

# Introdução ao Angular

Usabilidade, desenvolvimento web, mobile e jogos

Prof. Me. Gustavo Torres Custodio  
gustavo.custodio@ulife.com.br

# Conteúdo

## AngularJS

### Introdução

### Data-Binding

### Diretivas



Introducao ao Angular

AngularJS

# AngularJS

- O **AngularJS** é um framework de JavaScript.
- Ele é distribuído por meio de um arquivo JavaScript.
  - Assim como código JavaScript comum, o código pode ser adicionado a uma página HTML utilizando a tag `<script>`.
- A página HTML se torna um template que permite conteúdo mais complexo do que HTML comum.

# História

- O AngularJS é um framework de front-end para desenvolver aplicações *single-page*.
  - A aplicação atualiza elementos da página sem precisar recarregar toda a página.
- O AngularJS foi originalmente desenvolvido em 2009 por Misko Hevery na Brat Tech LLC.
  - Eventualmente o AngularJS foi disponibilizado como uma biblioteca de código aberto.

# História

- O AngularJS era mantido pelo Google.
  - Atualmente ele foi descontinuado.
  - Os desenvolvedores recomendam **atualizar para o Angular**.

## Comparação

AngularJS	Angular
JavaScript Distribuído com um arquivo .js Sem suporte para mobile	TypeScript Distribuído pelo npm Suporte para mobile



Introducao ao Angular

# Introdução



## Download AngularJS

- Vamos criar um projeto incluindo o AngularJS:
  - Disponível em <https://angularjs.org/>.
  - Salve a página como `angular.js` no diretório do seu projeto.

# A Primeira Página

- Vamos criar um “Hello, World!” utilizando o AngularJS

```
<!DOCTYPE html>
<html ng-app="helloWorld">
  <head>
    <script src="angular.js"></script>
    <script>
      // Criação do app
      var app = angular.module("helloWorld", []);
      // Criação do controller
      app.controller("HelloController", function ($scope) {
        $scope.boasvindas = 'Hello, World!';
      });
    </script>
  </head>
  <body>
    <div ng-controller='HelloController'>
      <p>{{boasvindas}}</p>
    </div>
  </body>
</html>
```

## A Primeira Página

- O comando `angular.module` cria um módulo chamado `helloWorld`.
- O `controller` controla dados da aplicação e o comportamento dela.
  - Adicionar propriedades dentro da variável **\$scope** permite que elas sejam visualizadas pela página (*view*).



Introducao ao Angular

# Data-Binding

## Data Binding

- Bibliotecas como o jQuery permitem modificar parte do conteúdo de uma página HTML sem recarregar a página toda.
  - No entanto, esse dinamismo se perde quando tentamos realizar tarefas como **manter alguma informação sempre atualizada** ou então **mudar algo baseada no input do usuário**.
- Existe a necessidade de “ligar” elementos da página HTML com propriedades do JavaScript.
  - Chamamos isso de *Data Binding*.

# Data Binding

- Vamos criar uma página que muda dinamicamente o conteúdo de um texto de acordo com o valor de um campo de texto.
  - Reutilizaremos o código usado anteriormente.

# Data Binding

```
<!DOCTYPE html>
<html ng-app="helloWorld">
  <head>
    <script src="angular.js"></script>
    <script>
      // Criação do app
      var app = angular.module("helloWorld", []);
      // Criação do controller
      app.controller("HelloController", function ($scope) {
        $scope.boasvindas = 'Hello, World!';
      });
    </script>
  </head>
  <body>
    <div ng-controller='HelloController'>
      <input ng-model='boasvindas'>
      <p>{{boasvindas}}</p>
    </div>
  </body>
</html>
```

- Neste código foi adicionado um campo de texto ligado ao atributo “boasvindas”.
- Se mudarmos o valor do campo de texto, o valor do atributo também muda e é imediatamente atualizado na página.

# Data-Binding

- Podemos também adicionar operações por data-binding.

```
<!DOCTYPE html>
<html ng-app="contas">
  <head>
    <script src="angular.js"></script>
    <script>
      // Criação do app
      var app = angular.module("contas", []);
      // Criação do controller
      app.controller("OpController", function ($scope) {
        $scope.tarefa = "Vamos fazer contas.";
        $scope.termo1 = 1;
        $scope.termo2 = 2;
      });
    </script>
  </head>
  <body>
    <div ng-controller='OpController'>
      <h1>O que vamos fazer?</h1>
      <h2>{{tarefa}}</h2>
      <p>Soma: {{termo1}} + {{termo2}} = {{termo1 + termo2}}</p>
      <p>Subtração: {{termo1}} - {{termo2}} = {{termo1 - termo2}}</p>
      <p>Multiplicação: {{termo1}} * {{termo2}} = {{termo1 * termo2}}</p>
      <p>Divisão: {{termo1}} / {{termo2}} = {{termo1 / termo2}}</p>
    </div>
  </body>
</html>
```



## Exercício

- Crie uma página com seu nome, a data de hoje e hora de acesso.
- Utilize as funções de data do JavaScript para recuperar a data de hoje.



