

Sumário

1	Programação para a Web	2
1.1	O que é SERVER-SIDE?	2
1.2	Introdução ao PHP	2
1.3	O que é necessário para executar o PHP?	3
2	Comandos Básicos	3
2.1	Operadores Lógicos e Matemáticos	4
2.2	Criando uma calculadora interativa através do método GET	6
2.3	Formulário HTML	7
3	Integrando PHP com MySQL	8
3.1	O que é MySQL?	8
3.2	Estabelecendo uma conexão	9
3.3	Criando um Sistema de Noticias	10
3.3.1	Inserindo Noticias	10
3.3.2	Listando as Noticias	11
3.3.3	Excluindo Noticias	12
3.3.4	Alterando Noticias	13
3.3.5	Finalizando o sistema de Noticias	15
4	Exercicios	16
5	Referências Bibliográficas	16

Lista de Códigos

1	“Oi Mundo” e Hora do Servidor	3
2	Variáveis	4
3	Laço de repetição for	4
4	Estruturas de controle e operadores lógicos	5
5	Uma Calculadora	6
6	Formulário para a Calculadora	8
7	Formulário para inserir Noticia	10
8	Código para inserir Noticia	11
9	Código que gera uma lista com as ultimas 10 noticias	12
10	Excluindo Noticias	12
11	Formulário para alterar Noticia	13
12	Código de alteração de noticia	14
13	Exibindo a lista de noticias no site	15
14	Exibindo noticia no site	15

1 Programação para a Web

A estrutura de um Web site é contruída através de códigos HTML. O HTML é uma linguagem de marcação de texto, ela descreve a formatação que o documento possuirá.

HTML não é uma linguagem de programação, pois não realiza nenhum tipo de processamento sobre dados de entrada fornecidos pelos usuários. Os códigos HTML são estáticos, Para suprir a necessidade do dinamismo, diversas linguagens de programação são usadas em conjunto com o HTML para transformar um Web site em um objeto que possua interatividade e dinamismo.

Dentre algumas destas linguagens podemos mencionar como as mais utilizadas:

1. **JavaScript:** linguagem de programação que usa o próprio *Browser* do usuário como interpretador para executar seus códigos, infelizmente, a linguagem é muito dependente do *Browser* em que o Web site é acessado, se o *Browser* não possui funções de interpretação ou esta foi desabilitada o código *JavaScript* não será executado;
2. **PHP:** é uma linguagem interpretada, *open-source*, possui uma sintaxe semelhante a da linguagem C. Ela tem uma série de funções como: criptografia, geração de documentos PDF, ordenação de dados, entre outras, que facilitam a criação de Web sites dinâmicos;
3. **ASP:** é uma linguagem interpretada de propriedade da Microsoft, sendo assim, fortemente ligada ao sistema operacional *Windows*, o que a torna pouco portátil;
4. **JSP:** A tecnologia JSP é uma extensão da classe dos Servlets . Ela foi desenvolvida para facilitar a integração entre códigos Java e HTML.

1.1 O que é SERVER-SIDE?

As linguagens de programação para a Web que possuem seu processamento realizado localmente, ou seja, na máquina do usuário que acessou o site, são conhecidas como *CLIENT-SIDE*. *JavaScript*, *Macros*, *Applets*, fazem parte deste grupo de linguagens.

Em contrapartida, existem linguagens que realizam a etapa de processamento no servidor Web em que estão hospedadas, este tipo de linguagem é conhecida como *SERVER-SIDE*. Estas linguagens ganharam força no últimos anos pois tornam os Web sites verdadeiras aplicações.

Tanto as linguagens *SERVER-SIDE* como *CLIENT-SIDE*, pertencem ao grupo das linguagens de programação interpretadas, elas necessitam de um sistema que receba o código escrito na linguagem específica e o processa de acordo com seu funcionamento interno e retorna ao *Browser* do usuário o resultado.

Como exemplo de tecnologias *SERVER-SIDE* temos o PHP, ASP, JSP, *Servlets*.

1.2 Introdução ao PHP

PHP é uma linguagem de *script* embutida no código HTML. Muito de sua sintaxe foi inspirada em outras linguagens como: *C*, *Java* e *Perl*. O objetivo da linguagem PHP é permitir que desenvolvedores Web construam páginas dinâmicas com certa velocidade.

Por ser uma linguagem *SERVER-SIDE* PHP é independente de plataforma, ou seja, para ser executado um *script* em um servidor web qualquer é necessário que este servidor possua o interpretador PHP, independente de qual o Sistema Operacional que este rode.

Veja um exemplo de um arquivo .php que irá realizar a inserção da frase "Oi Mundo", de uma quebra de linha `
` e data no código HTML.

