



Introdução

Prof. Anderson Henrique

Capítulo 5

Persistência dos dados (cookies e session)

Se você precisa manter informações sobre seus usuários enquanto eles navegam pelo seu sistema, ou até quando eles saem do sistema, é importante que você saiba lidar com cookies e sessões. Cookies são arquivos-texto que podem ser armazenados no computador do usuário para serem recuperados posteriormente pelo servidor no qual seu sistema está hospedado.

Sessões são recursos que podemos utilizar para manter uma conexão com o usuário, enquanto ele estiver navegando no site. Entre algumas utilidades de cookies e sessões, podem ser citadas:

Autenticação de usuários: criação de um sistema envolvendo login, autenticação e logout, o que garante o acesso do conteúdo somente aos usuários autorizados.

Carrinho de compras: utilizando nos sites de comércio eletrônico para armazenar todos os produtos já selecionados pelo cliente para compra, enquanto ele navega pelo site da loja.

Exibição de anúncios ou imagens: para não exibir mais de uma vez um mesmo anúncio ou uma imagem para o usuário é necessário manter informações sobre as imagens que já foram exibidas.

Personalização de páginas: por exemplo, uma livraria virtual poderia exibir o anúncio de um livro de culinária, caso o usuário tivesse feito uma pesquisa pela palavra “culinária” na última vez que acessou o site.

Utilizando cookies

Um cookie é formado pelo par nome/valor, ou seja, possui um nome pelo qual ele é referenciado e um valor associado a esse nome. Podem ser utilizados em qualquer aplicação que necessite compartilhar dados entre diferentes páginas ou até entre diferentes acessos (em dias e horários distintos).

O PHP nos oferece a função **setcookie**, que envia cookies para o computador do usuário. Essa função é usada tanto para definir um cookie como para excluí-lo. Sua sintaxe é a seguinte:

```
setcookie(NOME_COOKIE,VALOR_COOKIE);
```

```
6  setcookie("Login", "Admin");  
7  setcookie("Senha", "123abc");
```

Criando cookie válido por 2 dias, utilizando a função `time()` para obter o tempo atual e somamos 172.800 segundos, que equivalem a 48 horas. Função `time()` trabalha com o padrão UNIX (desde 1 de janeiro de 1970, à 0h).

```
14    setcookie("Autenticado", TRUE, time()+172800);
```

Obtendo o valor do cookie

O array superglobal **`$_COOKIE`** no PHP moderno não é mais utilizado por questões de segurança, agora utilizamos `filter_input(INPUT_COOKIE, NOME_COOKIE)`, recuperando o valor dessa forma:

```
22    echo "Login: " . filter_input(INPUT_COOKIE, "Login") . "<br>";  
23    echo "Senha: " . filter_input(INPUT_COOKIE, "Senha") . "<br>";
```

```
25  [-] if(filter_input(INPUT_COOKIE, "Autenticado")){  
26      echo "Usuário foi autenticado! <br>";  
27  [-] }else{  
28      echo "Usuário não foi autenticado! <br>";  
29  }
```

Excluindo cookies

Basta definirmos novamente com o mesmo nome identificador e não passar valores no segundo parâmetro, ou passar valor NULL.

```
36  setcookie("Login", NULL);  
37  setcookie("Senha", NULL);
```

Criando sessões e utilizando

Para utilizarmos sessões, primeiramente devemos inicializar utilizando a função **session_start()**, criamos a sessão utilizando a variável superglobal **\$_SESSION[Identificador]**.

No próximo exemplo, iremos criar duas sessões para exemplificar o uso em nossos sistemas:

Inicializa a sessão

```
6 session_start();
```

Criando sessões

```
11 $_SESSION["Usuario"] = "Jair da Silva";  
12 $_SESSION["Autenticado"] = true;
```


Obtendo o valor da sessão utilizando o array superglobal \$_SESSION[]

```
17 if($_SESSION["Autenticado"]){  
18     echo "O usuário {$_SESSION['Usuario']} foi autenticado! <br>";  
19 }else{  
20     echo "O usuário {$_SESSION['Usuario']} não foi autenticado! <br>";  
21 }
```

Para finalizar, podemos apagar todas as variáveis da sessão, e ou destruir os dados de uma sessão, garantindo assim que os dados não serão mais acessados, pois a persistência foi perdida.

```
28 session_unset();  
29 session_destroy();
```