

Rodrigo Cichetto Monteiro

# **Desenvolvimento de aplicações com JavaScript**

Brasil

2018, v-1.0.0



Rodrigo Cichetto Monteiro

## **Desenvolvimento de aplicações com JavaScript**

Trabalho apresentado a UNIP - UNIVERSIDADE PAULISTA como pré-requisito para obtenção da Certificação de Conclusão do Curso de Bacharelado em Ciência da Computação. Orientador: Prof. Leandro Carlos Fernandez

Universidade Paulista – UNIP  
Faculdade de Ciência da Computação  
Programa de Graduação

Orientador: Leandro Carlos Fernandez  
Coorientador: Danielle Colturato

Brasil  
2018, v-1.0.0

---

Rodrigo Cichetto Monteiro

Desenvolvimento de aplicações com JavaScript/ Rodrigo Cichetto Monteiro. –  
Brasil, 2018, v-1.0.0-

39 p. : il. (algumas color.) ; 30 cm.

Orientador: Leandro Carlos Fernandez

Tese (Graduação) – Universidade Paulista – UNIP

Faculdade de Ciência da Computação

Programa de Graduação, 2018, v-1.0.0.

1. JavaScript. 2. Desenvolvimento de aplicações. I. Orientador Leandro Carlos Fernandez. II. Universidade Paulista – UNIP. III. Faculdade de Ciência da Computação. IV. Desenvolvimento de aplicações com JavaScriptV. Rodrigo Cichetto Monteiro

CDU 02:141:005.7

---

Rodrigo Cichetto Monteiro

## **Desenvolvimento de aplicações com JavaScript**

Trabalho apresentado a UNIP - UNIVERSIDADE PAULISTA como pré-requisito para obtenção da Certificação de Conclusão do Curso de Bacharelado em Ciência da Computação. Orientador: Prof. Leandro Carlos Fernandez

Trabalho aprovado. Brasil, 24 de novembro de 2018:

---

**Leandro Carlos Fernandez**  
Orientador

---

**Professor**  
Convidado 1

---

**Professor**  
Convidado 2

Brasil  
2018, v-1.0.0



*“Cada sonho que você deixa pra trás,  
é um pedaço do seu futuro que deixa de existir.  
(Steve Jobs)*





# Resumo

Fazendo parte das três principais tecnologias que movem a internet, sendo elas HTML, CSS e claro JavaScript, não é mais uma linguagem voltada somente para desenvolvedores front-end, mas sim uma obrigação para todo desenvolvedor, que tem como obrigação conhecer pelo menos o básico de JS. Nos últimos tempos a linguagem JavaScript ganhou muita importância em quaisquer cenários que envolvam programação, e vem sendo utilizada em sites, aplicações, servidores, automação de testes, automação de tarefas, internet das coisas, entre outros. Este trabalho tem como principal objetivo atualizar o leitor através de um compilado de informações sobre as tendências do que mais vem sendo utilizado para a construção de aplicações utilizando a linguagem, mostrando os frameworks mais recentes e mais famosos. Mas lembre-se com grandes poderes vem grandes responsabilidades.

**Palavras-chaves:** javascript, typescript, frameworks



# Abstract

Being part of the three main technologies that move the internet, being HTML, CSS and clear JavaScript, is no longer a language aimed only at front-end developers, but rather an obligation for every developer, who has to know at least the basics of JS. In recent times the JavaScript language has gained a lot of importance in any scenarios involving programming, and has been used in websites, applications, servers, automation of tests, automation of tasks, internet of things, among others. This work has as main objective to update the reader through a compilation of information about the trends of what is being used to build applications using the language, showing the latest and most famous frameworks. But remember with great powers comes great responsibilities.

**Key-words:** javascript, typescript, frameworks



# Lista de ilustrações

Figura 1 – <i>Carrousel</i> - Um componente de apresentação de slides para percorrer imagens ou slides de texto - como um carrossel. – Exemplo de aplicação do JS . . . . .	23
Figura 2 – <i>Validation</i> - Validação de formulários – Exemplo de aplicação do JS . .	23
Figura 3 – <i>Modal</i> - Caixas de diálogo para notificações ao usuário. – Exemplo de aplicação do JS . . . . .	23
Figura 4 – <i>Brendan Eich</i> - Criador do JavaScript. . . . .	25



# Lista de abreviaturas e siglas

JS	JavaScript
HTML	Abreviação para <i>HyperText Markup Language</i> , que em português significa Linguagem de Marcação de Hipertexto.
DOM	Abreviação para <i>Document Object Model</i> , que em português significa Modelo de Objetos e Documentos.
CSS	Abreviação para <i>Cascading Style Sheets</i> , que em português significa Folhas de Estilo em Cascata.
ECMA	Acrônimo para <i>European Computer Manufacturers Association</i> , que em português significa Associação Européia de Fabricantes de Computadores.