Código 1: "Oi Mundo" e Hora do Servidor

```
1 <html>
2   <body>
3     <?php echo "Oi Mundo"?>
4     <?php echo "<br>"?>
5     A data atual do servidor é <br>
6     <? echo date('M d Y')?>
7   </body>
8 </html>
```

Percebam que o arquivo é praticamente todo composto por *tags* HTML, com exceção das linhas 3,4 e 6. Todos os comandos em php devem estar entre delimitadores especiais que o identificam, no código 1 foram utilizados dois demarcadores diferentes o `<?php ?>` e a forma contraída `<? ?>`.

Na linha 3 temos o comando **echo** que realiza a escrita dentro do código HTML, neste caso a frase "Oi Mundo" foi escrita, comando em HTML podem ser inseridos dentro de um bloco php desde que seja escrito dentro de um **echo**.

1.3 O que é necessário para executar o PHP?

O PHP é uma linguagem interpretada, sendo assim, para que um arquivo .php seja executado corretamente por um servidor Web, é necessário que este possua instalado o interpretador php.

Atualmente o PHP está na versão 5.0 e pode ser baixado gratuitamente no endereço www.php.net

Uma das combinações mais comuns em servidores Web é a utilização do *Apache* como servidores de HTTP e o PHP como linguagem para a construção de sites dinâmicos.

2 Comandos Básicos

Toda linguagem de programação trabalha com o conceito de variáveis, em PHP alguns pontos devem ser ressaltados.

- A declaração de uma variável acontece com o uso do \$, assim o texto `$x` em PHP realiza a criação de uma variável;
- Para definir um valor a uma variável basta usar o sinal de =, `$x=10`, diz que a variável `x` irá armazenar o valor 10;
- As variáveis não possuem tipo previamente definidos, seu tipo é definido pelo seu conteúdo, em php esta combinação é permitida `$x=10` e logo em seguida `$x="string"`;
- A declaradas de uma variável pode acontecer em qualquer parte do código;
- Seu tempo de vida é igual ao tempo de vida da página;
- PHP é *case sensitive*, acessar a variável `$Arvore` não é igual a acessar a variável `$arvore`.

Veja no código 2 o uso de variáveis em PHP.

Código 2: Variáveis

```
1 <html>
2 <body>
3     <?php
4         $musica = "Moda de Viola";
5         // escreve o valor da variável na tela
6         echo $musica;
7     ?>
8 </body>
9 </html>
```

Na linha 4 foi declarada uma variável cujo nome é *\$musica*, logo em seguida foi atribuída, "*Sinal de atribuição =, Sinal de comparação ==*", uma *string* com o texto "Moda de Viola" a ela.

Na linha 5 observa-se o uso do *//*, este é um dos demarcadores de comentários de uma linha. Para definir várias linhas de comentário usa-se o os demarcadores */* */*.

Na linha 6 aparece o comando **echo**, esta é a função realiza a inserção de algum conteúdo no código HTML de uma página, outra função que pode ser usada para o mesmo fim é o **print**.

Outra operação muito usada na programação é o uso dos laços de repetição, PHP possui basicamente 3 deles:

1. *while (condição verdadeira) { Bloco de instrução}*
2. *do { Bloco de instrução} while (condição verdadeira);*
3. *for (contador; condição verdadeira; variação do contador) { Bloco de instrução}*

Segue no código 3 o uso do laço **for**.

Código 3: Laço de repetição for

```
1 <html>
2 <body>
3     <?
4     for ($i=0; $i<10; $i++) {
5         echo $i;
6         /* procure evitar o uso de HTML junto com os
7            comando em html, prefira fechar a tag do php
8            e reabre depois de usar o HTML
9         */
10        ?><br><?
11    } ?>
12 </body>
13 </html>
```

2.1 Operadores Lógicos e Matemáticos

No PHP há diversos operadores matemáticos que podem ser utilizados, sendo que alguns destes operadores básicos que são conhecidos por qualquer pessoa. O PHP também oferece diversos operadores para o controle de fluxo de dados (também conhecidos como operadores Lógicos).

Observe abaixo a tabela com alguns dos operadores lógicos.

e lógico	and
ou lógico	or
negação	!
ou exclusivo	xor

Veja a seguir a tabela com os operadores de comparação.

igual a	==
diferente de	!=
menor que	<
maior que	>
menor ou igual	<=
maior ou igual	>=

Tabelas com operadores matemáticos

incrementa 1 a	++
decrementa 1	--
soma	+
subtração	-
divisão	/
resto da divisão	%

Outra questão importante quanto ao desenvolvimento de Web sites é o uso dos vetores também conhecidos como *array*. Veja o código 4 para uma visão sobre vetores, operações lógicas e operadores matemáticas.

