

CRIAÇÃO DE SITES EM

PHP

Programação em PHP

- Principais comandos
- Uso de arrays, operadores e variáveis
- Estruturas de controle
- Criação de classes

Edson J. R. Lobo

Recursos de interatividade

- Programação do banco de dados em MySQL
- Criação de tabelas
- Alimentação de tabelas com páginas de administração
- Recuperação de registros
- Criação de áreas restritas no site
- Imagens automáticas

Criação de um site completo na prática

- Sistema de login
- Desenvolvimento de menus
- Sistema de notícias
- Inclusão de conteúdo



DADOS DE COPYRIGHT

Sobre a obra:

A presente obra é disponibilizada pela equipe [Le Livros](#) e seus diversos parceiros, com o objetivo de oferecer conteúdo para uso parcial em pesquisas e estudos acadêmicos, bem como o simples teste da qualidade da obra, com o fim exclusivo de compra futura.

É expressamente proibida e totalmente repudável a venda, aluguel, ou quaisquer uso comercial do presente conteúdo

Sobre nós:

O [Le Livros](#) e seus parceiros disponibilizam conteúdo de domínio público e propriedade intelectual de forma totalmente gratuita, por acreditar que o conhecimento e a educação devem ser acessíveis e livres a toda e qualquer pessoa. Você pode encontrar mais obras em nosso site: [LeLivros.link](#) ou em qualquer um dos sites parceiros apresentados [neste link](#).

"Quando o mundo estiver unido na busca do conhecimento, e não mais lutando por dinheiro e poder, então nossa sociedade poderá enfim evoluir a um novo nível."



CRIAÇÃO DE SITES EM

PHP

Programação em PHP

- Principais comandos
- Uso de arrays, operadores e variáveis
- Estruturas de controle
- Criação de classes

Edson J. R. Lobo

Recursos de interatividade

- Programação do banco de dados em MySQL
- Criação de tabelas
- Alimentação de tabelas com páginas de administração
- Recuperação de registros
- Criação de áreas restritas no site
- Imagens automáticas

Criação de um site completo na prática

- Sistema de login
- Desenvolvimento de menus
- Sistema de notícias
- Inclusão de conteúdo



C R I A Ç Ã O D E S I T E S E M

PHP



© 2007 by Digerati Books

Todos os direitos reservados e protegidos pela Lei 9.610 de 19/02/1998.

Nenhuma parte deste livro, sem autorização prévia por escrito da editora, poderá ser reproduzida ou transmitida sejam quais forem os meios empregados: eletrônicos, mecânicos, fotográficos, gravação ou quaisquer outros.

Diretor Editorial

Luis Matos

Revisão

Sirlene Farias

Assistência Editorial

Monalisa Neves

Erika Sá

Diagramação

Rogério Chagas

Edição de Arte

Daniele Fátima

Capa

Marcos Mazzei

Preparação dos originais

Cristine Vecchi

Dados Internacionais de Catalogação na Publicação (CIP)
(Câmara Brasileira do Livro, SP, Brasil)

L799c Lobo, Edson Junio Rodrigues.

Criação de sites em PHP / Edson Junio Rodrigues Lobo. São Paulo Digerati Books, 2007.
96 p.

ISBN 978-85-60480-40-1

1. PHP (Linguagem de programação de computador). 2. Sites da Web. I. Título.

CDD 005.13

Universo dos Livros Editora Ltda.

Rua Tito, 1.609

CEP 05051-001 • São Paulo/SP

Telefone: (11) 3648-9090 • Fax: (11) 3648-9083

www.universodoslivros.com.br

e-mail: editor@universodoslivros.com.br

Conselho Administrativo: Alessandro Gerardi, Alessio Fon Melozo, Luis Afonso G. Neira, Luis Matos e William Nakamura.

Sobre o autor

É desenvolvedor de aplicações desde 1999. Iniciou seu conhecimento em micro informática em 1993 na linguagem CLIPPER 4.1. Desde então, adquiriu conhecimento vasto em várias outras linguagens de programação como Visual Basic, Delphi, Visual Fox Pro, Java, PHP, entre outras.

Possui vasto conhecimento em metodologia de desenvolvimento de sistemas e utiliza como ferramenta as linguagens Delphi, Visual Basic, Java, PHP e SQL.

Ministra aulas e consultoria em informática de nível avançado envolvendo variadas tecnologias e sistemas de informação.

Graduado em Tecnologia em processamento de dados pela Faculdade Anhanguera (Centro Universitário Goiás-Anhanguera), Goiânia - Go.

Autor de diversas aplicações de gerenciamento integrado para automação comercial e de uso empresarial.

Parte 1

Introdução

Para quem é este livro?

Caro (a) leitor (a). Você já deve ter pensado um dia: "Será que este livro foi escrito para mim?".

Para que o estudo dos fundamentos apresentados seja bem utilizado, é necessário que você conheça HTML básico, pois esta é a linguagem que utilizaremos para formatar nossas páginas.

Todos os recursos apresentados nesta obra serão cuidadosamente explicados de forma clara e fácil, desde o nível mais básico até o avançado. Ter conhecimento de lógica de programação é muito útil, assim como da linguagem SQL que será utilizada para manutenção do banco de dados. Com esses detalhes em dia, você estará pronto para começar a programar em PHP.

Começaremos apresentando exemplos simples e depois, trabalharemos em um exemplo completo com PHP, que é uma linguagem desenvolvida especialmente para Internet, muito objetiva e eficaz.

Para que você tenha melhor aproveitamento do que vamos estudar, trabalharemos com uma prática muito objetiva e que representa a vida real de um programador Web.

O mundo WWW

Um dos grandes veículos de comunicação que temos hoje é a Internet. Podemos encontrar nela grande parte daquilo que está no nosso dia-a-dia, como lojas, mídias, marketing, artes visuais, escolas, suporte técnico, correspondências - os famosos e-mails - e, principalmente, muita informação.

A Internet está cada vez mais presente em nossa vida atual. Observamos também que este veículo é de grande e fácil acesso pela maioria das pessoas.

Além de tudo, trabalhar pela Internet significa trabalhar para o mundo, pois tudo o que publicamos em um servidor Web pode ser acessado por qualquer computador da rede. Por esse e outros motivos, a Internet é visivelmente fascinante!

Você, que deseja desenvolver aplicativos para esse mundo, sabe que todo trabalho produzido poderá ser visto mundialmente. Quanto mais conhecimento técnico, teórico e atualizado, melhor será o seu trabalho de desenvolvedor. Vale lembrar que o internauta

pretende encontrar sites bem elaborados visualmente, com design objetivo e principalmente, um ambiente que funcione, ou seja, que possa ser explorado sem interrupções com mensagens de erro.

Se você gosta de desafios, dinamismo e muita tecnologia, o desenvolvimento Web lhe dará tudo isso e muito mais.

O desenvolvimento Web

O interesse dos programadores pelo desenvolvimento para Internet já atinge um nível muito alto desde que o uso da Web se popularizou. Isso ocorre, em grande parte, pela complexidade desses sistemas e pelo envolvimento cada vez maior de tecnologias variadas.

Os ambientes programados na Internet aumentam a cada dia. Um site comum necessita hoje de programação feita em uma linguagem como o PHP, que facilita a vida tanto para o usuário como para o administrador do site. Imagine uma página de notícias a qual, a cada momento, deva existir um registro novo para ser visto pelos usuários do site. Se este não for programado, o webmaster e sua equipe terão de editar o texto HTML constantemente para manter o site atualizado. Isso seria terrível!

Utilizando o PHP, podemos criar um banco de dados no qual cada notícia estará cadastrada em uma tabela de notícias. Com isso, a empresa pode ter pessoas exclusivamente responsáveis para atualizar esses dados, com tela de administração específica para tal finalidade. A partir de então, a equipe de programação poderá focar seus trabalhos em outra parte do sistema.

Outra grande importância de um ambiente programado é que uma linguagem server-side, como o PHP, roda do lado do servidor e encapsula o código-fonte. Essas linguagens só retornam o HTML puro, ou seja, só envia para o navegador o que o programador permitir e o conteúdo restrito aos usuários fica gravado no banco de dados.

Parte 2

Comandos do PHP

Consideramos que o leitor está pronto para trabalhar com PHP. Para isso, é necessário que se tenha um documento em HTML puro, podendo ser mesmo um documento de seu site. Para que não utilize ainda seus documentos, criaremos vários exemplos.

Veremos a seguir, os comandos do PHP e como eles podem ser utilizados em conjunto com o HTML. A intenção deste livro é fornecer ao leitor a estrutura da linguagem PHP e apresentar técnicas de desenvolvimento Web.

Notaremos que o desenvolvimento para Internet exige uma certa criatividade, sendo que os recursos disponibilizados pelo servidor não são e, por uma questão de segurança, não haveriam de ser iguais aos das máquinas locais.

Todo documento PHP será salvo com a extensão PHP. Isto fará com que o servidor entenda que esse é um documento de script em PHP.

Para que um site seja disponibilizado na Internet, é necessário que seja criado em um provedor de acesso uma conta na qual você colocará os documentos HTML e PHP para que o mundo todo acesse. Esses documentos são enviados por um programa de FTP que pode ser encontrado em vários sites de download.

Procure fazer uma conta em um provedor que disponibilize suporte a PHP. Normalmente, esses sistemas são em Linux e você poderá utilizar o Linux, Windows ou qualquer outro sistema para enviar seus documentos via FTP para o provedor.

Os delimitadores do script PHP

Delimitadores do script PHP são tags colocadas em qualquer parte do HTML para que o interpretador possa identificar o que realmente é código PHP e, a partir daí, executá-lo.

As linguagens de script são utilizadas em conjunto com o HTML. Existem vários programas que permitem manipular códigos PHP, mas aqui utilizaremos um editor de texto simples, que nos permitirá perfeitamente fazer isso. O objetivo deste livro é apresentar PHP e então, depois de adquirir uma certa experiência, você mesmo poderá procurar outras ferramentas para edição de códigos PHP.

Será possível utilizar qualquer uma das quatro sintaxes de delimitadores apresentadas a seguir, lembrando que a quarta normalmente é a preferida pelos desenvolvedores. Esse formato será utilizado nos próximos exemplos. Veja quais são elas.

```
<?php  
    comandos;  
?>  
<script language="php">  
    comandos;  
</script>  
<%  
    comandos;  
%>  
<?  
    Comandos;  
?>
```

Variáveis

Se você já utilizou uma linguagem de programação, deve ter ouvido falar em variável.

As variáveis de sistema são endereços de memória representados por um identificador, que podemos até mesmo chamar de nome. Estes nomes permitem que identifiquemos a variável para guardar e depois recuperar um valor qualquer.

Uma vez criada a variável, podemos armazenar nela algum dado que poderá ser utilizado ao longo do sistema.

As variáveis no PHP são definidas por um \$, seguido por uma seqüência de caracteres que pode ser iniciada por qualquer letra do alfabeto ou com o caractere - (underline).

Sendo o PHP case sensitive, você deve ter cuidado com a caixa alta, pois, para o PHP, \$Ontem é totalmente diferente de \$ONTEM.

A seguir, veremos um exemplo de criação de variáveis em PHP. E depois, você verá como utilizá-las.

```
<?  
    $paragrafo = "Trabalhando com PHP";  
?>
```

O comando echo

O comando echo é um dos mais utilizados em sistemas PHP. É ele o responsável por enviar dados para a tela.

