

Aula 11

Professor:

Claudio Esperança

Sessões

Conteúdo:

- Aplicações com estado
- Exemplo: Carrinho de Compras
- Cookies
- Gerenciamento de Sessões em PHP
- Exemplo: Carrinho de Compras usando Sessões

Aplicações com estado

- Interações via web são tipicamente *sem estado*
- Protocolo *http*:
 - ▶ Navegador faz um pedido ao servidor
 - ▶ Servidor responde com uma página
 - ▶ Resultado *não depende* (em geral) de interações anteriores
- Adequado para aplicações onde dados são somente inspecionados
 - ▶ Interações mais longas e complexas requerem *estado*
- *Estado* encapsula dados que o usuário especificou em interações anteriores. Ex.:
 - ▶ Código de *login* do usuário
 - ▶ Cesta de compras

Mantendo o estado

- Aplicações web com estado precisam manter dados ao longo de diversas interações
 - ▶ Estado ≈ variáveis globais
- Uma maneira de fazer isso é enviar todos os dados do estado em cada interação
- Por exemplo:
 - ▶ Dados são enviados do navegador (cliente) para o servidor através do método POST ou GET
 - ▶ Servidor gera uma nova página com esses dados em campos de um formulário
 - Pode-se usar campos escondidos (`type = "hidden"`) para guardar informações que não necessitam ser exibidas

Exemplo: Carrinho de Compras

The screenshot shows a Mozilla Firefox browser window titled "Loja X - Mozilla Firefox". The main content area displays a "Cesta de Compras" (Shopping Cart) table with four items: Item A, Item B, Item C, and Item D. Each item has a price and a quantity input field set to 0. Below the table, the total amount is shown as "Total: R\$0.00". At the bottom are two buttons: "Atualizar" (Update) and "Comprar" (Buy). The browser's menu bar includes File, Edit, View, Go, Bookmarks, Tools, and Help.

Item	Preço	Quantidade
Item A	10.50	0
Item B	9.25	0
Item C	7.25	0
Item D	7.32	0

Total: R\$0.00

Atualizar Comprar

Abrir editor

Abrir navegador



Exemplo: Carrinho de Compras

Loja X - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

Cesta de Compras

Item	Preço	Quantidade
Item A	10.50	0
Item B	9.25	0
Item C	7.25	0
Item D	7.32	0

Total: R\$0.00

Atualizar Comprar

- Inicialmente, todos os itens com quantidade 0

Abrir editor

Abrir navegador



Exemplo: Carrinho de Compras

Loja X - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

Cesta de Compras

Item	Preço	Quantidade
Item A	10.50	<input type="text" value="0"/>
Item B	9.25	<input type="text" value="0"/>
Item C	7.25	<input type="text" value="0"/>
Item D	7.32	<input type="text" value="0"/>

Total: R\$0.00

Atualizar Comprar

- Inicialmente, todos os itens com quantidade 0
- Usuário preenche formulário indicando quantidades de cada item que deseja comprar

Abrir editor

Abrir navegador



Exemplo: Carrinho de Compras

Loja X - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

Cesta de Compras

Item	Preço	Quantidade
Item A	10.50	<input type="text" value="0"/>
Item B	9.25	<input type="text" value="1"/>
Item C	7.25	<input type="text" value="2"/>
Item D	7.32	<input type="text" value="3"/>

Total: R\$0.00

[Atualizar](#) [Comprar](#)

- Inicialmente, todos os itens com quantidade 0
- Usuário preenche formulário indicando quantidades de cada item que deseja comprar

Abrir editor

Abrir navegador



Exemplo: Carrinho de Compras

Loja X - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

Cesta de Compras

Item	Preço	Quantidade
Item A	10.50	<input type="text" value="0"/>
Item B	9.25	<input type="text" value="1"/>
Item C	7.25	<input type="text" value="2"/>
Item D	7.32	<input type="text" value="3"/>

Total: R\$0.00

Atualizar **Comprar**

- Inicialmente, todos os itens com quantidade 0
- Usuário preenche formulário indicando quantidades de cada item que deseja comprar
- O Botão **Atualizar** recalcula o total devido

Abrir editor

Abrir navegador



Exemplo: Carrinho de Compras

Loja X - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

Cesta de Compras

Item	Preço	Quantidade
Item A	10.50	<input type="text" value="0"/>
Item B	9.25	<input type="text" value="1"/>
Item C	7.25	<input type="text" value="2"/>
Item D	7.32	<input type="text" value="3"/>

Total: R\$45.71

[Atualizar](#) [Comprar](#)

- Inicialmente, todos os itens com quantidade 0
- Usuário preenche formulário indicando quantidades de cada item que deseja comprar
- O Botão **Atualizar** recalcula o total devido

Abrir editor

Abrir navegador



Exemplo: Carrinho de Compras

- Formulário formatado usando uma tabela html
- Cada item corresponde a uma linha da tabela que tem um formato semelhante a:

```
<tr>
    <td>%s</td>
    <td align="right">% .2f</td>
    <td><input name="qtd[%d]" value="%d" size=2></td>
</tr>
```

- Vamos assumir que o trecho html acima foi guardado numa variável chamada `$linhaTabela`
- Ao programa cabe substituir os itens marcados **em azul** com valores reais para cada item, a saber:
 - ▶ Nome do item
 - ▶ Preço do item
 - ▶ Número do item
 - ▶ Quantidade do item

Exemplo: Carrinho de Compras

- Lista com nomes e preços de cada item é guardada numa variável array:

```
$tabelaItens =  
    array(array ("Item A", 10.50),  
          array ("Item B", 9.25),  
          array ("Item C", 7.25),  
          array ("Item D", 7.32));
```

- O array **\$qtd** deve conter as quantidades de cada item que foram preenchidas no formulário
 - ▶ Se a configuração **register_globals** do PHP estiver em uso, isto é feito automaticamente
 - ▶ Senão, precisamos obter os valores da variável **\$_POST**:

```
if (IsSet($_POST['qtd']))  
    $qtd = $_POST['qtd'];  
else $qtd = array ();
```