Código 4: Estruturas de controle e operadores lógicos

```

1 <html>
2 <body>
3 <table border="1">
4 <?
5     // cria um array com os dias da semana
6     $cor = array ("#666666", "#999999");
7     $dia = array (1=> "Domingo", "Segunda", "Terça",
8                   "Quarta", "Quinta", "Sexta", "Sábado");
9     // cont($array) é uma função que retorna a quantidade de elementos no vetor
10    for ($i=1; $i <= count($dia); $i++){
11    ?><tr bgcolor="<? echo $cor[$i%2]?>">
12        <td><?
13            if ($i == 7) {
14                // o . é usado para concatenar duas Strings
15                echo "Nada melhor que um ".$dia[$i];
16            } else {
17                // acessa a posição $i do array $dia
18                echo $dia[$i];
19            }
20        ?></tr></td><?
21    }
22 ?>
23 </table>
24 </body>
25 </html>

```

Na linha 6 podemos observar a criação de um vetor chamado *\$cor* com dois elementos, é importante observar que um vetor por padrão começa a sua indexação a partir do valor 0, a menos que se faça de maneira explícita um pedido para que o primeiro índice seja outro valor como aconteceu na criação do vetor *\$dia* na linha 7, neste o primeiro elemento será indexado pelo número 1.

Em muitos momentos como desenvolvedores Web, necessitamos criar tabela que tenham cores alternadas em suas linhas, um exemplo de como fazer isto está na linha 11. Veja que os comandos em HTML aparecem normalmente, dentro da *tag bgcolor* temos um comando **echo \$cor[*\$i*%2]**, a função **echo** irá escrever no HTML o resto da divisão da variável *\$i* por dois, esta operação pode ter apenas dois resultados, 0 se *\$i* for par ou 1 se *\$i* for ímpar, e é este valor é usado com índice do vetor *\$cor*[], para escolher entre a cor de índice 0 ou 1.

Na linha 13 temos uma comparação entre a variável *\$i* e o número 7, é possível ter várias comparações sendo feitas dentro do mesmo comando *if*, para isto, basta colocar todas dentro de parênteses e uni-las com os operadores lógicos, exemplo `if((a==10) and (b !=1))`. Também é possível utilizar a cláusula *else* como mostra a linha 16, o bloco de comando do *else* sempre será executado quando a verificação do *if* retornar falso.

Para finalizar o estudo deste exemplo temos na linha 15 uma concatenação entre duas variáveis, esta concatenação é feita utilizando o operador "ponto .", note que elas são de tipos diferentes de dados, o PHP faz a conversão, *casting*, automático das variáveis.

2.2 Criando uma calculadora interativa através do método GET

Nesta Seção será criada uma aplicação simples que responderá como uma calculadora comum. Os objetivos desta atividade são: conhecer outra estrutura de seleção, o *switch*, e entender como podemos "pegar" através do PHP variáveis passadas pela URL. segue a aplicação no código 2.4.

Neste exemplo temos nas linhas 2, 3 e 4 a criação de variáveis que receberão seus valores a partir de variáveis passadas através da URL da página. Se a URL passada for, por exemplo, *exemplo5.php?n1 = 10&n2 = 5&op = adicao*, a variável *\$numero1* receber o valor 10, a variável *\$numero2* receberá o valor 5 e a variável *\$op* receberá o valor "adição".

Em situações que o número de opções diferentes é muito grande o uso da instrução *if* torna-se inviável, assim, para solucionar este problema pode-se utilizar a estrutura *switch*, ela se comporta como o *case* do *Pascal* ou o comando de mesmo nome nas linguagens *C* e *Java*. Para cada caso é necessário criar uma estrutura *case* e logo em seguida a comparação que irá ser verificada para a execução do bloco do *case*. É de suma importância que ao final de cada bloco *case* coloque-se o comando *break;*, para que os demais blocos não sejam executados.

Na linha 26 temos o comando *default*, isto dentro de um comando *switch* significa que, se nenhuma das condições estabelecidas anteriormente for tida como verdadeira então o bloco da opção *default* será executada.

A variável *\$flag* é usada como um marcador, uma bandeira sinalizadora sobre o andamento do processamento, em muitos casos faz-se necessária o conhecimento se uma determinada ação aconteceu ou não, a forma mais comum é a de estabelecer uma variável que sinaliza este acontecimento, no nosso caso se *\$flag == 0* então os dados de entrada eram válidos e puderam ser processados, se *\$flag == 1* então ocorreu um erro de divisão por zero e se *\$flag == 2* então a operação passada não era válida.