A seguir, está um documento que será utilizado como exemplo. Poderá ser chamado de index.php. A estrutura dele é de um HTML. Será utilizado dessa forma para ilustrar como o PHP será induzido dentro de um HTML. Suponhamos que você já tenha um site todo em HTML e deseje incluir agora nele o PHP. Não será necessário criar outro site, incrementá-lo

com o PHP pode ser o suficiente. É para isso que existem as linguagens de script.

Para ilustrar o uso do comando echo, utilizaremos a variável criada no exemplo anterior. Então, o arquivo index.php ficará da seguinte forma:

```
<HTML>
  <HEAD>
    <TITLE>Primeiro exemplo em PHP</TITLE>
  </HEAD>
  <BODY>
    <font color="#909090" face = "Lucida Console">
      <h2>
        <?
          $paragrafo = "Trabalhando com PHP";
          echo '<p align="center">';
          echo $paragrafo;
          echo '</p>';
        ?>
      </h2>
    </font>
  </BODY>
</HTML>
```

Este comando permite enviar tags HTML para o browser. Basta colocar as tags no conteúdo de uma variável e enviar para o browser através do comando echo. Quando utilizar o banco de dados, você perceberá a importância deste comando e verá, na prática, como o uso dele é intenso na programação com PHP.

Veremos agora, a explicação dos comandos.

- echo '<p align="center">': o comando echo envia para o browser a seqüência de caracteres '<p align="center">'. Quando o browser recebe essa String, reconhece-a como uma tag HTML, como de fato é;
- echo \$paragrafo: envia o valor da variável \$parágrafo para o browser que, no caso, é "Trabalhando com PHP";
- echo '</p>': o browser recebe uma tag HTML e faz o tratamento dela.

Rode esse documento em um servidor que tenha suporte a PHP e veja o resultado. Este é o início de nossos estudos em PHP.

Comentários

Quando estamos trabalhando em programação, necessitamos colocar alguma anotação junto ao código para que outras pessoas que fazem parte do desenvolvimento possam entender

aquele trecho do código.

Se colocarmos qualquer palavra que não possa ser interpretada pela linguagem corretamente, ocorrerá erro no sistema. Para que isso não ocorra, o PHP tem alguns comandos que são colocados antes e/ou depois desses textos, que permitirão ao interpretador descartar as linhas onde estão os comentários. Os comandos mais utilizados são // para comentários de uma linha e/* */ para um comentário de várias linhas.

Podemos verificar como isso ficaria em um trecho do exemplo anterior:

```
<?
    $paragrafo = "Trabalhando com PHP"; // Exemplo
        do comando echo.
    /*
        A variável anterior está sendo criada com um valor
        que será passado ao browser!
        As "" representam que este valor é uma sequência
            de caracteres.
    */
?>
```

Nesse exemplo, o interpretador descartará o que está logo após o // e também o que está entre /* */.

Arrays

Quando falamos de array, estamos falando de uma matriz de variáveis. Matriz nada mais é que uma estrutura encadeada de alguma coisa, em nosso caso, de variáveis.

As arrays podem ser unidimensionais ou multidimensionais. No PHP são definidas da seguinte forma:

```
<?
    $frase[1] = "Trabalhando ";
    $frase[2] = "com ";
    $frase[3] = "PHP";
?>
```

Para prática, podemos substituir no exemplo anterior a variável \$paragrafo por uma seqüência de comandos echo que utilizará elementos de uma matriz para escrever a mesma frase na tela.

```
<?
    echo $frase[1];
    echo $frase[2];
    echo $frase[3];
?>
```

No código anterior, o conteúdo do elemento 1 da matriz frase será enviado pelo comando echo, e depois, o conteúdo dos elementos 2 e 3, sucessivamente.

Podemos substituir o trecho de código em nosso index.php. O resultado final será o mesmo, o único detalhe é que agora estamos trabalhando com array.

```
<?
$paragrafo = "Trabalhando com PHP";
echo '<p align="center">';
echo $paragrafo;
echo '</p>';
?>
```

O código anterior agora ficará assim:

```
<?
$frase[1] = "Trabalhando ";
$frase[2] = "com ";
$frase[3] = "PHP";
echo $frase[1];
echo $frase[2];
echo $frase[3];
?>
```

Considerando a teoria das matrizes, podemos representar graficamente o exemplo anterior pela Tabela 2.1.

Trabalhando	com	PHP
-------------	-----	-----

Tabela 2.1: Uma array unidimensional.

Operadores

Aritméticos

Como conhecemos da matemática, os operadores aritméticos são utilizados em operações matemáticas e no PHP são utilizados em conjunto com os tipos integer ou float.

Uma coisa muito importante em PHP é a possibilidade de utilizar uma operação aritmética com valores integer, float ou até mesmo Strings. Levando em conta que estamos programando com textos, essa facilidade pode simplificar bastante o código.

Os operadores aritméticos são:

+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Módulo

Tabela 2.2

A seguir alguns exemplos de operações aritméticas:

```
<?
$vlMercadoria = 1550;
$vlDesconto = 138;
$vlPagar = $vlMercadoria - $vlDesconto;

echo '<table border="1" width="400">';
echo '<tr><td width="300">Valor da mercadoria</td>';
echo '<td>'.$vlMercadoria.'</td></tr>';

echo '<tr><td width="300"> Valor do desconto</td>';
echo '<td>'.$vlDesconto.'</td></tr>';

echo '<tr><td width="300"> Total a pagar </td>';
echo '<td><b>'.$vlPagar.'</b></td></tr>';

echo '</table>';
?>
```

Em programação, necessitamos somar (concatenar) strings a todo momento, seja para montar um comando ou para mostrar informações ao usuário. No PHP é possível encontrar um operador de concatenação de strings que é o. (ponto).

Veja a tabela a seguir:

.	Adição
---	--------

Tabela 2.3

Podemos verificar que o exemplo anterior usa este recurso.

- '<td>' \$vlPagar: esta seqüência soma (junta) a string '<td>' com o valor atual da variável \$vlPagar. Tudo isso é enviado para o browser.

Relacionais

Os operadores relacionais são utilizados pela linguagem para fazer comparações entre valores. Isso é utilizado para criar condição verdadeira ou falsa, um recurso em linguagem de programação muito utilizado e que serve para tomar decisões no fluxo do código.

A seguir, é possível observar uma tabela com os operadores relacionais do PHP.

==	Igual a
>	Maior que
<	Menor que
>=	Maior ou igual a
<=	Menor ou igual a
!=	Diferente (Não igual)

Tabela 2.4

Vejamos alguns exemplos que utilizam os operadores mostrados. No código a seguir utilizaremos o comando estrutural if. Esse será explicado detalhadamente no próximo capítulo, utilizaremos aqui para que possamos exemplificar o uso dos operadores relacionais e lógico.

```
<?
$Mercadoria = 1557;
$Desconto = 625;
$Porcentual = $Desconto / $Mercadoria;
if ($Porcentual > 0.30)
{
    echo 'O desconto deve ser até 30%!';
}
else
{
    echo 'Ok, desconto permitido!';
}
?>
```

Teste o código anterior e veja qual será o trecho de código executado.

Ao fazer a comparação `$Porcentual > 0.30`, criamos uma condição para executar o primeiro trecho de código entre `{}`. Caso a condição anterior seja falsa, ou melhor, o desconto for menor que 30%, o segundo trecho de código entre `{}` é que será escolhido pelo sistema.

Se você trocar o valor da variável `$Desconto`, poderá verificar outros resultados.

Esse é um exemplo simples de utilização de operadores lógicos. Utilize o exemplo anterior

e modifique-o utilizando outros operadores, assim, treinará outras formas de fazer comparações, pois essa é uma das operações básicas do computador.

Nesta parte do livro, utilizaremos exemplos pequenos e individuais. Já na terceira parte, modificaremos um site anteriormente criado em HTML puro e faremos a programação necessária para adicionar nele um sistema que permitirá controlar os dados compartilhados nas páginas, com a utilização de banco de dados, administração remota, áreas restritas com login de usuários e páginas dinâmicas.

Lógicos

Estes operadores são utilizados para juntar dois ou mais resultados de operações de comparação. Digamos que queremos testar duas comparações ao mesmo tempo e queremos juntar os resultados (verdadeiro ou falso) destas comparações. É aí que entram os operadores lógicos. Veja o quadro com estes operadores, e teste o exemplo a seguir:

and	"e"
or	"ou"
!	"Não"
xor	"ou" exclusivo

Tabela 2.5

```
<?
$CordaBandeira = "Vermelho";
if ($CordaBandeira == "Verde" or
    $CordaBandeira == "Amarelo" or
    $CordaBandeira == "Azul" or
    $CordaBandeira == "Branco")
{
    echo $CordaBandeira;
    echo ' é uma das cores de nossa bandeira!';
}
else
{
    echo $CordaBandeira;
    echo ' não é uma cor de nossa bandeira!';
}
?>
```

Quando utilizamos o operador or, unimos o resultado de ambas as comparações e, caso alguma delas seja verdadeira, o interpretador entenderá que o resultado final das comparações é verdadeiro.

Podemos também utilizar o operador `and`. Neste caso, para que o resultado das comparações seja verdadeiro e, consequentemente, o interpretador execute o primeiro bloco de código entre `{}`, é necessário que todas as comparações sejam verdadeiras. Vejamos como isso funciona.

```
<?
$Aluno1 = "Aprovado";
$Aluno2 = "Reprovado";

if ($Aluno1=="Aprovado" and $Aluno2=="Aprovado")

{
    echo 'Ambos estão aprovados!';
}
else
{
    echo 'Um dos alunos foi reprovado!';
}
?>
```

Podemos observar que, no exemplo anterior, criamos uma condição de igualdade entre as comparações quando utilizamos o operador `and`. As duas comparações devem ser verdadeiras para que o computador aceite a expressão total como verdadeira. Se mudarmos o valor da variável `$Aluno2` para `Aprovado`, veremos que o segundo trecho de código será executado.

Estruturas de controle

Quando falamos em estruturas de controle, estamos falando em comandos utilizados normalmente por toda linguagem que possui bloco estruturado, para o controle do fluxo de execução do sistema.

Os sistemas de informação possuem a necessidade de tomar decisões a todo momento. Por isso, precisamos ter comandos que permitem controlar isso de forma automática, ou seja, quando o software for executado, ele considerará críticas que lhe permitirão "tomar decisões".

Este capítulo, dedicaremos ao estudo dessas estruturas e, com certeza, você encontrará estruturas parecidas ou até mesmo idênticas a essas em outras linguagens. O PHP, por exemplo, tem uma estrutura de controle idêntica à da linguagem C. Podemos dizer que quem programa ou programou em C, conhece parte do PHP. É natural que linguagens de programação tenham sido inspiradas em linguagens anteriores.

Veremos a seguir a estrutura `if`, que já utilizamos em nossos exemplos. Esse comando será muito utilizado, seja com PHP, C ou qualquer outra linguagem.

If (<condição>) { }

Este comando é composto por dois blocos, dos quais apenas um será executado em um determinado momento. Para que o primeiro seja interpretado pelo sistema, a condição colocada entre () deve ter resultado verdadeiro. Caso contrário, será o segundo que receberá o fluxo do sistema.

Existem também situações em que queremos fazer que um código seja executado perante uma condição e apenas isso. Vamos ver um exemplo:

```
<?
$Descontos = 315;
if ($Descontos <> 0)
{
    echo 'Descontos deste mês.....: $ '. $Descontos;
}
?>
```

O sistema só executará o comando echo quando o valor da variável desconto for maior que 0. Agora, ficou claro que este comando é muito útil para dar ao sistema uma espécie de "poder de decisão". Na verdade, estamos programando o sistema para cada tipo de situação e então o software agirá conforme a necessidade de cada problema, situação ou estado do sistema naquele momento.