Exemplo: Carrinho de Compras

- Podemos agora gerar as linhas da tabela para a próxima interação, calculando em `$total` o valor das compras:

```
$total = 0;  
foreach ($tabelaItens as $indice => $item) {  
    list($nome, $preco) = $item;  
    if (IsSet($qtd[$indice]))  
        $q = $qtd[$indice];  
    else $q = 0;  
    $total += $q * $preco;  
    echo sprintf ($linhaTabela,  
                 $nome, $preco, $indice, $q);  
}
```

- Note que o valor das compras também deve aparecer na página ao final da tabela, isto é, o html da página deve conter algo como:

Total: R\$ <?php echo sprintf ("% .2f", \$total); ?>

Cookies

- Um *cookie* (biscoito, em inglês) é um dado simples que é armazenado no navegador
- Assim como os dados enviados através dos métodos GET ou POST, os cookies são enviados no cabeçalho (*header*) de um documento servido através do protocolo HTTP. Ex.:
 - ▶ Ex.: O cabeçalho abaixo define o cookie contador com o valor 0:
HTTP/1.x 200 OK
Date: Sat, 29 Jul 2006 19:19:32 GMT
Set-Cookie: contador=0; expires=Sat, 29 Jul 2006
19:29:32 GMT; path=/
...
<html> ... </html>
- Cookies podem também ser definidos e manipulados pelo próprio navegador através de *JavaScript*

Cookies

- Uma vez recebido um cookie, o navegador lembra esse valor e automaticamente o reenvia ao servidor em novos pedidos ao *mesmo servidor*
- Por exemplo, ao reacessar o mesmo endereço que gerou o cabeçalho anterior, um novo pedido (GET) terá o valor do cookie contador expresso num campo **Cookie** enviado ao servidor:

```
GET /~esperanc/aula12/cookiesimples.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US;
rv:1.8.0.1) Gecko/20060124 Firefox/1.5.0.1
...
Cookie: contador=0
...
```

Parâmetros para campo *Set-Cookie*

- ***expires=data hora***
 - ▶ Uma data e hora além da qual o cookie expira.
 - ▶ Se não especificado, o navegador lembra o cookie apenas enquanto estiver rodando
- ***domain=domínios***
 - ▶ Domínios para os quais o cookie será reenviado
 - ▶ Caso não especificado, o cookie é reenviado apenas para o servidor que enviou o cookie originalmente
- ***path=caminho***
 - ▶ Cookie será reenviado apenas para páginas dentro desse caminho
 - ▶ caminho / indica o endereço que enviou o cookie originalmente
- ***secure***
 - ▶ Cookie será enviado apenas se comunicação é feita através do Secure Sockets Layer (SSL)

Cookies em PHP

- É possível especificar Cookies usando a função `header()` do PHP
 - ▶ Requer que o usuário conheça a sintaxe do campo Set-Cookie do HTTP
- Uma maneira mais simples é usar a função `setcookie()`:
`int setcookie (name, value, expire, path, domain, secure)`
 - ▶ `name` : nome da cookie (string)
 - ▶ `value` : valor do cookie (string, opcional)
 - ▶ `expire`: data de expiração (inteiro, opcional)
 - ▶ `path`: caminho (string, opcional)
 - ▶ `domain`: domínio (string, opcional)
 - ▶ `secure`: indica uso apenas com SSL (booleano, opcional)

