

Pós-graduação em Desenvolvimento Web e Aplicativos Móveis

Frameworks e Bibliotecas JS



Fábio Rodrigues Jorge
fabinhojorgenet@gmail.com



<https://github.com/fabinhojorge/aula-javascript>



Agenda

- HTML 5 - Validações
- Frameworks
 - Server side
 - Client side
 - Full Stack Framework
- Meteor (Full Stack Framework)
- Algumas Bibliotecas
- Tendências para 2016 / 2017

HTML 5 e o Javascript



HTML5

Validar campos de um formulário com HTML

O HTML 5 é ultima versão do HTML

Essa versão trouxe vários recursos e tecnologias (*HTML 5 & friends*) para a criação de sites mais poderosos.



HTML5 e o Javascript

Antes de pensar em **Javascript (ou jQuery)** para criar componentes, procure pelos componentes do HTML 5.

- **Não é necessário** construir componentes em Javascript se existem no HTML 5
- Funcionalidades nativas
- Renderização nativa nos Browsers

HTML5 Novidades interessantes

Algumas novidades interessantes:

- Novos *<inputs/>* e Atributos
- Tags de multimídia (*<video>*, *<audio>*, *<embed>*)
- Tags de desenho (*<canvas>* e *<svg>*)
- *LocalStorage* e *SessionStorage*

Validação HTML 5

```
<input type="[TIPO]" required></input>
```

Atributos: *required*, *title*, *min*, *max* e *pattern*

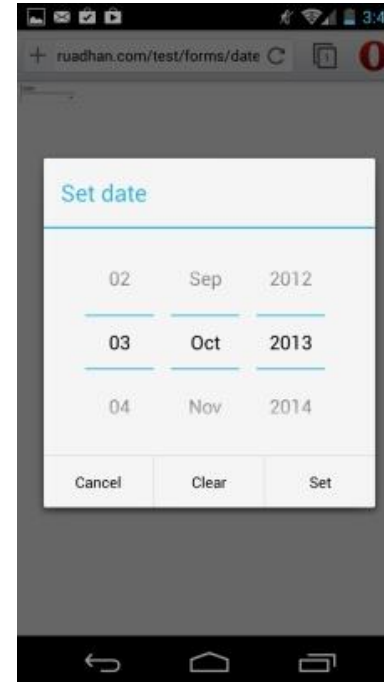
Tipos de input: *text*, *mail*, *tel*, *date*, *time*

Validação HTML 5

Qual a principal vantagem em usar os campos específicos?

- Componentes **Nativos do Browser**
- Nos **mobiles**, muitas vezes os componentes são nativos do **SO**.
- Comportamentos e interações com os **teclados de mobile**

Validação HTML 5



Storage do HTML 5

O **Storage** é uma “base” de dados compartilhada no browser. Outras páginas de mesmo domínio tem acesso a essa base.

- **window.sessionStorage**: Descarrega quando fechar o browser
- **window.localStorage**: Não descarrega mesmo se fechar o navegador



Vue.js

Frameworks

e Bibliotecas JS



Telerik Kendo UI



Bibliotecas vs Frameworks

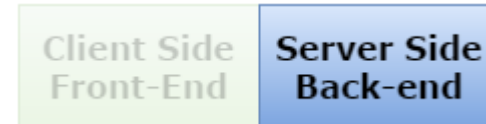
Bibliotecas: Resolvem algum problema específico

Frameworks: Conjunto de ferramentas genéricas.
Dita o fluxo da aplicação, geralmente impondo uma estrutura de desenvolvimento