Podemos dar ao fluxo mais uma opção utilizando a cláusula e/se. Nesse caso, o fluxo executará o bloco antes ou depois desta cláusula. No exemplo anterior, podemos incrementar o código com mais uma situação.

```
<?
$Descontos = 315;
if ($Descontos <> 0)
{
    echo 'Descontos neste mês.....: $ '. $Descontos;
}
else
{
    echo ' *** Não há descontos este mês *** ';
}
?>
```

Para entender melhor o exemplo, troque o valor da variável \$Desconto para zero e teste novamente.

A cláusula e/se/if nos permite ter várias expressões dentro do mesmo comando if. Isso permite escolher entre várias situações e, caso nenhuma retorne verdadeiro, o fluxo irá para a cláusula e/se, se esta for utilizada. Veja:

```

<?
$Salario = 1550;
if ($Salario < 1250)
{
    echo 'Remuneração isenta de imposto!';
}

ElseIf ($Salario >= 1250 and $Salario < 1800)
{
    echo 'Retenção de 5% para imposto mensal';
}
ElseIf ($Salario >= 1800 and $Salario < 2200)
{
    echo 'Retenção de 7% para imposto mensal';
}
else
{
    echo 'Retenção de 9% para imposto mensal';
}
?>

```

Podemos observar que a cláusula elseif pode ser utilizada várias vezes, permitindo calcular várias faixas para um valor de salário. Aqui, utilizamos a variável \$Salario, mas este valor pode vir de um banco de dados que será alimentado por usuários do sistema, em momento totalmente distinto a sua programação. Por isso, os sistemas são bastante flexíveis aos dados e devem tratá-los de acordo com as normas e legislações vigentes. Podemos verificar isso no exemplo anterior, no qual para cada faixa salarial é aplicado um imposto que, em um software oficial, deve estar de acordo com as normas legais.

Switch (<condição>) {}

Este comando permite que várias comparações sejam feitas e, no final, apenas uma seja escolhida para executar um código específico da condição selecionada. Ele funciona igualmente a vários ifs, mas agora ele testa os valores de uma mesma variável.

Vejamos um exemplo:

```

<?
$valor = 2;
Switch($valor)
{
    case 0: echo 'O valor da variável é 0<BR>';
    case 1: echo 'O valor da variável é 1<BR>';
    case 2: echo 'O valor da variável é 2<BR>';
}
?>

```

```

        case 3: echo 'O valor da variável e 3<BR>';
        case 4: echo 'O valor da variável e 4<BR>';
    }
?>

```

No exemplo anterior, o comando switch testará linha a linha, em busca de um case que corresponda ao valor da variável. Quando encontrar o valor correspondente, executará os comandos que virão logo após este case. Esses comandos, estarão entre {}, se for mais que um. Teste o código e veja como fica.

Quando precisamos executar apenas o case que corresponda ao valor da variável, utilizaremos um comando que tem a característica de parar fluxos de execução, o comando break. Caso esse não seja utilizado, todos os cases após o case que corresponda ao valor da variável de referência serão executados.

Utilizando o exemplo anterior, veremos como ficaria o código com o uso do comando break:

```

<?
$valor = 2;
Switch($valor)
{
    case 0: echo 'O valor da variável e 0';
    break;
    case 1: echo 'O valor da variável e 1';
    break;
    case 2: echo 'O valor da variável e 2';
    break;
    case 3: echo 'O valor da variável e 3';
    break;
    case 4: echo 'O valor da variável e 4';
    break;
}
?>

```

No exemplo anterior, apenas os comandos que correspondam ao case que tem o mesmo valor da variável serão executados. Teste os exemplos anteriores, veja a diferença entre eles e descobrirá que o exemplo que possui o break é, neste caso, o correto.

While (<condição>) {}

Encontrar uma situação na qual seja necessário executar um mesmo código várias vezes não é muito difícil em programação. Para isso, temos o comando while. Ele permite que um código seja aproveitado e repetido várias vezes até que uma condição seja satisfeita.

Essa repetição é chamada de loop. Um loop pode ser executado infinitas vezes até que seja

interrompido, ou pela mudança de um valor que faz parte de uma condição de teste ou até mesmo pelo comando break, que foi visto anteriormente.

O nosso exemplo de loop pode ser verificado a seguir.

```
<?
  While (True)
  {
    echo 'Este é um loop infinito<BR>';
    echo 'Se não fosse utilizado o comando break,
      seria executado infinitamente';
    break;
  }
?>
```

No exemplo anterior, o comando break é o responsável pela parada do loop. Se ele não fosse utilizado, a página PHP nem mesmo seria mostrada; não seria possível mostrar uma página de "montagem infinita". O sistema travaria.

Quando temos uma condição para a repetição, podemos utilizar uma estrutura na qual o break não será mais necessário. A seguir, veremos como normalmente utilizamos este comando.

```
<
$Valor=1;
While ($Valor<=10)
{
  echo 'Neste momento, o valor é: '.$Valor.'<BR>';
  $Valor++;
}

?>
```

Quando executar este bloco de código, verá que será escrita na tela várias linhas, onde apenas o valor da variável é mudado a cada ciclo. Isso porque, a cada ciclo do comando while, incrementamos o valor da variável em 1 pelo comando \$Valor++.

O sistema executou o que está dentro do bloco entre {} e como o valor da variável foi mudado em cada momento de repetição, quando atingiu um valor maior que 10, a nossa condição para execução retornou False. Então, o loop foi interrompido e o fluxo passou para a próxima linha após o bloco {}.

É essa a função do comando while, permitir que um mesmo código seja executado várias vezes e que possamos pará-lo pelo resultado de uma condição que será mudado a cada ciclo ou pela utilização do comando break.

Do {} While (<condição>)

Semelhantemente ao comando anterior, temos o do while, que se diferencia apenas pelo fato de executar o bloco pelo menos uma vez.

Como podemos observar, o teste é feito no final, permitindo que o fluxo seja repetido apenas se a condição for verdadeira, mas tudo isso após a primeira execução dos comandos entre o bloco. Veja como funciona:

```
<?
$Cabecalho = True;
$Valor=1;
Do
{
    if ($Cabecalho)
    {
        $Cabecalho=False;
        echo 'Código      Filme<BR><BR>';
    }
    echo $Valor.' ';
    echo 'Sinopse do filme '.$Valor.'<BR>';

    $Valor++;
}
While ($Valor<=5)
?>
```

Neste exemplo, criamos uma variável com o nome \$Cabecalho, com seu valor igual a True. Isso fará o if ser executado, pois no lugar da expressão do comando if, colocamos o valor desta variável, que é True, ou seja, verdadeiro.

Ao executar o código que consta dentro do comando if, o valor da variável \$Cabecalho será trocado para False. Com isso, na próxima execução do loop, o cabeçalho não será mais impresso.

Como o valor da variável \$Valor é 5, o loop executará cinco vezes. Mas se você trocar o valor inicial desta variável para um valor maior que cinco, pelo menos o cabeçalho e um comando echo serão executado. Isto ocorre porque o bloco de código do comando do while vem antes do teste para repetição. Com esta função, podemos utilizar o while sempre que for necessário executar um código pelo menos uma vez.

For(<início>;<condição>;<incremento>) { }

Ainda com nosso estudo de repetições de código, podemos caracterizar este comando que vamos estudar agora como sendo uma estrutura de repetição finita.

Anteriormente, para finalizar nosso loop, foi utilizado uma expressão lógica para que o fluxo fosse interrompido. Utilizamos também o comando break. Agora, nosso loop já tem

valores de repetição definidos e até mesmo o valor de incremento será previamente estipulado.

Para entender melhor e de forma prática, vejamos este exemplo. Faremos a mesma coisa que o anterior, só que agora utilizando o comando for.

```
<?
$Cabecalho = True;
For($Valor=1;$Valor<=5;1)

{
    if ($Cabecalho)
    {
        $Cabecalho=False;
        echo 'Código Filme<BR><BR>';
    }
    echo $Valor.' ';
    echo 'Sinopse do filme '.$Valor.'<BR>';
    $Valor++;
}
?>
```

Parte 3

Sistemas on-line

Até aqui, vimos os comandos estruturais do PHP. Agora, iremos passar para a prática desses comandos. Utilizaremos um site em HTML puro, para a utilização do PHP em sua definição de sistema.

No momento, existem basicamente quatro tipos de profissionais diretamente ligados ao ambiente de desenvolvimento de software para Internet. São eles:

- Analista de requisitos;
- Analista projetista;
- Webdesign;
- Analista programador.

O analista programador é o responsável por programar o site em uma linguagem de programação. Esta linguagem pode ser de script (PHP, ASP) ou uma plataforma como o Java, que nos permite criar software para acesso remoto via Internet. A escolha desta linguagem pode ser discutida por toda a equipe. O PHP é uma linguagem de script que pode ser utilizada para criar vários recursos, como links dinâmicos; imagens dinâmicas; sessão, manutenção e alimentação de banco de dados; telas de administração de banco de dados; entre outros, com a vantagem de ser uma linguagem muito simples e dinâmica.

A seguir, apresentaremos estes recursos, os quais poderão ser utilizados para a programação de grande parte dos sites hoje existentes na Internet.

Criando um modelo de site

Quando chegamos à fase de programação de um site, é importante que os requisitos necessários para este serviço, seu projeto e seu design já estejam concluídos.

Então, vale dizer que este momento de programação é o momento final, e não definitivo, da jornada de desenvolvimento para Web. Quando digo que não é definitivo, estou dizendo que este mesmo projeto pode sofrer alterações após sua conclusão e, até mesmo, o ideal é que isso ocorra. Um site que tem sempre novas áreas, novos serviços e atrativos, os quais são constantemente ampliados e cada vez mais atualizados, está no caminho certo, tanto para os objetivos como para o conforto dos internautas cada vez mais informados e exigentes.

Neste capítulo criaremos um site que, primeiramente, estará formatado em HTML puro e

aos poucos iremos utilizar os recursos do PHP já apresentados. Este site terá sua página inicial e, como é um padrão, seu nome será index.php. Sua extensão será em php porque logo terá seus scripts.

Criaremos um menu e depois incluiremos no site conteúdos de banco de dados. Isso lhe permitirá fazer o mesmo com seu projeto, não importando quantas seções a mais terá nele, visto que a técnica aqui apresentada pode ser utilizada de várias formas.

Então, vamos ao nosso site exemplo.

Definindo uma área de cabeçalho

Este site possui uma página inicial chamada index.php e você poderá criá-la com os códigos a seguir. Observe que a primeira letra deste nome é minúscula. Isto porque os provedores costumam ser configurados para buscar nomes de páginas iniciais em minúsculo. Crie um arquivo chamado de index.php e coloque nele o código a seguir:

```
<html>
  <head>
    <title>Criando Sites Profissionalmente</title>
  </head>
  <body>
    <table border="0" width="100%" colspacing = 0>
      <tr>
        <td width = 150>
        </td>
        <td>
        </td>
        <td width = 150>
        </td>
      </tr>
      <tr>
        <td>
        </td>
        <td>
        </td>
        <td valign = "top">
        </td>
      </tr>
    </table>
  </body>
</html>
```

Como podemos observar, se esta página for rodada em um servidor PHP, não será apresentado nada no browser. Isto porque, estes comandos, neste momento, só servem para

definir a estrutura inicial da página. Note que existem outras formas de estruturar um documento. Em nosso caso, serão utilizadas tabelas HTML.