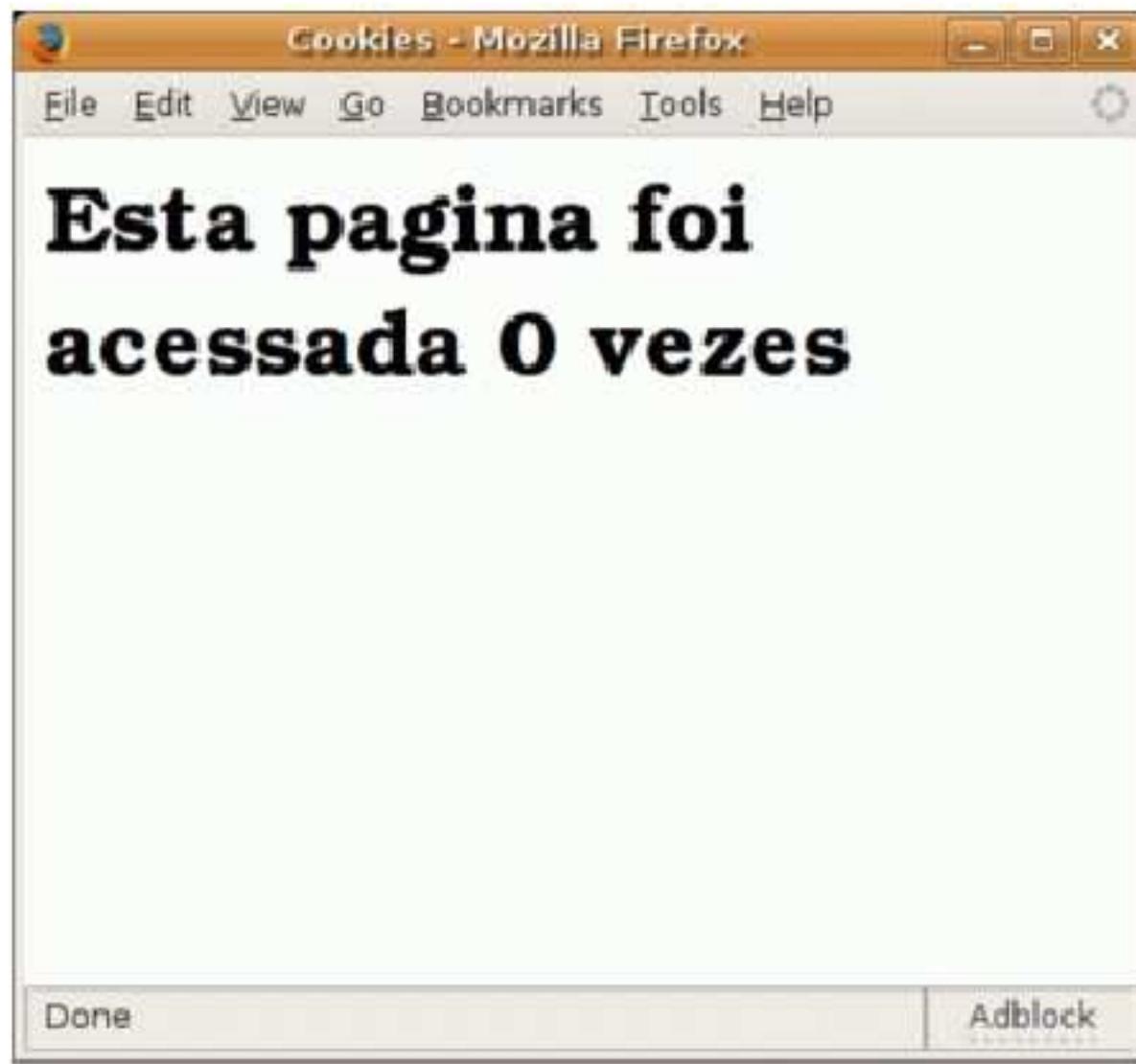
Cookies em PHP

- O cookie informado por um pedido HTTP precisa ser capturado pela aplicação
- Se o PHP foi configurado com `register_globals` a aplicação terá variáveis correspondentes aos nomes dos cookies com valores automaticamente definidos
 - ▶ Esta configuração não é mais encorajada a partir da versão 4.1 do PHP
- Caso contrário, é necessário obter os valores dos cookies examinando a array `$HTTP_COOKIE_VARS`
 - ▶ A partir da versão 4.1 do PHP, pode-se usar o array "super-global" `$_COOKIE` para a mesma finalidade
 - ▶ A vantagem desta é que pode ser usada em funções sem precisar ser importada usando uma declaração `global`

Exemplo: Contador com Cookies

```
<?php  
// Verificamos se o pedido HTTP contém um cookie  
// chamado 'contador'  
if (isset($_COOKIE['contador']))  
    $contador = $_COOKIE['contador'] + 1;  
else $contador = 0;  
  
// Enviar um novo valor para o cookie válido por 10  
// minutos  
setcookie("contador", $contador, time()+600, "/");  
?>  
<html>  
    <head><title>Cookies</title></head>  
    <body>  
        <h1> Esta pagina foi acessada <?php echo  
$contador; ?> vezes </h1>  
    </body>  
</html>
```

