

# AngularJS: Controllers

Desenvolvimento de Aplicativos Web  
João Eduardo Montandon

COLÉGIO TÉCNICO

# AngularJS: Controllers

# Controllers

- **O que é??** Módulo que contém as regras de negócios a serem implementadas por trás da visão do sistema
- **Propósito:** Expor variáveis, funções e procedimentos para uso na visão
- **Consequência:** Separação da lógica das regras de negócios e apresentação dos dados

# Controllers: Criação

```
var app = angular.module('myApp', []);
app.controller('myController', function() {

    // acoes e propriedades do meu controller
    this.myVariable = "variable";

    this.myFunction = function() {
        //...
    };

    this.anotherFunction = function(param) {
        //...
    };
});
```

# Controllers: Utilização

```
<!DOCTYPE html>
<html ng-app="myApp">
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.6/
angular.min.js"></script>
    <script src="app.js"></script>
  </head>
  <body>
    <div ng-controller="myController as myCtrl">
      <!-- Interação com o controller -->
    </div>
  </body>
</html>
```

# Controllers: Um exemplo

**Invoice:**  
Quantity:   
Costs:  EUR   
**Total:**  
  
USD2.70  
  
EUR2.00  
  
CNY16.46

<http://plnkr.co/edit/5duSVXJvka7IPh4NrxyR>

# Controllers: Mudanças

- Novo arquivo: invoice1.js
  - Controller InvoiceController
  - Definição de variáveis e funções para realizar determinado processamento

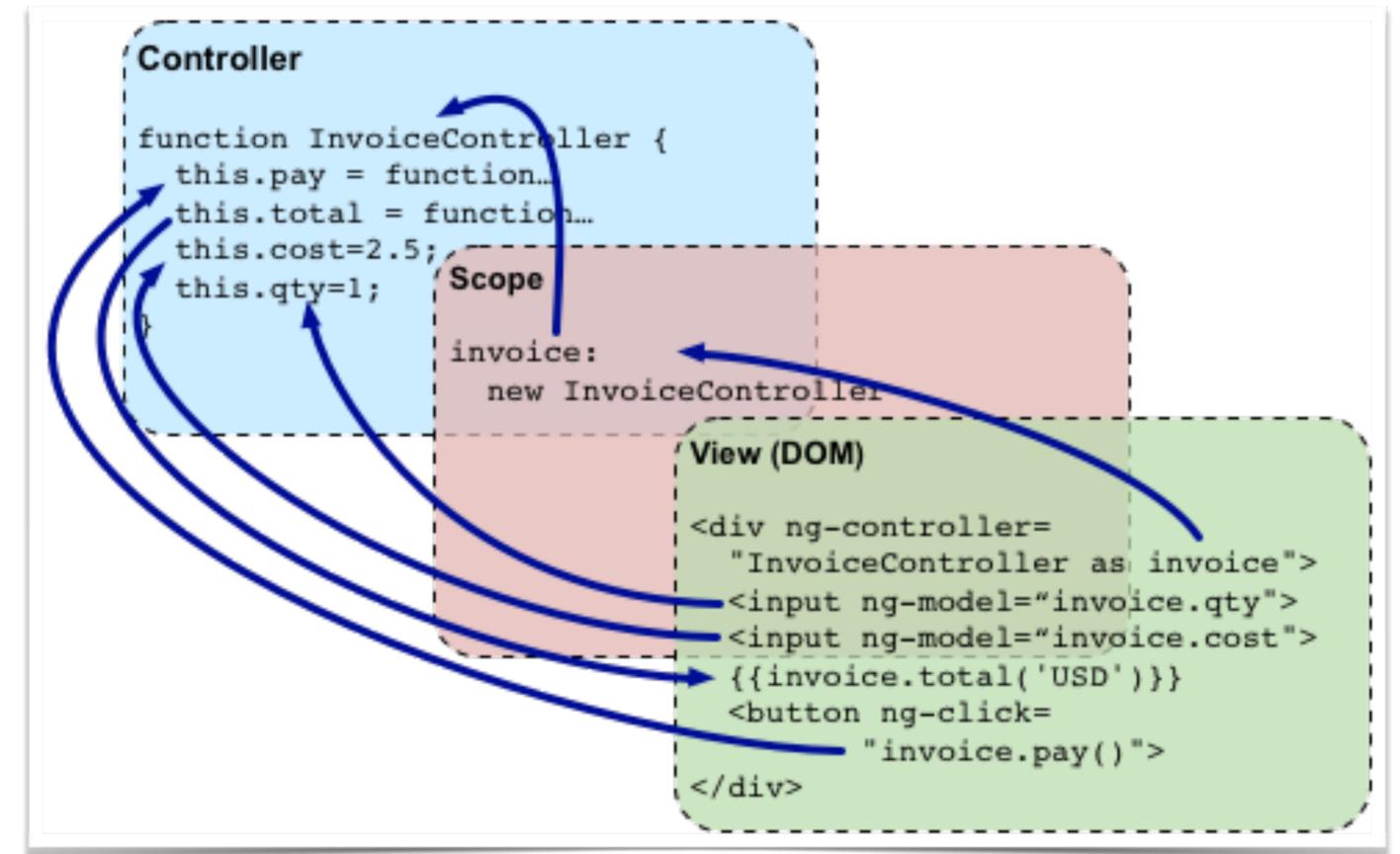
# Controllers: Mudanças

- index.html: Adição de novas diretivas
  - ng-app: Agora define qual módulo será carregado
  - ng-controller: Especifica o controller que será carregado
  - ng-model: Binding com os atributos definidos no controller
  - ng-repeat: diretiva para repetição
  - ng-click: Evento de click

# Controllers: Exemplo

- Comunicação entre Controller e View
- Sincronização entre controller e visão se dá pelo **scope**

**Invoice:**  
Quantity:   
Costs:  EUR   
**Total:**  
USD2.70  
EUR2.00  
CNY16.46



# Controllers: \$scope

- As vezes, precisamos acessar variáveis que são criadas na view e acessadas no controller ou vice-versa.
- Tais variáveis não pertencem ao controller ou a view, mas sim ao **\$scope**
- **O que é??** Serviço que liga dados entre visão e controller
- O **\$scope** permite o armazenamento tanto de variáveis quanto de funções

# Controllers: Scope

app.js

```
myApp.controller('DoubleController', ['$scope', function($scope) {  
    $scope.double = function(value) { return value * 2; };  
}]);
```

```
<div ng-controller="DoubleController">  
    Two times <input ng-model="num"> equals {{ double(num) }}  
</div>
```

index.html

De onde vem  
esse **\$scope**??

<http://plnkr.co/edit/vqNjmWvE7SaJI0v6Ek6P>



# Controllers: Herança

- Em determinadas situações, a mesma view pode ter vários controllers

```
<body ng-app="myApp">  
  <div ng-controller="firstController">  
    ... acesso ao firstController  
  </div>  
  <div ng-controller="secondController">  
    ... acesso ao secondController  
  </div>  
</body>
```

# Controllers: Herança

- Os controllers também podem ser organizados de forma aninhada
- Quando for o caso, lembrar sempre de dar um nome aos controllers por meio da notação **as**

```
<body ng-app="exemploHeranca">
  <div ng-controller="firstController as firstCtrl">
    {{firstCtrl.nome}}
    <div ng-controller="secondController as secCtrl">
      {{secCtrl.nome}}
    </div>
  </div>
</body>
```