# Sumário

1	INTRODUÇÃO . . . . .	17
I	A LINGUAGEM JAVASCRIPT	19
2	O BÁSICO SOBRE JS . . . . .	21
3	JAVASCRIPT NÃO É JAVA . . . . .	25
4	A LINGUAGEM NOS DIAS ATUAIS . . . . .	27
4.1	JavaScript além dos navegadores . . . . .	27
4.2	ECMA . . . . .	28
II	O TYPESCRIPT	29
5	O JS COM PODERES . . . . .	31
III	MEAN STACK	33
6	A PILHA . . . . .	35
6.1	MongoDB . . . . .	35
6.2	Express . . . . .	35
6.3	Angular . . . . .	35
6.4	Node.js . . . . .	35
IV	CONCLUSÃO	37
	Considerações finais . . . . .	39



# 1 Introdução

Inicialmente implementada com o objetivo no desenvolvimento web para o lado do cliente, a linguagem criada por Brendan Eich enquanto trabalhou na Netscape se tornou uma das linguagens mais populares atualmente, sendo a terceira camada do bolo quando se fala de tecnologias web, das quais HTML e CSS também fazem parte.

Com navegadores cada vez mais modernos a tendência é que nossos sites também fiquem cada vez mais sofisticados.

Todo projeto que iniciamos existe uma origem, uma necessidade e um problema a ser resolvido. Há algum tempo, um grande problema ganhou atenção: a reutilização de trechos de código. São módulos de código (trechos em HTML, CSS, JavaScript, etc) que juntos criam padronizações e organizações, mas sobretudo flexibilidade. São os chamados frameworks.

Nos últimos anos a linguagem JavaScript ganhou maior importância, com o surgimento de bibliotecas e frameworks que possibilitaram o desenvolvimento de não somente web sites mas também aplicativos, single page applications, programas desktop, progressive web apps e muito mais. Sendo alguns deles Angular, React, Vue, jQuery e Node.js.

Talvez nenhuma outra linguagem tenha conseguido ganhar tanta atenção dos desenvolvedores como o JavaScript. Em busca de sua identidade a linguagem foi a única que conseguiu se enraizar nos navegadores, e atualmente também passou a se empoderar dos servidores de alta performance através do Node.js.

O surgimento do Node.js possibilitou um novo mundo para os desenvolvedores, levando a linguagem para um novo patamar. Criado com um modelo não bloqueante na entrada e saída de dados, possibilitou que a linguagem fosse levada agora também para aplicações back-end, ou seja, para o server-side (lado do servidor).

Com o tempo a Microsoft desenvolveu o TypeScript, que de uma forma simples é o JavaScript acrescido de tipagem de dados, ou seja, agora erros podem ser detectados durante a digitação do código, sem dúvidas podemos considerar o TypeScript como uma evolução do JavaScript.

Este trabalho tem como objetivo atualizar o leitor do que há de mais novo no desenvolvimento com a linguagem JavaScript, para melhor compreensão do conteúdo apresentado é necessário conhecimento prévio básico da linguagem.



Parte I

A Linguagem JavaScript



## 2 O Básico sobre JS

JavaScript é uma linguagem de programação dinâmica interpretada, inicialmente utilizada pelos navegadores para execução de *scripts* no lado do cliente, ou seja, no seu *browser*. Os *scripts* são incorporados a páginas HTML tendo como função adicionar interatividade para o usuário.

Atualmente, é praticamente impossível imaginar a internet sem a existência do JavaScript, certamente você já se deparou com alguns dos exemplos como os famosos carrosseis (Figura 1, página 23), as tão importantes validações de formulários (Figura 2, página 23), ou até mesmo o *modal* (Figura 3, página 23) que mostram a aplicação da linguagem em *web sites*.

A web está infestada de códigos JavaScript, pode ter certeza de que diariamente muitos *scripts* estão sendo executados durante uma pesquisa no Google, seu tempo no Facebook, acesso a seu banco na internet, ou até mesmo para leitura de uma notícia em um portal. Seu uso primário na web é de funções incluídas no HTML que interagem com o Modelo de Objetos e Documentos (DOM) da página.