Exemplo: Contador com *Cookies*

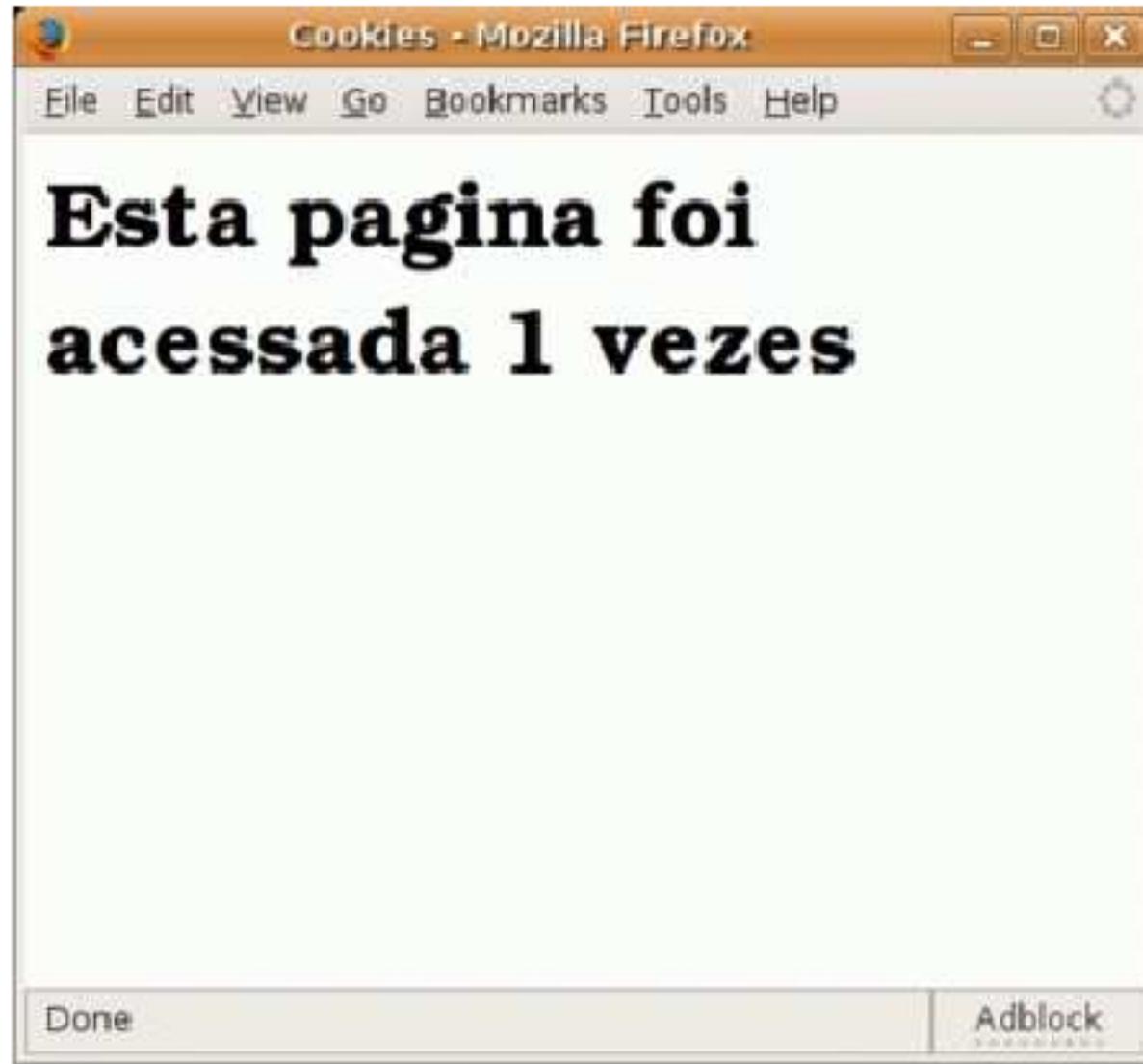


Abrir editor

Abrir navegador



Exemplo: Contador com Cookies

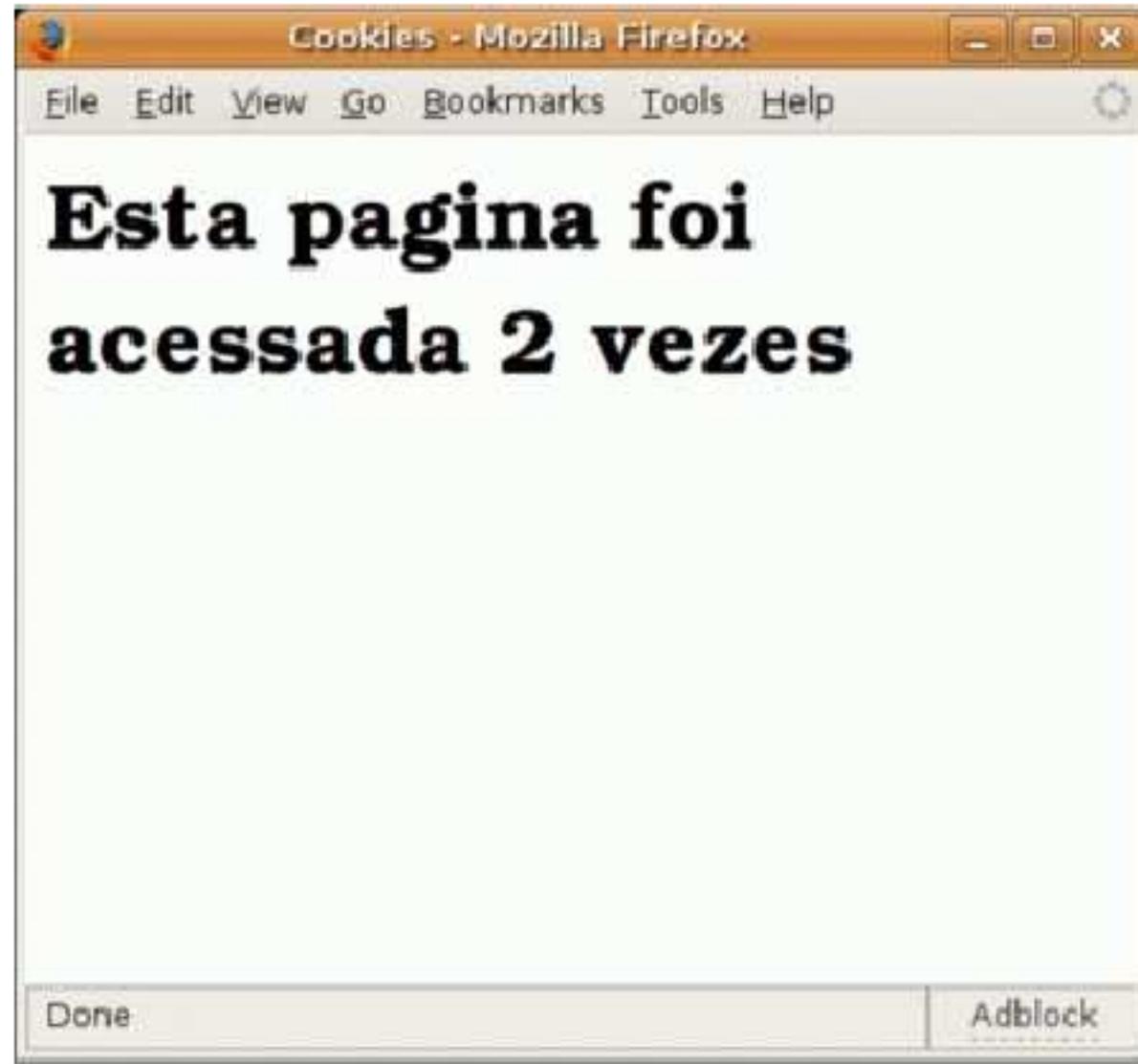


Abrir editor

Abrir navegador



Exemplo: Contador com *Cookies*



Abrir editor

Abrir navegador



Sessões

- Ao invés de transmitir repetidamente todas as variáveis do estado do navegador para o servidor e vice-versa, pode-se manter as variáveis do estado apenas no servidor
 - ▶ O navegador mantém apenas um *identificador de sessão* (um cookie, normalmente)
 - É usado pelo servidor para distinguir aquele cliente entre todos os clientes sendo servidos
 - ▶ A sessão é associada com o tempo que o servidor mantém os dados do cliente
- Uma implicação importante é que o servidor usa recursos (memória/disco) para manter dados de *todos* os clientes sendo servidos
 - ▶ Não há como saber se um cliente abandonou a sessão
 - ▶ Sessões devem ter um tempo de duração limitado para evitar desperdício de recursos no servidor