O nosso documento foi estruturado com uma tabela de duas linhas com três colunas cada. A primeira linha será utilizada para acoplar o cabeçalho da página e a outra, o conteúdo.

Para que nosso cabeçalho seja exibido, utilizaremos dois frames que ficarão nas extremidades do cabeçalho. Crie um arquivo com o nome Banner1.php e coloque nele o código a seguir:

```
<html>
<head><title>Digerati Books</title></head>
<body bgcolor = "#FF8080">
<p align="center">
    <font style = "Tahoma" color="#0000FF" size = 6>
        Digerati<BR>Books
    </font>
</p>
</body>
</html>
```

Quando rodar o código anterior, verá que será apresentada uma página com a marca Digerati Books, a qual será mostrada em nossa página inicial. Agora vamos criar o segundo banner, que terá o nome Banner2.php. Coloque neste arquivo o código seguinte.

```
<html>
<head><title>Hoje</title></head>
<body bgcolor = "#E6E6E6">
<?
$mes = date('n');
switch ($mes)
{
    case 1: $mes = 'janeiro'; break;
    case 2: $mes = 'fevereiro'; break;
    case 3: $mes = 'março'; break;
    case 4: $mes = 'abril'; break;
    case 5: $mes = 'maio'; break;
    case 6: $mes = 'junho'; break;
    case 7: $mes = 'julho'; break;
    case 8: $mes = 'agosto'; break;
    case 9: $mes = 'setembro'; break;
    case 10: $mes = 'outubro'; break;
    case 11: $mes = 'novembro'; break;
    case 12: $mes = 'dezembro'; break;
```

```

}
$dia = date('d');
$ano = date('Y');
echo '<p align = "center">';
    echo '<font style = "Tahoma" color="#0000FF" size=4>';
echo 'Hoje:</p></font>';
echo '<p align = "center">';
    echo '<font style = "Tahoma" color="#0000FF" size=2>';
echo $dia.' / '.$mes.' / '.$ano.'</p></font>';
?>
</body>
</html>

```

Para que o cabeçalho do site possa mostrar a data de hoje, precisamos utilizar alguns recursos do PHP para mostrar esta data conforme a data do sistema. Para isso, utilizamos a função date().

Toda função é uma rotina predefinida que retorna valores, neste caso, serão valores de data hora. A forma com que estes valores serão retornados depende do parâmetro que enviamos entre () (parênteses).

Criamos, então, uma variável chamada \$mes, e atribuímos o mês atual do sistema utilizando o parâmetro n da função date(). Veja o código:

```
$mes = date('n');
```

Como o valor numérico do mês será retornado pela função date(), converteremos este para extenso, utilizando o comando switch case que estudamos anteriormente.

- switch (\$mes): iniciamos o comando utilizando como parâmetro de comparação a variável \$mês;
- case 1: \$mes = janeiro'; break; : este é o primeiro teste do comando switch case, cuja finalidade é testar se o valor da variável \$mês é igual a 1. Caso isso seja verdadeiro, os comandos seguintes serão executados;
- Nas outras comparações case, só muda o comando seguinte;
- \$dia = date('d');: com o atributo d da função date() podemos atribuir a variável \$dia, o valor do dia atual do sistema;
- \$ano = date('Y'); : de forma análoga ao comando anterior, atribuímos agora o ano da data atual, utilizando o parâmetro Y. Observe que estamos finalizando cada comando com um (ponto e vírgula);

- A seqüência de comandos echo que vem logo após o comando switch case enviará ao browser algumas formatações do HTML;
- echo \$dia.'/'.\$mes.'/'.\$ano: para que a data seja mostrada na tela no formato dd/mm/aaaa, utilizamos o recurso de concatenação. Juntamos os valores das variáveis anteriormente definidas com a String '/' para ter este formato de data;

Ao executar Banner2.php em um servidor PHP, uma data formatada aparecerá na tela, e esta será utilizada para compor o cabeçalho do site. Nosso arquivo index.php ficará da seguinte forma:

```

<html>
  <head>
    <title>Criando Sites Profissionalmente</title>
  </head>
  <body>
    <table border="0" width="100%" colspacing = 0>
      <tr>
        <td width = 150>
          <IFRAME align="center" name=
            Banner1 frameBorder=0 src=
            "Banner1.php" scrolling=no height=
            "110" width="100%">
            Seu navegador não oferece suporte a quadros
          </IFRAME>
        </td>
        <td bgcolor = "#FFFFC0" align = "center">
          <font style = "Tahoma" color="#0000FF" size = 6>
            Criando Sites Profissionalmente
          </font>
        </td>
        <td width = 150>
          <IFRAME align="center" name=
            Banner2 frameBorder=0 src=
            "Banner2.php" scrolling=no height=
            "110" width="100%">
            Seu navegador não oferece suporte a quadros
          </IFRAME>
        </td>
    </table>
  </body>
</html>

```

```
</td>
</tr>
<tr>
<td>
</td>
<td>
</td>
<td valign = "top">
</td>
</tr>
</table>
</body>
</html>
```

Ao rodar o código anterior, surgirá uma página com dois frames que representam o cabeçalho de nosso site exemplo. Observe que para o código PHP funcionar corretamente, é necessário rodar esta página em um servidor PHP previamente configurado. Se você já tem um provedor que forneça o serviço, teste a página nele.

Login

Quando desenvolvemos um site com informações que devem ser restritas a um grupo de pessoas, necessitamos criar um sistema de segurança em que cada usuário tenha uma senha e um nome, o qual é chamado de login.

A forma que o login será utilizado depende de cada site. O administrador pode definir críticas para a criação de páginas dinâmicas nas quais cada usuário terá o direito de acessar informações somente se uma senha for utilizada. Isto poderá ser melhor compreendido no próximo capítulo.

Agora, criaremos um formulário que utilizaremos para registrar o usuário no site. No momento, somente estamos criando o layout em HTML. Em breve criaremos o sistema de controle de usuários. O código grifado a seguir deve ser incluído no arquivo index.php. Preste atenção para colocar o trecho no local correto.

```
<html>
  <head>
    <title>Criando Sites Profissionalmente</title>
  </head>
  <body>
    <table border="0" width="100%" colspacing = 0>
      <tr>
        <td width = 150>
          <IFRAME align="center" name=
            Banner1 frameBorder=0 src=
            "Banner1.php" scrolling=no height=
            "110" width="100%">
            Seu navegador não oferece suporte a quadros
          </IFRAME>
        </td>
        <td bgcolor = "#FFFFC0" align = "center">
          <font style = "Tahoma" color="#0000FF" size = 6>
            Criando Sites Profissionalmente
          </font>
        </td>

        <td width = 150>
          <IFRAME align="center" name=
            Banner2 frameBorder=0 src=
            "Banner2.php" scrolling=no height=
            "110" width="100%">
            Seu navegador não oferece suporte a quadros
          </IFRAME>
        </td>
      </tr>
      <tr>
        <td valign = "top">
          <table border="0" width="100%">
            <tr>
              <td align = "center" bgcolor = "#C0C0FF">
                <font style = "Tahoma" size = 2>Login
              </font></td>
            </tr>
          </table>
        </td>
      </tr>
    </table>
  </body>
</html>
```

```

</tr>
<tr>
    <td bgcolor = "#D0D0D0">
        <form method = "post" name=
        "Logar" action = "">

            <font style = "Tahoma" size = 1>
                Nome: <input type="text" name=
                    "nome" value="" size="15">
                <BR>
                Senha: <input type="text" name=
                    "senha" value="" size="15">
                <BR>
                <p align="center">
                    <input onmouseover=this.style.cur-
sor= "hand" type="submit" name=
                    "entrar" value="Entrar!" STYLE=
                    "border:0; height: 18;
                    width:50; background: #E8F3FD">
                </p>
            </font>
        </form>
    </td>
</tr>
</table>
</td>
<td>
</td>
<td valign = "top">
</td>
</tr>
</table>
</body>
</html>

```

Agora o index.php já está com o formulário de login. O novo código está grifado e deve ser colocado no mesmo local do exemplo. Estamos apenas criando a formatação do site que utilizaremos como exemplo em todo o restante do livro.

Vamos agora para a próxima formatação, que é a criação da área de menu.

Menu

Logo abaixo do formulário de login criaremos o menu. Nele, estarão os links para as

principais seções do site. Utilizaremos três seções para ilustrar os exemplos. O index.php ficará da seguinte forma:

```
<html>
  <head>
    <title>Criando Sites Profissionalmente</title>
  </head>
  <body>
    <table border="0" width="100%" colspacing = 0>
      <tr>
        <td width = 150>
          <IFRAME align="center" name=
            Banner1 frameBorder=0 src=
            "Banner1.php" scrolling=no height=
            "110" width="100%">
            Seu navegador não oferece suporte a quadros
          </IFRAME>
        </td>
        <td bgcolor = "#FFFFC0" align = "center">
          <font style = "Tahoma" color="#0000FF" size = 6>
            Criando Sites Profissionalmente
          </font>
        </td>
        <td width = 150>
          <IFRAME align="center" name=
            Banner2 frameBorder=0 src=
            "Banner2.php" scrolling=no height=
            "110" width="100%">
            Seu navegador não oferece suporte a quadros
          </IFRAME>
        </td>
      </tr>
      <tr>
        <td valign = "top">
          <table border="0" width="100%">
            <tr>
              <td align = "center" bgcolor = "#C0C0FF">
                <font style = "Tahoma" size = 2>Login
              </font></td>
```

```
</tr>
<tr>
<td bgcolor = "#D0D0D0">
<form method = "post" name=
"Logar" action = ""
<font style = "Tahoma" size = 1>
    Nome: <input type="text" name=
"nome" value="" size="15"><BR>
    Senha: <input type="text" name=
"senha" value="" size="15"><BR>
<p align="center">
    <input onmouseover=this.style.cursor=
"hand" type="submit" name=
"entrar" value="Entrar!" STYLE=
"border:0; height: 18;
width:50; background: #E8F3FD">
</p>

    </font>
</form>
</td>
</tr>
</table>
<table border="0" width="100%">
<tr>
<td align = "center" bgcolor = "#C0FFFF">
<font style = "Tahoma" size = 2>
    Quem somos
</font>
</td>
</tr>
<tr>
<td align = "center" bgcolor = "#C0FFFF">
<font style = "Tahoma" size = 2>
    Fale Conosco
</font>
</td>
</tr>
<tr>
```

```
<td align = "center" bgcolor = "#C0FFFF">
    <font style = "Tahoma" size = 2>Imagens
    </font>
</td>
</tr>
</table>
</td>
<td>
</td>
<td valign = "top">
</td>
</tr>
</table>
</body>
</html>
```

