

# Univiçosa

## ADS 103 – Desenvolvimento web back-end

Prof. Hermes Nunes Pereira Júnior

### CodeIgniter, exibição dos registros cadastros no bd

Na aula de hoje trataremos da exibição dos registros que foram inseridos no mysql através da view “Conteudo.php”. Criaremos a function “**exibir()**” e a view para mostrar os registros armazenados no bd.

#### 1. Criando o model MuseuModel.php

O CodeIgniter usa o modelo MVC para fazer todo o controle dos dados que passam pelo sistema. Seguindo este modelo, para acessar o banco de dados, suas tabelas e campos, o framework precisa ter um arquivo específico para isso, armazenado no diretório “**app/Models**”. Neste arquivo contém todas as informações necessárias para que o sistema acesse o banco de dados.

Vamos criar o arquivo, mas isso é feito acessando o servidor via ssh.

```
ssh aXXXXX@192.168.51.5
```

Depois de logar no servidor acesse o diretório da disciplina e do sistema.

Quando estiver na raiz do sistema, digite:

```
php spark make:model MuseuModel
```

Explicando o comando.

O php executará o programa spark. Este programa é parte do framework e é responsável por várias ações, dentre elas criar automaticamente Models e Controllers.

O parâmetro make, seguido do model (make:model) criará um Model padrão chamado “MuseuModel”. O nome do model precisa seguir este padrão: Primeira letra maiúscula, depois a palavra Model também com a primeira letra em maiúscula, sem espaços. No nosso caso usamos o nome “MuseuModel”.

Após digitar o comando será criado automaticamente o arquivo “MuseuModel.php” dentro do diretório “app/Models”.

Ao abrir este arquivo teremos alguns dados a serem digitados.

Observe o quadro abaixo.

```
class MuseuModel extends Model
{
    protected $table      = "";
    protected $primaryKey = "";
    protected $useAutoIncrement = true;
    protected $returnType  = 'object';
    protected $useSoftDeletes = false;
    protected $protectFields = true;
    protected $allowedFields = ['Id','Titulo','Conteudo'];
}
```

Neste arquivo vamos mexer inicialmente apenas nas 3 linhas que estão marcadas com a cor vermelha.

A primeira linha é o nome da tabela do seu banco de dados. Coloque o nome entre aspas, exatamente como foi criada no banco de dados.

A segunda linha é o nome do campo da tabela que é a chave primária. Observe que o nome do campo deve ser exatamente igual ao da tabela.

A terceira linha é a forma como os dados serão manipulados: object ou array. Como estamos trabalhando com orientação a objetos, o padrão usado é o “object”.

A quarta linha é onde serão colocados os campos da tabela, entre aspas simples e separados por vírgula. Ex.: ['Campo01', 'Campo02', 'Campo03'].

Com esta configuração feita o sistema saberá qual tabela e quais campos poderão ser usados na operação.

## 02. Criando a função para buscar os dados no bd.

Toda a regra de negócios do sistema é feito dentro dos arquivos do diretório “app/Controllers”. Por padrão neste diretório há o controller Home.php, que é o controller inicial do sistema. Vamos usá-lo para criar a função que fará a seleção dos dados na tabela.

Ao abrir o controler “Home.php” temos a função index(), esta função foi usada na última aula para chamar uma view criada manualmente.

Vamos criar uma nova função chamada “exibir”. Sua função primordial será fazer a seleção no banco de dados, usando o model “**MuseuModel**” para fazer o select e depois chamar a view que exibirá o que foi selecionado.

Observe o quadro abaixo

```
public function exibir() : string
{
    // vai chamar uma view que exhibe todas os registros

    // criar uma variável para ter acesso à MuseuModel
}
```

```

    $museuModel = new \App\Models\MuseuModel();

    // vamos armazenar o select na variavel $data
    // o método find() faz o select de todos os campos listados no model
    // MuseuModel
    $data['dados'] = $museuModel->find();

    // o controller vai chamar a view "Conteudo.php"
    echo view('Conteudo', $data);
}

```

## 2. Criando a view

Vamos criar a view Conteudo.php, dentro do diretório /app/Views. Este arquivo será chamado pelo function criada anteriormente.

Basicamente é uma página html com uma seção php para torná-la dinâmica.