Código 5: Uma Calculadora

```

2  $numero1 = $_GET["n1"];
3  $numero2 = $_GET["n2"];
4  $operacao = $_GET["op"];
5
6  $resultado = 0;
7  $flag = 0;
8  switch ($operacao) {
9      case $operacao=="adicao":
10         $resultado = $numero1 + $numero2;
11         break;
12     case $operacao=="subtracao":
13         $resultado = $numero1 - $numero2;
14         break;
15     case $operacao=="multiplicacao":
16         $resultado = $numero1 * $numero2;
17         break;
18     case $operacao=="divisao":
19         // verificar se número dois não é zero
20         if ($numero2==0) {
21             $flag = 1;
22         } else {
23             $resultado = $numero1 / $numero2;
24         }
25         break;
26     default:
27         echo "Operação não conhecida";
28         $flag = 2;
29         break;
30 }
31 if ($flag == 1) {
32     echo "Erro de divisão por 0";
33 }
34 if ($flag == 2) {
35     echo "Operação desconhecida use:
36         <br> - adicao <br> - subtração <br>
37         - divisão <br> - multiplicação";
38 }
39 if ($flag==0) {
40     echo "O Resultado é igual a ".$resultado;
41 }
42 ?>

```

2.3 Formulário HTML

Um formulário HTML é uma das formas mais comuns de realizar a interligação entre o usuário e o Web site. Duas coisas são importantes quando se cria um formulário.

1. A *tag action* define qual será o arquivo ou aplicação que irá receber os dados do formulário, este arquivo pode ser uma aplicação CGI, um *Servlet*, um arquivo ASP, no exemplo, os dados serão submetidos a um arquivo PHP.
2. Os dados de um formulário podem ser enviado através de dois métodos o *post* e o *get*, a diferença entre eles é que enquanto o *get* envia os dados através da URL o método *post* anexa os dados contidos nos elementos do formulário ao cabeçalho HTTP.

Código 6: Formulário para a Calculadora

```
1 <html>
2 <form name="calculadora" method="get" action="exemplo5.php">
3 <table>
4   <tr>
5     <td>Número 1</td>
6     <td><input name="n1"></td>
7   </tr>
8   <tr>
9     <td>Número 2</td>
10    <td><input name="n2"></td>
11  </tr>
12  <tr>
13    <td>Operação</td>
14    <td>
15      <select name="op">
16        <option value="adicao">+</option>
17        <option value="subtracao">-</option>
18        <option value="multiplicacao">*</option>
19        <option value="divisao">/</option >
20      </select>
21    </td>
22  </tr>
23  <tr>
24    <td colspan="2"><input type="submit" value="Calcular"></td>
25  </tr>
26 </table>
27 </form>
28 </html>
```

Percebam que no código 6, não existe nenhum comando em PHP, nele estão apenas código HTML, o tratamento dos dados será realizado no arquivo *exemplo5.php* como mostra a *tag action* na linha 2. O método de envio será o *get* também descrito na linha 2.

Quando se utiliza o componente *select* do HTML, o valor que é passado pelo formulário é o que está contido dentro do atributo *value* de cada um dos *option* que compoem o *select*.

Para que nossa aplicação funcione com o envio dos dados através do método *post* teríamos que realizar duas alterações:

1. no arquivo *exemplo6.php* na linha 2 a *method* ficará *method="post"*;
2. no arquivo *exemplo5.php* nas linhas 2, 3 e 4, o comando `$ _get` deve ser substituído por `$ _post`

3 Integrando PHP com MySQL

3.1 O que é MySQL?

O MySQL é um sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL (Structured Query Language - Linguagem de Consulta Estruturada) como interface. É atualmente um dos bancos de dados mais populares, com mais de 4 milhões de instalações pelo mundo.

Principais características do MySQL:

- Portabilidade (suporta praticamente qualquer plataforma atual)
- Compatibilidade (existem drivers ODBC, JDBC e .NET e módulos de interface para diversas linguagens de programação, como C/C++, Python, PERL, PHP e Ruby)
- Excelente desempenho e estabilidade
- Pouco exigente em termos de recursos de hardware
- Facilidade de uso
- É um Software Livre
- Suporte a vários engines (como MyISAM e InnoDB)

O MySQL foi criado na Suécia por dois suecos e um finlandês: David Axmark, Allan Larsson e Michael "Monty" Widenius, que trabalham juntos desde a década de 1980. Hoje emprega aproximadamente 70 profissionais no mundo inteiro, e mais de mil contribuem testando o software, integrando-o a outros produtos, e escrevendo a respeito do mesmo.

O sucesso do MySQL deve-se em grande medida à fácil integração com o PHP incluído, quase que obrigatoriamente, nos pacotes de hospedagem de sites da Internet oferecidos atualmente. Empresas como Yahoo! Finance, MP3.com, Motorola, NASA, Silicon Graphics e Texas Instruments usam o MySQL em aplicações de missão crítica.

O MySQL hoje suporta Unicode, Full Text Indexes, replicação, Hot Backup, GIS, OLAP e muitos outros recursos.

Outra grande vantagem é a de ter código aberto e funcionar em, quase, qualquer plataforma e sistema operacional : Windows, Linux, FreeBSD, BSDI, Solaris, Mac OS X, SunOS, SGI, etc.

É reconhecido pelo seu desempenho e robustez e também por ser multi-tarefa e multi-usuário. No entanto, devido a não possuir funcionalidades consideradas essenciais em muitas áreas, como stored procedures e two-fase commit, é frequentemente considerado um sistema mais "leve" para aplicações menos exigentes, sendo preterido por outros sistemas como o PostgreSQL.