Notícias

Estamos concluindo a formatação do site, então, adicionaremos uma seção de notícias. Ajuste index.php conforme mostrado a seguir:

```
<html>
  <head>
    <title>Criando Sites Profissionalmente</title>
  </head>
  <body>
    <table border="0" width="100%" colspacing = 0>
      <tr>
        <td width = 150>

<IFRAME align="center" name=
  Banner1 frameBorder=0 src=
  "Banner1.php" scrolling=no height=
  "110" width="100%">
  Seu navegador não oferece suporte a quadros
</IFRAME>
</td>
<td bgcolor = "#FFFFC0" align = "center">
  <font style = "Tahoma" color="#0000FF" size = 6>
    Criando Sites Profissionalmente
  </font>
</td>
<td width = 150>
<IFRAME align="center" name=
  Banner2 frameBorder=0 src=
  "Banner2.php" scrolling=no height=
```

```
"110" width="100%">>
    Seu navegador não oferece suporte a quadros
  </IFRAME>
</td>
</tr>
<tr>
  <td valign = "top">
    <table border="0" width="100%">
      <tr>
        <td align = "center" bgcolor = "#C0C0FF">
          <font style = "Tahoma" size = 2>Login
          </font></td>
      </tr>
    </table>
  </td>
</tr>
<tr>
  <td bgcolor = #D0D0D0>
    <form method = "post" name=
      "Logar" action = ">
        <font style = "Tahoma" size = 1>
          Nome: <input type="text" name=
            "nome" value="" size="15">
          <BR>
          Senha: <input type="text" name=
            "senha" value="" size="15">
          <BR>
        <p align="center">
```

```
<input onmouseover=this.style.cursor=
    "hand" type="submit" name=
    "entrar" value="Entrar!" STYLE=
    "border:0;height: 18;
    width:50; background: #E8F3FD">
</p>
</font>
</form>
</td>
</tr>
</table>
<table border="0" width="100%">
<tr>
    <td align = "center" bgcolor = "#C0FFFF">
        <font style = "Tahoma" size = 2>
            Quem somos
        </font>
    </td>
</tr>
<tr>
    <td align = "center" bgcolor = "#C0FFFF">
        <font style = "Tahoma" size = 2>
            Fale Conosco
        </font>
    </td>
</tr>
<tr>
    <td align = "center" bgcolor = "#C0FFFF">
        <font style = "Tahoma" size = 2>
            Imagens
        </font>
    </td>
</tr>
</table>
</td>
```

```

<td>
</td>
<td valign = "top">
<table border="0" width="100%">
<tr>
<td align = "center" bgcolor = "#C0C0FF">
<font style = "Tahoma" size = 4>
    Destaques
</font>
</td>
</tr>
<tr>
<td align = "center" bgcolor = "#C0FFF">
<font style = "Tahoma" size = 2>
    Previsão do tempo
</font>
</td>

</tr>
<tr>
<td align = "center" bgcolor = "#FFFFDD">
<font style = "Tahoma" size = 2>
    Veja a atuação de um <BR>
    furação flagrado por<BR>
    um cinegrafista amador
</font>
</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>

```

Agora iremos para o último item da formatação. É o local onde ficará o conteúdo principal do site.

Conteúdo

Este local é a área reservada para mostrar o conteúdo do site. Utilizaremos um frame para direcionar a ele o conteúdo das outras páginas. Antes, criaremos uma página chamada QuemSomos.php que será apresentada inicialmente, quando o site for carregado. QuemSomos.php terá o código a seguir.

```

<html>
  <head>
    <title>Quem Somos</title>
  </head>
  <body>
    <font color = "#000080" style = "Tahoma" size = 3>
      Olá, caro leitor(a). Agora chegou a hora de praticar
      os comandos até aqui apresentados. Utilizaremos este
      exemplo para ilustrar estes recursos.<BR><BR>
      Quando colocamos um site na Web, devemos nos preocupar, primeiramente,
      com o seu projeto, no qual definimos o que ele
      vai oferecer aos seus usuários.<BR><BR>
      Iniciaremos falando que, para criar um site,
      existem várias ferramentas disponíveis que podem
      ser encontradas na Web.<BR><BR>
      Como nosso objetivo aqui é apresentar os recursos
      de PHP, utilizaremos um modelo de site comum e com
      formatação em HTML e, a partir daí, utilizaremos o PHP
      para criar recursos que poderão ser utilizados em
      outras páginas.</font>
    </body>
  </html>

```

O conteúdo desta página é informativo. Agora vejamos como fica nosso index.php:

```

<html>
  <head>
    <title>Criando Sites Profissionalmente</title>
  </head>

  <body>
    <table border="0" width="100%" colspacing = 0>
      <tr>
        <td width = 150>
          <IFRAME align="center" name=
            Banner1 frameBorder=0 src=
            "Banner1.php" scrolling=no height=
            "110" width="100%">
            Seu navegador não oferece suporte a quadros
          </IFRAME>
        </td>
        <td bgcolor = "#FFFFC0" align = "center">

```

```
<font style = "Tahoma" color="#0000FF" size = 6>
    Criando Sites Profissionalmente
</font>
</td>
<td width = 150>
    <IFRAME align="center" name=
        Banner2 frameBorder=0 src=
        "Banner2.php" scrolling=no height=
        "110" width="100%">
        Seu navegador não oferece suporte a quadros
    </IFRAME>
</td>
</tr>
<tr>
    <td valign = "top">
        <table border="0" width="100%">
            <tr>
                <td align = "center" bgcolor = "#C0C0FF">
                    <font style = "Tahoma" size = 2>Login
                </font></td>
            </tr>
        </table>
    <td bgcolor = #D0D0D0>
        <form method = "post" name=
            "Logar" action = "">
            <font style = "Tahoma" size = 1>
                Nome: <input type="text" name=
                    "nome" value="" size="15"><BR>
                Senha: <input type="text" name=
                    "senha" value="" size="15"><BR>
            <p align="center">
                <input onmouseover=this.style.cursor=
                    "hand" type="submit" name=
                    "entrar" value="Entrar!" STYLE=
                    "border:0;height: 18;
                    width:50; background: #E8F3FD">
            </p>
    </td>
</tr>
```

```
</font>
</form>
</td>
</tr>
</table>
<table border="0" width="100%">
<tr>
<td align = "center" bgcolor = "#C0FFFF">
<font style = "Tahoma" size = 2>
    Quem somos
</font>
</td>
</tr>
<tr>
<td align = "center" bgcolor = "#C0FFFF">
<font style = "Tahoma" size = 2>
    Fale Conosco
</font>
</td>
</tr>
<tr>
<td align = "center" bgcolor = "#C0FFFF">
<font style = "Tahoma" size = 2>
    Imagens
</font>
</td>
</tr>
</table>
</td>
<td>
<IFRAME align = "center" name=Display
src="QuemSomos.php" frameBorder=
0 scrolling=1 height="280" width = "100%">
    Seu navegador não oferece suporte a quadros
</IFRAME>
```

```

</td>
<td valign = "top">
<table border="0" width="100%">
<tr>
<td align = "center" bgcolor = "#C0C0FF">
<font style = "Tahoma" size = 4>
    Destaques
</font>
</td>
</tr>
<tr>
<td align = "center" bgcolor = "#C0FFFF">
<font style = "Tahoma" size = 2>
    Previsão do tempo
</font>
</td>
</tr>
<tr>
<td align = "center" bgcolor = "#FFFDDA">
<font style = "Tahoma" size = 2>
    Veja a atuação de um <BR>
    furação flagrado por<BR>
    um cinegrafista amador
</font>
</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>

```

Este capítulo foi dedicado à criação do site exemplo. Agora, incrementá-lo-emos com o PHP.

Programando um site com PHP

Quando trabalhamos com um sistema, necessitamos que este faça uso de um banco de dados. Agora você aprenderá a trabalhar com o MySQL, um banco de dados muito utilizado no momento para este fim.

A técnica atual para armazenamento de dados é voltada para um SGBD, ou seja, Sistema de Gerenciamento de Banco de Dados.

SGBD é um sistema com vários recursos para armazenamento e tratamento de dados, e nele

constam vários recursos para controlar e manipular esses dados.

Quando temos uma conta de Internet, normalmente utilizamos um software para gerenciamento do banco de dados fornecido pelo provedor.

Aqui, utilizaremos o PHP para criar as tabelas, visto que podemos criar um script em PHP para fazer isto perfeitamente, com a vantagem de poder utilizar este script quantas vezes quisermos, e até mesmo em outra conta de provedor.

Consideraremos o usuário padrão do MySQL para nossos exemplos e você poderá alterá-lo para o fornecido pelo seu provedor.

O MySQL possui um usuário padrão chamado root e senha nula, ou seja, em branco. O servidor padrão do MySQL é o localhost. Trabalharemos com esses dados.

Para que nosso site tenha conexão a um banco de dados, precisamos criar uma página com script em PHP que faça esta conexão. Criaremos uma página chamada Base.php, que conterá nossos comandos de acesso ao banco de dados e esta será adicionada por um comando especial em cada página. Crie Base.php com o código a seguir.

```
<?
$conexao = mysql_connect('localhost','root','');
mysql_select_db('criandosites',$conexao);
?>
```

- connect('localhost','root','');:\$conexao=mysql a variável \$conexao recebe uma conexão que será aberta pelo comando mysql_connect(). Os parâmetros desta função devem ser mudados para os fornecidos pelo seu provedor. Se for um servidor local, poderá ficar conforme nosso exemplo;

- mysqlselectdb('criandosites,\$conexao);: este comando define o banco de dados para nosso identificador \$conexao. A string 'criandosites' deve ser modificada para o nome do banco de dados que seu provedor lhe fornecerá.

Para que esta conexão seja utilizada pelo site, poderemos incluir um comando no início da página index.php e nas páginas de administração que serão criadas. Isto evitará a repetição destes comandos. Na primeira linha de index.php coloque o seguinte:

```
<? include "Base.php"; ?>
```

- <? include "Base.php", ?>: adiciona o conteúdo da página Base.php. Executa o código desta página neste local.

Criando tabelas no banco de dados

No site que iremos programar, criaremos inicialmente duas tabelas, que serão utilizadas para o cadastramento de usuários e a atualização de notícias. Vamos utilizar a linguagem SQL para a criação dessas tabelas. O comando SQL será o create table, que tem sua sintaxe a seguir:

```
create table <nome da tabela>(<Campo1> <Tipo1>,
<Campo2> <Tipo2>...)
```

Utilizando a sintaxe anterior, poderemos criar facilmente uma tabela no MySQL.

Crie um novo arquivo chamado Scriptl.php, salve-o e inclua nele o código a seguir.

```
<?
include "Base.php";

$usuarios = mysql_query('CREATE TABLE usuarios(
    codigo integer primary key auto_increment,
    log_nome varchar(20), senha varchar(14));
    $conexao );

$noticias = mysql_query('CREATE TABLE noticias(
    registro integer primary key auto_increment,
    titulo varchar(50), conteudo blob);
    $conexao );
if ($usuarios)
{
    echo 'A tabela de usuários foi criada<BR>';
}
else
{
    echo 'Erro tabela usuários. Veja sintaxe!<BR>';
}
if ($noticias)
{
    echo 'A tabela de notícias foi criada<BR>';
}
else
{
    echo 'Erro tabela notícias. Veja sintaxe!<BR>';
}
?>
```

•mysqlquery: esta função executa uma instrução SQL. Veja que estamos utilizando o identificador de conexão anteriormente criado no arquivo Base.php. Isto é possível pela utilização do comando include.

Tendo um identificador de conexão já definido, que em nosso caso é a variável \$conexao,

agora é só utilizar a função mysql query para executar o comando SQL desejado.

- CREATE TABLE: cria uma nova tabela ao banco de dados;
- integer: define o tipo do campo como inteiro;
- primary key: define o campo da tabela como um campo primário, ou seja, este não aceitará valores duplicados e será utilizado como campo chave de expressões;
- auto_increment: permite que, ao incluir registros nesta tabela, este campo não precise receber valores porque será incrementado automaticamente;
- varchar(20): este campo será de caracteres com tamanho máximo de 20;
- blob: tipo de campo texto.

