

M V C

Danielle

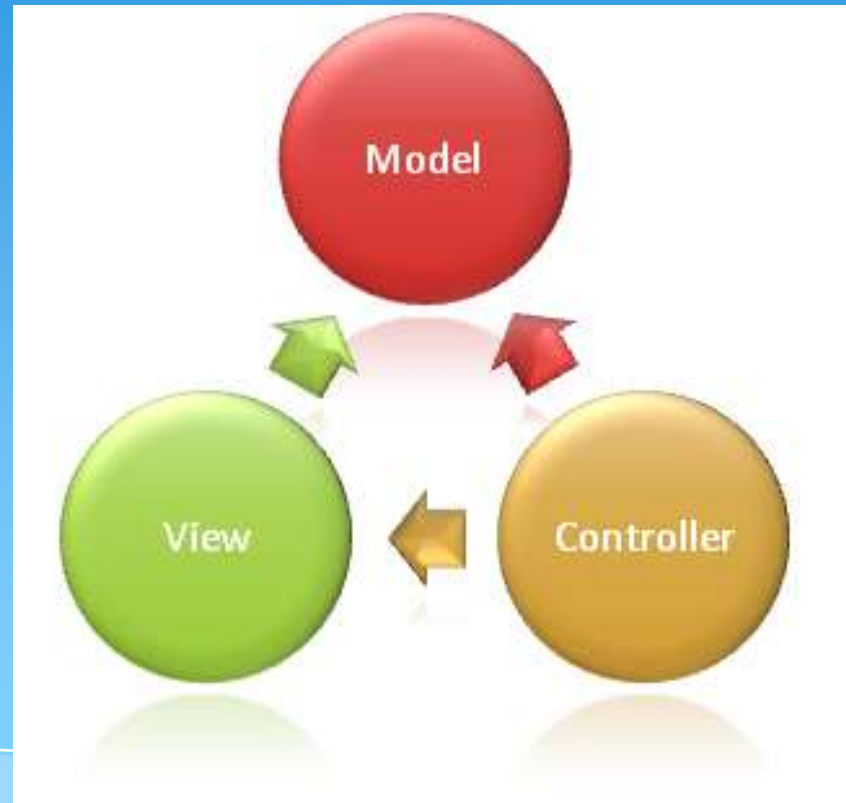
Elizângela

Leonardo Pereira

Marcelo Helt

Marcelo Junio

Wellington



MVC

- * **Model**
- * **View**
- * **Controller**

MVC

- * É um padrão de arquitetura de software usado para dividir a aplicação (software) em camadas, sendo a *Model* a responsável pelos dados (camada de banco de dados), a *View* que é a interface de usuário e a *Controller* que intermedia as duas.
- * Pode ser visto como um divisor de tarefas em um aplicativo.

MVC

- * O MVC surgiu em 1979
- * Trygve M. H. Reenskaug criou o padrão como uma solução óbvia para o problema geral de oferecer maior controle sobre suas informações.
- * Foi usado pela primeira vez no Smalltalk