3.2 Estabelecendo uma conexão

Para que o PHP possa interagir com o MySQL, é necessário realizar uma conexão com o mesmo. Isso é feito através do seguinte comando:

```
MySql_connect(host,login,senha);
```

Essa função recebe como parâmetros o host do MySQL, e o login e a senha para conexão. Esses parâmetros podem ser passados tanto diretamente como em variáveis. Caso o seu servidor Mysql não use a porta e o soquete padrão, é possível especificá-los dentro desta função. (**host:porta:soquete'**)

Após criar a conexão, é necessário selecionar o Banco de Dados onde esta as tabelas com quais o PHP irá interagir. Isso é feito através do comando:

```
MySql_select_bd(database);
```

Este deve ser executado sempre apos estabelecer uma conexão com o MySQL, e recebe como parâmetro o nome do banco de dados.

Usa-se o @ prefixando cada uma dessas duas funções caso for necessário utilizá-las em modo silencioso, ou seja, sem retornar nenhuma mensagem de erro. Ex.: "@MySql_select_bd(database);"

3.3 Criando um Sistema de Noticias

Para melhor entender como o PHP se interage com o MySQL, vamos ilustrar a criação de um sistema de noticia simples, onde poderemos exemplificar os comandos PHP Mysql para a consulta, inserção, alteração e exclusão de dados do Banco de dados.

O PHP passa todas as instruções de manipulação de dados para o MySQL através de códigos em SQL (Structures Query Language), e para isso, usa-se a função `Mysql_Query()`, que tem o trabalho de enviar uma consulta SQL ao Banco de Dados MySQL e recebe como parâmetro uma string contendo o comando SQL a ser executado.

```
Mysql_Query('Select * from tabela');
```

Primeiramente, vamos trabalhar na parte por trás do sistema de noticias, que é a parte onde iremos manipular os dados no banco de dados. A tabela que irá armazenar os dados no banco de dados MySQL é composta pelos campos (todos em minúsculo):

- id: Numero de identificação Único da noticia;
- titulo: Titulo que descreve a noticia;
- corpo: Conteúdo da noticia;
- autor: Autor da noticia;
- data: Data de inserção da noticia;

3.3.1 Inserindo Noticias

Tendo as informações sobre quais informações serão armazenadas dentro da Tabela, podemos então criar um formulário para realizar a inserção de noticias no MySQL. Podemos notar que os dados para os campos id e data não precisam ser lidos de um formulario, portanto, os dados que necessitam ser lidos são: Titulo, Corpo e Autor e, sendo assim, vamos criar um arquivo chamado *inserir_noticia.php*, e dentro dele vamos colocar o seguinte formulário:

Código 7: Formulário para inserir Noticia

```
1 <html>
2 <head>
3   <title>Administração de Noticia – Inserir Noticia</title>
4 </head>
5 <body>
6 <h2> Inserir Noticia </h2>
7   <form name="form1" method="post" action="inserir_noticia.php">
8     <p>Titulo:<br>
9     <input name="not_titulo" type="text" id="not_titulo"></p>
10    <p>Corpo:<br>
11    <textarea name="not_corpo" cols="40" rows="6" id="not_titulo">
12    </textarea></p>
13    <p>Autor:<br>
14    <input name="not_autor" type="text" id="not_autor"></p>
15    <p><input name="not_inserir" type="submit" value="Inserir"></p>
16  </form>
17 </body>
18 </html>
```

Agora com o Formulário de Inserção criado, podemos criar o código que irá pegar os dados do formulário e inseri-los na tabela MySQL. Notem no formulário que a página que ira receber os dados enviados é a mesma em que o formulário esta inserido, portanto, vamos inserir todo o código de inserção de noticias antes do formulário, no mesmo arquivo.

Código 8: Código para inserir Noticia

```
1 <?php
2 $inserir = $_POST['not_inserir'];
3 if (IsSet($inserir)){
4     //Conexão com o Banco de Dados Mysql
5     mysql_connect("localhost","root","");
6     mysql_select_db("curso");
7     $data      = getdate();
8     $not_dia   = $data['mday']+ "/" + $data['mon']+ "/" + $data['year'];
9     $not_titulo = $_POST['not_titulo'];
10    $not_corpo  = $_POST['not_corpo'];
11    $not_autor  = $_POST['not_autor'];
12    $sql_inserir = "INSERT INTO noticia VALUES (0,'$not_titulo','$not_corpo','$not_autor','$not_dia)";
13    if (Mysql_Query($sql_inserir)){
14        print "<h4> Noticia Inserida com Sucesso </h4>\n";
15    }else{
16        print "<h4> Erro ao inserir Noticia </h4>\n";
17    }
18 }
19 ?>
```

Observando o código de inserção, pode-se notar que na linha 5, a variável \$inserir recupera o valor "not_inserir" enviado pelo formulário, e na linha 6 é feita a verificação se ela foi ou não setada, ou seja, se tem algum valor armazenado dentro da variavel ou não, se tiver, significa que o formulário foi enviado (Essa verificação pode ser feita com qualquer variável presente no formulário). Na linha 8 e 9 é feita a conexão com o banco de dados, e logo em seguida temos a variavel \$data recebendo o valor da função **getdate()**; que retorna um array contendo os valores correspondentes a data atual. Na linha seguinte, criamos uma string contendo a data através dos valores armazenados em \$data.

