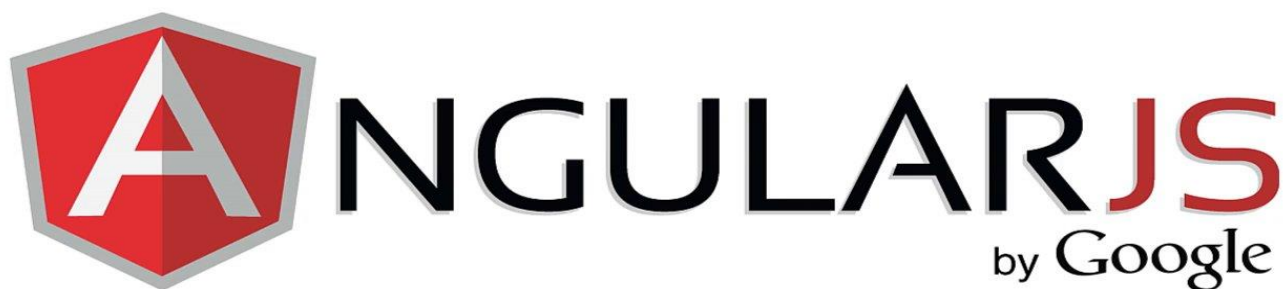


AngularJS



“Is what HTML would have been”

Guia Introdutório





Guia Introdutório ao AngularJS

Com a leitura, você terá uma introdução completa ao AngularJS.

Você ganhará intimidade com conceitos deste Framework em em [JavaScript](#).

Você encontrará os caminhos e indicações de conteúdos necessários para avançar seus estudos no tema.

Jorge Massari

Versão 0.1 (Beta) - Setembro de 2017



1)Índice

1) Índice	3
2) O que é AngularJS?	6
3) Como Usar o Angular	8
4) Motivação do AngularJS	9
5) Principais Características e Recursos do AngularJS	10
5.1. Padrão MVC (Model View Controller)	11
O que é Model-View-Controller (MVC)	11
O que é Arquitetura de Aplicação	11
Model-View-Controller (MVC) na Prática	13
As camadas do Modelo MVC	14
MVC no AngularJS	16
5.2. SPA: Single Page Application	17
O que é Single Page Application (SPA)	17
O que Significa SPA na Prática?	18
Tendência	19
O Papel do JavaScript, dos Navegadores e do HTML no Single Page Application (SPA)	20
Pioneiros no Uso de SPA	21



O que Não é Single Page Application (SPA)	22
Vantagens do Single Page Application (SPA)	25
Problemas Comuns com o Single Page Application (SPA)	26
5.3. O Recurso Two-Way Data Binding	27
5.4. Injeção de Dependência	29
5.5. Diretivas no AngularJS	30
O que São Diretivas no AngularJS	30
Exemplos de Diretivas no AngularJS	31
Exemplos de Uso das Diretivas no AngularJS	33
6) Angular Expressions	34
5.6. O que é Angular Expressions?	34
5.7. Exemplo de Angular Expression	34
7) Código fonte do AngularJS	35
8) Mais Sobre AngularJS	36



Observação Importante

Neste Ebook, você notará que existem **Links para conteúdos externos**.

Os links direcionam para páginas da web que escalarem os conceitos e expressões, e alguns casos a exemplos práticos / *templates*.



1) O que é AngularJS?

AngularJS é um framework em JavaScript de código aberto e que é mantido pelo Google. Seu objetivo é aumentar aplicativos que podem ser acessados por um navegador web e tem como padrão o [MVVM \(Model-View-View-Model\)](#), em um esforço para facilitar tanto o desenvolvimento quanto o teste dos aplicativos.

O framework AngularJS funciona através da leitura de páginas HTML, que tem embutido nelas atributos adicionais personalizados em suas tags. Angular interpreta esses atributos como as diretivas para ligar partes de entrada ou saída de página para um modelo que é representado por variáveis em padrão javascript. Os valores dessas variáveis javascript podem ser configurados manualmente no código ou recuperado a partir de recursos JSON estáticos ou dinâmicos.



O Angular JS disponibiliza recursos completos para facilitar a criação de um aplicativo CRUD:

- vinculação de dados,
- diretrizes básicas de modelos,
- validação de formulários,
- roteamento,
- componentes reutilizáveis
- injeção de dependência.

Angular segue o padrão MVC da engenharia de Software e encoraja o baixo acoplamento entre apresentação, dados e componentes lógicos.