DOM é uma interface de programação utilizada para documentos HTML, ele fornece a representação e interação com objetos os estruturando em nós de uma árvore chamada de Árvore DOM. É assim que o JavaScript consegue manusear as estruturas, estilos e conteúdos dos elementos presentes no HTML. Vale ressaltar que ele não faz parte da linguagem JavaScript, pois também pode ser acessado por outras linguagens.

O JavaScript apresenta uma sintaxe simples que facilita o aprendizado, mas não confunda pois a primeira vista muitos desenvolvedores podem acreditar que a linguagem é defeituosa ou esquisita, pois não compreendem o real poder que se esconde por trás desta simplicidade.

Inicialmente classificada como linguagem do tipo *client side*, é por si só uma linguagem compacta, mas muito flexível, com comportamentos diferenciados das demais ela permite, por exemplo, que um objeto tenha seus atributos adicionados ou removidos em tempo de execução, o que não é muito comum para desenvolvedores de outras linguagens.

É uma linguagem interpretada, pois seus comandos são executados sem que haja necessidade de compilação, tendo como interpretador de *script* o *browser* do usuário. Sendo assim independente de plataformas, como os comandos são interpretados pelo navegador do usuário é irrelevante se o usuário está utilizando Windows, Linux ou Mac OS.

Sua tipagem é dinâmica, ou seja, tipos são associados com valores. Por exemplo, uma variável pode ser associada a um número e posteriormente associada a um texto.

Baseada em objetos, no JavaScript os objetos são matrizes associativas, ou seja `obj.a = 10` e `obj["a"] = 10` são equivalentes, sendo a única diferença a própria sintaxe.

As funções em JS são tratadas como objetos, possuindo métodos e propriedades, podendo assim serem atribuídas a variáveis, retornadas como objetos, ou até mesmo passadas como argumentos para outras funções, o que é chamado de aninhamento de funções.

Com JavaScript também é possível a detecção de eventos, sempre que algo de importante acontece é disparado um evento, o clique de um botão, o preenchimento de um campo de formulário, a movimentação do mouse, são alguns exemplos dos eventos que são disparados. Isso nos permite reagir a estes eventos deixando assim que nossa aplicação deixe de ser estática.

A linguagem possui grande tolerância a erros, uma vez que conversões automáticas são realizadas durante suas operações.

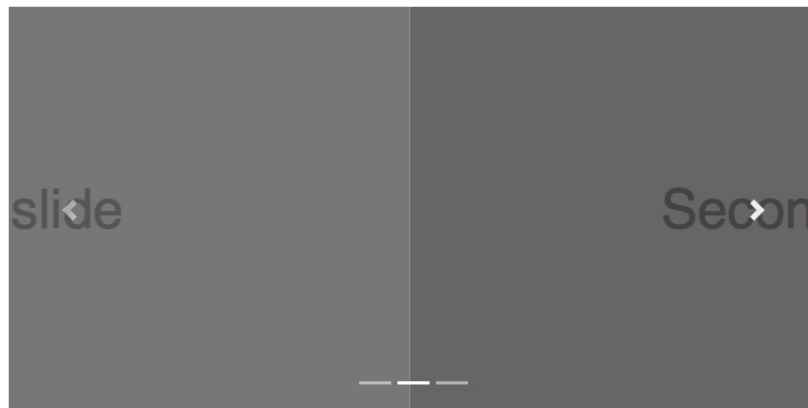
Podemos executar códigos JS de várias formas na web, sendo uma delas pelo próprio console do navegador pressionando as teclas F12, importando um script em uma página HTML, ou até mesmo envolvendo o trecho de código na tag `<script>`.

No início da internet as páginas não eram nada interativas, documentos apresentavam seu conteúdo exatamente como foram criados para serem exibidos no navegador e só. O JavaScript revolucionou o que podemos fazer, hoje não só na web, mas praticamente em todas as áreas que se possa programar.

No próximo capítulo abordaremos de como surgiu a linguagem, como ela evoluiu e atua no presente e a necessidade de padronizações.

