

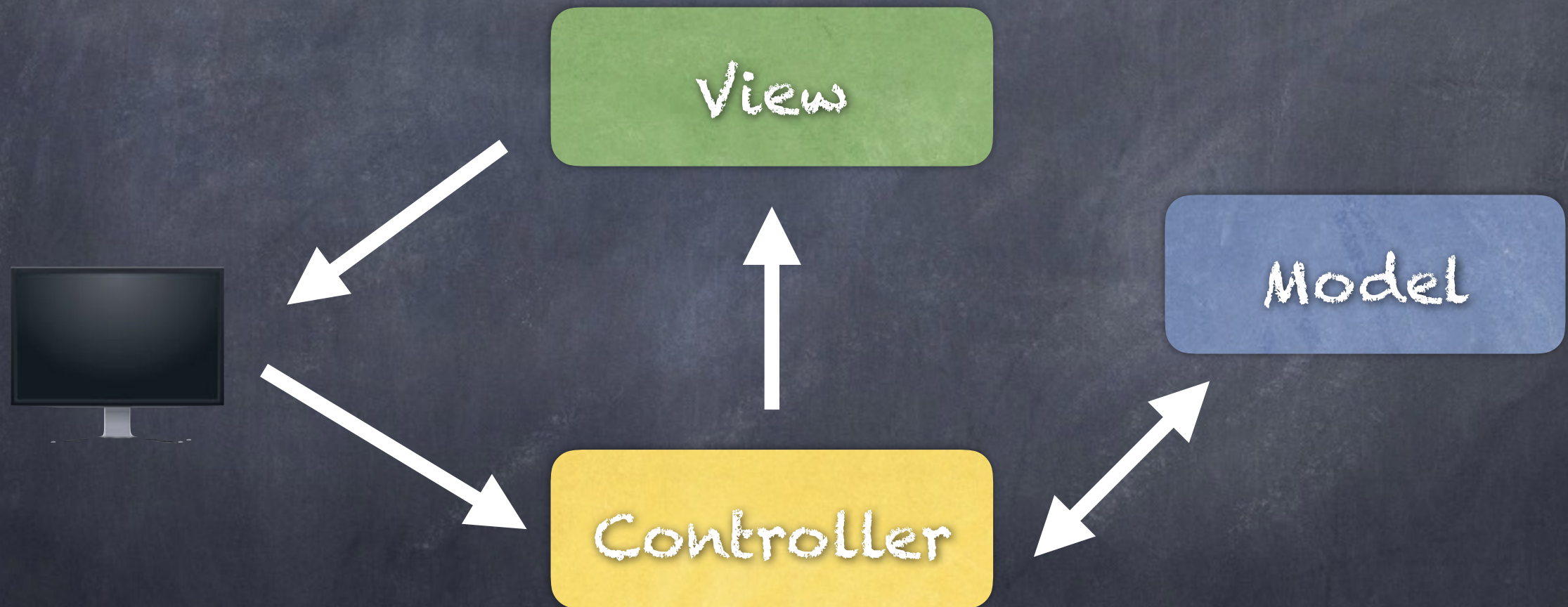


TOTVS

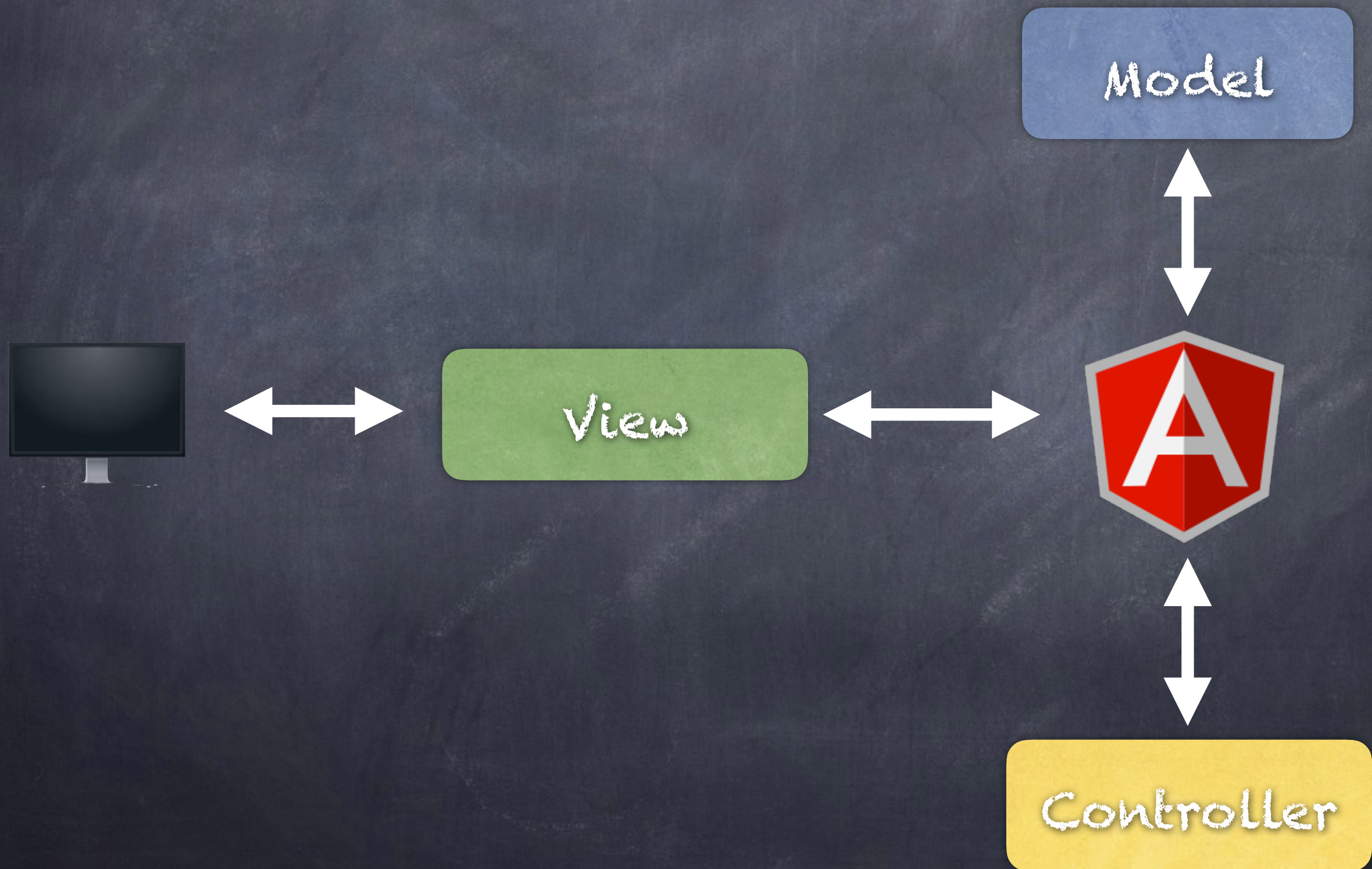
Angular é um framework
client-side MVC/MVVM feito
em JavaScript

MVC

Tradicional



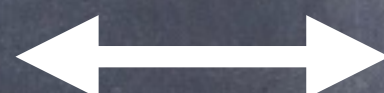
M V C



M V W



View



Model



Controller

Directive

Unit Test



Whatever

- Programação declarativa
 - Interface do usuário
 - Conexão entre componentes

- Programação imperativa
 - Lógica de negócios

```
<html ng-app="ProdutosApp">
  <head>
    <title>Exemplo</title>
  </head>
  <body ng-controller="ProdutosController">
    <h1>Lista Produtos</h1>
    <ul>
      <li ng-repeat="produto in produtos">
        {{produto.nome}} - {{produto.preco}}
      </li>
    </ul>
  </body>
  <script type="text/javascript" src="angular.js"></script>
</html>
```



```
<script type="text/javascript">
  var app = angular.module('ProdutosApp', []);

  app.controller('ProdutosController', ['$scope', function ($scope) {
    $scope.produtos = [
      {nome: 'Café', preco: 1.50},
      {nome: 'Açúcar', preco: 3.50},
      {nome: 'Adoçante', preco: 8.50}
    ];
  }]);
</script>
```


Principais Conceitos

- Módulos
- Two-way data-binding
- Diretivas
- Controladores
- Filtros
- Expressões
- Fábricas
- Injeção de Dependências

Two-way data-binding

- "Ligação em duas direções"
- É possível conectar elemento visuais com dados do escopo:
 - A alteração feita em um lado refletirá automaticamente no outro lado;
- Como fazer:
 - Na view, pode-se conectar um elemento de entrada com um valor do escopo: `<input type="text" ng-model="nome"/>`
 - Tudo o que for digitado nesse campo, será armazenado no valor nome do escopo, e se esse valor for alterado, o conteúdo do campo também será;

Módulos

- Uma aplicação Angular é dividida em módulos
- Cada módulo pode conter controladores e fábricas de componentes;
 - `var modulo = angular.module("nome do módulo", [dependências]);`
- Um módulo pode depender de outros;
 - `var dados = angular.module("Dados", []);`
 - `var financeiro = angular.module("Financeiro", ["Dados"]);`

Diretivas

- Diretivas são como Angular dá superpoderes ao browser.

