

# Introdução ao AngularJS

Professor Anderson Henrique



# Baixando o AngularJS

Acesse o site <https://angularjs.org>



## Download AngularJS



Branch

1.8.x (latest)

1.2.x (legacy)



Build

Minified

Uncompressed

Zip



CDN

`https://ajax.googleapis.com/ajax/libs/angularjs/1.8.0/angular.min.js`



Bower

`bower install angular#1.8.0`



npm

`npm install angular@1.8.0`

Extras

[Browse additional modules](#)

[Previous Versions](#)



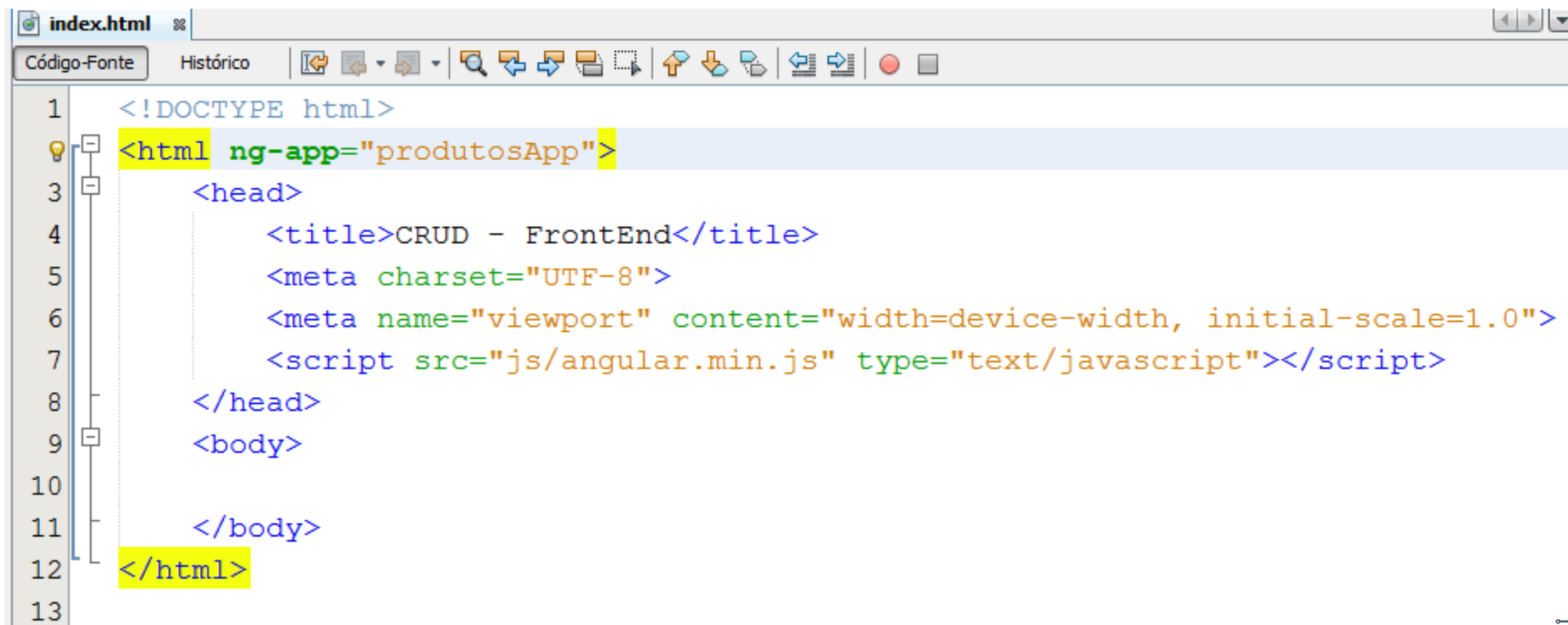
Download

Professor Anderson Henrique

# Utilizando Diretivas

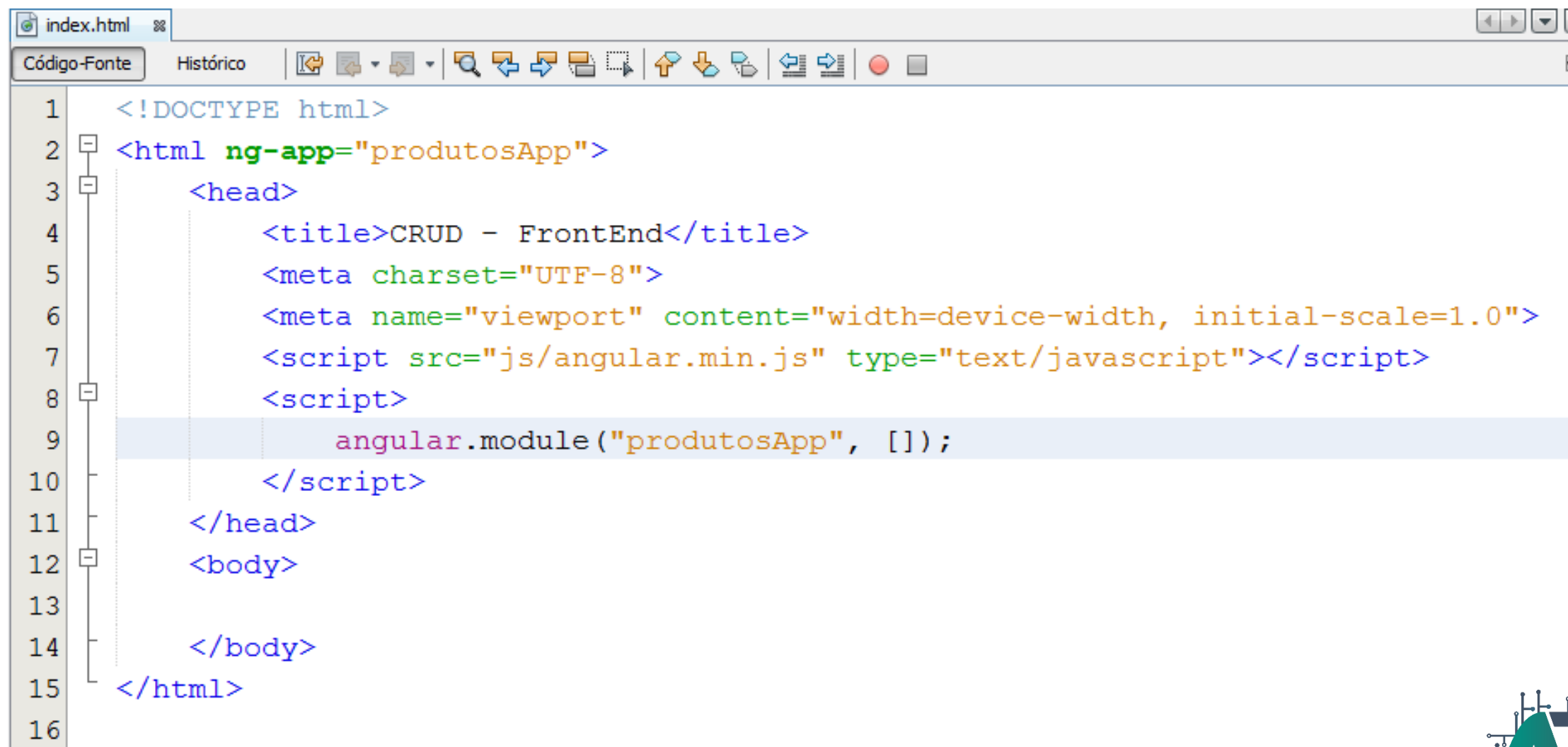
Diretivas são extensões da linguagem HTML que permitem a implementação de novos comportamentos de forma declarativa

# ngApp – definindo as fronteiras da aplicação



The screenshot shows a code editor window titled 'index.html'. The editor has a toolbar with various icons for file operations and a sidebar with a file tree. The main area displays the following HTML code:

```
1  <!DOCTYPE html>
2  <html ng-app="produtosApp">
3      <head>
4          <title>CRUD - FrontEnd</title>
5          <meta charset="UTF-8">
6          <meta name="viewport" content="width=device-width, initial-scale=1.0">
7          <script src="js/angular.min.js" type="text/javascript"></script>
8      </head>
9      <body>
10
11      </body>
12  </html>
```



```
1 <!DOCTYPE html>
2 <html ng-app="produtosApp">
3   <head>