2) Como Usar o Angular

AngularJS é distribuído como um arquivo JavaScript e pode ser adicionado a uma página da web com uma tag de script:

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.4/angular.min.js"></script>
```



3) Motivação do AngularJS

O AngularJS surgiu a partir da necessidade de evolução dos frameworks de [Javascript](#). O Javascript é uma linguagem de programação web frontend, isso é, programação ligada à interface com o usuário.

Não é o primeiro framework popular que surgiu neste sentido. Muitos outros são utilizados por desenvolvedores, a exemplo do jQuery, que é uma biblioteca que facilita o controle de elementos de uma página HTML, também chamado de DOM. Outros exemplos de frameworks de Javascript:

- [JavascriptMVC](#),
- [Backbone.js](#),
- [Knockout](#).

Com o passar dos anos, tornou-se tendência que linguagem que é executada no lado do cliente (navegador) pudesse também ser executada no lado do servidor, gerando a plataforma [NODE.JS](#) (um interpretador de código [JavaScript](#)), hoje em dia mundialmente conhecida e utilizada.

Para saber mais sobre NODE.JS, acesse: [O que é NODE.JS](#)



4) Principais Características e Recursos do AngularJS

- Utiliza [Padrão MVC \(Model View Controller\)](#)
 - É Baseado no Conceito [SPA: Single Page Application](#)
 - Utiliza Recurso Two-Way Data Binding
 - Utiliza o Conceito de Injeção de Dependência
- Utiliza [Recurso de Diretivas](#)

5.1. Padrão MVC (Model View Controller)

O que é Model-View-Controller (MVC)

É um padrão de arquitetura de aplicações que divide a aplicação em três camadas: a visão (view), o modelo (model), e o controlador (controller). Traduzido para o português, a expressão significa: modelo-visão-controlador.

O padrão MVC foi desenvolvido em 1979 por [Trygve Reenskaug](#) com a finalidade de ser utilizado como arquitetura para aplicativos desktop. Entretanto, o padrão se popularizou para uso em sistemas web, a partir da adesão de milhares de Frameworks de mercado.

O que é Arquitetura de Aplicação

A arquitetura de uma aplicação nada mais é do que um modelo que define as suas estruturas. Tal estrutura engloba:

- componentes de software,
- propriedades dos componentes / elementos,
- os relacionamento entre os componentes / elementos,



- e , enfim, todos os elementos que fazem parte da estrutura básica padrão do software e como estes elementos interagem / interagem entre si.
- Alguns exemplos de elementos: utilitários, elementos de interação, elementos de conexão, elementos de persistência.

Existem arquiteturas padrão de mercado e arquiteturas específicas, criadas e usadas por empresas que desenvolvem software (neste segundo caso, configura o papel de um arquiteto de software).



Model-View-Controller (MVC) na Prática

Em termos práticos, e de forma resumida, utilizar do padrão MVC significa:

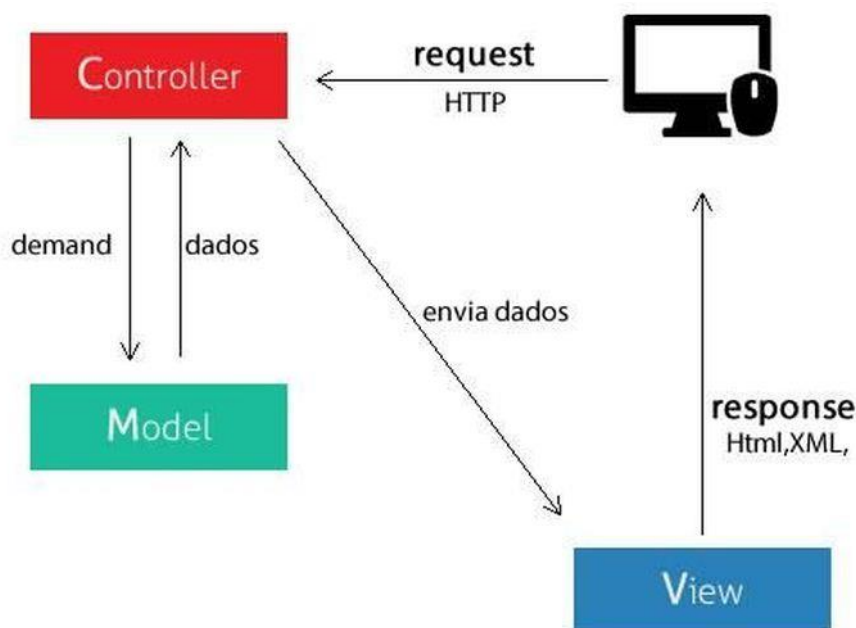
Dividir a aplicação em camadas: uma da interface do usuário denominada View, uma para manipulação lógica de dados chamada Model, e uma terceira camada de fluxo da aplicação chamada Control)

Criar a possibilidade de exibir uma mesma lógica de negócios através de várias interfaces.

Isolar a camada de negócios (Model) das demais camadas do sistema, de forma a facilitar a sustentabilidade do código

A implementação do controlador deve permitir que esta camada receba os eventos da interface e os converta em ações no modelo.

As camadas do Modelo MVC



A Camada View do MVC

- É a camada que exibe uma representação dos dados.
- Realiza a interface com usuário (view).
- Também conhecida como cliente-side.
- Faz a exibição dos dados, utilizando-se de #HTML e/ou XML.
