

# NODE.JS: DESENVOLVIMENTO FÁCIL DE UM CHAT WEB

Felipe Ferreira Campos <sup>□</sup>

Instituto de Informática – Universidade Federal de Goiás (UFG)  
Avenida Esperança, Vila Itatiaia – Câmpus Samambaia – Goiânia – Goiás – Brasil

## RESUMO

Este artigo focaliza a metodologia de escrita de código utilizando a plataforma Node.js para o desenvolvimento de aplicações web, cada vez mais utilizada na programação de sistemas distribuídos, tornando-se uma ferramenta poderosa para desenvolvedores de todo o mundo. Em particular, é mostrado um exemplo prático de uso dessa tecnologia. O exemplo apresenta a criação de um chat onde várias pessoas conectadas à rede podem acessar e conversar umas com as outras de forma simples e rápida.

**Palavras-chave:** Node.js. Desenvolvimento. Sistemas Distribuídos.

## 1 INTRODUÇÃO

O Node.js é uma plataforma para desenvolvimento de aplicações servidoras baseadas em rede utilizando JavaScript, ou seja, com Node.js podemos criar uma variedade de aplicações web utilizando apenas código JavaScript.

As principais diferenças entre o Node.js e os servidores tradicionais como apache e IIS são: Simplicidade, por exigir menos linhas de codificação e velocidade por ser menos robusto que os demais servidores.

“Node.js é uma plataforma construída sobre o motor JavaScript do Google Chrome para facilmente construir aplicações de rede rápidas e escaláveis. Node.js usa um modelo de I/O direcionada a evento não bloqueante que o torna leve e eficiente, ideal para aplicações em tempo real com troca intensa de dados através de dispositivos distribuídos”.(NodeBR.com: Repositório de artigos da comunidade brasileira de Node.js).

Uma aplicação distribuída é qualquer aplicação projetada para executar em dois ou mais computadores conectados através de uma rede.

## 2 OBJETIVO

O objetivo deste trabalho é mostrar um exemplo prático do uso de Node.js no desenvolvimento web. Essa plataforma utiliza uma abordagem orientada a eventos, isso significa que é necessário conhecer os eventos que serão emitidos em diferentes momentos da execução e também saber ouvi-los para executar as operações necessárias. A construção da interface com o usuário permite visualizar os elementos onde os eventos serão aplicados, por exemplo os botoes onde podemos executar o evento do tipo click. Ao final deste artigo estará concluído um chat web para comunicação em rede entre vários usuários utilizando o servidor criado com Node.js.

## 3 MATERIAIS E MÉTODOS

Para escrita do código foi utilizado o notepad++ por ser uma ferramenta simples, assim como o bloco de notas ou qualquer outro editor também pode ser utilizado, o primeiro passo é instalar o Node.js no site [nodejs.org](https://nodejs.org). Após a instalação no Node.js basta instalar o socket.io para que seja possível realizar a comunicação entre hosts. A instalação é feita automaticamente após digitar o seguinte comando no prompt: `$ npm install socket.io`