4     <title>CRUD - FrontEnd</title>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <script src="js/angular.min.js" type="text/javascript"></script>
8     <script>
9       angular.module("produtosApp", []);
10    </script>
11  </head>
12  <body>
13
14  </body>
15 </html>
16
```

## ngController – vinculando um elemento da View ao Controller

```
22 <body ng-controller="produtosController">
23   <h2>Cadastro de Produto</h2>
24   <form>
25     <input type="text" placeholder="Código:"><br><br>
26     <input type="text" placeholder="Descrição:"><br><br>
27     <input type="text" placeholder="Preço:"><br><br>
28     <button> Salvar </button>
29   </form>
30   <br>
```

```
8 <style>
9   table{
10     width: 90%;
11     text-align: center;
12   }
13   table thead{
14     background: #333;
15     color: #fff;
16   }
17 </style>
18 <script>
19   angular.module("produtosApp", []);
20   angular.module("produtosApp").controller("produtosController",function($scope) {
21
22     });
23 </script>
```



Criamos um array produtos e vinculamos a view através do parâmetro \$scope do nosso controller

```
18 <script>
19     angular.module("produtosApp", []);
20     angular.module("produtosApp").controller("produtosController", function($scope) {
21         var produtos = [
22             {id: 1, descricao: "Arroz", preco: 12.50},
23             {id: 2, descricao: "Feijão", preco: 5.50}
24         ];
25
26         $scope.produtos = produtos;
27     });
28 </script>
```