Frameworks

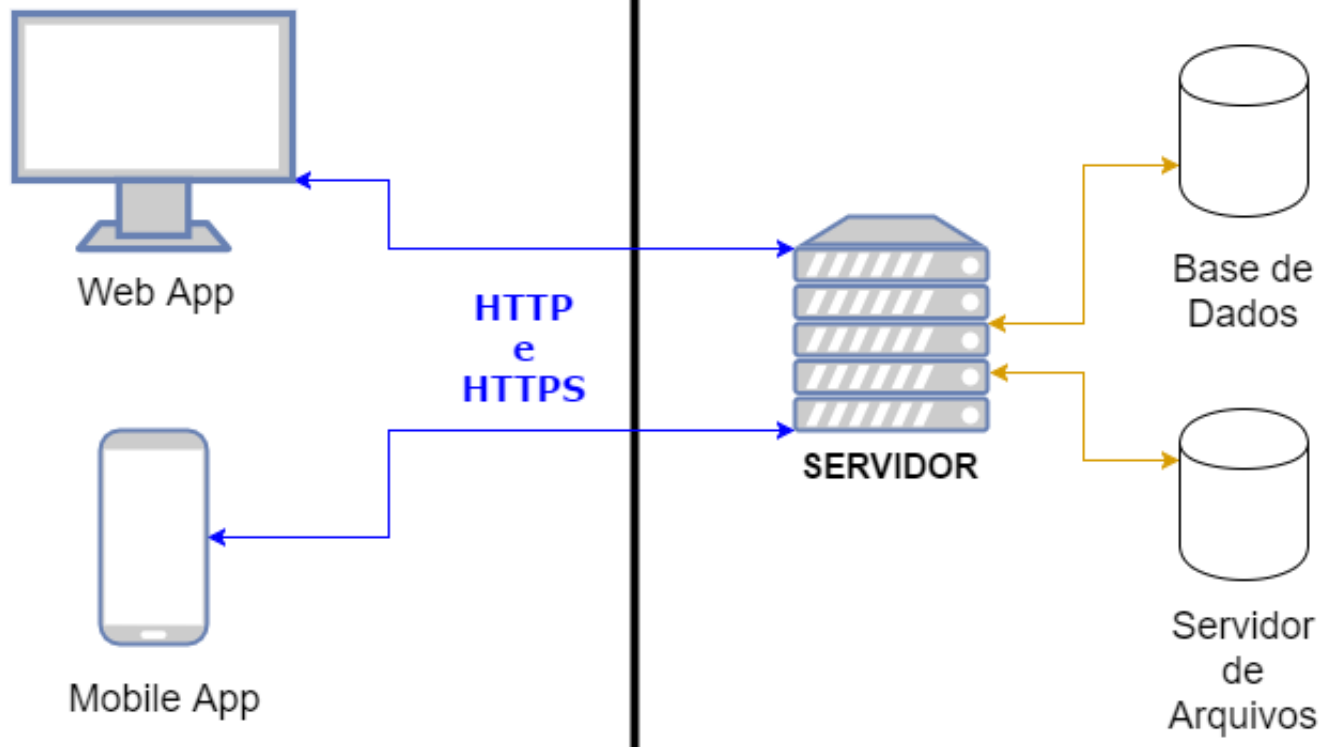
Divisão dos Frameworks em

- Client Side
- Server Side
- Full Stack



Front-end

Back-end



Frameworks Javascript - Aplicações

SPA (Single Page Applications) são aplicações completas, desenvolvidas em Javascript, e executadas no Navegador do Cliente.

- Web 2.0
- Aplicações Complexas
- Um dos maiores exemplos é o Gmail.
- Hoje temos muitos: Netflix, Whatsapp, Telegram,

Client Side
Front-End

Server Side
Back-end

Frameworks – Server Side

Frameworks – Server Side

Client Side
Front-End

Server Side
Back-end

No **lado do Servidor** temos aplicações responsáveis pelo acesso aos recursos do servidor e base de dados.

Existem inúmeras linguagens para se trabalhar:
PHP, ASP, JSP, Perl, Ruby, Python, entre outras.

E agora com **Javascript**

Frameworks – Server Side

Client Side
Front-End

Server Side
Back-end

Problema: Um projeto com diversas linguagens envolvidas atrasa o desenvolvimento. Cada linguagem precisava de ferramentas de teste e documentação.

