02

Introdução ao Code Igniter

Bem vindo ao nosso curso sobre Code Igniter, nele veremos como criar um sistema completo que funcionará como um sistema de venda e compra de produtos, onde será possível:

- Cadastrar usuários
- Cadastra produtos
- Usuários se cadastram
- Notificações
- Envio de e-mails

Neste caminho veremos vários conceitos teóricos importantes do Code Igniter na versão 2.x e do MVC, além de ver muita prática.

Primeiro fazemos o download do Code Igniter: https://s3.amazonaws.com/caelum-online-public/codeigniter-1/CodeIgniter-2.2-stable.zip (https://s3.amazonaws.com/caelum-online-public/codeigniter-1/CodeIgniter-2.2-stable.zip)

Depois vamos clicar em download. Após baixar o zip vamos descompactá-lo. Agora vamos ao diretório htdocs dentro da sua pasta do XAMPP, e criaremos uma pasta chamada mercado, logo após copiamos os arquivos deszipados para este diretório.

Vamos acessar http://localhost/mercado (http://localhost/mercado) e vemos a tela de bem vindo do Code Igniter:

Ao acessarmos essa url, ele executa uma lógica e mostra uma visualização. O servidor recebe uma requisição do cliente e executa uma lógica, como por exemplo, buscar dados no banco e mostra eles de volta para os clientes, dois passos importantes: executar e mostrar.

Vamos dentro da nossa aplicação, nos controladores e lá encontramos o bem vindo `welcome.php` .

Dentro deste welcome, por padrão, ele executa a função index, esta função carrega a visualização, a view chamada `welcome-message` , que se encontra no diretório `views/welcome-message.php` .

Mas ainda não é bem isso que gostariamos de fazer, gostariamos de criar uma lógica nossa, portanto vamos no diretório controllers e criamos um arquivo que sera nossa lógica, chamado `Produtos.php` .

Agora vamos escrever uma classe `Produtos` que extende o controller do Code Igniter:

```
<?php
class Produtos extends CI_Controller{
}
```

Com a função index:

```
<?php
class Produtos extends CI_Controller{
    public function index(){
}
}
```

Agora vamos buscar alguns produtos meio fakes, de mentira, usando um array:

```
public function index(){
    $produtos = array();
    $bola = array("nome" => "Bola de futebol", "descricao" => "Bola de futebol assinada pelo Zi
    $hd = array("nome" => "HD Externo usado", "marca" => "Mega HD 300 teras" ,"preco" => 400);
    array_push($produtos, $bola, $hd);
}
```

No próximo capítulo usaremos um banco de verdade, vamos agora mostrar isso na nossa camada de visualização. Faremos igual ao welcome, passando o nome da view:

```
$this->load->view("produtos/index.php");
```

Portanto nosso código inicializa uma array, coloca a bola e o hd nela, e por fim redireciona para a página `index.php` :

```
    public function index(){
        $produtos = array();
        $bola = array("nome" => "Bola de futebol", "descricao" => "Bola de futebol assinada pel
        $hd = array("nome" => "HD Externo usado", "marca" => "Mega HD 300 teras" ,"preco" => 40
        array_push($produtos, $bola, $hd);
        $dados = array("produtos" => $produtos);
        $this->load->view("produtos/index.php",$dados);
    }
```

Vamos criar agora um diretório chamado produtos, dentro da view, e dentro de produtos, criamos um `index.php` , e usamos o vardump para jogar os produtos.

Dentro deste index.php teremos:

```
<html lang="en">
    <body>
        <?= var_dump($produtos);?>
    </body>
</html>
```

Mas onde está nossa váriavel produtos? Quem nos passou ela? Ninguém! Vamos fazer isto agora, vamos pegar ele da nossa lógica e enviar para a view. Para isso iremos criar um array com todos os dados que queremos enviar para a view:

```
$dados = array("produtos" => $produtos);
```

Vamos agora executar o produtos controller, função index, através da URL:

http://localhost/mercado/index.php/produtos/index (http://localhost/mercado/index.php/produtos/index)

Deste modo ele mostra os produtos, sempre isolando a minha lógica dentro de um controller e a minha view, dentro de um arquivo `.php` no diretório views.

Vale lembra que o index é o padrão, então podemos acessar: http://localhost/mercado/index.php/produtos (http://localhost/mercado/index.php/produtos)

Teremos exatamente o mesmo resultado.

Agora vamos alterar a tela padrão, vamos colocar a nossa lista, para isto, vamos no diretório config, abriremos o arquivo `routes.php` , onde definimos as rotas, as rotas são URIs que chegam e redirecionam para nossos controllers.

Para mudar a rota padrão, iremos alterar a linha:

```
$route['default_controller'] = "welcome";
```

para

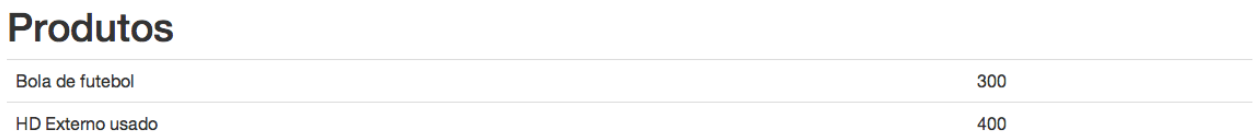
```
$route['default_controller'] = "produtos";
```

Agora acessamos a url abaixo e verificamos que a lista é a tela padrão. http://localhost/mercado/index.php (http://localhost/mercado/index.php)

Vamos mostrar de forma mais bonita nossos produtos, criando uma tabela, no `index.php` :

```
<h1> Produtos</h1>
<table>
    <?php foreach($produtos as $produto) : ?>
    <tr>
        <td><?=$produto["nome"] ?></td>
        <td><?=$produto["preco"] ?></td>
    </tr>
<?php endforeach ?>
</table>
```

Veja novamente no seu navegador, como ficou a tela.



Produtos

Bola de futebol	300
HD Externo usado	400

Agora vamos dar uma cara mais bonita ainda para nossa lista, vamos usar o getbootstrap, um CSS que dará uma cara bem mais bonita, vamos baixa-lo em http://www.getbootstrap.com (http://www.getbootstrap.com)

Após baixá-lo, vamos descompactá-lo e copiamos os 3 diretórios para dentro do diretório mercado. Pois na raiz do mercado, é onde ficam os arquivos que o navegador acessa.

Vamos agora importar o css no nosso index e envolver nosso conteúdo do body em uma div com a class container e colocar a class table na nossa tabela:

```
<html lang="en">
    <head>
        <link rel="stylesheet" href="css/bootstrap.css">
    </head>
    <body>
    <div class="container">

        <h1> Produtos</h1>
        <table class="table">
            <?php foreach($produtos as $produto) : ?>
            <tr>
                <td><?=$produto["nome"] ?></td>
                <td><?=$produto["preco"] ?></td>
            </tr>
        <?php endforeach ?>
        </table>
        <div>
    </body>
</html>
```

Veja no seu navegador como a aparência melhorou, no próximo capitúlo continuaremos a evoluir nossa aplicação.