- É responsável por usar as informações modeladas para produzir interfaces de apresentação conforme a necessidade.



A Camada Model do MVC

- É a camada que contém a estrutura de dados de uma parte específica da aplicação
- Usualmente portada em [JSON](#).
- Responsável pela leitura, manipulação e validação de dados, e também de suas validações.
- Responsável por tratar as regras de negócio.
- Obtém os dados e os traduz em informações relevantes para serem exibidas pela View.
- Notifica a view e o controlador associados quando há uma mudança em seu estado.

A Camada Controller do MVC

- É a camada de controle.
- Exerce o controle de qual modelo deverá ser aplicado e qual view será mostrado ao usuário.
- Podemos dizer que esta camada faz uma gerência das outras duas camadas.
- O controller manipula e roteia as requisições dos usuários.
- Interpreta as requisições submetidas pelo usuário e traduz em comandos que são enviados para o (Model) e/ou para a View) .



- Valida as requisições dos usuários de acordo com as regras de autenticação e autorização.

MVC no AngularJS

O AngularJS utiliza da seguinte forma estas camadas:

- A camada de interação (**view**) é o próprio DOM (Document Object Model)
- A camada de manipulação (**model**) é formada por dados armazenados em formato de objetos
- A camada de controle (**controllers**) é concebida por classes [Javascript](#)



5.2. SPA: Single Page Application

O que é Single Page Application (SPA)

SPA significa Aplicação de Página Única, e trata-se de um método - relativamente novo - para o desenvolvimento web baseado em técnicas de front-end, usando sobretudo de JavaScript e seus frameworks.

As praticas baseadas em SPA envolvem codificar menos no server-side e mais no client-side, proporcionando ao usuário uma aplicação dinâmica, que carrega recursos conforme necessário. Quando um usuário acessa um site construído dentro do conceito de SPA, a aplicação e seus componentes são armazenados no lado cliente, diferente da forma tradicional, quando o usuário precisa visitar várias páginas diferentes e recarregá-las para visualizar atualizações.

A ideia de uma página SPA é desenvolver uma em sua totalidade (ou quase totalidade) no cliente-side. É praticamente uma aplicação Desktop rodando no navegador / App. O server-side muitas vezes se encarrega apenas das funções de comunicação com o banco de dados



O que Significa SPA na Prática?

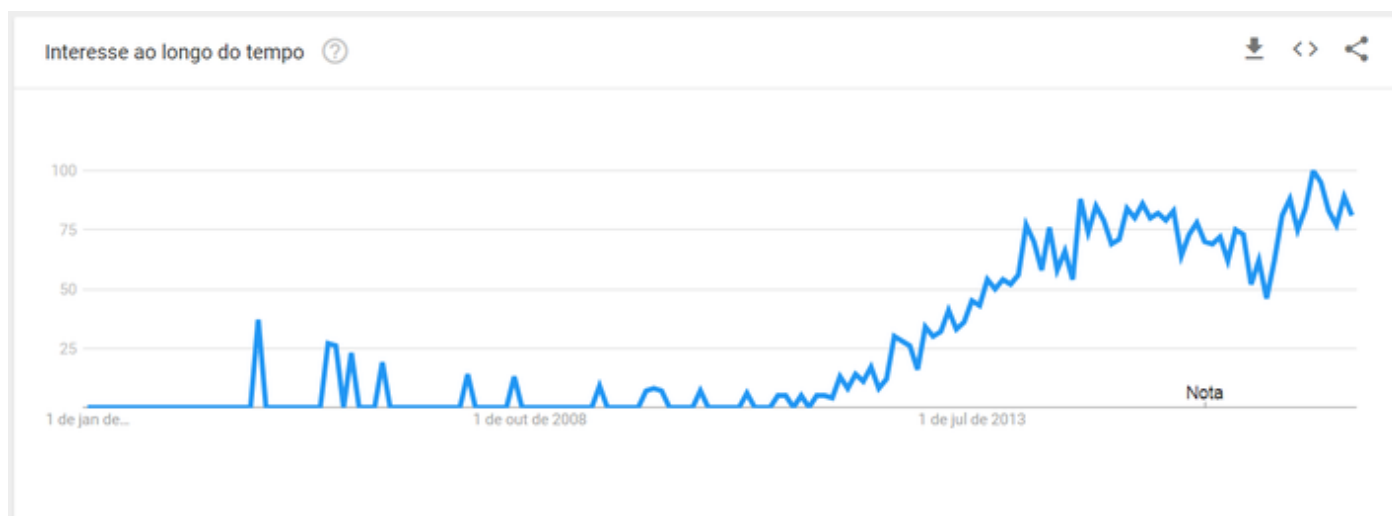
Em termos práticos, SPA trata-se de adotar padrões de desenvolvimento tais como:

- Priorizar e técnicas que incrementem a performance dos scripts
- Construir aplicações inteiramente contidas no browser que não façam requisições de novas páginas no servidor
- "Usar e abusar" de JavaScript e seus frameworks
- Usar padrões como Model-View-Controller (MVC), Model-View-Presenter (MVP) ou ou Model-View-ViewModel (MVVM)
- Criar diversas Views (camada de interface) para um mesmo modelo de dados (camada de manipulação dos dados)
- Separar de forma bem clara a interface visual e persistência de dados



Tendência

Conforme imagem a seguir, através uma simples busca no Google Trends pela expressão Single Page Application (SPA), já é possível ter uma noção do crescimento de sua popularidade.



Pesquisa "Single Page Application (SPA)" no Google Trends



O Papel do JavaScript, dos Navegadores e do HTML no Single Page Application (SPA)

JavaScript

Pode-se dizer que a evolução do JavaScript teve um papel fundamental na chegada do conceito de SPA.

Inicialmente, esta linguagem tinha um uso bastante restrito às poucas funções dentro da camada de interação com o usuário (view), tais como habilitar botões, validar preenchimento de campos em formulários e desenvolver menus. Hoje, assumiu um papel mais significativo no [Desenvolvimento](#) web.