O arquivo Scriptl.php que acabamos de criar, quando executado, deverá produzir duas tabelas em nosso banco de dados. Note que para isto acontecer, nossos comandos de conexão do arquivo Base.php devem estar funcionando perfeitamente. Caso ocorra algum erro de conexão com o MySQL, verifique se o nome do servidor, usuário e senha estão corretos.

Se tudo correu bem, agora já temos em nosso banco de dados uma tabela de usuários e uma de notícias. No próximo item deste capítulo, abordaremos a criação das páginas que serão responsáveis pela manutenção destas tabelas. Estas páginas são chamadas de páginas de administração.

Até o momento, criamos somente as páginas. No item Criando áreas restritas no site, você verá como criar restrições de uso do conteúdo do site. Com o uso de senhas de usuário, você poderá definir uma senha de administrador que terá permissão para o uso dessas páginas de administração. Poderá ser utilizado o comando if para executar os comandos da página somente se o usuário logado no sistema for o administrador.

Alimentando tabelas com páginas de administração

Para que nosso sistema esteja sempre atualizado, poderemos ter uma equipe que seja responsável por sua manutenção. Isto implica em registrar novas informações e retirar outras.

Como estamos trabalhando com banco de dados, será perfeitamente possível que isso seja feito com páginas de administração, as quais são criadas em PHP especificamente para isso.

Iniciaremos criando a página de administração de usuários. Crie um arquivo chamado AdmUsuarios.php e inclua nele o código a seguir:

```

<html>
  <head>
    <title>Administrando usuários</title>
  </head>
  <body>
    <font style = "Tahoma" size=5>
      <p align="center">
        Página de Administração de Usuários
      </p>
    </font>
    <table border="0" width="100%">
      <tr>
        <td width = "50%" valign="top" bgcolor=
          "#DFDFBF">
          <h3 align="center">Usuários</h3>
        </td>
        <td valign="top" align="center">
        </td>
      </tr>
    </table>
  </body>
</html>

```

Criamos a formatação do documento, na qual temos uma tabela com duas colunas. A seguir, incluiremos o primeiro formulário que será utilizado para a inclusão de um novo usuário. Agora nosso arquivo AdmUsuarios.php fica da seguinte forma:

```

<html>
  <head>
    <title>Administrando usuários</title>
  </head>
  <body>
    <font style = "Tahoma" size=5>
      <p align="center">
        Página de Administração de Usuários

      </p>
    </font>
    <table border="0" width="100%">
      <tr>
        <td width = "50%" valign="top" bgcolor=
          "#DFDFBF">
          <h3 align="center">Usuários</h3>
          <form method = "post" name=
            "IncluirUsuario" action=
            "IncluirUsuario.php" target=
            "QuadroUsuarios">

```

```

<font style = "tahoma" size = 1>
    Nome: <input type="text" name=
        "log_nome" value="" size = "15">
    Senha: <input type="text" name=
        "senha" value="" size = "10">
    <input onmouseover=this.style.cursor=
        "hand" type="submit" name=
        "Cadastrar" value="Cadastrar" STYLE=
        "border : 0; height: 20; width:72;
        background: #E8F3FD">
</font>
</form>
</td>
<td valign="top" align="center">
</td>
</tr>
</table>
</body>
</html>

```

- action="IncluirUsuario.php" target="QuadroUsuarios": a ação deste formulário aponta para um arquivo chamado IncluirUsuario.php. Esta página terá um código em PHP que será responsável pelo cadastramento do novo usuário. Ela receberá as variáveis do formulário e as utilizará na inclusão do novo registro.

O código do arquivo IncluirUsuario.php pode ser conferido a seguir.

```

<html>
  <head>

```

```

<title>Incluir usuário</title>
</head>
<body>
<?
    include "Base.php";
    $result = mysql_query('insert into usuarios
        (log_nome,senha) values("'.$log_nome.'","'.
        $senha.'");,$conexao);
    if ($result)
    {
        echo 'Usuário '.$log_nome.
            ' cadastrado com sucesso!';
    }
    else
    {
        echo 'Usuário '.$log_nome.
            ' não pode ser cadastrado!';
    }
?>
</body>
</html>

```

- include "Base.php";: utilizamos os comandos deste arquivo para fazer a conexão com o banco de dados;

- \$result = mysql query('insert into usuarios (log_nome,senha) values(-.\$log_nome.—.-.

\$senha.");,\$conexao);: neste momento, executaremos o comando SQL por meio da função mysql query. Veja que utilizamos duas variáveis: \$log_nome e \$senha. Esses valores foram enviados pela ação do formulário da página AdmUsuarios.php. Como esta página que estamos editando foi acionada por AdmUsuarios.php, temos estes nomes de variáveis disponíveis, mas agora com um \$ no início de

seu nome;

- if (\$result): é utilizado para mostrar o resultado da operação.

Agora, rode a página AdmUsuarios.php e veja o resultado. Podemos digitar um nome e senha de usuário nas caixas de texto e clicar no botão Incluir para cadastrar este registro na tabela de usuários. Você pode estar se perguntando: ver onde?

Neste momento, se utilizar um software de administração de banco de dados, verá o registro incluído, mas não deve ser só isso que você quer. Precisamos então, criar uma rotina que mostre nesta mesma página os dados cadastrados na tabela de usuários. Mas isto é o assunto do próximo capítulo.

Recuperando registros

Recuperar registros significa resgatar os dados que estão armazenados em um banco de dados. Como estamos trabalhando com a página de manutenção de usuários, vamos criar agora um arquivo chamado ListarUsuarios.php. Ele será o responsável por mostrar os dados que estão armazenados na tabela de usuários. Seu código pode ser conferido a seguir:

```
<html>
  <head>
    <title>Listar usuários</title>
  </head>
  <body>
    <table border="0" width=
      "100%" colspacing = "0">
      <tr>
        <td>
          <?
            include "Base.php";
            $result = mysql _ query(
              "select codigo,
                log _ nome from usuarios order by codi-
go;"
              , $conexao);
            $Quantos = mysql _ num _ rows($result);
            $i=0;

echo '<table border="0" width="100%">';
echo '<tr><td width = 30" bgcolor=
  "#FFFFC0">Código';
echo '</td><td bgcolor="#FFFFC0">
  Usuário</td>';
for ($i==0; $i<$Quantos; $i++)
{
  echo '<tr><td>';
  echo '<span style="font-size:8.0pt;
    font-family:Verdana;color:#000080">';


```

```

        echo mysql_result($result,$i,codigo);
        echo '</td><td>';
        echo mysql_result($result,$i,log_
nome);
        echo '</td></tr>';
    }
echo '</table>';
?>
</td>
</tr>
</table>
</body>
</html>

```

- \$Quantos = mysql_num_rows(\$result);: este comando armazena na variável \$Quantos, o número de registros da tabela de usuários. Observe que estamos utilizando o identificador \$result, que naquele momento representa o resultado da última query executada pelo comando mysql query, que, em nosso caso, faz referência à tabela de usuários;
- \$i=0,: criamos esta variável para ser utilizada dentro de um ciclo for, e assim representar cada linha da consulta feita pelo comando mysql query anterior;
- echo '<table border="0" width="100%">': utilizamos o comando echo para enviar ao browser uma string de comandos HTML;
- for (\$i==0, \$i<\$Quantos, \$i++): para que seja possível listar na tela todos os registros da tabela de usuários, precisamos utilizar o comando for. Neste caso, ele executará o comando mysql result inicialmente na linha zero da tabela de usuários e só terminará a repetição quando a variável \$i for menor que \$Quantos;
- echo mysql_result(\$result,\$i,codigo); : este comando envia para a tela o valor do campo codigo. A variável \$i define qual linha será enviada, que também representa o registro na tabela. Como estamos dentro de um ciclo for que pega desde o início até o fim da tabela, todas as linhas serão enviadas formando, assim, nossa lista de usuários.

No código anterior, podemos observar que foi utilizado o comando mysql result em conjunto com o echo, o qual também foi utilizado para enviar formatações em HTML.

O nosso exemplo até aqui apresentou formas de incluir dados em uma tabela no MySQL e também como listar estes dados em uma página. Criamos então os arquivos IncluirUsuarios.php e ListarUsuarios.php, que são as páginas responsáveis por realizar estas operações. Mas não são estas páginas que iremos acessar, e sim AdmUsuarios.php. Esta terá o seu código novamente modificado para que liste também os usuários cadastrados. Veja a seguir:

```
<html>
  <head>
    <title>Administrando usuários</title>
  </head>
  <body>
    <font style = "Tahoma" size=5>
      <p align="center">
        Página de Administração de Usuários
      </p>
    </font>
    <table border="0" width="100%">
      <tr>

<td width = "50%" valign="top" bgcolor=
"#DFDFBF">
  <h3 align="center">Usuários</h3>
  <form method = "post" name=
"IncluirUsuario" action=
"IncluirUsuario.php" target=
"QuadroUsuarios">
    <font style = "tahoma" size = 1>
      Nome: <input type="text" name=
"log_nome" value="" size = "15">
      Senha: <input type="text" name=
"senha" value="" size = "10">
      <input onmouseover=this.style.cursor=
"hand" type="submit" name=
"Cadastrar" value="Cadastrar" STYLE=
"border : 0; height: 20; width:72;
background: #E8F3FD">
    </font>
  </form>
</td>
<td valign="top" align="center">
```

```

<form method = "post" name=
      "ListarUsuarios"
      action = "ListarUsuarios.php" target=
      "QuadroUsuarios">
    <input onmouseover=this.style.cursor=
          "hand" type="submit" name=
          "Listar" value="Atualizar!" STYLE=
          "border : 0; height: 18; width:72;
          background: #E8F3FD">
</form>
<IFRAME align="center" name=
        QuadroUsuarios frameBorder=0 valign=
        "top" scrolling=1 height="240" src=
        "ListarUsuarios.php" width="100%">
  Seu navegador não oferece suporte a quadro
</IFRAME>
</td>
</tr>
</table>
</body>
</html>

```

- <form method="post" name="ListarUsuarios": este formulário que incluímos tem a função de acionar a página que listará os usuários, aquela que acabamos de programar. Podemos observar isto pelo comando action = "ListarUsuarios.php";

- target="QuadroUsuarios": direciona a página para o frame que acabamos de criar.

Até o momento, aprendemos como incluir e listar dados de uma tabela com o PHP. Em Criando áreas restritas no site, produziremos outra página de administração, parecida com a que acabamos de programar. A novidade é que ela terá também um formulário para excluir dados.

Implementando uma sessão de login

Agora que temos usuários cadastrados em nossa tabela, utilizaremos uma sessão de login para que o acesso a um determinado conteúdo do site seja restrito a esses usuários.

