**Passando parâmetros para a próxima requisição**

Ainda há algumas coisas estranhas em nosso código, por exemplo, se tentamos nos logar no sistema com um usuário e senha inválido. Lá em cima, na URI, aparece **login=0**, e o nosso usuário final pode brincar com isso, ele pode alterar esse parâmetro e mostrar uma mensagem indesejada. Temos que tomar muito cuidado com esses parâmetros. Imagina se passamos um parâmetro:

http://localhost/loja/index.php?mensagem=Logado com sucesso!

Mas um usuário espertinho poderia colocar um palavrão. Por mais que não seja uma brecha de segurança, é uma brecha que pode causar vergonha para uma empresa. Evitamos essas situações. O que devemos fazer então? Quando o usuário é logado ou o login fracassa queremos marcá-lo com essa informação, para que no redirect, quando ele fizer a próxima requisição como cliente, nós possamos mostrar para ele a mensagem que armazenamos. Isto é queremos colocar a mensagem na session. E na próxima requisição queremos tirar ela da session e mostrar para o usuário.

Vamos começar? Primeiro no formulário de produto e no adiciona produto temos a chamada para **verificaUsuario**, no arquivo **logica-usuario.php**. Essa função faz:

function verificaUsuario() {

if(!usuarioEstaLogado()) {

header("Location: index.php?falhaDeSeguranca=true");

die();

}

}

Primeiro deixamos de passar a falha de segurança como parâmetro:

function verificaUsuario() {

if(!usuarioEstaLogado()) {

header("Location: index.php");

die();

}

}

E agora adicionamos uma variável na sessão, algo que será um aviso de erro, portanto um *danger*:

function verificaUsuario() {

if(!usuarioEstaLogado()) {

$\_SESSION["danger"] = "Você não tem acesso a esta funcionalidade.";

header("Location: index.php");

die();

}

}

Agora no nosso arquivo **index.php** não precisamos mais ler o parâmetro falha de segurança. Ao invés daquele **if**, colocamos um **if** para o danger, e mostramos a mensagem do servidor, da sessão, que não foi para o cliente:

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

?>

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

<?php

}

Mas precisamos remover a variável da sessão. Para remover, usamos o **unset**:

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

?>

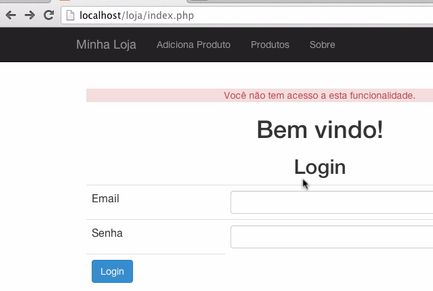
<p class="alert-danger"><?= $\_SESSION["danger"]?></p>

<?php

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

}

Testamos a aplicação e tudo funciona, mas agora não temos mais o parâmetro lá em cima:



Agora devemos fazer o mesmo processo de usar a sessão em outros casos que usávamos parâmetros de URI para trocar informações. Vamos ao **login.php** e no caso de usuário com senha inválida usamos a mesma mensagem de danger:

if($usuario == null) {

$\_SESSION["danger"] = "Usuário ou senha inválido.";

header("Location: index.php");

} else {

E agora no **index.php** não precisamos mais do caso **login==false** já que nunca mais enviamos ele. Enviamos agora pelo atributo da sessão. Então removemos o **if** do caso de **login==false**.

Seguindo o padrão, farei a mesma coisa no login com sucesso. A diferença é que agora configuraremos a mensagem de **success\* ao invés de**danger\*\*:

if($usuario == null) {

$\_SESSION["danger"] = "Usuário ou senha inválido.";

header("Location: index.php");

} else {

$\_SESSION["success"] = "Usuário logado com sucesso.";

logaUsuario($usuario["email"]);

header("Location: index.php");

}

E no **index.php** podemos remover completamente os **ifs** de login. Precisamos adicionar um *if* para o caso de**success**:

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

?>

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

<?php

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

}

Só falta agora o *if* de logout. Vamos no **logout.php** e trocamos para usar a sessão:

$\_SESSION["success"] = "Deslogado com sucesso.";

header("Location: index.php");

die();

E no index.php removemos este caso. Testamos o login e tudo está funcionando, perfeito. Mas o logout não funciona. A mensagem não aparece. O que aconteceu com ele? Vamos olhar seu código:

logout();

$\_SESSION["success"] = "Deslogado com sucesso.";

header("Location: index.php");

die();

Repare que nós matamos a sessão e depois colocamos algo nela, mas ela será jogada fora! Portanto dentro do método logout devemos destruir a sessão atual e criar uma nova sessão:

function logout() {

session\_destroy();

session\_start();

}

Existe mais um lugar onde fazemos o if, a listagem de produto. Assim como no **produto-lista.php**, faremos o if de sucesso nele:

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

?>

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

<?php

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

}

Perfeito, mas agora no **remove-produto** devemos enviar esse parâmetro **success** na session:

$id = $\_POST['id'];

removeProduto($conexao, $id);

$\_SESSION["success"] = "Produto removido com sucesso.";

header("Location: produto-lista.php");

die();

Como o arquivo passa a usar a sessão, precisamos ter certeza que ela existe:

include("logica-usuario.php");

Talvez fosse um caso interessante colocar o **session\_start** em outro lugar, tudo depende de cada projeto: no cabeçalho, em uma lógica etc. Devemos colocar o include também no produto-lista.php.

Testamos e o produto é removido com sucesso, sem passar parâmetros na URI.