

02

Logout e FlashData

Nosso sistema já evoluiu bastante, temos até sistema de login. Mas precisamos nos deslogar, então vamos implementar isto agora.

Primeiro removemos o profiling do nosso controller, para não ficar poluindo a tela, removendo a linha que ativa ele, nós temos que fazer isto em dois dos nossos controllers `usuarios`, método `novo` e no `produtos`, método `index`.

Lembra que temos um `if` que controla na view se estamos logados, escondendo os formulários:

```
<?php if(!$this->session->userdata("usuario_logado")) : ?>
formulários aqui
<?php endif ?>
```

Então, vamos alterá-lo, para caso estejamos logados, ele mostrar também um link para nos deslogarmos:

```
<?php if($this->session->userdata("usuario_logado")) : ?>
<a href="login/logout">Logout</a>
<?php endif ?>
```

Vamos melhorar ainda mais usando um helper chamado `anchor`, ele nos ajuda com links, este helper recebe primeiro o link que queremos formar, neste caso `login/logout`, seguido do título do link `Logout`.

```
<?php if($this->session->userdata("usuario_logado")) : ?>
<?= anchor('login/logout', 'Logout')?>
<?php endif ?>
```

Por último podemos passar opções de CSS, vamos colocar a class `btn btn-primary` para deixarmos o link bonito:

```
<?php if($this->session->userdata("usuario_logado")) : ?>
<?= anchor('login/logout', 'Logout', array("class" => "btn btn-primary"))?>
<?php endif ?>
```

Agora vamos atualizar e veremos o botão de logout:

Produtos

Bola de futebol	R\$ 300,00
HD externo usado	R\$ 400,00
Garrafa de 2 litros	R\$ 10,50

[Logout](#)

Legal, vamos alterar nosso `if`, para se tornar um `if-else`, pois do modo que fizemos o `if`, ele apenas está colocando o botão, mas não está escondendo o formulário:

```
<?php if($this->session->userdata("usuario_logado")) : ?>
<?= anchor('login/logout','Logout', array("class" => "btn btn-primary"))?>
<?php else : ?>
    formulários aqui
<?php endif ?>
```

Agora é hora de criar o método de `logout` no nosso controller `login.php`:

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

class Login extends CI_Controller {
public function autenticar() {
    //implementação do método
}

public function logout(){
}
```

Para colocarmos o `usuario` na session nós usamos o `set_userdata()`, para removermos iremos usar o `unset_userdata()`

```
public function logout(){
$this->session->unset_userdata("usuario_logado");
}
```

Agora vamos confirmar para o usuário que ele foi deslogado, carregando uma view para isto

```
public function logout(){
$this->session->unset_userdata("usuario_logado");
$this->load->view('login/logout');
}
```

Vamos agora criar, dentro do diretório login, o arquivo `logout.php`:

```
<html>
<body>
    Deslogado com sucesso.
</body>
</html>
```

Vamos testar o logout:

Deslogado com sucesso.

Você não acha estranho, toda vez que nos logarmos, caímos em uma página e temos que voltar para a página inicial, depois carregarmos ela novamente, para ai sim conseguirmos ver que estamos logados? Não seria interessante que já fossemos direto para a página inicial e lá mesmo confirmasse nosso login?

Para isto vamos precisar usar o helper de `url`, ele tem um método chamado `redirect()`, que redireciona o usuário para uma outra uri. Lembra que usamos o helper `url` uma vez já? Pelo jeito usaremos ele em vários locais, então vamos adicioná-lo no `autoload.php`:

```
$autoload['helper'] = array('url');
```

No nosso `produtos controller`, no método `index`, removemos o `load`, pois já esta sendo feito automaticamente, onde temos:

```
$this->load->helper(array("url", "currency", "form));
```

Ficará:

```
$this->load->helper(array("currency", "form));
```

Pronto, agora temos o helper de `url` em todos os controllers, podemos alterar o `autentica` do `login controller`, onde fazemos o `load` da view para ser um redirect para a página inicial, o redirect faz outra requisição para a url especificada.