Necessidade

- * Facilitar o desenvolvimento
- * Garantir uma estrutura
- * Fácil Manutenção

MVC

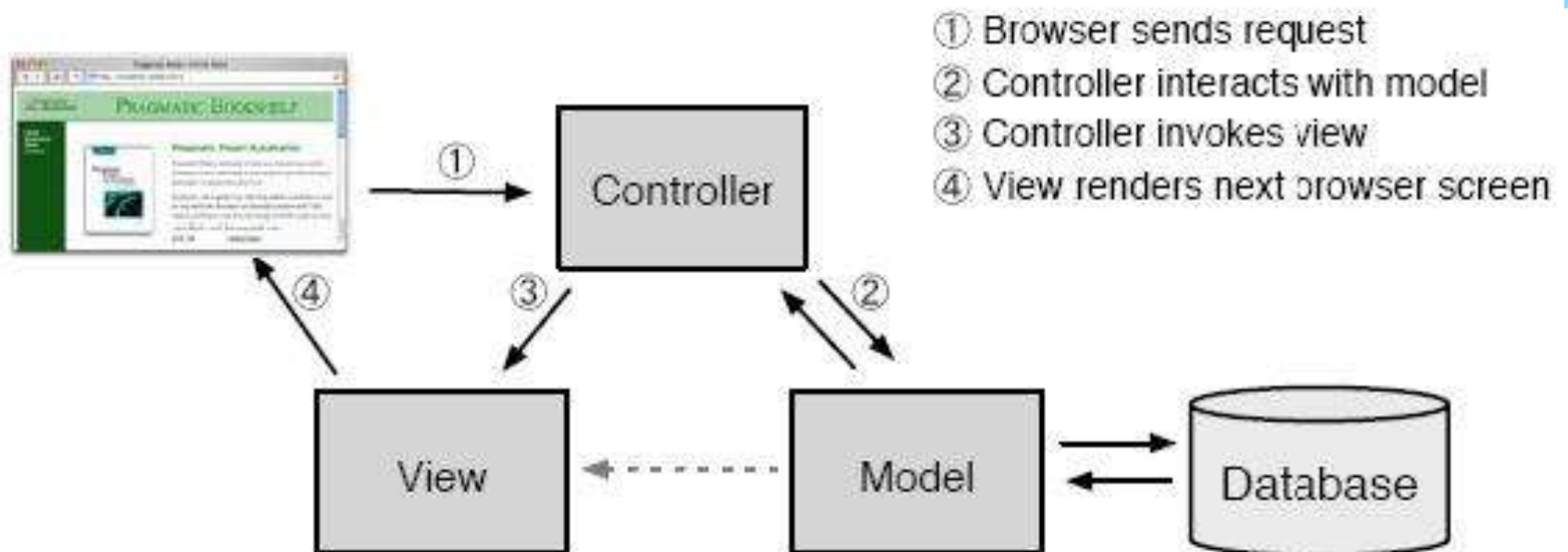
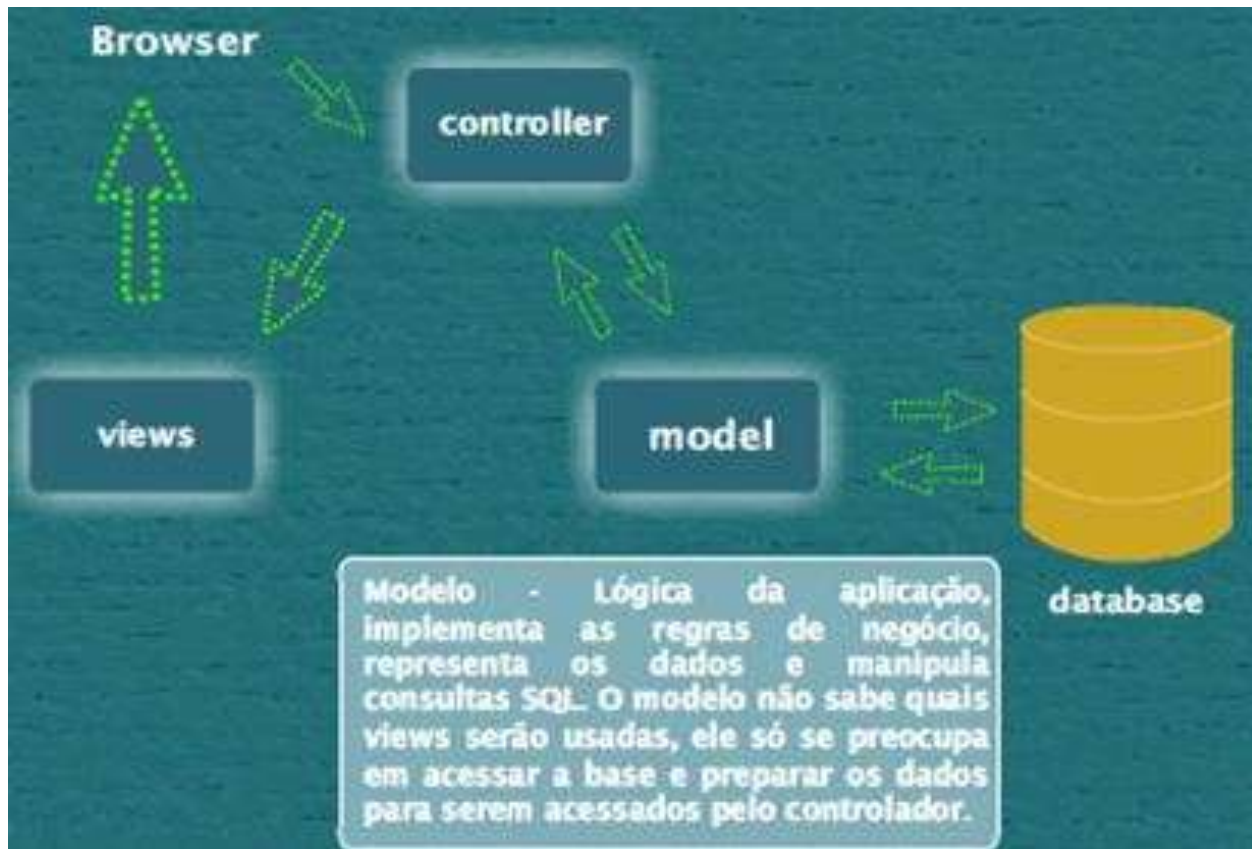