Solução: Usar a mesma linguagem que se usa no lado do Cliente => **Javascript**

Frameworks – Server Side

Client Side
Front-End

Server Side
Back-end

Usando o servidor Node.js temos diversos frameworks:

- Express.js
- Hapi.js
- Total.js



Mais Frameworks aqui:

<http://nodeframework.com/>

Client Side
Front-End

Server Side
Back-end

Frameworks – Client Side

E Bibliotecas

Frameworks – Client Side

Client Side
Front-End

Server Side
Back-end

No **lado do Cliente** temos aplicações responsáveis pelo acesso aos recursos da máquina do cliente.

Frameworks fornecem diversas ferramentas para:

- Manipulação de DOM
- Comunicação Assíncrona
- Estruturação de código (Padrões MVC, MVP, MVVM)
- Desenvolvimento ágil

Frameworks – Client Side

Client Side
Front-End

Server Side
Back-end

Ember (<http://emberjs.com/>)



- O Ember é um projeto sólido
- Está começando a trabalhar com releases LTS (A cada 6 meses lança uma LTS)
- Estabilidade da API e Segurança

E.g.: [Twitch.tv](https://www.twitch.tv/), [Vine](https://vine.co/), Apple

Frameworks – Client Side



AngularJS (<https://angularjs.org/>)



- Criado e mantido pelo Google
- Estende o HTML (Linguagem Declarativa)
- Objetivo: Fácil de aprender e que funcione
- Two way data bind

<https://angular-ui.github.io/>

E.g.: **Telegram**, youtube(ps3), Google Analytics,

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

Frameworks – Client Side

Client Side
Front-End

Server Side
Back-end

AngularJS 2.0 (<https://angular.io/>)

- Não tem nada a ver com o Angular 1
- Usa TypeScript por padrão
- Trouxe melhoras no fluxo e uso de rotas
- Usa muitos dos benefícios do ES6 (Performance)
- Foi feito para aplicações grandes (modular)

<http://builtwithangular2.com/>



Frameworks – Client Side

Client Side
Front-End

Server Side
Back-end

Vue.js (<http://vuejs.org/>)



Vue.js

O Objetivo do Vue é ser simples e elegante.

- Implementa conceitos do Angular 1 porém diz ser mais elegante e robusto em apps maiores
- Cria web componentes parecido com o React. A vantagem é um código menor e mais limpo.

<https://vuejs.org/guide/comparison.html>

Frameworks – Client Side

Client Side
Front-End

Server Side
Back-end

Kendo-UI (<http://www.telerik.com/kendo-ui>)

- Framework pago e mantido pela Telerik (empresa de criação de ferramentas e softwares para desenvolvimento). Baseado em jQuery
- Framework com diversas ferramentas e que promete desenvolvimento ágil
- Conjunto compatível com outros frameworks
- Componentes prontos (<http://demos.telerik.com/kendo-ui/>)



 **Telerik** Kendo UI

Web Component

Client Side
Front-End

Server Side
Back-end

- Componentes personalizados ou criados por você, que encapsulam código HTML, Css e Javascript.
- Devem seguir o padrão da W3C para webComponents
- Reuso de componentes
- Facilita testes e desenvolvimento ágil

```
<google-map latitude="37.77493" longitude="-122.41942"></google-map>
```

<https://elements.polymer-project.org/elements/google-map>

Bibliotecas – Client Side

Client Side
Front-End

Server Side
Back-end

Polymer 1.0 (<https://www.polymer-project.org/1.0/>)

- Biblioteca do Google para Web Components (Google I/o 2015)
- Web Components: Reuso de código.
- Leva os padrões de Web Components para os browsers que não fazem isso nativamente
- Parecido com Angular1: Permite “two way data bind”