Para termos uma ideia desta evolução - de qual falo no parágrafo anterior - [o ranking de linguagens da TIOBE](#) aponta que o Javascript pulou da 12ª para a 6ª posição entre 2001 e 2016, e hoje está na 8ª colocação (Agosto/2017).



Navegadores e o HTML

A evolução dos browsers e do [HTML](#), por sua vez, deram espaço para que o JavaScript desenvolvesse poder de processamento necessário para oferecer mais recursos aos programadores front-end. Hoje, navegadores web deixaram de ser uma simples janela para a visualização da aplicação, e se comportam como uma parte essencial no processamento, manipulação e controle das informações.

Pioneiros no Uso de SPA

- [Google Gmail](#)
- [Facebook](#)
- [Twitter](#)
- [Instagram](#)
- [Trello](#)
- [Foursquare](#)



O que Não é Single Page Application (SPA)

O conceito de SPA pode ser confundido com outros ligados a programação. Para que isso não aconteça com você, seguem dois outros conceitos com os respectivos pontos que os distinguem de SPA.

Efeito Parallax

É um efeito essencialmente gráfico, muito usado em em sites de uma única página que disponibiliza um conteúdo que você vê parte a parte, à medida em que rolar a página, enquanto o fundo é fixado. O parallax é, portanto, uma técnica que gera como resultado um conteúdo estático, ao contrário do SPA que disponibiliza recursos de forma dinâmica, à medida em que o usuário cria necessidade. [Veja aqui um exemplo de página que usa o efeito Parallax.](#)

Infinite Scroll

Muito menos o SPA pode ser comparado a um simples Scroll infinito. O Infinite Scroll já é utilizado para reduzir o tempo de carregamento de páginas na web, fracionando-o à medida em que você navega top-down.



Ficou confuso?

A grande diferença para estas duas técnicas, é que em uma SPA o que irá ou não ser carregado, a medida em que você navega em uma página, é disponibilizado de forma dinâmica.

Além do mais, uma SPA possui diversas outras propriedades de dinamicidade, que vão além de uma simples disponibilização de conteúdo a partir da navegação de Scroll.

Exemplos:

- A timeline no Facebook é um exemplo legítimo de uso de práticas de SPA.
- Tomando a mesma rede social como exemplo, um comentário novo que surge em sua publicação - sem que você tenha precisado carregar a página - é um exemplo de uso das propriedades do conceito de SPA.
- Se você digita caracteres em um campo de pesquisa (em um determinado site / sistema web) e o site exibe os resultados à medida que você digita, então este site está usando uma técnica de busca que se



enquadra nas boas práticas de Single Page Application (SPA)

- Navegando [no próprio Portal GSTI](#), você poderá identificar algumas técnicas que se enquadram nas práticas de uma SPA: atualizações, inserção dinâmica de comentários, alertas, dentre outras ações que ocorrem no cliente-side dinamicamente, sem que você precise dar um refresh na página.



Vantagens do Single Page Application (SPA)

Melhor experiência do usuário

O usuário vive uma experiência muito parecida com a de utilizar um sistema desktop, já que não precisa realizar refreshs na página.

Melhor Performance

O conteúdo é carregado de forma faseada, sem que o usuário tenha que aguardar pelo carregamento completo de uma página.

Sustentabilidade

As divisões em camadas, aplicadas por práticas de SPA, facilitam a manutenção de sistemas / websites.

Menor consumo de banda

As cargas de dados são realizadas conforme a necessidade, o que economiza o carregamento de informações que não são / seriam utilizadas pelo usuário.