Após concluir os downloads e as instalações basta abrir o editor desejado e criar o servidor Node.js com meras 14 linhas de código, como mostrado abaixo, o nome deste arquivo será ServerChat.js.

```
var fs = require('fs');
var http = require('http');
var socketio = require('socket.io');

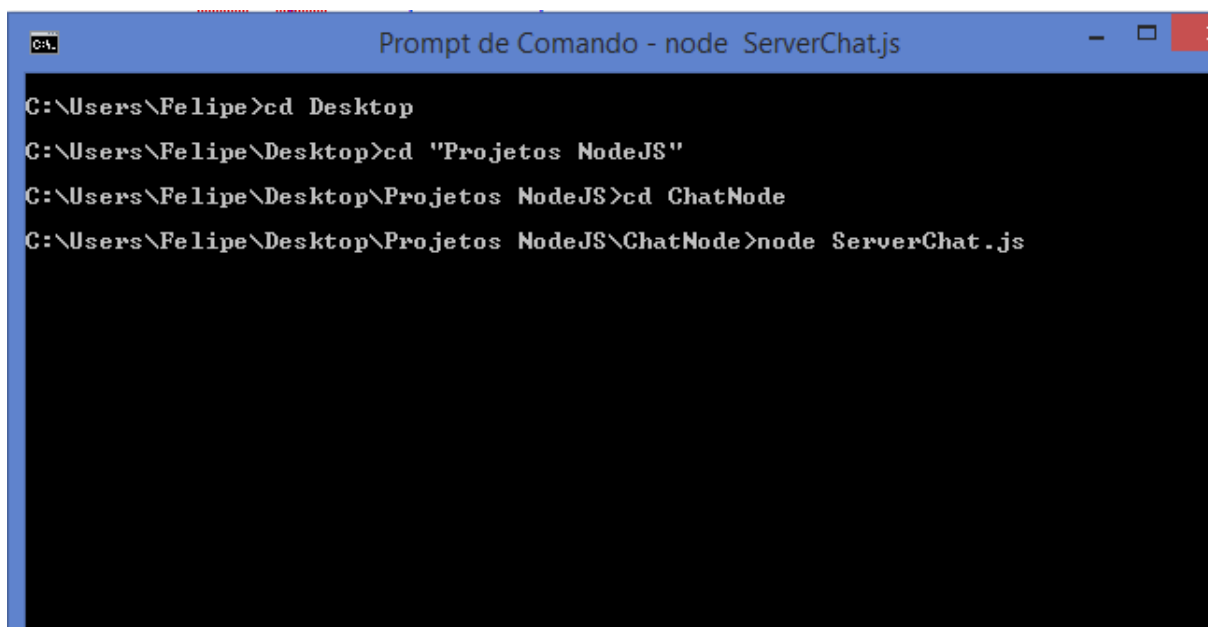
var Server = http.createServer(function (requisicao, resposta) {
  resposta.writeHead(200, { 'Content-type': 'text/html' });
  resposta.end(fs.readFileSync(__dirname + '/index.html'));
}).listen(80);

socketio.listen(Server).on('connection', function (socket) {
  socket.on('message', function (msg) {
    socket.broadcast.emit('message', msg);
  });
});
```

Figura 1 – ServerChat.js

Esse é o servidor Node.js utilizando Hypertext Transfer Protocol (http) e já está pronto para aguardar conexões na porta padrão do seu computador (porta 80), e já está apontado para um arquivo index.html, o FileSystem (fs) se encarrega dessa parte, para que o desenvolvedor se sinta à vontade para criar a interface com o cliente da forma que desejar. O socket.io trata às conexões ao servidor criado e o envio e recebimento das mensagens.

Para executar o servidor criado, basta abrir o prompt, ir até o local do arquivo e escrever o comando: `$ node nomedoarquivo.js` conforme demonstrado abaixo.



```
C:\Users\Felipe>cd Desktop
C:\Users\Felipe\Desktop>cd "Projeto NodeJS"
C:\Users\Felipe\Desktop\Projeto NodeJS>cd ChatNode
C:\Users\Felipe\Desktop\Projeto NodeJS\ChatNode>node ServerChat.js
```

Figura 2 – Executando o ServerChat.js

Agora será criado um arquivo index.html para criar uma interface com o usuário conforme desejar o desenvolvedor, este terá a utilização de estilos com css e também jQuery para proporcionar interatividade, css e jQuery podem ser estudados mais a fundo em: [jquery.com](http://jquery.com) e [w3schools.com/css/](http://w3schools.com/css/).

A próxima imagem contém a criação das tags iniciais de html e os estilos css que serão utilizados para modificar a aparência da aplicação:

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>Chat</title>

  <style type="text/css">
    body {background-color:#000000;}
    input{border:solid 2px #1dd7f1; margin-left:3px; border-radius:4px;}
    h3{padding:5px 30px 2px 50px; position:relative; margin:auto;}
    #Chat {
      background-color:aquamarine;
      width:200px;
      position:relative;
      border-radius:4px;
      border: 3px solid #00ff21;
      margin:auto;
    }
  </style>

```

Figura 3 – Início do index.html

A seguir é feita a Importação dos scripts do jQuery e do Socket.io utilizados e a criação de um script próprio utilizando notações jQuery para que as mensagens sejam colocadas na página umas em baixo das outras sempre que a tecla enter for pressionada no navegador:

```

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js"></script>
<script src="/socket.io/socket.io.js"></script>
<script type="text/javascript">
  $(function () {
    var iosocket = io.connect();

    iosocket.on('connect', function (){
      $('#incomingChatMessages').append('<li> Connected</li>');

      iosocket.on('message', function (message) {
        $('#incomingChatMessages').append('<li></li>').text(message);
      });
    });

    $('#outgoingChatMessage').keypress(function (event) {
      if (event.which == 13) {
        event.preventDefault();
        iosocket.send($('#outgoingChatMessage').val());
        $('#incomingChatMessages').append('<li></li>').text($('#outgoingChatMessage').val());
        $('#outgoingChatMessage').val('');
      }
    });
  });
</script>
</head>

```

Figura 4 – Meio do index.html

Assim encerra-se o cabeçalho do html e para finalizar basta criar o corpo do html conforme abaixo.

```
<body>
  <div id="Chat">
    <h3>Chat Web</h3>
    <ul id="incomingChatMessages"></ul>
    <input type="text" id="outgoingChatMessage">
  </div>
</body>
</html>
```

Figura 5 – Fim do index.html

## 5 RESULTADOS E DISCUSSÕES

Para acessar a aplicação criada basta digitar <http://localhost> no navegador desejado, rapidamente o chat abrirá conforme a seguinte figura:

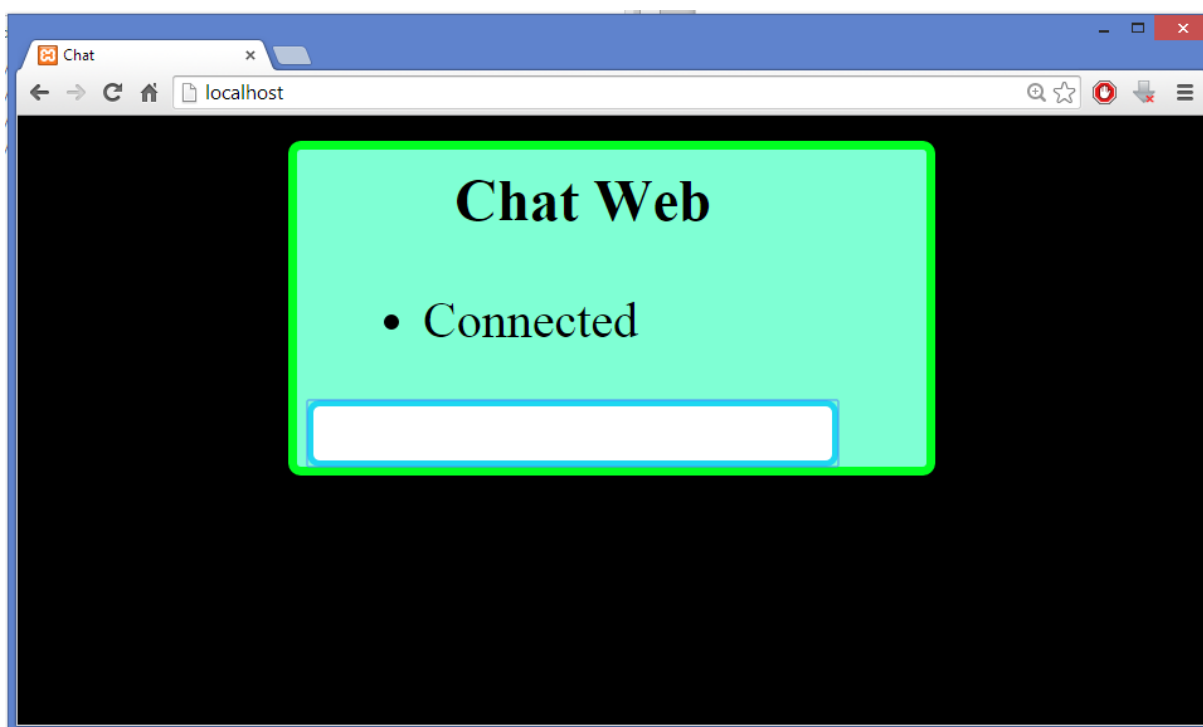