# ngRepeat – iterando sobre itens de uma coleção ou de um objeto

```
48 <tbody>  
49   <tr ng-repeat="produto in produtos track by $index">  
50     <td></td>  
51     <td></td>  
52     <td></td>  
53   </tr>  
54 </tbody>  
55 </table>
```

## ngBind - substituindo o elemento por uma expressão

```
48 <tbody>
49   <tr ng-repeat="produto in produtos track by $index">
50     <td ng-bind="produto.id"></td>
51     <td ng-bind="produto.descricao"></td>
52     <td ng-bind="produto.preco"></td>
53   </tr>
54 </tbody>
```

## Interpolação da expression – {{expression}} – [ng-bind]

```
50 <td>{{produto.id}}</td>
51 <td>{{produto.descricao}}</td>
52 <td>{{produto.preco}}</td>
```

## ngModel – vinculando uma propriedade ao objeto \$scope

```
33 <form>
34   <input type="text" placeholder="Código:" ng-model="produto.id"><br><br>
35   <input type="text" placeholder="Descrição:" ng-model="produto.descricao"><br><br>
36   <input type="text" placeholder="Preço:" ng-model="produto.preco"><br><br>
37   <button> Salvar </button>
38 </form>
```

## ngClick – atribuindo um comportamento (function) ao evento

```
37   <button ng-click="salvar(produto)"> Salvar </button>
38 </form>
```

Criamos a função e vinculamos a view através do parâmetro \$scope do nosso controller

```
28  
29     $scope.salvar = function(produto) {  
30         produtos.push(produto);  
31         $scope.produto = {};  
32     };  
33     });  
34 </script>
```

Vamos adicionar uma coluna na nossa tabela e criar um link para excluir produto, utilizando o **ng-click** para vincular o evento a função no \$scope

```
68     <td><a href="" ng-click="excluir(produto)">[Excluir]</a></td>  
69 </tr>  
70 </tbody>
```

Criamos a função e vinculamos a view através do parâmetro \$scope do nosso controller

```
45 $scope.excluir = function(produto){
46     for(var i = 0; i < produtos.length; i++){
47         if(produtos[i].id == produto.id){
48             produtos.splice(i, 1);
49             break;
50         }
51     }
52 };
```

Vamos adicionar uma coluna na nossa tabela e criar um link para editar produto, utilizando o **ng-click** para vincular o evento a função no \$scope

```
69 <td><a href="" ng-click="excluir(produto)">[Excluir]</a></td>
70 <td><a href="" ng-click="editar(produto)">[Editar]</a></td>
71 </tr>
```

Criamos a função e vinculamos a view através do parâmetro \$scope do nosso controller

```
54 | $scope.editar = function(produto) {  
55 |     $scope.produto = produto;  
56 | };  
57 |  
58 | });
```

Precisamos alterar o método salvar para verificar se o **id** do produto já existe, se **sim**, atualizar o produto, se **não**, adicionar novo produto

```
29 $scope.salvar = function(produto){
30     var produtoEncontrado = false;
31     for(var i = 0; i < produtos.length; i++){
32         if(produtos[i].id == produto.id){
33             produtos[i].descricao = produto.descricao;
34             produtos[i].preco = produto.preco;
35             produtoEncontrado = true;
36             break;
37         }
38     }
39     if(!produtoEncontrado){
40         produtos.push(produto);
41     }
42     $scope.produto = {};
43 }
```

Professor Anderson Henrique



**Two-way databinding** – é um ciclo de notificação, ele escuta a diretiva ngModel e exibe a informação atualizada

```
54 |         $scope.editar = function(produto) {  
55 |             $scope.produto = angular.copy(produto);  
56 |         };  
57 |  
58 |     });
```

Vamos adicionar um botão cancelar no nosso formulário e criar um link para cancelar, utilizando o **ng-click** para vincular o evento a função no \$scope

```
67 |         <button ng-click="salvar(produto)"> Salvar </button>  
68 |         <button ng-click="cancelar()"> cancelar </button>
```

```
58 $scope.cancelar = function() {  
59     $scope.produto = {};  
60 }
```

## ngdblClick – atribuindo um comportamento (function) ao evento

```
87 <tr ng-repeat="produto in produtos track by $index">  
88     <td ng-dblclick="editar(produto)">{{produto.id}}</td>  
89     <td ng-dblclick="editar(produto)">{{produto.descricao}}</td>  
90     <td ng-dblclick="editar(produto)">{{produto.preco}}</td>  
91     <td><a href="" ng-click="excluir(produto)">[Excluir]</a></td>  
92     <td><a href="" ng-click="editar(produto)">[Editar]</a></td>  
93 </tr>
```

**Dúvidas?**

**Professor Anderson Henrique**



## Para a próxima aula

01 – Melhorando nosso projeto AngularJS

02 – Trabalhando com Service (Comunicando com Back-End) [Controller e Model]

Professor Anderson Henrique

