

02

Conectando no Banco de Dados

Já temos em nossa aplicação uma tela que mostra uma lista de produtos, que após acessar um controller, ele redireciona para visualização. Vamos agora incrementar a aplicação trazendo os produtos do banco de dados.

O código que acessa o banco deve ficar isolado em outra classe, outro modelo, específico para isso, mas antes vamos criar as tabelas necessárias(Usuário e Produto).

Acessamos <http://localhost/xampp/index.php> ,vamos clicar em phpMyAdmin e em seguida clicamos em databases, definimos um nome para o banco, neste caso mercado, e criamos ele.

Agora acessamos o database mercado através da barra lateral esquerda, depois vamos em SQL para criarmos nossas tabelas, lá vamos executar os comandos SQLs:

```
create table usuarios (id integer auto_increment primary key, nome varchar(255), email varchar(255));
create table produtos(id integer auto_increment primary key, nome varchar(255), descricao text,
```

Para executar, basta clicarmos em Go.

Agora vamos inserir alguns produtos e vendedores. Lembrando que como a senha é criptografada sempre, usamos o md5 da string 123456 . Primeiro o usuário:

```
insert into usuarios values(1,'Guilherme', 'guilherme.silveira@alura.com.br','e10adc3949ba59abbe56e0
```

E agora dois produtos:

```
insert into produtos values(1, 'Bola de Futebol', 'Bola de futebol assinada pelo Zico', 300, 1);
insert into produtos values(2, 'HD Externo', 'Marca HD-Mega', 400, 1);
```

Criaremos o modelo, aquele que representa o meu objeto de negócio, dentro da pasta models, crie um arquivo chamado produtos_model.php que extende do Code Igniter Model, com uma função que busque todos os produtos do banco dados:

```
<?php
class Produtos_model extends CI_Model {

    public function buscaTodos() {
        return $this->db->get("produtos")->result_array();
    }
}
```

Este código pega no banco de dados tudo que está na tabela definida como parâmetro para a função `get`, neste caso `produtos` e retorna um array com tudo que tem lá.

Agora vamos editar nosso controller `produtos.php` para pegar produtos do banco e não mais do array:

```
public function index()
{
    $this->load->model("produtos_model");
    $produtos = $this->produtos_model->buscaTodos();

    $dados = array("produtos" => $produtos);
    $this->load->view("produtos/index.php", $dados);
}
```

Esta linha informa que iremos carregar o `produtos_model`:

```
$this->load->model("produtos_model");
```

Agora com o modelo carregado, temos acesso a função que definimos para buscar todos os produtos do banco, justamente, o que esta linha faz:

```
$produtos = $this->produtos_model->buscaTodos();
```

Mas ao acessarmos nossa página inicial, teremos um erro avisando que o banco de dados não foi carregado.

A PHP Error was encountered

Severity: Notice

Message: Undefined property: Produtos::\$db

Filename: core/Model.php

Line Number: 51

Para arrumar isso devemos alterar o `database.php` dentro da pasta config, mudando as seguintes linhas:

```
$db['default']['database'] = '';
```

para

```
$db['default']['database'] = 'mercado';
```

e

```
$db['default']['username'] = '';
```

para

```
$db['default']['username'] = 'root';
```

Vale lembrar que o database.php é o responsável por todas as configurações de conexão com o banco de dados, como por exemplo senha e usuário.

Precisamos também avisar o `index` que iremos usar o banco, para isso alteramos a função `index` do controller `produtos` :

```
public function index(){
    $this->load->database();
    //resto do código
}
```

Podemos inserir mais um produto para testar

```
insert into produtos values(3, 'Garrafa de 2 Litros', 'Um refrigerante', 10.50, 10);
```

Verifique que agora a garrafa aparece na listagem.