Na linha 15, usamos o comando SQL INSERT para inserir os dados do formulário na tabela. Este comando tem a seguinte sintaxe:

INSERT INTO tabela VALUES (valor1,valor2,..,valorn)

sendo que a ordem dos valores é equivalente a ordem das colunas da tabela onde os dados serão inseridos (no exemplo: id,titulo,corpo,autor,data). No campo id, o valor inserido foi 0, pois este é o campo auto incrementável que identifica unicamente os dados da tabela, assim, ele coloca como valor o numero seguinte ao ultimo da seqüência. O if da linha 17 verifica se a inserção foi realizada, mostrando uma mensagem indicando se houve ou não sucesso na operação.

3.3.2 Listando as Noticias

Agora que já se pode inserir dados na tabela, podemos listar os dados presente dentro dela. Para isso, vamos criar um arquivo chamado "admin.php", este arquivo será a pagina principal da administração do sistema de noticias. O comando SQL que é usado para fazer as consultas é o SELECT, e ele possui a seguinte sintaxe:

SELECT colunas FROM tabela WHERE restrições/operações

onde podemos selecionar as colunas com os dados desejados (usa-se * quando se deseja selecionar todas as colunas), a tabela onde estão os dados e ainda, pode-se filtrar os dados através das restrições, ou realizar operações como por exemplo a ordenação. A seguir temos o código que irá listar as últimas 10 notícias, junto com a opção para excluir e alterar:

Código 9: Código que gera uma lista com as últimas 10 notícias

```
1 mysql_connect("localhost","root","");
2 mysql_select_db("curso");
3 $sql = "Select id,titulo from noticia order by id desc limit 0,10";
4 $executa = Mysql_Query($sql);
5 if (Mysql_num_rows($executa) == 0){
6     print "<h4> Não existe noticias Cadastradas </h4>\n";
7 }else{
8     while ($result = Mysql_fetch_array($executa)){
9         print "$result[titulo]\n";
10    }
11 }
```

Na linha 3 temos o comando SQL para a nossa busca, nele a gente seleciona as colunas id e titulo, que serão necessárias para a exclusão e alteração (coluna id) e também para exibição na tela (coluna titulo). A restrição feita nessa consulta de quantidade máxima de linhas que virá na consulta, através do comando Limit x,y, onde o x é ordem do primeiro registro a ser selecionado e y é a quantidade de registros seguintes. Também realizamos a operação que ordena os registros segundo o campo id através do comando **order by id**, também indicamos que é necessário que a ordenação seja feita em ordem decrescente, para que a listagem comece da última notícia inserida, então, inserimos o comando **desc**.

Na linha 4 realizamos a consulta no banco de dados MySQL e guardamos o resultado na variável \$resultado. Na linha seguinte, verificamos a quantidade de linhas retornadas pela consulta através do comando **Mysql_num_rows(\$executa)**, e não tiver retornado nenhum resultado, exibe-se na tela uma mensagem de aviso. Porém, se encontrar uma ou mais linhas que satisfaça a consulta, realiza-se um while, armazenando na variável \$result o comando **Mysql_fetch_array(\$executa)**, que retorna um array associativo com uma linha de resultado a cada iteração até não tenha mais nenhum resultado a retornar. Assim, na linha 9, escreve-se os valores contidos dentro de \$result no documento HTML.

3.3.3 Excluindo Notícias

O comando DELETE do SQL é o responsável por realizar as exclusões na tabela do Banco de Dados e sua sintaxe é:

DELETE FROM tabela WHERE restrições

No caso do nosso sistema de notícias, temos que usar uma restrição que nos permita excluir apenas uma notícia por vez, ou seja, que nos dê a certeza de que não estamos excluindo notícias indesejadas. Portanto, usaremos o campo id como restrição, levando que cada notícia tem um id diferente. Temos então o seguinte código, que deve ser inserido no arquivo admin.php após a conexão com o Banco de Dados e Antes do código de listagem de notícias:

Código 10: Excluindo Notícias

```
1 $excluir = $_GET['excluir'];
2 if (IsSet($excluir)){
```

```

3  $sql_excluir = "Delete from noticia where id = '$excluir'";
4  if (Mysql_Query($sql_excluir)){
5      print "<h4> Noticia Excluida com sucesso </h4>\n";
6  }else{
7      print "<h4> Erro ao excluir noticia </h4>\n";
8  }
9  }

```

O id da noticia a ser excluída é passado por método GET, e armazenado na variável \$excluir. Tendo essa variável, e ela estando setada, armazenamos na variável \$sql_excluir o comando SQL que irá excluir na tabela os valores onde o campo id seja igual ao id passado como parâmetro e armazenado em \$excluir. Na linha 4 realiza-se a operação no MySQL retornando para o usuário uma mensagem indicando o erro ou o sucesso da exclusão.