Para que isso seja possível, precisamos criar um sistema de acesso que tenha o objetivo de registrar os usuários logados. Voltaremos então ao nosso index.php, pois é lá que está nosso formulário de login. A seguir está o código do arquivo index.php que será substituído em breve.

```

<form method = "post" name=
"Logar" action = ""
<font style = "Tahoma" size = 1>
    Nome: <input type="text" name=
"nome" value="" size="15"><BR>
    Senha: <input type="text" name=
"senha" value="" size="15"><BR>
    <p align="center">
        <input onmouseover=this.style.cursor=
"hand" type="submit" name=
"entrar" value="Entrar!" STYLE=
"border:0; height: 18;
width:50; background: #E8F3FD">
    </p>
</font>
</form>

```

O formulário anterior foi criado em nosso arquivo index.php. Substitua este trecho de código pelo que vem a seguir.

```

<?
if ($HTTP_POST_VARS['entrar']==
'Entrar!')
{
    @session_start();
    $result = mysql_query(
        'select log_nome,
        senha from usuarios where log_nome
        =="'.trim($nome)."';',$conexao);
    if (mysql_num_rows($result)>0)
    {
        $usuSenha=mysql_result($result,
        0,senha);
    }
    else
    {
        $usuSenha = 'notuser';
    }
    if ($result and ($senha==$usuSenha

```

```

) and ($usuSenha<>'notuser'))
{
@session_register($nome);
echo '<p align="center">';
echo '<font style="Tahoma" size=
4>';
echo $nome.'</font><BR>';
echo '<font style="Tahoma" size=
2>';
echo 'Você está on-line</font>';
echo '</p>';

}
else
{
    echo 'Usuário não autorizado';
}
}
else
{
    echo '<form method = "post" name=
"Logar" action = "">
<font style="Tahoma" size = 1>
Nome: <input type="text" name=
"nome" value="" size = "15"><BR>
Senha: <input type="text" name=
"senha" value="" size = "15"><BR>
</font>
<p align="center">
<input onmouseover=
this.style.cursor=
"hand" type="submit"
name="entrar" value="Entrar!"'
STYLE = "border : 0; height: 18;
width:50; background: #E8F3FD">
</p>
</form>';
}
?>
```

Observe que o código anterior ficará no lugar apenas do formulário de login. Para que não reste dúvida, colocaremos a seguir todo o código do index.php com o novo trecho grifado, o qual será explicado em seguida.

```
<? include "Base.php"; ?>
<html>
```

```
<head>
  <title>Criando Sites Profissionalmente</title>
</head>
<body>
  <table border="0" width="100%" colspacing = 0>
    <tr>
      <td width = 150>
        <IFRAME align="center" name=Banner1
          frameBorder=0 src="Banner1.php"
          scrolling=no height="110" width="100%">
          Seu navegador não oferece suporte a quadros
        </IFRAME>

      </td>
      <td bgcolor = "#FFFFC0" align = "center">
        <font style = "tahoma" color="#0000FF" size = 6>
          Criando Sites Profissionalmente
        </font>
      </td>
      <td width = 150>
        <IFRAME align="center" name=Banner2
          frameBorder=0 src="Banner2.php"
          scrolling=no height="110" width="100%">
          Seu navegador não oferece suporte a quadros
        </IFRAME>
      </td>
    </tr>
    <tr>
```

```

<td valign = "top">
    <table border="0" width="100%">
        <tr>
            <td align = "center" bgcolor = "#C0C0FF">
                <font style = "tahoma" size = 2>Login
            </font></td>
        </tr>
        <tr>
            <td bgcolor = #D0D0D0>
                <?
                    if ($HTTP_POST_VARS['entrar']==='Entrar!')
                    {
                        @session_start();
                        $result = mysql_query(
                            'select log_nome,
                                senha from usuarios where log_
nome
                                =' . trim($nome) . ',$conexao);
                        if (mysql_num_rows($result)>0)
                        {
                            $usuSenha=mysql_result($result,
                                0,senha);
                        }
                    }
                else
                {
                    $usuSenha = 'notuser';
                }
                if ($result and ($senha==$usuSenha
                    ) and ($usuSenha<>'notuser'))
                {
                    @session_register($nome);
                    echo '<p align="center">';
                    echo '<font style="Tahoma" size=
                        4>';
                    echo $nome.'</font><BR>';
                    echo '<font style="Tahoma" size=
                        2>';
                    echo 'Você está on-line</font>';
                    echo '</p>';
                }
            else

```

```

        {
            echo 'Usuário não autorizado';
        }
    }
else
{
    echo '<form method = "post" name=
"Logar" action = "">
<font style="Tahoma" size = 1>
Nome: <input type="text" name=
"nome" value="" size = "15"><BR>
Senha: <input type="text" name=
"senha" value="" size =
"15"><BR>
</font>
<p align="center">

<input onmouseover=
this.style.cursor=
"hand" type="submit"
name="entrar" value="Entrar!"'
STYLE = "border : 0;height: 18;
width:50; background: #E8F3FD">
</p>
</form>';
}
?>
</td>
</tr>
</table>

<table border="0" width="100%">
<tr>
<td align = "center" bgcolor = "#C0FFFF">
<font style = "tahoma" size = 2>
    Quem somos
</font>
</td>
</tr>
<tr>
<td align = "center" bgcolor = "#C0FFFF">
<font style = "tahoma" size = 2>
    Fale Conosco
</font>

```

```
</td>
</tr>
<tr>
  <td align = "center" bgcolor = "#C0FFFF">
    <font style = "tahoma" size = 2>
      Imagens
    </font>
  </td>
</tr>
</table>
</td>
<td>
<IFRAME align = "center" name=Display
  src = "QuemSomos.php" frameBorder=0
  scrolling=1 height="280" width = "100%">
  Seu navegador não oferece suporte a quadros
</IFRAME>

</td>
<td valign = "top">
  <table border="0" width="100%">
    <tr>
      <td align = "center" bgcolor = "#C0C0FF">
        <font style = "tahoma" size = 4>
          Destaques
        </font>
      </td>
    </tr>
    <tr>
      <td align = "center" bgcolor = "#C0FFFF">
        <font style = "tahoma" size = 2>
          Previsão do tempo
        </font>
      </td>
    </tr>
    <tr>
      <td align = "center" bgcolor = "#FFFFDD">
```

```

<font style = "tahoma" size = 2>
    Veja a atuação de um <BR>
    furação flagrado por<BR>
    um cinegrafista amador
</font>
</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>

```

Este novo código é iniciado com um comando if que testará se a ação do botão Entrar foi acionada. Ao abrir a página pela primeira vez, será enviado para tela, via comando echo, um formulário em HTML que nos permitirá digitar os dados do usuário e senha. Ao acionar o botão Enviar, a página será novamente carregada, só que agora não executará a condição e/se como anteriormente, mas sim terá a condição do if validada, já com os valores das variáveis \$nome e \$senha enviados pelo formulário.

Vamos estudar os novos comandos utilizados.

- @session_start : inicia uma seção;
- if (\$HTTP_POST_VARS['entrar']=='Entrar!'): quando a página é carregada pela primeira vez, esta expressão retornará falso, pois as variáveis de formulário ainda não foram enviadas. Após enviar dados pelo formulário de login, esta página será novamente carregada, porém a expressão retornará verdadeiro, permitindo assim a execução dos comandos seguintes para o registro da seção;
- \$result = mysql_query('select...': executa uma consulta ao banco de dados, retornando o usuário de log_nome igual ao digitado na caixa de texto nome do formulário;
- if (mysql_num_rows(\$result)>0): esta expressão testa se o resultado da consulta retornou algum registro. Em caso negativo, significa que o usuário digitado no formulário não consta na tabela de usuários do banco de dados;
- \$usuSenha =mysql_result(\$result, 0,senha); : se o if anterior permitir a execução deste comando, é porque o usuário digitado no formulário de login existe. Neste caso, a variável \$usuSenha receberá a senha cadastrada no banco de dados;
- \$usuSenha = 'notuser', : este comando será executado caso o usuário não esteja cadastrado. Como o valor desta variável será utilizado na expressão que define o registro de seção, esta não será registrada quando o valor de \$usuSenha for 'notuser';

- if (\$result and (\$senha == \$usuSenha) and (\$usuSenha- <>'notuser')): esta expressão permitirá o registro de seção caso a consulta de usuário no banco de dados tenha sido efetuada com sucesso, e somente se a senha digitada no formulário for igual à cadastrada no banco de dados. Observe que a senha digitada no formulário está neste momento representada pela variável \$senha, a qual foi definida pelo envio do formulário;
- @session_register(\$nome);: registra uma seção com o log_nome do usuário logado. Esta seção será utilizada pelo sistema para o mecanismo de restrição que será apresentado no item Criando áreas restritas no site deste capítulo. Lembre-se de que um sistema deve ter mecanismos para não permitir o cadastramento de usuários com o mesmo log_nome. Basta fazer uma consulta no banco de dados antes de cadastrar, caso o log_nome já exista, não cadastrar e enviar uma mensagem pelo comando echo;
- echo '<p align="center">...': a seqüência de comandos echo que vem a seguir serve para registrar na tela o nome do usuário atualmente logado no sistema.

Links automáticos

Neste momento, trabalharemos com links. Estes serão criados com base nos registros incluídos na tabela de notícias do banco de dados. Para que isso seja possível, criaremos uma página de administração muito semelhante à de administração de usuários. Esta página será para cadastrar as notícias que serão apresentadas no site.

Como o código é muito semelhante ao anteriormente explicado, sofrendo algumas variações no SQL, não entraremos em muitos detalhes.

Crie um arquivo chamado AdmNoticias.php e salve-o com o código a seguir.

```
<html>
  <head>
    <title>Administrando notícias</title>
  </head>
  <body>
    <font style = "Tahoma" size=5>
      <p align="center">
        Página de Administração de Notícias
      </p>
    </font>
    <table border="0" width="100%">
      <tr>
```

```
<td width = "50%" valign="top" bgcolor=
 "#DFDFBF">
<h3 align="center">Notícias</h3>
<form method = "post" name=
 "IncluirNoticia" action=
 "IncluirNoticia.php" target=
 "QuadroNoticias">
<font style="Tahoma" size = 1>
    Título: <input type="text" name=
        "titulo" value="" size = "50">
    <BR>Campo da notícia:<BR>
    <textarea name = "conteudo" rows=
        "5" cols="42">

    </textarea>
    <BR><BR>
    <input onmouseover=this.style.cursor=
        "hand" type="submit" name=
        "Incluir" value="Incluir" STYLE=
        "border:0;height:20;width:72;
        background: #E8F3FD">
    </font>
</form>
<form method = "post" name=
 "ExcluirNoticia" action=
 "ExcluirNoticia.php" target=
 "QuadroNoticias">
<font style = "tahoma" size = 1>
    Registro: <input type="text" name=
        "registro" value="" size = "10">
    <input onmouseover=this.style.cursor=
        "hand" type="submit" name=
        "Excluir" value="Excluir" STYLE=
        "border:0;height: 20;width:72;
        background: #E8F3FD">
    </font>
</form>
</td>
```

```

<td>
<form method = "post" name=
"ListarNoticias" action =
"ListarNoticias.php" target=
"QuadroNoticias">

<p align="center"><input onmouseover=
this.style.cursor="hand" type=
"submit" name="Listar" value=
"Atualizar!" STYLE="border:0;
height: 18; width:72;
background: #E8F3FD">
</p>

</form>
<IFRAME align="center" name=
QuadroNoticias frameBorder=0 valign=
"top" scrolling=1 height="240" src=
"ListarNoticias.php" width="100%">
    Seu navegador não oferece suporte a qua-
dro
</IFRAME>
</td>
</tr>
</table>
</body>
</html>

```

Para que nossa administração de notícias funcione, criaremos outros arquivos que serão utilizados por este que acabamos de criar. Iniciaremos com o IncluirNoticia.php. Veja como ficará o código:

```

<html>
<head>

```

```

<title>Incluir notícia</title>
</head>
<body>
    <?
    include "Base.php";
    $result = mysql_query('insert into noticias
        (titulo,conteudo) values ("'.$titulo.'","'.
        $conteudo.'");', $conexao);
    if ($result)
    {
        echo 'Notícia incluída no Banco de Dados!';
    }
    else
    {
        echo 'Esta notícia não pode ser incluída!';
    }
    ?>
</body>
</html>