Gerenciamento de Sessões

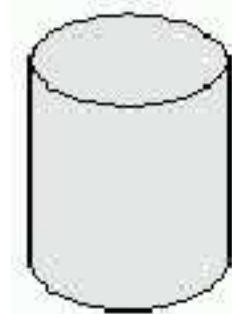
Cliente



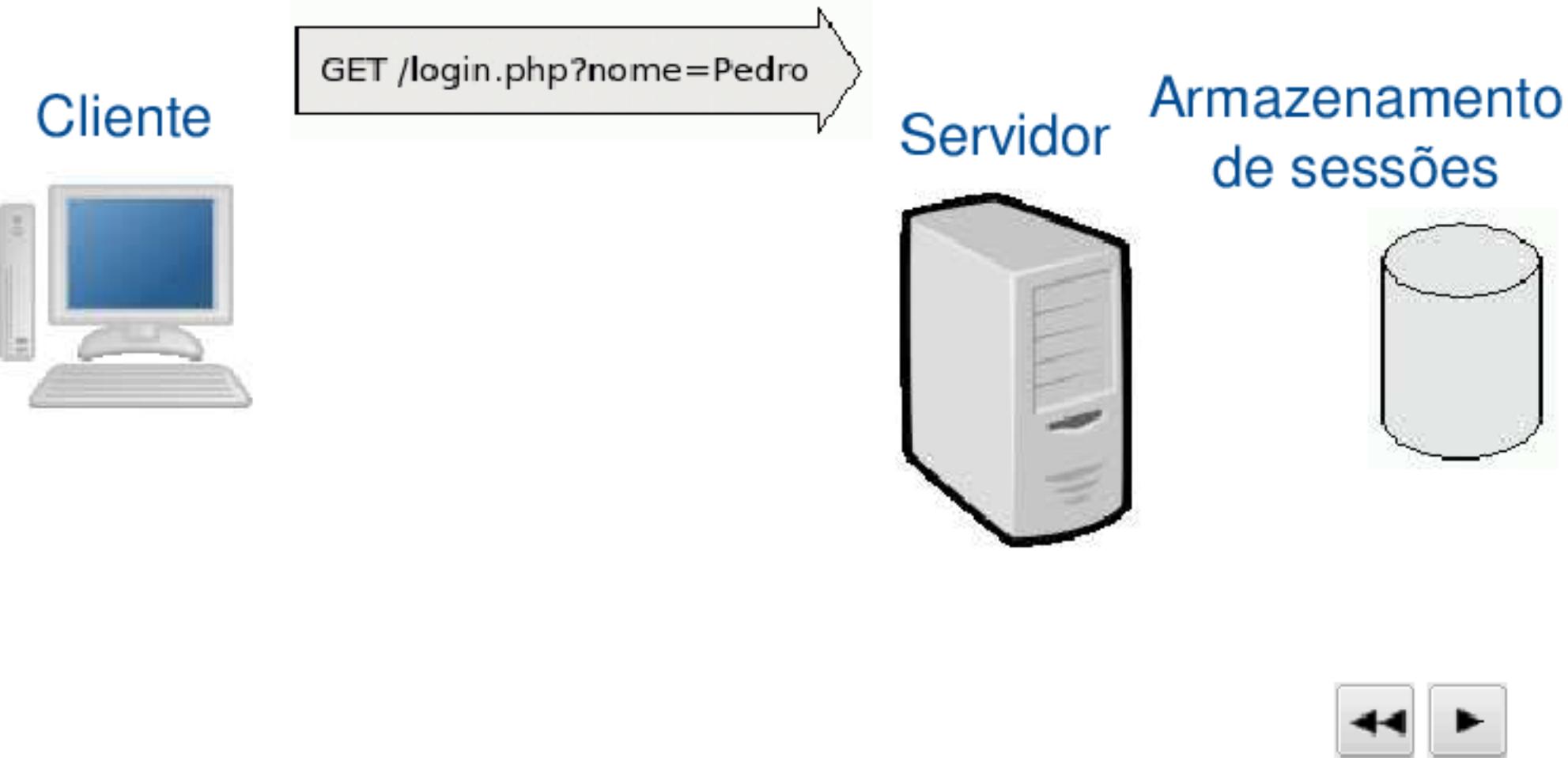
Servidor



Armazenamento
de sessões



Gerenciamento de Sessões



Gerenciamento de Sessões

Cliente



Servidor



Armazenamento
de sessões



Gerenciamento de Sessões

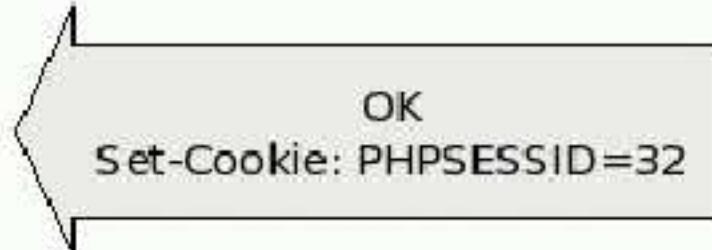
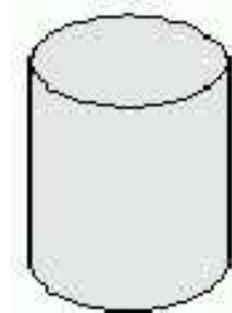
Cliente



Servidor



Armazenamento
de sessões



Gerenciamento de Sessões

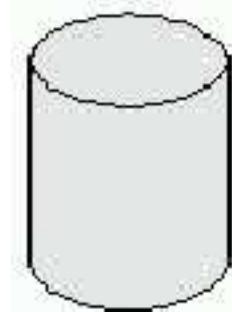
Cliente



Servidor



Armazenamento
de sessões



GET /alo.php
Cookie: PHPSESSID=32



Gerenciamento de Sessões

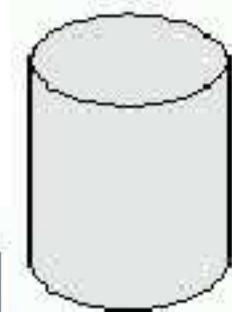
Cliente



Servidor



Armazenamento
de sessões



Nome=
Pedro



Gerenciamento de Sessões

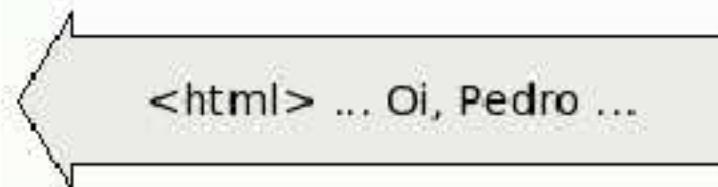
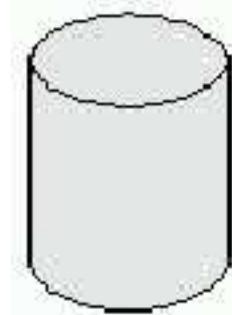
Cliente



Servidor



Armazenamento
de sessões



Iniciando uma sessão

- Para iniciar uma sessão ou registrar uma sessão em curso, a aplicação deve chamar a função `session_start()`
 - ▶ Caso seja a primeira interação da sessão, um novo cookie `PHPSESSID` é gerado e enviado ao cliente e um novo arquivo de sessão é criado
 - O cookie `PHPSESSID` é uma string com 32 dígitos hexadecimais aleatórios
 - Normalmente, o arquivo de sessão é guardado em algum diretório temporário e tem nome com prefixo `sess_` e o número da sessão
 - ▶ Caso o cookie `PHPSESSID` já tenha sido recebido junto com o pedido, os dados da sessão do usuário são recuperados do arquivo de sessão
 - Se o arquivo correspondente não for encontrado, um novo arquivo vazio é criado
 - ▶ Não é necessário chamar `header()` ou `setcookie()`