Falta ainda adequar a listagem das noticias para que, através dela, possamos escolher qual noticia deve ser excluída. Para isso, adicionaremos um link para a pagina contendo os comandos de exclusão na frente do titulo da noticia que vai ser exibido:

```
<a href="admin.php?excluir=$result[id]">Excluir</a>
```

3.3.4 Alterando Noticias

A ultima instrução de manipulação de dados do SQL que nos falta ver é a instrução UPDATE, que nos permite realizar alterações no conteúdo dos dados presentes dentro da tabela. Sua sintaxe é simples, e segue o mesmo padrão dos outros comandos SQL:

UPDATE tabela SET coluna1=valor1 ,... , colunan=valorn WHERE restrições

Esse comando é muito flexível, pois nos permite alterar o valor de qualquer numero de colunas e também, dependendo da restrição, pode-se alterar mais de uma linha da tabela. No exemplo a seguir temos uma alteração de preço para todos os pertencentes a categoria 'A':

UPDATE catalogo SET preco=10 WHERE categoria = 'A'

Em nosso sistema de noticia os únicos dados que podem sofrer alterações são os que foram inseridos pelo usuário (titulo, corpo e autor), assim, o formulário de alteração necessita ter somente esses campos. Vamos então criar um novo documento, que vamos chamar de alterar_noticia.php, e nele vamos inserir o seguinte código:

Código 11: Formulário para alterar Notícia

```

1  mysql_connect("localhost","root","");
2  mysql_select_db("curso");
3  $id_alterar = $_GET['alterar'];
4  if (IsSet($id_alterar)){
5      $sql = "Select id,titulo,corpo,autor from noticia where id='$id_alterar'";
6      $acao = Mysql_Query($sql);
7      if (Mysql_num_rows($acao) == 0){
8          print "<h4> Voce precisa selecionar uma noticia Válida </h4>\n";
9      }else{
10         $resultado = Mysql_fetch_array($acao);
11     ?>
12
13     <form method="post" action="alterar_noticia.php?alterar=?php print $id_alterar;
14         ?>">
15         <p>Titulo:<br>
16         <input name="not_titulo" type="text" value="<?php print $resultado['titulo
17             '];?>">

```

```

16 </p>
17 <p>Corpo:<br>
18 <textarea name="not_corpo" cols="40" rows="6">
19 <?php print $resultado['corpo'];?>
20 </textarea>
21 </p>
22 <p>Autor:<br>
23 <input name="not_autor" type="text" value="<?php print $resultado['autor
24 >'];?>">
25 </p>
26 <input name="not_id" type="hidden" value="<?php print $resultado['id']; ?>"
27 >
28 <p><input name="not_alterar" type="submit" id="not_alterar" value="Alterar">
29 </p>
30 </form>
31 <?php
32 }
33 }else{
    print "<h4> Voce precisa selecionar uma noticia Válida </h4>\n";
}

```

Este código recebe o numero de identificação, e cria um formulário para realizar a alteração com os valores atuais da noticia que se deseja alterar, sendo que, se passarmos o id inexistente de uma noticia, mostra uma mensagem de erro. Notemos que o comando `Mysql_fetch_array` é usado fora de um while, isso porque se deseja recuperar apenas uma linha da tabela e exibir no formulário. Com os dados atuais dentro do formulário, fica mais fácil realizar as alterações necessárias da noticia. Podemos observar que na linha 23 do formulário temos um campo escondido hidden, ele é necessário para que possamos saber na hora da alteração qual noticia estaremos atualizando. Outro aspecto importante que é bom observar, é a alternância entre código PHP e HTML realizado na criação do formulário, onde, é bom ressaltar que todo o conteúdo do formulário HTML esta dentro de um comando if, e pode acontecer do formulário não ser exibido caso o if não seja satisfeito.

O código que recebe uma noticia e realiza a alteração no banco de dados é muito semelhante ao de inserção. Vamos inseri-lo no mesmo documento `alterar_noticia.php`, antes do formulário.

