**Refatoração, escopo de flash e controle de erro**

Nosso código ainda está muito espalhado. Temos dois trechos muito parecidos no **index.php** e um outro igual no**produto-lista.php**. Vamos extrair este código?

Pegamos o código de sucesso:

if(isset($\_SESSION["success"])) {

?>

<p class="alert-success"><?= $\_SESSION["success"]?></p>

<?php

unset($\_SESSION["success"]);

}

E extraímos para uma função que mostra um alerta de acordo com o tipo:

function mostraAlerta($tipo) {

}

Agora queremos verificar se este tipo de alerta está setado:

function mostraAlerta($tipo) {

if(isset($\_SESSION["success"])) {

}

}

E aí dentro, em caso positivo, queremos mostrar a mensagem e remover o campo da sessão:

function mostraAlerta($tipo) {

if(isset($\_SESSION[$tipo])) {

?>

<p class="alert-<?= $tipo ?>"><?= $\_SESSION[$tipo]?></p>

<?php

unset($\_SESSION[$tipo]);

}

}

Perfeito. Basta invocarmos esta função com o parâmetro **success** para que executemos o código de mostrar alerta de sucesso:

mostraAlerta("success");

E queremos mostrar também o alerta de danger:

mostraAlerta("success");

mostraAlerta("danger");

O código fica bem mais simples, mas desejamos usar ele não só no index.php, mas também em outro arquivo. Portanto extraímos essa função para outro arquivo, um chamado **mostra-alerta.php**:

<?php

function mostraAlerta($tipo) {

if(isset($\_SESSION[$tipo])) {

?>

<p class="alert-<?= $tipo ?>"><?= $\_SESSION[$tipo]?></p>

<?php

unset($\_SESSION[$tipo]);

}

}

Agora no **index.php** e no **produto-lista.php** basta importar e usar:

<?php mostraAlerta("success"); ?>

<?php mostraAlerta("danger"); ?>

Mas é assim mesmo que queremos trabalhar? Toda página que precisa mostrar um alerta de sucesso ou erro tem que lembrar de chamar essa função? Não, ao invés de usarmos a invocação em cada página, jogamos ela no fim de nosso **cabecalho.php**:

<?php mostraAlerta("success"); ?>

<?php mostraAlerta("danger"); ?>

E no início do cabecalho:

<?php include("mostra-alerta.php"); ?>

Perfeito. Mas repare que o **mostra-alerta** usa a sessão, certo? E já chamamos o **session\_start()**? Parece que não, então vamos jogar o **session\_start** dentro de nosso arquivo **mostra-alerta.php**. Assim sempre que carregar o arquivo, a sessão já existe:

<?php

session\_start();

function mostraAlerta($tipo) {

if(isset($\_SESSION[$tipo])) {

?>

<p class="alert-<?= $tipo ?>"><?= $\_SESSION[$tipo]?></p>

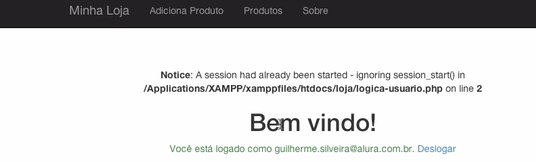
<?php

unset($\_SESSION[$tipo]);

}

}

Testamos o login mas dá um erro estranho, um tal de sessão já foi inicializada:



É só uma informação, um notice, não significa que é um erro, mas que eu possa estar fazendo algo que achava que funcionava de outra maneira. O que acontece é que chamamos o **session\_start** duas vezes. Uma primeira dentro do**mostra-alerta** e outra no **logica-usuario**. No nosso caso queremos deixar de visualizar esse tipo de erro, erros do tipo **E\_NOTICE**. Como desejamos mostrar todos os tipos de erro (**E\_ALL**) exceto **E\_NOTICE**, podemos fazer a exclusão de bits:

E\_ALL ^ E\_NOTICE

Essa configuração pode ser feita no **php.ini** ou via programática. Como em alguns deploys você não tem acesso ao arquivo de configuração, faremos dessa vez via código, chamando o **error\_reporting**:

error\_reporting(E\_ALL ^ E\_NOTICE);

Fazemos isso em nosso cabeçalho:

<?php

error\_reporting(E\_ALL ^ E\_NOTICE);

include("mostra-alerta.php"); ?>

Testamos novamente e sucesso!

Agora que refatoramos nosso arquivo de mostrar alertas podemos analisar melhor o que fizemos: setamos um atributo no escopo de sessão que será removido e na próxima requisição. Esse escopo que criamos, que dura mais do que uma requisição (dura exatamente duas requisições, a atual e a próxima) e menos do que uma sessão (dura diversas requisições), é chamado de escopo de **flash**. Na prática, se não usamos nenhum framework temos que criá-lo na mão, ou se usamos frameworks eles já fornecem esse escopo para nós, da mesma maneira que acessamos o escopo de cookies ou de session.