responsável por criar o servidor
var http = require('http');
var contador = 0;
//cria uma instância do servidor e deixa uma resposta padrão (Hello World!!)
var server = http.createServer(function(request, response){
  console.log("Nova conexão!");
  contador++;
  response.writeHead(200, {"Content-Type":"text/plain"});
  response.write("Hello World!!\n");
  response.end("Visitante número "+contador);
});
// server está acessível no localhost (127.0.0.1) na porta 8000
server.listen(8000);
//mensagem que será exibida no terminal assim que o servidor for iniciado
console.log("Rodando em localhost:8000/");|
```



Hello World





Chat (Back-End)

```
scratchpad.txt x index.js x index.html x package.json x
1 var app = require('express')();
2 var http = require('http').Server(app);
3 var io = require('socket.io')(http);
4
5 app.get('/', function(req, res){
6   res.sendFile('index.html');
7 });
8
9 io.on('connection', function(socket){
10   socket.on('mensagem', function(msg){
11     io.emit('mensagem', msg);
12   });
13 });
14
15 http.listen(3030, function(){
16   console.log('Disponível na porta :3030');
17 });|
```



Chat (Front-End)

```
17 <ul id="messages"></ul>
18 <form action="">
19   <input id="m" autocomplete="off" /><button>Enviar</button>
20 </form>
21 <script src="/socket.io/socket.io.js"></script>
22 <script src="http://code.jquery.com/jquery-1.11.1.js"></script>
23 <script>
24   var socket = io();
25   $('form').submit(function(){
26     socket.emit('mensaje', $('#m').val());
27     $('#m').val('');
28     return false;
29   });
30   socket.on('mensaje', function(msg){
31     $('#messages').append($('- ').text(msg));
32   });
33 </script>

```



Chat (Tela)

localhost:3030

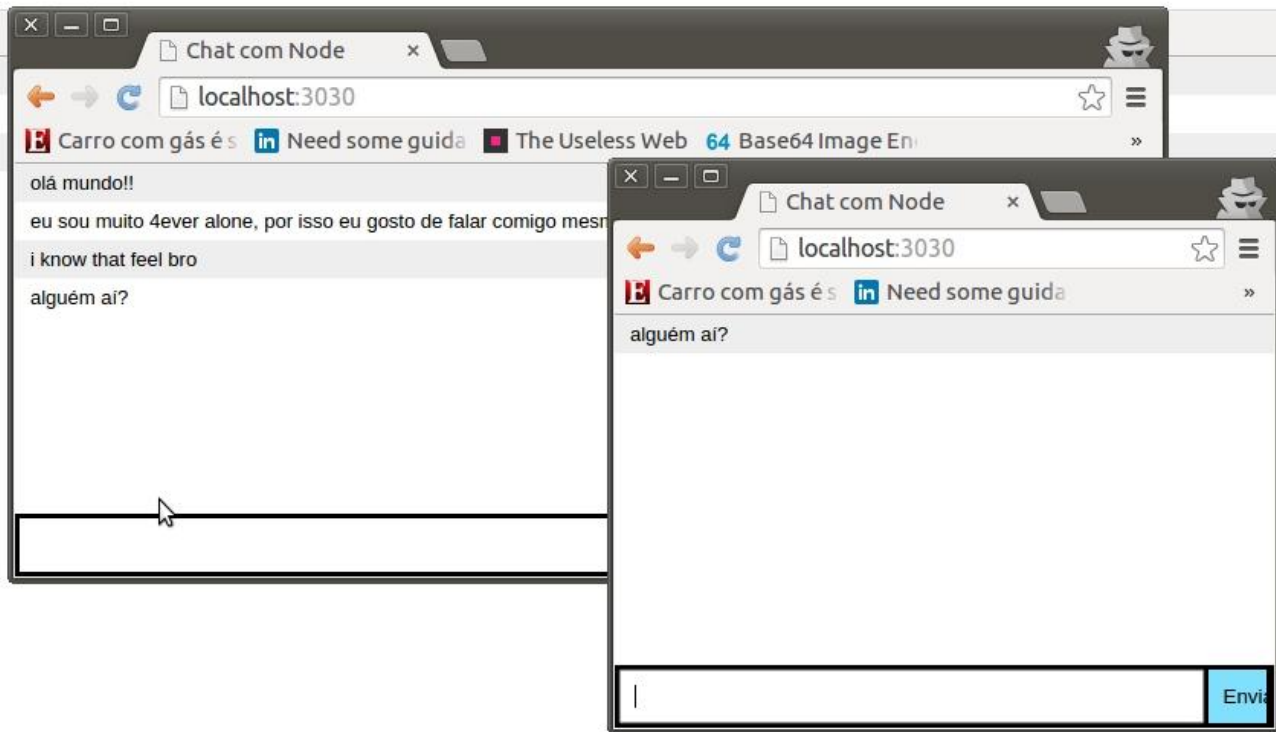
Aplicativos Carro com gás é s Need some guida

olá mundo!!

eu sou muito 4ever alone, por isso eu gosto de falar comigo mesmo

i know that feel bro

alguém aí?





Dicas

- Nodemon / PM2: Monitoram mudanças e dão deploy automático no caso de falhas
- Async: Paralelismo na aplicação
- Node Inspector: Debugger eficiente
- Nodefly: Monitora performance
- Buster.js: Módulo de testes



Links úteis

<http://nodejs.org/>

<https://npmjs.org/>

<https://www.nodejitsu.com/>

<http://howtonode.org/>

<http://nodebr.com/>

<http://udgwebdev.com/nodejs/>



Perguntas ?

Email:

raul@raulferreira.com.br

LinkedIn: <https://www.linkedin.com/in/raulsenaferreira>

Exemplos (GitHub):

<https://github.com/raulsenaferreira/Playing-with-Node.js>