```
redirect("/");
```

Agora ao usarmos o `login`, iremos cair na página inicial, mas falta ainda avisarmos o usuário que ele está logado, que o `login` dele foi um sucesso. É sempre muito importante darmos um feedback sobre as ações que o usuário realiza no sistema.

Mas ao usarmos o `redirect()` o usuário faz uma requisição, então precisamos de algum modo marcar na sessão dele que na próxima requisição teremos uma mensagem para ser mostrada, aquela requisição depois do `autentica`, neste caso a home "/":

```
if($usuario) {
    $dados = array("mensagem" => "Logado com sucesso");
    $this->session->set_userdata("usuario_logado" , $usuario);
} else {
    $dados = array("mensagem" => "Usuário ou senha inválida.")
}
```

Agora não queremos mais por nos `$dados` e enviar para a página atual, queremos por na sessão, mas queremos que seja apenas na próxima ida e volta do cliente e não que fique todo o tempo na session, pois o período de vida dela é maior do que queremos que a mensagem dure, então ao invés de usarmos `set_userdata`, iremos usar `set_flashdata`, este `flashdata` só dura uma requisição a mais.

Vamos agora alterar o `if` para usar o `flashdata`:

```
if($usuario) {
    $this->session->set_userdata("usuario_logado" , $usuario);
    $this->session->set_flashdata("success" , "Logado com sucesso");
} else {
    $this->session->set_flashdata("success" , "Usuário ou senha inválida");
}
```

Então agora, por apenas uma requisição após o `autenticar`, nós temos o `success`, então agora no nosso `index.php`, vamos mostrar a mensagem de sucesso, vamos adicioná-la na página:

```
resto do código html
<div class="container">
    <?= $this->session->flashdata("success") ?>
    <h1> Produtos </h1>
    resto do código html
```

Se logue novamente para testar e veja a mensagem acima do index:

Logado com sucesso

Produtos

Bola de futebol	R\$ 300,00
-----------------	------------

HD externo usado	R\$ 400,00
------------------	------------

Garrafa de 2 litros	R\$ 10,50
---------------------	-----------

[Logout](#)

Se você se logar com um usuário inválido, verá que temos problemas de encoding, pois não setamos o `charset`, ele está apenas no arquivo `autenticar`. Vamos colocar a `metatag` lá no `index.php`:

```
<head>
    <link rel="stylesheet" href="=base_url("css/bootstrap.css")?"&gt;"
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" >
</head>
```

Vamos apagar o `autenticar.php`, pois não usaremos mais ele.

Seria interessante deixar a mensagem de confirmação um pouco mais bonita, então vamos colocar `success` dentro de um `p`, com uma classe do `bootstrap`:

```
<p class="alert-success"><?= $this->session->flashdata("success") ?></p>
```

Teste e veja que no caso de sucesso, ficou bem legal, mas no caso de falha ficou estranho, verde, parece que é uma mensagem apenas para sucesso:

Usuário ou senha inválida.

Produtos

Bola de futebol	R\$ 300,00
HD externo usado	R\$ 400,00
Garrafa de 2 litros	R\$ 10,50

Vamos alterar para usarmos dois estilos o `success` e o `danger`:

```
if($usuario) {
    $this->session->set_userdata("usuario_logado" , $usuario);
    $this->session->set_flashdata("success" , "Logado com sucesso");
} else {
    $this->session->set_flashdata("danger" , "Usuário ou senha inválida");
}
```

Na view também:

```
<p class="alert-success"><?= $this->session->flashdata("success") ?></p>
<p class="alert-danger"><?= $this->session->flashdata("danger") ?></p>
```

Agora no `logout` queremos fazer a mesma coisa, sem uma página intermediária, agora que já sabemos como fazer, vamos alterar nosso controller:

```
public function logout(){
    $this->session->unset_userdata("usuario_logado");
    redirect("/");
}
```

Agora precisamos adicionar a mensagem de confirmação:

```
public function logout(){
    $this->session->unset_userdata("usuario_logado");
    $this->session->set_flashdata("success" , "Deslogado com sucesso");
    redirect("/");
}
```

Vamos agora testar:

Deslogado com sucesso

Produtos

Bola de futebol	R\$ 300,00
HD externo usado	R\$ 400,00
Garrafa de 2 litros	R\$ 10,50