

02

## Adicionando usuários

Nosso próximo passo é a criação de um sistema de cadastro de usuário para que possamos nos logar. Vamos agora criar um formulário que permite adicionar usuários.

Dentro do nosso `produtos` na função `index` vamos usar mais um helper, que ajuda com a criação de formulários, o helper `form`.

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

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

    $dados = array("produtos" => $produtos);
    $this->load->helper(array("url"));
    $this->load->helper(array("currency"));
    $this->load->helper(array("form"));
    $this->load->template("produtos/index.php", $dados);
}
```

Com este helper, teremos várias funções disponíveis para ajudar a criarmos o formulário. Dentro do `index.php`, logo abaixo da tabela, iremos usar o helper para abrirmos um formulário, deste modo:

```
<?php
echo form_open("usuarios/novo");
echo form_close();
?>
```

Agora vamos ver como criar os campos do formulário.

Usaremos funções do helper, repare que podemos passar vários parâmetros para a função, como por exemplo a classe para aplicarmos estilo, o tamanho máximo de caracteres aceitos e um id, usaremos esse mesmo id também como parametro para o label, neste modo estamos ligando um ao outro.

```
echo form_label("Nome", "nome");
echo form_input(array(
    "name" => "nome",
    "id" => "nome",
    "class" => "form-control",
    "maxlength" => "255"
));
```

Vamos ver como ficou?

## Produtos

Bola de Futebol	Bola de futebol...	R\$ 300,00
HD Externo	Marca HD-Mega	R\$ 400,00
Garrafa de 2 Litros	Um refrigerante	R\$ 10,50

## Cadastro

Email

Agora iremos adicionar o campo de email.

```
<h1>Cadastro</h1>
<?php
echo form_open("usuarios/novo");

echo form_label("Nome", "nome");
echo form_input(array(
    "name" => "nome",
    "id" => "nome",
    "class" => "form-control",
    "maxlength" => "255"
));

echo form_label("Email", "email");
echo form_input(array(
    "name" => "email",
    "id" => "email",
    "class" => "form-control",
    "maxlength" => "255"
));

echo form_close();
?>
```

Vamos adicionar o campo de senha, ele é diferente, repare como ele é definido pela função `form_password`, que utiliza o modo que estamos acostumados em outros sites.

```
echo form_label("Senha", "senha");
echo form_password(array(
    "name" => "senha",
    "id" => "senha",
    "class" => "form-control",
    "maxlength" => "255"
));
```

E por fim um botão de submit, repare que aqui utilizamos o método `form_button`:

```
echo form_button(array(
    "class" => "btn btn-primary",
    "content" => "Cadastrar",
    "type" => "submit"
));
```

Se enviarmos esse formulário, iremos receber um erro 404. Faz todo sentido, pois ainda não temos o controller que cuidará disto, vamos criar ele agora mesmo, e se chamará `usuarios.php`. Lembre-se da linha de cabeçalho que impede o acesso direto pelo usuário:

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Usuarios extends CI_Controller {

    public function novo() {

    }
}
```

Agora queremos pegar os dados que foram enviados(`input`) via post e montar um usuário, também chamamos a função `md5()` para criptografar nossa senha:

```
public function novo() {
    $usuario = array(
        "nome" => $this->input->post("nome"),
        "email" => $this->input->post("email"),
        "senha" => md5($this->input->post("senha"))
    );
}
```

Agora vamos carregar o `usuarios_model` para podermos usar o método `salva` e em seguida redirecionaremos para a view `novo` que irá ser apenas uma confirmação:

```
$this->load->model("usuarios_model");
$this->usuarios_model->salva($usuario);
$this->load->view("usuarios/novo");
```

O código final ficará assim:

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Usuarios extends CI_Controller {

    public function novo() {
        $usuario = array(
            "nome" => $this->input->post("nome"),
            "email" => $this->input->post("email"),
        )
    }
}
```

```

"senha" => md5($this->input->post("senha"))
);
$this->load->database();
$this->load->model("usuarios_model");
$this->usuarios_model->salva($usuario);
$this->load->view("usuarios/novo");
}
}

```

Agora vamos criar nosso modelo `usuarios_model.php`, com o método salva

```

<?php
class Usuarios_model extends CI_Model {
    public function salva() {
    }
}

```

Vamos implementar o método salva que faz um insert no banco de dados, o primeiro parâmetro é o nome da tabela e o segundo o que iremos inserir no banco, vamos também receber o usuário como parâmetro do método salva.

```

<?php
class Usuarios_model extends CI_Model {
    public function salva($usuario) {
        $this->db->insert("usuarios", $usuario);
    }
}

```

Ainda nos falta a view, ela ficará dentro da pasta `view/usuarios` e se chamara `novo.php`, vamos apenas colocar uma confirmação de cadastro:

```

<html>
    Cadastrado com sucesso!
</html>

```

Vá no seu navegador e insira um usuário, você verá a confirmação:

## Cadastrado com sucesso!

Agora vamos ao `phpmyadmin` para conferir que os usuários realmente foram cadastrados, execute a seguinte query:

```
select * from usuarios;
```

Verifique que seu usuário está lá.