<https://builtwithpolymer.org/>



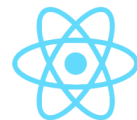
Bibliotecas – Client Side

Client Side
Front-End

Server Side
Back-end

React.js (<https://facebook.github.io/react-native/>)

- Biblioteca responsável pela *View Layer*.
- Facebook
- Web Components.
- Utiliza o Virtual DOM para renderizar apenas a sub-árvore e não toda a árvore DOM
- **React-Native** for mobile (ios and Android): Biblioteca para renderizar componentes nativos dos Browsers e SOs



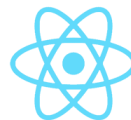
React

Bibliotecas – Client Side

Client Side
Front-End

Server Side
Back-end

React.js



React

Extensão JSX ajuda a escrever código Javascript com código HTML embutida

Editores: Atom, WebStorm, Sublime3(babel)

Plugin Chrome:

<https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi>

CDN: <https://cdnjs.com/libraries/react/>

Bibliotecas – Client Side

Client Side
Front-End

Server Side
Back-end



React

☐ Only show products in stock

Name	Price
------	-------

Sporting Goods	
----------------	--

Football	\$49.99
----------	---------

Baseball	\$9.99
----------	--------

Basketball	\$29.99
------------	---------

Electronics	
-------------	--

iPod Touch	\$99.99
------------	---------

iPhone 5	\$399.99
----------	----------

Nexus 7	\$199.99
---------	----------

☐ Only show products in stock

Name	Price
------	-------

Sporting Goods	
----------------	--

Football	\$49.99
----------	---------

Baseball	\$9.99
----------	--------

Basketball	\$29.99
------------	---------

Electronics	
-------------	--

iPod Touch	\$99.99
------------	---------

iPhone 5	\$399.99
----------	----------

Nexus 7	\$199.99
---------	----------

Client Side
Front-End

Server Side
Back-end

Frameworks – Full Stack

Frameworks – Full Stack

Client Side
Front-End

Server Side
Back-end

Objetivo: Criação de aplicações Isomórficas. Servidor e cliente compartilham o mesmo código

(<http://isomorphic.net/>)

Com um único Framework (conjunto de ferramentas) é possível desenvolver back-end e front-end.

- Desenvolvimento Ágil nos dois lados.
- Menos linguagens distintas envolvidas no projeto

Full Stack - MEAN

MEAN não é um framework, mas um conjunto de frameworks e ferramentas.

<http://mean.io/>

M – Mongo DB

E – ExpressJS

A – AngularJS

N – NodeJS



Meteor é um dos primeiros a oferecerem esse tipo de desenvolvimento.

<https://www.meteor.com/>
<http://pt.discovermeteor.com/>
<http://www.telescopeapp.org/>

- Inúmeras ferramentas para Deploy, Debug e Teste
- Desenvolvimento Ágil
- Utiliza muitos componentes do WebSocket (Sincronia com o servidor e outros clientes)
- Minimongo: Mini banco de dados implementado no cliente

Full Stack - Meteor



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

<https://www.meteor.com/developers>

<http://www.telescopeapp.org/>

<http://pt.discovermeteor.com/>

<https://github.com/SemDisquete/Hora-Do-Codigo>

Tendências 2016/2017

Tendências

Funções são objetos de primeira ordem:

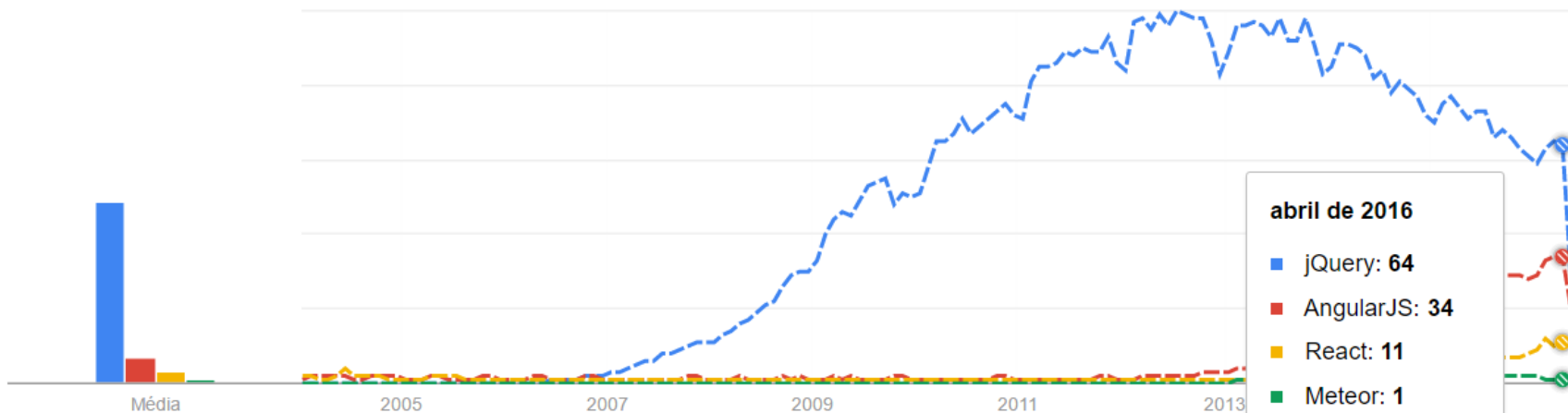
- React.js (React-Native and Redux)
- jQuery
- AngularJS 2.0
- Vue.js
- Meteor + Angular 2.0

Tendências

Interesse com o passar do tempo ?

☐ Títulos das notícias ?

☐ Previsão ?



<https://www.google.com/trends/explore#cmpt=q&geo=q=/m/0268gyp,+/m/0j45p7w,+/m/012l1vxv,+/m/0t545zt>

Tendências

Meteor + Angular 2.0



<http://angularjs.blogspot.com.br/2016/04/please-welcome-our-friend-uri.html>

React.js (React-Native and Redux)

Referências

- Rodrigo Branas. “***AngularJS Essentials Starting***”. Editora PACKT PUBLISHING. Agosto de 2014.
- Daniel Schmitz e Douglas Lira. “***AngularJS na prática***”. Editora Leanpub. Julho de 2015

Links úteis

- **HTML5, validação de formulários** (<http://blog.popupdesign.com.br/validando-formularios-like-a-boss-com-html5/>)
- **HTML5, validação de formulários 2** (<http://wbruno.com.br/html/validando-formularios-apenas-com-html5/>)
- **HTML5, validação de formulários 3** (<http://html5doctor.com/html5-forms-input-types/>)
- **HTML5 LocalStorage** (<https://zenorocha.com/html5-local-storage>)
- **Livro AngularJS na prática** (http://docplayer.com.br/278424-Pt_br-angularjs-na-pratica.html)
- **SPA e Modelos de dev** (<http://imasters.com.br/desenvolvimento/single-page-applications-e-outras-maravilhas-da-web-moderna/?trace=1519021197>)
- **Top Frameworks e quando usar** (<http://www.sitepoint.com/top-javascript-frameworks-libraries-tools-use/>)

Links úteis

- **Comparação de Angular, Ember e Backbone**

(<https://www.airpair.com/js/javascript-framework-comparison>)

- **Framework Kendo-ui** (<http://www.devmedia.com.br/apresentando-o-framework-kendo-ui/27285>)

- **WebComponents e Frameworks**

(<https://smthngsmwhr.wordpress.com/2015/04/13/web-components-and-friends-react-angular-polymer/>)

- **Prós e Contras do Meteor** (<http://www.slant.co/topics/3918/viewpoints/1/~full-stack-isomorphic-javascript-frameworks~meteor>)

- **Discussão sobre o estado da arte do JS 28/02/2016**

(<https://medium.com/javascript-and-opinions/state-of-the-art-javascript-in-2016-ab67fc68eb0b>)