Diretivas

- ◉ Diretiva ng-app
 - ◉ Usada para fazer o bootstrap de sua aplicação.
- ◉ Diretiva ng-controller
 - ◉ Define um controller para o trecho de código.
- ◉ Diretiva ng-bind
 - ◉ O mesmo que `{{var}}`
- ◉ Diretiva ng-model
 - ◉ "Amarra" um campo input a uma propriedade do controller.
- ◉ Diretiva ng-repeat
 - ◉ Itera sobre um array (for each).
- ◉ Diretiva ng-show
 - ◉ Deixa visível um elemento de acordo com uma expressão booleana.
- ◉ Diretiva ng-hide
 - ◉ Esconde um elemento de acordo com o valor de uma expressão booleana.
- ◉ Diretiva ng-click
 - ◉ Define o comportamento que um clique terá.

Controllers


- Classe que contém as regras de negócio da aplicação e gerencia a interação entre a view e o model
- Realizam o tratamento e o direcionamento da informação;
- São criados dentro dos módulos:
 - `modulo.controller('nome do controlador', function(dependências) { código });`

\$scope

- O \$scope se refere ao atual elemento/área do DOM (não, não é o mesmo que this)
- Encapsula um escopo inteligente e capaz de manter os dados e a lógica completamente situados dentro dos elementos.
- Ligação com todas as suas funções dentro dos controladores específicos.
- O conceito de \$scope pode ser assustador no começo, mas é a sua conexão com o DOM e com o servidor

Filters

- Filtro é um mecanismo que permite transformar os dados que são exibidos para o usuário
- {{ expressão | filtro :
parametro1 : ...paramentron }}



Filters

- currency

- Transforma um número em formato monetário.

- date

- Formata uma data (usando máscara ou não).

- orderBy

- Ordena um array de acordo com o predicado.

- filter

- Filtra dados de um array. Útil quando temos uma lista muito grande.

- json

- Exibe um json da expressão..

- lowercase

- Passa todas as letras para caixa baixa (minúsculas).

- Uppercase

- Passa todas as letras para caixa alta (maiúsculas).

- limitTo

- Limita os elementos de um array.

Exercícios

- Faça um script que leia o nome de uma pessoa e escreva o mesmo de traz para frente; Após, exiba o mesmo em caixa-alta;
- Crie um programa que de lista de compras que armazene o item e a quantidade a ser comprada. Crie também uma forma de marcar o item como comprado e uma forma de excluir o mesmo da lista.

Filters

```
app.filter('myFilter', function() {  
  return function(data) {  
    return data;  
  };  
});
```

- {{ var | myFilter }}

Exercícios

Faça um script que leia o nome de um produto e execute uma Busca Instantânea dentro de uma lista de produtos pré-definida

Services

- É onde vamos concentrar as regras de negocio e abstrações
- Você vai precisar de usar a injeção de dependência para dizer ao Angular que ele precisa ligar seu novo serviço
- Lembre-se: controllers são apenas a ponte de ligação entre o model e a view. Não devem ter lógica de negócio!

Services

```
app.service('MyService', [function () {  
    this.add = function (a, b){  
        return a + b;  
    }  
}])
```


As Organizações XPTO resolveram dar um aumento de salário aos seus colaboradores e lhe contrataram para desenvolver o programa que calculará os reajustes.

Faça um programa que recebe o salário de um colaborador e calcule reajuste segundo o seguinte critério, baseado no salário atual:

- Salários até R\$ 280,00 (incluindo): aumento de 20%
- Salários entre R\$ 280,00 e R\$ 700,00: aumento de 15%
- Salários entre R\$ 700,00 e R\$ 1500,00: aumento de 10%
- Salários de R\$ 1500,00 em diante: aumento de 5%

Após o aumento ser calculado, deverá ser informado: o salário antes do reajuste; o percentual de aumento aplicado; o valor do aumento; o novo salário, após o aumento.

\$http

- Para o consumo e envio rápidos de dados, pode-se utilizar o módulo \$http;
 - `$http.get(url).then(funcao_sucesso, funcao_erro);`
 - `$http.post(url).then(funcao_sucesso, funcao_erro);`
- <http://cep.correiocontrol.com.br/58402000.json>

\$http

```
$http('url').then(function successCallback(response) {  
    ...  
},  
function errorCallback(response) {  
    ...  
});
```


Routes

- ngRoute

- [https://docs.angularjs.org/api/
ngRoute](https://docs.angularjs.org/api/ngRoute)

Routes

```
.config(['$routeProvider', function($routeProvider) {  
    $routeProvider.when('/url/', {  
        templateUrl: 'arquivo.html',  
        controller: 'ArquivoController'  
    });  
});
```


Routes

```
controller('ArquivoController', ['$scope',  
'$routeParams', function($scope,  
$routeParams,) {
```

```
    $routeParams.name
```

```
}
```


Exercício

- Criar uma pagina que liste todos os repositórios de um usuário no GitHub. Ao clicar em um repositório, listar os detalhes do mesmo.

- URLs

- <https://api.github.com/users/centeno>
- <https://api.github.com/users/centeno/repos>
- [https://api.github.com/repos/centeno/
api_skatespotme_php](https://api.github.com/repos/centeno/api_skatespotme_php)