Figure 2.1: The Model-View-Controller Architecture

Model

- * Camada que manipula e modela os dados gerenciando o comportamento da aplicação.
- * Responsável pela validação dos dados.
- * Estabelece regras de negócio (acesso e modificação aos dados)
- * Uma consulta estará disponível para acessos vindos de requisições distintas.

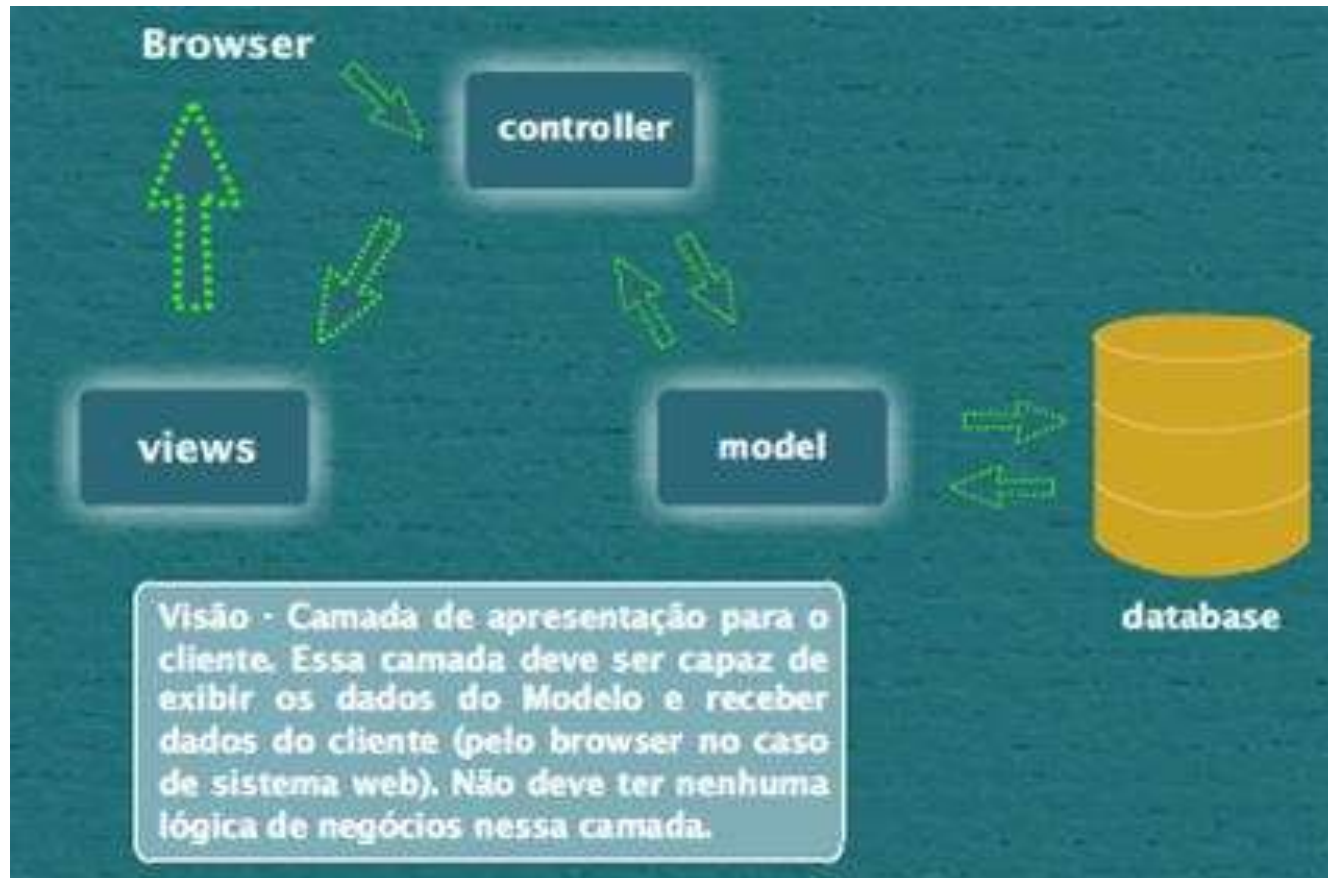
Modelo



View

- * Responsável pela camada de apresentação ao usuário.
- * Interação gráfica.
- * Disponibiliza também a entrada de dados para a model.
- * A view não tem ciência de qual será sua fonte de dados. (*Apenas exibe a informação no browser*)