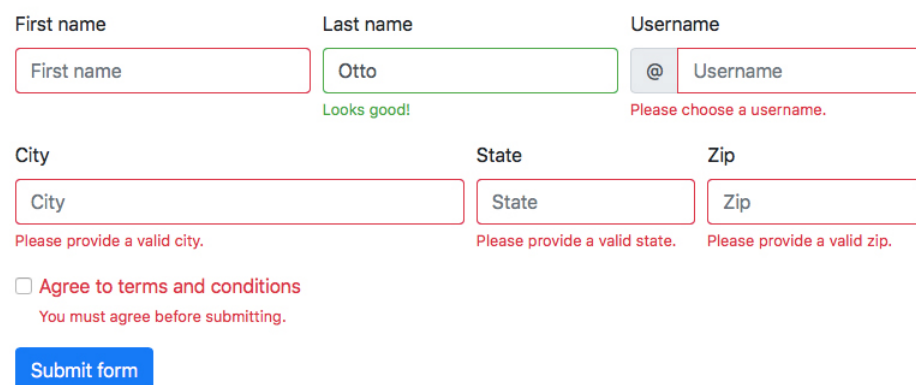


Figura 1 – *Carrousel* - Um componente de apresentação de slides para percorrer imagens ou slides de texto - como um carrossel. – Exemplo de aplicação do JS



Fonte: Bootstrap (<https://getbootstrap.com/docs/4.1/components/carousel/>)

Figura 2 – *Validation* - Validação de formulários – Exemplo de aplicação do JS



First name Last name Username

First name Otto @ Username

Looks good! Please choose a username.

City State Zip

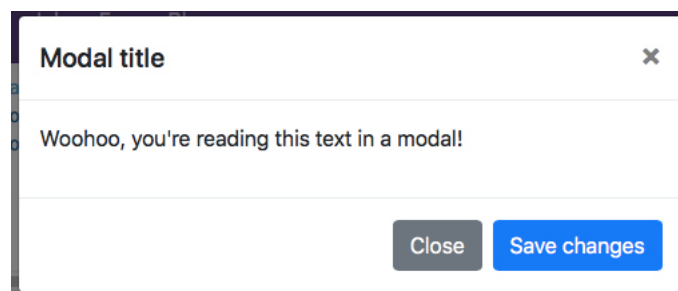
Please provide a valid city. Please provide a valid state. Please provide a valid zip.

☐ Agree to terms and conditions  
You must agree before submitting.

Submit form

Fonte: Bootstrap (<https://getbootstrap.com/docs/4.1/components/forms/>)

Figura 3 – *Modal* - Caixas de diálogo para notificações ao usuário. – Exemplo de aplicação do JS



Fonte: Bootstrap (<https://getbootstrap.com/docs/4.1/components/modal/>)



### 3 JavaScript não é Java

Acredite JavaScript não é Java, criada em 1995 por Brendan Eich a linguagem ganhou esse nome somente por estratégia publicitária, na época a recém lançada linguagem Java tinha muita sucesso. O que causou muitas confusões que existem até hoje para aqueles que iniciam no mundo da programação, pois esses indivíduos pensam que as linguagens são iguais ou que o JavaScript é uma versão mais simples do Java, o que não é verdade. É muito importante deixar claro que Java e JavaScript são linguagens diferentes e com propósitos também distintos, a única coisa parecida entre elas é o nome.

Brendan Eich, trabalhava na Netscape quando desenvolveu em apenas 10 dias uma linguagem de programação simples com o intuito de atrair novos programadores para ela.

Figura 4 – *Brendan Eich* - Criador do JavaScript.



Fonte: Wikipedia (<https://pt.wikipedia.org/wiki/>)

Originalmente desenvolvida com o nome Mocha (nome de um tipo de café), posteriormente teve seu nome modificado para LiveScript e, por fim, JavaScript em 4 de dezembro de 1995, nome pelo qual ficou tão conhecida.

O JavaScript teve grande influencia da linguagem C, suportando elementos de sintaxe de programação estruturada do C (por exemplo while, if, switch) ambas diferem expressões e comandos, temos como exceção o escopo em blocos, que em seu lugar o JS utiliza escopos a níveis de funções. Uma diferença curiosa do C é que a quebra de linha termina automaticamente o comando, sendo assim o ponto e vírgula opcional ao fim do comando.

Logo a linguagem adquiriu ampla aceitação e cresceu cada vez mais, desde então o JavaScript se tornou a linguagem de programação mais popular da web. Suportada por todos os navegadores, a linguagem é responsável por praticamente qualquer tipo de dinamismo que queiramos em nossas páginas.



## 4 A linguagem nos dias atuais

Conforme a linguagem evoluiu se tornou mais poderosa e independente do navegador. Isso possibilitou que a linguagem fosse utilizada não somente para a web como *client side* mas agora em vários lugares.