Introducao ao Angular

# Diretivas

## Diretivas

- As palavras-chave `ng-model` e `ng-controller` são chamadas de **diretivas**.
- Diretivas adicionam novos atributos além daqueles definidos pelo HTML comum.
- Elas são adicionadas dentro de tags HTML e permitem modificar seu comportamento.
  - Podemos, por exemplo, utilizar estruturas como *loops* de repetição dentro de nossas páginas HTML.

## Diretivas - ng-if e ng-repeat

- As diretivas ng-if e ng-repeat simulam o uso de estruturas de decisão e repetição.
  - O ng-if é especialmente útil para situações em que a informação mostrada é condicional.
  - O ng-repeat permite percorrer um conjunto de dados e mostrar eles na tela.

# Diretivas - ng-if e ng-repeat

```
<!DOCTYPE html>
<html ng-app="paísesApp" >
  <head>
    <style>
      table, td {
        border: 1px solid grey;
      }
    </style>
    <meta charset="utf-8">
    <script src="angular.js"></script>
    <script src="controller.js"></script>
    <title>Países</title>
  </head>
  <body>
    <div ng-controller="PaísesController">
      <table>
        <tr ng-repeat="x in capitais">
          <td ng-if="$odd" style="background-color:#f1f1f1">
            {{x.capital }}</td>
          <td ng-if="$even">
            {{x.capital }}</td>
          <td ng-if="$odd" style="background-color:#f1f1f1">
            {{x.pais }}</td>
          <td ng-if="$even">
            {{x.pais }}</td>
        </tr>
      </table>
    </div>
  </body>
</html>
```

```
var app = angular.module('paísesApp', []);
app.controller('PaísesController', function($scope) {
  $scope.capitais = [{capital: 'São Paulo', pais: 'Brasil'},
                    {capital: 'Madrid', pais: 'Espanha'},
                    {capital: 'Alemanha', pais: 'Munique'}];
});
```

- O código imprime uma tabela com as capitais dos países alternando entre cores.

# Diretivas - Carrinho de Compras

- Vamos criar um carrinho de compras.

## carrinho.html

```
<html ng-app='carrinhoApp'>
  <head>
    <title>Seu Carrinho de Compras</title>
    <script src="angular.js"></script>
    <script src="controller.js"></script>
  </head>
  <body ng-controller='CarrinhoController'>
    <h1>Seu Pedido</h1>
    <div ng-repeat='item in items'>
      <p>{{item.nome}}</p>
      <input ng-model='item.quantidade'>
      <!-- currency muda a apresentação para formato monetário -->
      <p>{{item.preco | currency}}</p>
      <p>{{item.preco * item.quantidade | currency}}</p>
      <!-- O índice está associado a cada elemento -->
      <button ng-click="remove($index)">Remover</button>
    <hr>
  </div>
</body>
</html>
```

## controller.js

```
// Criação do App
var app = angular.module("carrinhoApp", []);
app.controller("CarrinhoController", function ($scope) {
  // Criando um Array com os elementos
  $scope.items = [
    {nome: 'Cabo HDMI', quantidade: 1, preco: 22.00},
    {nome: 'Teclado', quantidade: 3, preco: 300.00},
    {nome: 'Pen Drive', quantidade: 2, preco: 100.00}
  ];
  $scope.remove = function(index) {
    $scope.items.splice(index, 1);
  }
});
```

# Diretivas - Carrinho de Compras

## Seu Pedido

Cabo HDMI

1

\$22.00

\$22.00

Remover

Teclado

3

\$300.00

\$900.00

Remover

Pen Drive

2

\$100.00

\$200.00

Remover

## Diretivas - Carrinho de Compras

- `items` é um array contendo um conjunto de produtos.
  - Quando um elemento é removido de `items`, imediatamente isso é atualizado no template HTML.
- A variável `$index` armazena a iteração atual do `ngRepeat`.
- Queremos a função `remove()` disponível na página HTML, então adicionamos ela no `$scope`.



## Exercício

- Crie uma aplicação com dois campos de texto: altura e peso.
- Mostre para o usuário o resultado do IMC, calculado por:

$$IMC = \frac{peso}{altura \times altura}$$

## Exercício

- Use a diretiva ng-if para dizer a situação do usuário de acordo com o IMC:
  - $\text{IMC} < 18$ : sobrepeso.
  - IMC maior que 18 e menor que 25: normal.
  - IMC maior ou igual a 25 e menor que 30: sobrepeso.
  - IMC maior ou igual a 30 e menor que 35: obesidade grau 1.
  - IMC maior ou igual a 35 e menor que 40: obesidade grau 2.
  - IMC maior que 40: obesidade mórbida.

Obrigado

[gustavo.custodio@ulife.com.br](mailto:gustavo.custodio@ulife.com.br)