Código 12: Código de alteração de noticia

```

1 $acao_alterar = $_POST['not_alterar'];
2 if (IsSet($acao_alterar)){
3     $not_titulo = $_POST['not_titulo'];
4     $not_corpo = $_POST['not_corpo'];
5     $not_autor = $_POST['not_autor'];
6     $not_id = $_POST['not_id'];
7     $sql_alterar = "Update Noticia Set titulo='$not_titulo',corpo='$not_corpo',
8     autor='$not_autor' Where id='$not_id'";
9     if (Mysql_Query($sql_alterar)){
10        print "<h4> Noticia Alterada com Sucesso </h4>\n";
11    }else{
12        print "<h4> Erro ao Alterar Noticia </h4>\n";
13    }
}

```

A restrição usada no comando SQL (linhas 7/8) é o numero de id da noticia, já que, cada noticia possui um numero de identificação diferente e único. Devido ao fato de não saber qual

campo foi alterado pelo usuário, o comando SQL atualiza todos os campos em que é possível fazer alteração pelo formulário. Na linha 9 é executado o comando no MySQL e é feita uma verificação de possíveis erros.

3.3.5 Finalizando o sistema de Noticias

Agora, temos a base de um sistema de noticias completo. Falta ainda a parte de exibição das noticias no site, que é um trabalho muito semelhante ao de criar o formulario de alteração, mudando somente a forma de exibir as informações. Vamos então criar um arquivo chamamo noticia.php, e nele vamos colocar o seguinte código:

Código 13: Exibindo a lista de noticias no site

```
1 <?php
2 mysql_connect("localhost","root","");
3 mysql_select_db("curso");
4 $id = $_GET['id'];
5 $sql = "Select * from noticia where id='$id'";
6 $acao = Mysql_Query($sql) or die("Não foi possivel conectar com o servidor!");
7     if (Mysql_num_rows($acao) == 0){
8         print "<h4> Erro ao exibir noticia</h4>\n";
9     }else{
10         $result = Mysql_fetch_array($acao);
11         $corpo = nl2br($result[corpo]);
12     ?>
13     <p><?php print "$result[data];?></p>
14     <p align="center"><?php print "$result[titulo]"; ?></p>
15     <p align="justify"><?php print "$result[corpo]"; ?></p>
16     <p><?php print "$result[autor]";?></p>
17 <?php
18     }
19 ?>
```

O comando SQL recebe como restrição id=\$id, que ira retornar todas os registros que tenham o id igual ao numero armazenado dentro da variavel \$id, como o campo garante que não existe dois registro com o mesmo numero id, então o MySQL irá retornar nessa consulta 0 ou 1 registros. Note que na linha 6 temos uma outra forma de identificar que houve um erro na hora de realizar a consulta no MySQL, que é através do comando or que chama a função die caso a consulta não seja realizada com sucesso, a função die é reponsavel por encerrar o processamento da pagina retornando uma mensagem ao usuario. Na linha 11 tambem fazemos um tratamento conteudo da noticia obtido na consulta. A função nl2br() utilizada troca todos as quebras de linha dentro do texto por tags de quebra de linha HTML (
).

Para finalizar, falta a listagem das noticias que ira chamar a pagina noticia.php. Esse código é praticamente igual ao codigo que lista as ultimas 10 noticias na pagina de administração, a unica linha que iremos alterar, é a linha que ira escrever na tela o conteudo da consulta. Vejamos o codigo com a linha alterada a seguir:

Código 14: Exibindo noticia no site

```
1 mysql_connect("localhost","root","");
2 mysql_select_db("curso");
3 $sql = "Select id,titulo from noticia order by id desc";
4 $executa = Mysql_Query($sql);
5 if (Mysql_num_rows($executa) == 0){
```

```
6     print "<h4> Não existe noticias Cadastradas </h4>\n";
7 }else{
8     while ($result = Mysql_fetch_array($executa)){
9         print "<a href=\"noticia.php?id=$result[id]\">$result[titulo]</a>\n";
10    }
11 }
```

Assim, concluímos este simples sistema de notícias. A partir dos conceitos básicos apresentados nesta apostila, pode-se criar uma variedade muito grande de aplicações para a Internet.

4 Exercícios

1. Crie um vetor com 10 posições, atribua valores a estes elementos e faça o cálculo da média aritmética, se ela for menor do que 7 escreva turma reprovada, se maior ou igual a 7 escreva turma aprovada.
2. Crie uma página com um formulário com perguntas, as respostas devem estar dentro de select, faça um outro arquivo que pegue as respostas dadas e diga se ele acertou ou errou a pergunta. Princípio básico do Quiz

5 Referências Bibliográficas

- CHOI, Wankyu; KENT Allan; LEA Chris; PRASAD; GANESH; ULMAN Chris, Beginning PHP 4 ed Wrox Press 2000
- CONVERSE, Tim; PARK, Joyce; PHP4 - A Bíblia ed, Campus, 2003
- WikiPedia - <http://pt.wikipedia.org/wiki/MySQL> - Acessado em 29/08/2005
- PHP.net - <http://www.php.net> - Acessado em 29/08/2005