Propagando variáveis

- Até a versão 4.1 do PHP, variáveis de sessão eram promovidas a variáveis globais automaticamente através do mecanismo `register_globals` e do comando PHP `session_register()`
 - ▶ A partir da versão 4.2, esse mecanismo foi descontinuado por representar risco de segurança
- Os dados globais de uma sessão são guardadas em um array "super-global" chamado `$_SESSION`
 - ▶ `$_SESSION` é global mesmo em funções.
 - Para compatibilidade com PHP 4.1 e anteriores, pode-se usar `$HTTP_SESSION_VARS` para o mesmo fim
 - ▶ Para guardar um contador, por exemplo, escreve-se
`$_SESSION["contador"] = 1`
 - ▶ Uma vez inicializada a variável, ela se propagará por todas as interações

Exemplo: Contador com Sessão

```
<?php  
session_start();  
if (!IsSet($_SESSION["contador"]))  
    $_SESSION["contador"]=0;  
$_SESSION["contador"]++;  
?  
  
<html>  
<head><title>Sessão Simples</title></head>  
<body>  
<h1>Esta página foi acessada  
<?php echo $_SESSION["contador"]; ?>  
vezes </h1>  
</body>  
</html>
```

[Abrir editor](#)[Abrir navegador](#)

Destruindo a Sessão

- Numa aplicação web usando sessões é importante fechar explicitamente uma sessão sempre que possível
- Para isso, use o comando `destroy_session()`
 - ▶ Normalmente associado com uma ação do tipo "sair do site"
 - ▶ O comando apaga o arquivo de sessão e destrói as variáveis de sessão
- Mesmo que uma sessão não seja explicitamente fechada, o servidor verifica periodicamente se existem sessões para as quais interações não ocorreram recentemente
 - ▶ Opção `session.gc_maxlifetime` do arquivo de configuração `php.ini` define o tempo máximo de inatividade
 - Default = 1440 segundos (24 minutos)
 - ▶ Decorrido o tempo máximo de inatividade, o arquivo de sessão correspondente é apagado

Exemplo: Carrinho de Compras com Sessões

- A aplicação de carrinho de compras pode ser melhorada com o uso de sessões e, desta vez, com os itens armazenados em um banco de dados:

```
CREATE TABLE produto (
    id int(11) NOT NULL auto_increment,
    descricao varchar(30) NOT NULL ,
    preco decimal(10,2) NOT NULL ,
    PRIMARY KEY ('id'))
```

- Vamos supor que o número total de produtos disponíveis seja grande demais para ser apresentado de uma só vez
 - ▶ Nossa aplicação deverá paginar a relação de itens de forma a permitir o exame de 5 produtos por vez
 - Novos botões: "Próxima Página" e "Página Anterior"
 - ▶ Número da página e as quantidades compradas de cada item são armazenadas como variáveis de sessão

Exemplo: Carrinho de Compras com Sessões

Loja X - Firefox

File Edit View Go Bookmarks Tools Help

Cesta de Compras

Item	Descrição	Preço	Quantidade
1	Televisão	950.00	<input type="text" value="0"/>
2	Monitor CRT	400.00	<input type="text" value="0"/>
3	Computador Pessoal A	1500.00	<input type="text" value="0"/>
4	Computador Pessoal B	2400.00	<input type="text" value="0"/>
5	Mouse Comum	20.00	<input type="text" value="0"/>

Página 1 de 3 [Atualizar](#) [Comprar](#) [Próxima Pg.](#)

Total : R\$0.00

[Abrir editor](#)
[Abrir navegador](#)


Exemplo: Carrinho de Compras com Sessões

Loja X - Firefox

File Edit View Go Bookmarks Tools Help

Cesta de Compras

Item	Descrição	Preço	Quantidade
1	Televisão	950.00	<input type="text" value="0"/>
2	Monitor CRT	400.00	<input type="text" value="0"/>
3	Computador Pessoal A	1500.00	<input type="text" value="0"/>
4	Computador Pessoal B	2400.00	<input type="text" value="0"/>
5	Mouse Comum	20.00	<input type="text" value="0"/>

Página 1 de 3 [Atualizar](#) | [Comprar](#) | [Próxima Pg.](#)

Total : R\$0.00

- Usuário preenche a quantidade de um item

[Abrir editor](#)
[Abrir navegador](#)


Exemplo: Carrinho de Compras com Sessões

Loja X - Firefox

File Edit View Go Bookmarks Tools Help

Cesta de Compras

Item	Descrição	Preço	Quantidade
1	Televisão	950.00	<input type="text" value="0"/>
2	Monitor CRT	400.00	<input type="text" value="0"/>
3	Computador Pessoal A	1500.00	<input type="text" value="0"/>
4	Computador Pessoal B	2400.00	<input type="text" value="1"/>
5	Mouse Comum	20.00	<input type="text" value="0"/>

Página 1 de 3 [Atualizar](#) | [Comprar](#) | [Próxima Pg.](#)

Total : R\$0.00

- Usuário preenche a quantidade de um item

[Abrir editor](#)

[Abrir navegador](#)



Exemplo: Carrinho de Compras com Sessões

Loja X - Firefox

File Edit View Go Bookmarks Tools Help

Cesta de Compras

Item	Descrição	Preço	Quantidade
1	Televisão	950.00	<input type="text" value="0"/>
2	Monitor CRT	400.00	<input type="text" value="0"/>
3	Computador Pessoal A	1500.00	<input type="text" value="0"/>
4	Computador Pessoal B	2400.00	<input type="text" value="1"/>
5	Mouse Comum	20.00	<input type="text" value="0"/>

Página 1 de 3 [Atualizar](#) | [Comprar](#) | [Próxima Pg.](#)

Total : R\$0.00

- Usuário preenche a quantidade de um item
- Botão Atualizar atualiza o preço

Abrir editor

Abrir navegador



Exemplo: Carrinho de Compras com Sessões

Loja X - Firefox

File Edit View Go Bookmarks Tools Help

Cesta de Compras

Item	Descrição	Preço	Quantidade
1	Televisão	950.00	<input type="text" value="0"/>
2	Monitor CRT	400.00	<input type="text" value="0"/>
3	Computador Pessoal A	1500.00	<input type="text" value="0"/>
4	Computador Pessoal B	2400.00	<input type="text" value="1"/>
5	Mouse Comum	20.00	<input type="text" value="0"/>

Página 1 de 3 [Atualizar](#) [Comprar](#) [Próxima Pg.](#)

Total : R\$2400.00

- Usuário preenche a quantidade de um item
- Botão Atualizar atualiza o preço

[Abrir editor](#)
[Abrir navegador](#)