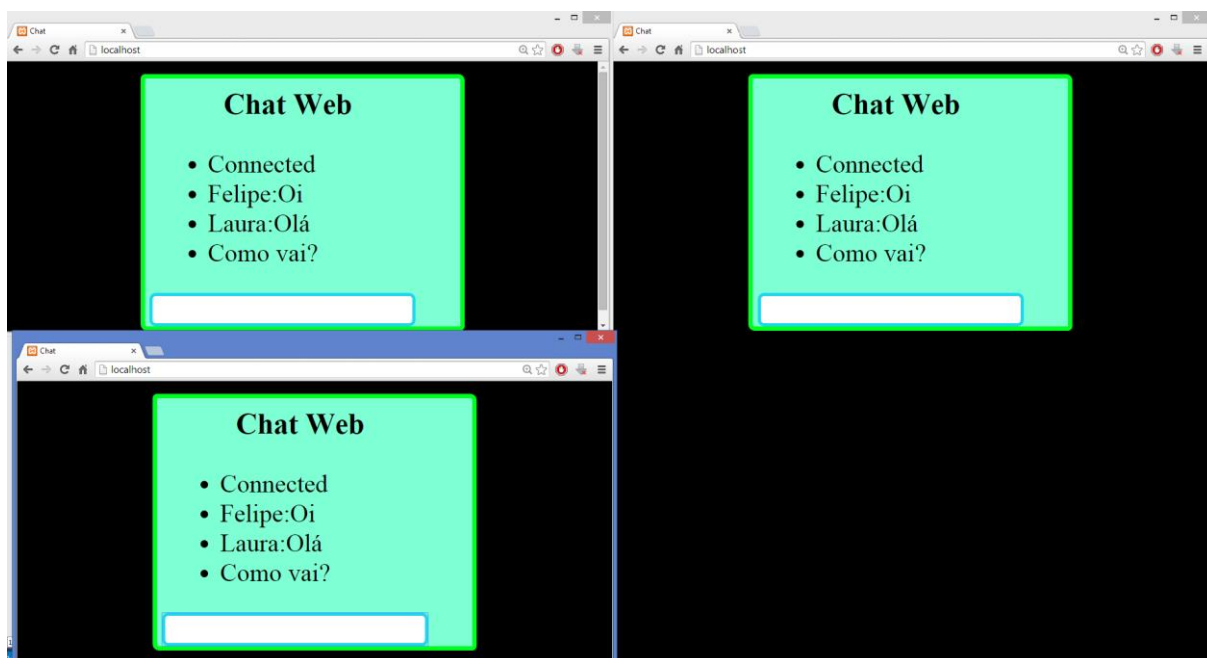


Figura 6 – <http://localhost>

Para testar o poder desse chat como aplicação distribuída basta que outras pessoas se conectem na aplicação, substituindo localhost pelo IP do computador onde se encontra o servidor Node.js.

Abaixo uma demonstração de dialogo com várias guias abertas na máquina local:



As mensagens podem ser enviadas anonimamente caso não se deseje escrever o nome nelas.

## 6 REFERÊNCIAS

Node.js. Disponível em: <<http://nodejs.org/>>. Acesso em: 03 dezembro 2014.

Comunidade brasileira de Node.JS . Disponível em: <<http://nodebr.com/>>. Acesso em: 03 dezembro 2014.

jQuery. Disponível em: <<http://jquery.com/>>. Acesso em: 03 dezembro 2014.

Css Tutorial. Disponível em: <<http://www.w3schools.com/css/>>. Acesso em: 03 dezembro 2014.

Getting-Started-Socketio. Disponível em: <<http://java.dzone.com/articles/getting-started-socketio-and>>. Acesso em: 03 dezembro 2014.

Primeiros Passos com Node.js. Disponível em: <<http://tableless.com.br/primeiros-passos-com-node-js/>>. Acesso em: 03 dezembro 2014.

Video-aulas. Disponível em:

<[https://www.youtube.com/watch?v=TAZaYjk7klo&list=PLz6D6n\\_hNk4x255Eo4EbPsD9Kpsr2H7yc](https://www.youtube.com/watch?v=TAZaYjk7klo&list=PLz6D6n_hNk4x255Eo4EbPsD9Kpsr2H7yc)

<http://nodebr.com/o-que-e-node-js/>>. Acesso em: 03 dezembro 2014.