Problemas Comuns com o Single Page Application (SPA)

Curva de Aprendizado

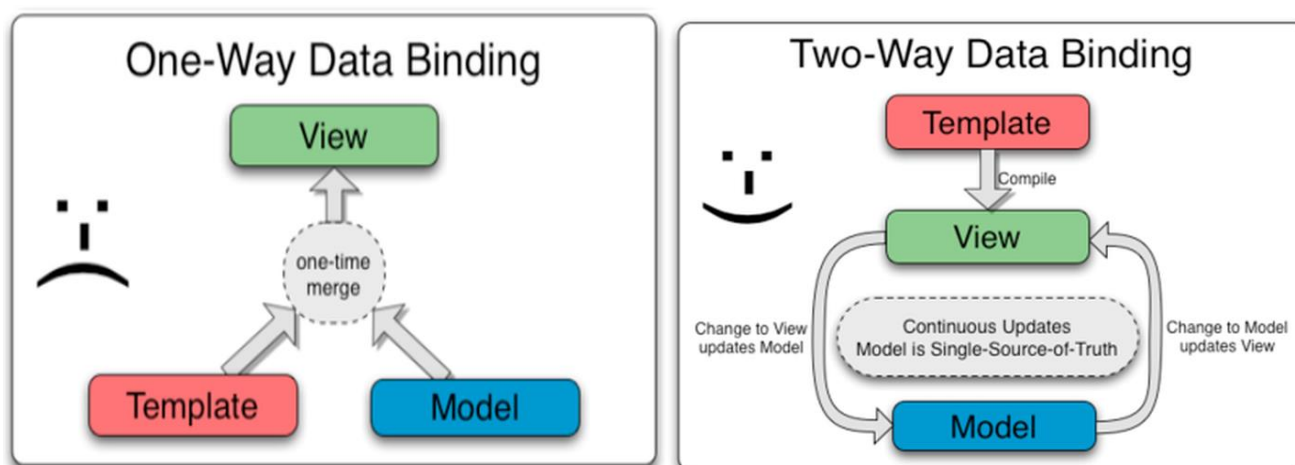
Assim como toda nova tendência, a utilização das práticas extremamente voltadas para front-end requer aprendizado. Exige um bom conhecimento de JavaScript e de seus Frameworks.

#SEO

Esta é uma provável desvantagem para quem mantém sites e/ou blogs: apesar do Google já ter um sistema que indexa o retorno das requisições AJAX, há dúvidas com relação a sua eficácia de indexação.

5.3. O Recurso Two-Way Data Binding

Aplicações tradicionais vinculam dados em em uma única direção, que passam pelo modelo até serem disponibilizadas na view (camada do usuário). O desenvolvedor precisa escrever um código que faça uma sincronia de maneira constante através do modelo para que a view esteja sempre atualizada.



One-Way e Two-Way Data Binding

O AngularJs, fornece um recurso de Data Binding que permite esta sincronização de dados de forma automática, entre os componentes do model e da view. O que é exibido na interface do usuário é uma projeção do model, atualizado a todo instante. Quando o modelo muda, a visão reflete esta mudança, e vice-versa.



Esta facilidade é possível através da adição de alguns controles (atributos) diretamente no HTML, de forma que seja possível identificar e controlar os atributos a serem atualizados.



5.4. Injeção de Dependência

Injeção de Dependência é um padrão de design de software que lida com a forma como os componentes adotam suas dependências. Com o AngularJ as dependências dos componente ficam explícitas, facilitando o gerenciamento. Você pode manipular explicitamente as dependências conforme necessidade. Desta forma, a Injeção de dependência tornar o código mais flexível, testável e organizado.

O AngularJS reconhece os parâmetros dos controllers e injeta as respectivas dependências. Existem 3 formas de declarar dependências: Arrays, \$inject ou usando diretamente os parâmetros de uma função.



5.5. Diretivas no AngularJS

O que São Diretivas no AngularJS

Diretivas são extensões da linguagem HTML, que fornecem a possibilidade de estender/ampliar o comportamento de elementos HTML. Este recurso permite a implementação de novos comportamentos de forma declarativa.

Ao seleccionar um elemento HTML, a diretiva pode ampliar seu comportamento de diversas formas:

- para adicionar um novo HTML,
- associar eventos à funções Javascript,
- manipular o DOM.



Exemplos de Diretivas no AngularJS

ng-app	Declara um elemento como o elemento raiz da aplicação, ocasionando a mudança do comportamento padrão da tag.
ng-bind	Muda o texto de um elemento HTML automaticamente, de acordo com o seu resultado, vindo das regras de negócio.
ng-model	É similar ao ng-bind, mas permite ligação direta bidirecional (two-way data binding) entre a view e o escopo do aplicativo.
ng-class	Permite atributos de classe serem carregados dinamicamente.
ng-click	Permite instanciar o evento de click, semelhante ao onclick.
ng-controller	Especifica um controller JavaScript para aquele HTML em questão.
ng-repeat	Instancia um elemento por item de um array.
ng-show & ng-hide	Mostra ou esconde um elemento HTML de acordo com o resultado de uma expressão booleana.
ng-switch	Instancia um template, em uma lista de escolhas, dependendo do valor obtido pela expressão.
ng-view	A diretiva base para manipulação de rotas, resolvendo um JSON antes de renderizar os modelos acionados por controladores especificados.



ng-if	Declaração básica de um 'if' que permite mostrar um elemento se a condição for verdadeira.
-------	--



Exemplos de Uso das Diretivas no AngularJS

A seguir, um exemplo de uso das diretivas ng-app, ng-bind e ng-model