Exemplo: Carrinho de Compras com Sessões

Loja X - Firefox

File Edit View Go Bookmarks Tools Help

Cesta de Compras

Item	Descrição	Preço	Quantidade
1	Televisão	950.00	<input type="text" value="0"/>
2	Monitor CRT	400.00	<input type="text" value="0"/>
3	Computador Pessoal A	1500.00	<input type="text" value="0"/>
4	Computador Pessoal B	2400.00	<input type="text" value="1"/>
5	Mouse Comum	20.00	<input type="text" value="0"/>

Página 1 de 3 [Atualizar](#) [Comprar](#) [Próxima Pg.](#)

Total : R\$2400.00

- Usuário preenche a quantidade de um item
- Botão Atualizar atualiza o preço
- Botão Próxima Pg mostra os 5 itens seguintes

[Abrir editor](#)
[Abrir navegador](#)


Exemplo: Carrinho de Compras com Sessões

Loja X - Firefox

File Edit View Go Bookmarks Tools Help

Cesta de Compras

Item	Descrição	Preço	Quantidade
6	Mouse Ótico	60.00	<input type="text" value="0"/>
7	PDA marca A	300.00	<input type="text" value="0"/>
8	PDA marca B	950.00	<input type="text" value="0"/>
9	Alto-falantes	100.00	<input type="text" value="0"/>
10	Telefone	50.00	<input type="text" value="0"/>

Página 2 de 3 [Atualizar](#) [Comprar](#) [Pg. Anterior](#) [Próxima Pg.](#)

Total : R\$2400.00

- Usuário preenche a quantidade de um item
- Botão Atualizar atualiza o preço
- Botão Próxima Pg mostra os 5 itens seguintes

[Abrir editor](#)
[Abrir navegador](#)


Exemplo: Carrinho de Compras com Sessões

Loja X - Firefox

File Edit View Go Bookmarks Tools Help

Cesta de Compras

Item	Descrição	Preço	Quantidade
6	Mouse Ótico	60.00	<input type="text" value="0"/>
7	PDA marca A	300.00	<input type="text" value="0"/>
8	PDA marca B	950.00	<input type="text" value="0"/>
9	Alto-falantes	100.00	<input type="text" value="0"/>
10	Telefone	50.00	<input type="text" value="0"/>

Página 2 de 3 [Atualizar](#) [Comprar](#) [Pg. Anterior](#) [Próxima Pg.](#)

Total : R\$2400.00

- Usuário preenche a quantidade de um item
- Botão Atualizar atualiza o preço
- Botão Próxima Pg mostra os 5 itens seguintes
- Usuário preenche mais um item

[Abrir editor](#)
[Abrir navegador](#)


Exemplo: Carrinho de Compras com Sessões

Loja X - Firefox

File Edit View Go Bookmarks Tools Help

Cesta de Compras

Item	Descrição	Preço	Quantidade
6	Mouse Ótico	60.00	<input type="text" value="0"/>
7	PDA marca A	300.00	<input type="text" value="0"/>
8	PDA marca B	950.00	<input type="text" value="0"/>
9	Alto-falantes	100.00	<input type="text" value="1"/>
10	Telefone	50.00	<input type="text" value="0"/>

Página 2 de 3 [Atualizar](#) [Comprar](#) [Pg. Anterior](#) [Próxima Pg.](#)

Total : R\$2400.00

- Usuário preenche a quantidade de um item
- Botão Atualizar atualiza o preço
- Botão Próxima Pg mostra os 5 itens seguintes
- Usuário preenche mais um item

[Abrir editor](#)
[Abrir navegador](#)


Exemplo: Carrinho de Compras com Sessões

Loja X - Firefox

File Edit View Go Bookmarks Tools Help

Cesta de Compras

Item	Descrição	Preço	Quantidade
6	Mouse Ótico	60.00	<input type="text" value="0"/>
7	PDA marca A	300.00	<input type="text" value="0"/>
8	PDA marca B	950.00	<input type="text" value="0"/>
9	Alto-falantes	100.00	<input type="text" value="1"/>
10	Telefone	50.00	<input type="text" value="0"/>

Página 2 de 3 [Atualizar](#) [Comprar](#) [Pg. Anterior](#) [Próxima Pg.](#)

Total : R\$2400.00

- Usuário preenche a quantidade de um item
- Botão Atualizar atualiza o preço
- Botão Próxima Pg mostra os 5 itens seguintes
- Usuário preenche mais um item
- Botão Próxima Pg mostra os itens seguintes com total atualizado

[Abrir editor](#)

[Abrir navegador](#)



Exemplo: Carrinho de Compras com Sessões

Loja X - Firefox

File Edit View Go Bookmarks Tools Help

Cesta de Compras

Item	Descrição	Preço	Quantidade
11	Impressora Laser	1200.00	<input type="text" value="0"/>
12	Memória Chaveiro 128 MB	140.00	<input type="text" value="0"/>

Página 3 de 3 [Atualizar](#) | [Comprar](#) | [Pg. Anterior](#)

Total : R\$2500.00

[Abrir editor](#)

[Abrir navegador](#)



Exemplo: Carrinho de Compras com Sessões

- Cabeçalho:

```
<?php  
header ("Content-Type: text/html; charset=ISO-8859-  
1",true);  
session_start();  
?>
```

- ▶ Envia um *header* especial para selecionar a codificação de caracteres ISO-8859-1
 - Importante, pois as descrições dos itens possuem caracteres acentuados usando esta codificação
 - Dependendo das configurações do servidor, este pode usar outra codificação por default, por exemplo, UTF-8
- ▶ Instrução para início da sessão

Exemplo: Carrinho de Compras com Sessões

- Código html em geral:

```
<html> <head> . . . </head>
<body>

<h2>Cesta de Compras</h2>

<form method="post" action="cestacomsessoes.php"
      name="frmCesta">
    <table cellpadding="5" cellspacing="5">
      <tr>
        <th width=20px> Item </th>
        <th width=250px> Descrição </th>
        <th width=80px> Preço</th>
        <th width=50px> Quantidade</th>
      </tr>

      <?php . . . ?>

    </form> </body> </html>
```