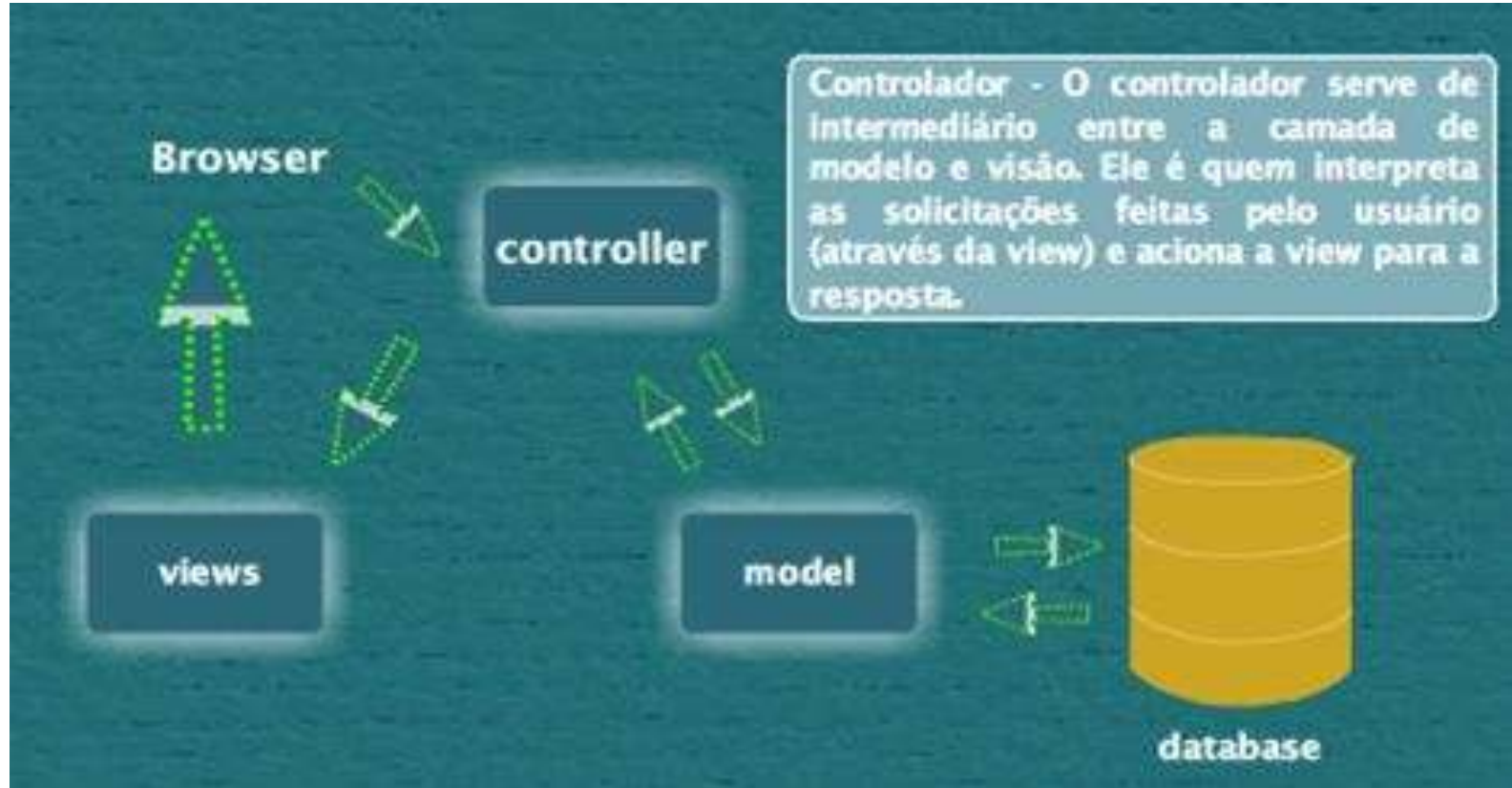
Visão



Controller

- * Camada intermediária entre a view e a model.
- * Controla todas as ações.
- * Organiza o MVC por meio de métodos, decisões e estratégias para atender às requisições dos usuários.
- * A View irá apenas exibir o que a controller definir.

Controlador



Alguns Frameworks

- * CodeIgniter

- * CakePHP

Framework -> é uma infra-estrutura ou esqueleto de uma família de aplicações projetado para ser reutilizado.

- Muitas funcionalidades já implementadas.
- Soluções bem testadas (muitos colaboradores)
- Diminuição linhas de códigos (menos erros)
- Urls inteligentes
(www.urlbase/controller/metodo/parametros)

CakePHP - Model

```
<?php

class Post extends AppModel {
    var $name = 'Post';
    var $validate = array(
        'title' => array(
            'rule' => 'notEmpty'
        ),
        'body' => array(
            'rule' => 'notEmpty'
        )
    );
}
```

CakePHP - Controller

```
<?php
class PostsController extends AppController {
    var $name = 'Posts';

    function index() {
        $this->set('posts', $this->Post->find('all'));
    }
}
```

CakePHP - View

```
<!-- Arquivo: /app/views/posts/index.ctp -->
<h1>Posts do Blog MVC- 8º Período</h1>
<table>
    <tr>
        <th>Id</th>
        <th>Título</th>
    </tr>
    <?php foreach ($posts as $post): ?>
    <tr>
        <td><?php echo $post['Post']['id']; ?></td>

        <td><?php echo $post['Post']['title']; ?></td>
    </tr>
    <?php endforeach; ?>
</table>
```


Vantagens - MVC

- * Nitidez no código.
- * Melhor visibilidade da camada de negócios .
- * Tratamento de segurança. (nos frameworks)
- * Integração de equipes e/ou divisão de tarefas.
- * Implementação de segurança.
- * Facilidade na alteração da interface da aplicação.

Desvantagens

- * Exige disciplina do programador.
- * Pode criar uma complexidade desnecessária em pequenas aplicações.

Considerações Finais

- * A arquitetura MVC oferece aos programadores e analistas uma separação nítida das responsabilidades na aplicação, facilitando a manutenção e a reutilização de códigos.

Referências

- * http://www.oficinadanet.com.br/artigo/1687/mvc_-_o_padrao_de_arquitetura_de_software
- * <http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>
- * <http://book.cakephp.org/1.3/pt/view/876/Manual>
- * <http://ead.prodemge.gov.br>