```
<!DOCTYPE html>
<html>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.4/angular.min.js"></script>
<body>

<div ng-app="">
  <p>Name: <input type="text" ng-model="name"></p>
  <p ng-bind="name"></p>
</div>

</body>
</html>
```

Explicando o Exemplo:

- O AngularJS é iniciado automaticamente quando a página da Web foi carregada.
- A diretiva ng-app diz ao AngularJS que o elemento <div> é o "proprietário" de um aplicativo AngularJS.
- A diretiva do modelo ng-model vincula o valor do campo de entrada ao nome da variável da aplicação.
- A diretiva ng-bind liga o innerHTML do elemento <p> ao nome da variável da aplicação.



5) Angular Expressions

5.6. O que é Angular Expressions?

São trechos de códigos em Javascript, que no caso do AngularJS devem ser colocados entre chaves: {{expression}}. As expressions em AngularJS também

podem ser escritas dentro de uma diretiva: ng-bind = "expression"

5.7. Exemplo de Angular Expression

```
<!DOCTYPE html>
<html>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.4/angular.min.js"></script>
<body>
<div ng-app="">
  <p>Exemplo de Expression: {{ 5 + 5 }}</p>
</div>
</body>
</html>
```



6) Código fonte do AngularJS

- [Acesse aqui o código fonte do AngularJS](#)



7) Mais Sobre AngularJS

Acesse os links a seguir e continue estudando o AngularJS

- [Videoaulas gratuitas de AngularJS](#)
- [Cursos online de AngularJS](#)
- [Ebooks e apostilas de AngularJS](#)
- [Comundiade AngularJS](#)
- [Faça uma pergunta sobre AngularJS](#)