Exemplo: Carrinho de Compras com Sessões

- Estabelecendo a conexão e realizando uma consulta para obter todos os produtos do banco de dados:

```
$servidor = endereço do servidor;  
$conexao =  
    mysql_connect ($servidor, "aluno", "aluno");  
mysql_select_db ("prog2", $conexao);  
$consulta = mysql_query ("select * from produto",  
    $conexao);
```

- ▶ No nosso código real, o *endereço do servidor* é
`$_SERVER["REMOTE_ADDR"]`, isto é, a máquina do próprio aluno, que deve ter a tabela de produtos instalada no database `prog2`

Exemplo: Carrinho de Compras com Sessões

- Obtendo o número total de páginas, a página atual e as quantidades de cada item:

```
// Numero de páginas depende de quantos
// produtos por página vamos listar
define (LinhasPorPag, 5);
// Arrendondo para cima
$numPaginas =
    ceil(mysql_num_rows ($consulta)/LinhasPorPag);
// Página e quantidades obtidos por funções
$pagina = pega_pagina ($numPaginas);
$qtd = pega_quantidades ();
```

Exemplo: Carrinho de Compras com Sessões

- Obtendo o número da página atual:

```
function pega_pagina ($numPaginas) {  
    // Obtenho página da variável de sessão  
    $pagina = IsSet ($_SESSION["pagina"]) ?  
              $_SESSION["pagina"] : 0;  
    // Ajusto o número da página  
    if (IsSet ($_POST["submit"])) {  
        if ($_POST["submit"] == 'prox')  
            $pagina = min ($pagina+1, $numPaginas-1);  
        if ($_POST["submit"] == 'prev')  
            $pagina = max ($pagina-1, 0);  
    }  
    // Atualizo a variável de sessão  
    $_SESSION["pagina"] = $pagina;  
    return $pagina;  
}
```

Exemplo: Carrinho de Compras com Sessões

- Obtém-se as quantidades totais combinando as quantidades guardadas na variável de sessão e as postadas na última interação

```
function pega_quantidades () {  
    // Obtem qtd da variavel de sessão se existir  
    $qtd = IsSet($_SESSION["qtd"]) ?  
        $_SESSION["qtd"] : array();  
    // Combinar com qtds postadas se existirem  
    if (IsSet($_POST["qtd"])) {  
        foreach ($_POST["qtd"] as $id => $q)  
            $qtd[$id] = $q;  
    }  
    // Atualizar variável de sessão  
    $_SESSION["qtd"] = $qtd;  
    return $qtd;  
}
```

Exemplo: Carrinho de Compras com Sessões

- Gera-se o restante da página em três etapas:

```
// Gerar as linhas da tabela para a página atual
formata_linhas_tabela ($consulta, $pagina, $qtd);
// Marcar o final da tabela
echo "</table>";
// Gerar botões de submissão do formulário
formata_botoes ($numPaginas, $pagina);
// Computar e imprimir o total das compras
formata_total ($consulta, $qtd);
// Encerrar conexão com o MySQL
mysql_close ($conexao);
```

Exemplo: Carrinho de Compras com Sessões

Loja X - Firefox

File Edit View Go Bookmarks Tools Help

Cesta de Compras

Item	Descrição	Preço	Quantidade
6	Mouse Ótico	60.00	0
7	PDA marca A	300.00	0
8	PDA marca B	950.00	0
9	Alto-falantes	100.00	1
10	Telefone	50.00	0

Página 2 de 3 [Atualizar](#) [Comprar](#) [Pg. Anterior](#) [Próxima Pg.](#)

Total : R\$2400.00

Exemplo: Carrinho de Compras com Sessões

- O padrão para as linhas da tabela é guardado em uma variável global:

```
$linhaTabela = <<<FIMLINHA
<tr>
  <td>%d</td>
  <td>%s</td>
  <td align="right">%.2f</td>
  <td><input name="qtd[%d]" value="%d"
            size=2></td>
</tr>
FIMLINHA;
```

Exemplo: Carrinho de Compras com Sessões

- Gerando as linhas da tabela:

```

function formata_linhas_tabela ($consulta,$pag,&$qtd) {
    // Avanço até o registro com primeira linha da pág
    mysql_data_seek($consulta, $pag * LinhasPorPag);
    // Variável com código html de uma linha da tabela
    global $linhaTabela;
    // Gero LinhasPorPag linhas no máximo
    for ($n = 0; $n < LinhasPorPag &&
        ($linha=mysql_fetch_array($consulta)); $n++) {
        // Dados obtidos da consulta
        $descricao = $linha["descricao"];
        $preco = $linha["preco"]; $id = $linha["id"];
        // Quantidades no array $qtd
        $q = IsSet($qtd[$id]) ? $qtd[$id] : 0;
        // Emito html substituindo campos em $linhaTabela
        echo sprintf ($linhaTabela, $id, $descricao,
                     $preco, $id, $q);
    }
}

```

Exemplo: Carrinho de Compras com Sessões

- Gerando os botões de submissão do formulário:

```

function formata_botoes ($numPg, $pagina) {
    // Número da página e total de páginas
    echo "<br> Página ".($pag+1)." de ".$numPg." ".
    // Botões atualizar e comprar
    '<button value="atualizar" name="submit">Atualizar' .
    '</button>  ' .
    '<button value="comprar" name="submit">Comprar' .
    '</button>  ' ;
    // Gero botão Pg anterior se não estou na 1a pag
    if ($pag > 0)
        echo '<button value="prev" name="submit">' .
        'Pg. Anterior</button>  ' ;
    // Gero botão Próxima Pg se não estou na ultima pg
    if ($pag < $numPg-1)
        echo '<button value="prox" name="submit">' .
        'Próxima Pg.</button>' ;
    echo "<br>\n";
}

```

Exemplo: Carrinho de Compras com Sessões

- Calcular e imprimir o total de compras:

```
function formata_total ($consulta, $qtd) {  
    // Retorno ao inicio da consulta de produtos  
    mysql_data_seek($consulta, 0);  
    // Somo o preço de cada produto multiplicado  
    // pela quantidade  
    $total = 0;  
    while (($linha = mysql_fetch_array($consulta))) {  
        $id = $linha["id"];  
        $total += $linha["preco"] * $qtd[$id];  
    }  
    // Escrevo total com duas casas decimais  
    echo "<br> Total : R\$" . sprintf ("%,.2f", $total)  
        "<br>\n";  
}
```