### 4.1 JavaScript além dos navegadores

Inicialmente tratada como um extra para os navegadores, podemos dizer que a linguagem caminhou com o avanço da tecnologia, isso possibilitou o uso da linguagem em diversas áreas sendo elas:

- a) Aplicações web
- b) Aplicativos mobile
- c) Automação de testes e de tarefas
- d) Controle de hardware
- e) Desenvolvimento de jogos
- f) Internet das coisas
- g) Realidade virtual e aumentada
- h) Softwares desktop
- i) Servidores

Praticamente tudo que envolve programação o JavaScript está presente, para criar um software desktop existe por exemplo o *framework* Electron (<https://electronjs.org>), desenvolvido pelo GitHub ele possibilita criar aplicativos desktop multiplataforma através do JavaScript. No desenvolvimento de jogos a *engine* Unity (<https://unity3d.com/pt>) é uma das plataformas que oferecem suporte a linguagem.

Hoje grandes empresas como Google, Microsoft, Netflix, Uber e LinkedIn usam JavaScript até mesmo no *back-end*.

Isso acabou impactando o mundo dos desenvolvedores, fazendo com que seja obrigatório todo programador saber pelo menos o básico da linguagem, mesmo atuando na área de *back-end* ou até mesmo de teste.

## 4.2 ECMA

Atualmente a linguagem se encontra na 8ª edição chamada de ECMAScript 2017 (<https://www.ecma-international.org/publications/standards/Ecma-262.htm>), finalizada em Junho de 2017, sua contribuição de maior expressão foi a inclusão dos operadores *await* e *async*, possibilitando agora que a linguagem trabalhe com funções assíncronas de uma maneira simples.

Pela linguagem rodar em ambientes que podem variar, algo importante a considerar é a compatibilidade entre os navegadores. Para isso é necessário um padrão a seguir, tal criado e mantido até hoje pelo ECMA Internacional.

Já em 1996 a Netscape, detentora do JavaScript, anunciou que submetia a linguagem para o ECMA Internacional como candidata a padrão industrial, resultando então no ECMAScript.

Padronização que define a estrutura da linguagem, seus comportamentos e comandos, dando assim um padrão aos interpretadores da linguagem.

Com participação colaborativa de empresas que implementam o *run-time* da linguagem, como Mozilla, Google, Microsoft e Apple, além da participação de desenvolvedores da comunidade, o ECMA coordena e faz o trabalho de desenvolvimento contínuo e descentralizado do JS.

Em seu site (<http://www.ecma-international.org/publications/standards/Ecma-262.htm>) é possível encontrar informações como versão atual e a documentação da linguagem, porém aconselho a buscar documentações na internet, uma boa referência a seguir é o site da Mozilla (<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>), ou também o site da W3Schools (<https://www.w3schools.com/jsref/default.asp>).

A 9ª edição já está em desenvolvimento, devendo chegar até o final deste ano de 2018, sendo chamada de ECMAScript 2018.

Parte II

O TypeScript





## 5 O JS com poderes



Parte III

MEAN Stack



## 6 A pilha

6.1 MongoDB

6.2 Express

6.3 Angular

6.4 Node.js



Parte IV

Conclusão





## Considerações finais

Sed consequat tellus et tortor. Ut tempor laoreet quam. Nullam id wisi a libero tristique semper. Nullam nisl massa, rutrum ut, egestas semper, mollis id, leo. Nulla ac massa eu risus blandit mattis. Mauris ut nunc. In hac habitasse platea dictumst. Aliquam eget tortor. Quisque dapibus pede in erat. Nunc enim. In dui nulla, commodo at, consectetur nec, malesuada nec, elit. Aliquam ornare tellus eu urna. Sed nec metus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Phasellus id magna. Duis malesuada interdum arcu. Integer metus. Morbi pulvinar pellentesque mi. Suspendisse sed est eu magna molestie egestas. Quisque mi lorem, pulvinar eget, egestas quis, luctus at, ante. Proin auctor vehicula purus. Fusce ac nisl aliquam ante hendrerit pellentesque. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Morbi wisi. Etiam arcu mauris, facilisis sed, eleifend non, nonummy ut, pede. Cras ut lacus tempor metus mollis placerat. Vivamus eu tortor vel metus interdum malesuada.

Sed eleifend, eros sit amet faucibus elementum, urna sapien consectetur mauris, quis egestas leo justo non risus. Morbi non felis ac libero vulputate fringilla. Mauris libero eros, lacinia non, sodales quis, dapibus porttitor, pede. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Morbi dapibus mauris condimentum nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam sit amet erat. Nulla varius. Etiam tincidunt dui vitae turpis. Donec leo. Morbi vulputate convallis est. Integer aliquet. Pellentesque aliquet sodales urna.