Dentro de <body> está toda a página, observe que dentro das tags html há algumas palavras reservadas: class e id que chamam as configurações para aquele componente. Veremos a construção de um css mais adiante.

Observe que dentro da <table> há estas linhas:

```

<tr>
<th>Código ou Id</th>
<th>Titulo</th>
<th>Conteudo</th>
</tr>

```

Elas são responsáveis pela primeira linha da tabela. A <tr> significa Table Row, ou seja, cria uma linha na tabela. O <th> significa Table head, é a célula que é o cabeçalho da tabela. Toda tag de abertura deve ser fechada no final. Ex.: <th>Código da Pessoa</th>.

A tag <td></td> significa Table Data, são as células da tabela que terão os dados. A diferença da <th> e da <td> é a formação que o html atribui a elas.

Logo depois há esta sequência:

```

<?php foreach ($dados as $dado) : ?>
<tr>
<td><?php echo $dado->Id ?></td>
<td><?php echo $dado->Titulo ?></td>
<td><?php echo $dado->Conteudo ?><td>
</tr>
<?php endforeach ?>

```

Há o uso do comando **foreach**. No php este comando fará um loop pelo conjunto de dados (pode ser objeto ou array) que vier através da variável dentro dos parênteses. Enquanto o loop não retornar o valor vazio o foreach será executado.

Observe que que a variável **\$dado** é um objeto, ela chama o campo através da expressão “->”. Lá no arquivo MuseuModel.php nós definimos que o \$returnType é “object” ou “array”.

O código final da página está no quadro abaixo.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Página do museu</title>
</head>
<body>

<table>

<tr>
<th>Código ou Id</th>
<th>Titulo</th>
<th>Conteudo</th>
</tr>


<?php foreach ($dados as $dado) : ?>
<tr>
<td><?php echo $dado->Id ?></td>
<td><?php echo $dado->Titulo ?></td>
<td><?php echo $dado->Conteudo ?></td>
</tr>
<?php endforeach ?>

</table>

</body>
</html>
```

### 03. Criando a rota para acessar a página

Por questão de segurança o framework não permite que páginas ou views sejam acessadas diretamente. Para isso um arquivo é o responsável por organizar todas as rotas do sistema: “app/Config/Routes.php”.

Vamos inserir a seguinte linha no final do arquivo

```
$routes->get('exibir','Home::exibir');
```

ao escrever na url do sistema a palavra “exibir”, será chamado o controller “Home” e a função “exibir”.

A url ficará como no quadro abaixo.

```
http://192.168.51.5/aXXXXX/adsXXX/museu/public/index.php/exibir
```

Aí será exibida a página com todos os dados que foram gravados no banco de dados. Observe a imagem abaixo.



The screenshot shows a web browser window with the address bar displaying 'localhost/hermes/ads103/museu/public/index.php/exibir'. The page content is a table with two columns: 'Código ou Id' and 'Conteúdo'. The first row has the value '1' in the first column and a detailed definition of an operating system in the second column.

Código ou Id	Conteúdo
1	Definição de sistema operacional O sistema operacional é um software, ou conjunto de softwares, cuja função é administrar e gerenciar os recursos de um sistema, desde componentes de hardware e sistemas de arquivos a programas de terceiros, estabelecendo a interface entre o computador e o usuário. Entenda como um “computador” qualquer máquina de processamento automático de dados, como um desktop, notebook ou celular e um console de videogame, por exemplo. O sistema operacional introduz uma camada de abstração entre o hardware e o usuário, que transforma comandos no mouse, teclado e solicitações do sistema, como gerenciamento de recursos (CPU, memória RAM), em linguagem de máquina, enviando instruções ao processador. Este último os traduz para código binário, executa os comandos e envia as respostas como informações que aparecem na sua tela. Um sistema operacional contém componentes divididos entre os para o usuário (como bibliotecas, programas e interface) e as instruções que compõem o seu núcleo (kernel).

Não houve nenhum tratamento visual na página, foi criada uma tabela e os dados colocados em linhas e colunas. Para que a página tome forma é necessário criar as estruturas para cada um dos campos e fazer as devidas formatações.

### 4) Considerações Finais sobre a aula

Vimos nesta como a programação web tem várias tecnologias aplicadas. Além de fazer o select através do “find()”, vimos uma estrutura de controle, o foreach, usamos os recursos nativos do framework junto com uma estrutura do php.

Bons estudos.