```

Agora criaremos o ExcluirNoticia.php. Observe que este arquivo tem uma novidade em seu código: o comando SQL agora será o delete from para excluir registros do banco de dados.

```

<html>
    <head><title>Excluir notícia</title></head>
    <body>
        <?
        include "Base.php";
        $result = mysql_query(
            'delete from noticias where registro='.
            $registro.';', $conexao);
        if ($result)
        {
            echo 'Notícia excluída!';
        }
        else
        {
            echo 'Notícia não encontrada!';
        }
        ?>
    </body>
</html>

```

Vamos criar, então, o ListarNoticias.php. Seu código vem logo a seguir:

```
<html>
  <head>
    <title>Listar Notícias</title>
  </head>
  <body>
    <table border="0" width="100%
      " colspacing = "0">
      <tr>
        <td>
          <?
            include "Base.php";
            $result = mysql_query(
              "select * from noticias order by titu-
lo;" 
            , $conexao);
            $Quantos = mysql_num_rows($result);
            $i=0;

            echo '<table border="0" width="100%">';
            echo '<tr><td width = 15" bgcolor=
              "#FFFFC0">Cod.';
            echo '</td><td width = 50" bgcolor=
              "#FFFFC0">Título';
            echo '</td><td bgcolor="#FFFFC0">
              Notícia</td>';

            for ($i==0; $i<$Quantos; $i++)
            {
              echo '<tr><td>';
              echo '<span style="font-size:8.0pt;
                font-family:Verdana;color:#000080">';
              echo mysql_result($result,$i,registr
o);
              echo '</td><td>';
              echo mysql_result($result,$i,titulo);
              echo '</td><td>';
              echo mysql_result($result,$i,conteud
o);
              echo '</td></tr>';
            }
            echo '</table>';
          ?>
        </td>
      </tr>
    </table>
  </body>
</html>
```

Observe que nesses arquivos apresentados foram utilizados comandos SQL relacionados à tabela notícias.

Para que as notícias sejam apresentadas conforme o que está cadastrado no banco de dados, modificaremos nosso index.php para que os títulos cadastrados na tabela notícias sejam colocados na tela e com links para apresentação de seu conteúdo. Veja como fica o código de nosso index.php.

```
<? include "Base.php"; ?>
<html>
  <head>
    <title>Criando Sites Profissionalmente</title>
  </head>
  <body>
    <table border="0" width="100%" colspacing = 0>
      <tr>
        <td width = 150>

<IFRAME align="center" name=Banner1
  frameBorder=0 src="Banner1.php"
  scrolling=no height="110" width="100%">
  Seu navegador não oferece suporte a quadros
</IFRAME>
</td>
<td bgcolor = "#FFFFC0" align = "center">
  <font style = "tahoma" color="#0000FF" size = 6>
    Criando Sites Profissionalmente
  </font>
</td>
<td width = 150>

<IFRAME align="center" name=Banner2
  frameBorder=0 src="Banner2.php"
  scrolling=no height="110" width="100%">
  Seu navegador não oferece suporte a quadros
</IFRAME>
</td>
</tr>
<tr>
  <td valign = "top">
    <table border="0" width="100%">
      <tr>
        <td align = "center" bgcolor = "#C0C0FF">
```

```

<font style = "tahoma" size = 2>Login
</font></td>
</tr>
<tr>
<td bgcolor = #D0D0D0>
<?
if ($HTTP_POST_VARS['entrar']==='Entrar!')
{
    @session_start();
    $result = mysql_query(
        'select log_nome,
            senha from usuarios where log_
nome
            =' . trim($nome) . ',$conexao);
    if (mysql_num_rows($result)>0)
    {
$usuSenha=mysql_result($result,
0,senha);
    }
else
{
    $usuSenha = 'notuser';
}
if ($result and ($senha==$usuSenha
) and ($usuSenha<>'notuser'))
{
    @session_register($nome);
    echo '<p align="center">';
    echo '<font style="Tahoma" size=
        4>';
    echo $nome.'</font><BR>';
    echo '<font style="Tahoma" size=
        2>';
    echo 'Você está on-line</font>';
    echo '</p>';
}

```

```
else
{
    echo 'Usuário não autorizado';
}
}
else
{
echo '<form method = "post" name=
"Logar" action = ">
<font style="Tahoma" size = 1>
Nome: <input type="text" name=
"nome" value="" size = "15"><BR>

Senha: <input type="text" name=
"senha" value="" size = "15"><BR>
</font>
<p align="center">
<input onmouseover=
this.style.cursor=
"hand" type="submit"
name="entrar" value="Entrar!"'
STYLE = "border : 0; height: 18;
width:50; background: #E8F3FD">
</p>
</form>';
}
```

```
?>
</td>
</tr>
</table>
<table border="0" width="100%">
<tr>
<td align = "center" bgcolor = "#C0FFFF">
<font style = "tahoma" size = 2>
    Quem somos
</font>
</td>
</tr>
<tr>
<td align = "center" bgcolor = "#C0FFFF">
<font style = "tahoma" size = 2>
    Fale Conosco
</font>
</td>
</tr>
<tr>
<td align = "center" bgcolor = "#C0FFFF">
<font style = "tahoma" size = 2>
    Imagens
</font>
</td>
</tr>
</table>
</td>
<td>
<IFRAME align = "center" name=Display
src = "QuemSomos.php" frameBorder=0
scrolling=1 height="280" width = "100%">
    Seu navegador não oferece suporte a quadros
</IFRAME>
</td>
<td valign = "top">
<table border="0" width="100%">
<tr>
<td align = "center" bgcolor = "#C0C0FF">
<font style = "tahoma" size = 4>
Destaques
</font>
</td>
</tr>
<?>
```

```

$result = mysql_query('select titulo,
    conteudo from noticias;',$conexao);
$Quantos = mysql_num_rows($result);
$i=0;
for ($i==0; $i<$Quantos; $i++)
{
    echo '<tr><td bgcolor="#C0FFFF">';
    echo '<span style="font-size:8.0pt;
        font-family:Verdana;color:#000080">';
    echo '<a target = "Display" href=
        "Conteudo.php?Texto='.$mysql_result
        ($result,$i,Conteudo).'">';
    echo mysql_result($result,$i,titulo);
    echo '</a></td></tr>';
}
?>
</table>
</td>
</tr>
</table>
</body>
</html>

```

Ao observar o código anterior pela primeira vez, pode ser um pouco complicado entender tudo. Como estamos criando este index.php desde o início do livro, será fácil observar que um arquivo de script é inicialmente criado com um código pequeno e ao longo do desenvolvimento, seu código ficará mais robusto e complexo. Para você que está trabalhando com este código desde o início, será fácil observar que o código grifado substitui as linhas da tabela destaque.

Vamos à explicação do novo código.

- echo '; ao adicionar a tag <a> ao título da notícia, criamos um link para um arquivo chamado Conteudo.php. Enviamos para este arquivo, através do parâmetro ?, o conteúdo da notícia que está gravado no banco de dados. Para enviar ao browser o conteúdo da notícia, utilizamos o comando mysql result(\$resu lt,\$i,Conteudo).

Observe que por este comando estar no mesmo ciclo for, será enviado o conteúdo correspondente ao título da notícia. Isto ocorre, pois estamos utilizando a mesma variável \$i para fazer referência ao registro na tabela. Utilizamos, mais uma vez, o recurso de concatenação de strings para enviar HTML através do comando echo do PHP.

O arquivo Conteudo.php ficará conforme código a seguir. A variável \$Texto utilizada pode ser encontrada no código anterior.

```

<html>
  <head><title></title></head>
  <body>
    <?
      echo '<span style="font-size:8.0pt;
        font-family:Verdana;color:#000080">';
      echo $Texto;
    ?>
  </body>
</html>

```

Criando áreas restritas no site

A última alteração no arquivo index.php, feita no item Links automáticos deste capítulo, listou os títulos das notícias cadastradas e estabeleceu os links para a apresentação de seus respectivos conteúdos. Isto era feito de forma livre.

Agora criaremos uma restrição em que somente os usuários logados poderão visualizar o conteúdo das notícias.

O index.php vem a seguir, com o novo código grifado.

```

<? include "Base.php"; ?>
<html>
  <head>
    <title>Criando Sites Profissionalmente</title>
  </head>
  <body>
    <table border="0" width="100%" colspacing = 0>
      <tr>
        <td width = 150>
          <IFRAME align="center" name=Banner1
            frameBorder=0 src="Banner1.php"
            scrolling=no height="110" width="100%">
            Seu navegador não oferece suporte a quadros
          </IFRAME>
        </td>
        <td bgcolor = "#FFFFC0" align = "center">
          <font style = "tahoma" color="#0000FF" size = 6>
            Criando Sites Profissionalmente
          </font>
        </td>
      </tr>
    </table>
  </body>
</html>

```

```
</font>
</td>
<td width = 150>
<IFRAME align="center" name=Banner2
frameBorder=0 src="Banner2.php"
scrolling=no height="110" width="100%">
Seu navegador não oferece suporte a quadros
</IFRAME>
</td>
</tr>
<tr>
<td valign = "top">
<table border="0" width="100%">
<tr>
<td align = "center" bgcolor = "#C0C0FF">
<font style = "tahoma" size = 2>Login
</font></td>
</tr>
<tr>
<td bgcolor = #D0D0D0>
<?
if ($HTTP_POST_VARS['entrar']==='Entrar!')
{
@session_start();
$result = mysql_query(
'select log_nome,
senha from usuarios where log_
nome
=".trim($nome).";',$conexao);
if (mysql_num_rows($result)>0)
{
$usuSenha=mysql_result($result,
0,senha);
}
}
```

```

else
{
    $usuSenha = 'notuser';
}
if ($result and ($senha==$usuSenha
    ) and ($usuSenha<>'notuser'))
{
    @session_register($nome);
    echo '<p align="center">';
    echo '<font style="Tahoma" size=
        4>';
    echo $nome.'</font><BR>';
    echo '<font style="Tahoma" size=
        2>';
    echo 'Você está on-line</font>';
    echo '</p>';
}
else
{
    echo 'Usuário não autorizado';
}
}

else
{
    echo '<form method = "post" name=
        "Logar" action = "">
<font style="Tahoma" size = 1>
Nome: <input type="text" name=
        "nome" value="" size = "15"><BR>
Senha: <input type="text" name=
        "senha" value="" size = "15"><BR>
</font>
<p align="center">
    <input onmouseover=
        this.style.cursor=
        "hand" type="submit"
        name="entrar" value="Entrar!"'
        STYLE = "border : 0; height: 18;
        width:50; background: #E8F3FD">
</p>
</form>';
}

```

```
?>
</td>
</tr>
</table>
<table border="0" width="100%">
<tr>
<td align = "center" bgcolor = "#C0FFFF">
<font style = "tahoma" size = 2>
    Quem somos
</font>
</td>
</tr>
<tr>
<td align = "center" bgcolor = "#C0FFFF">
<font style = "tahoma" size = 2>
    Fale Conosco
</font>
</td>
</tr>
<tr>
<td align = "center" bgcolor = "#C0FFFF">
<font style = "tahoma" size = 2>
    Imagens
</font>
</td>
</tr>
</table>
</td>
<td>
<IFRAME align = "center" name=Display
src = "QuemSomos.php" frameBorder=0
scrolling=1 height="280" width = "100%">
    Seu navegador não oferece suporte a quadros
</IFRAME>
</td>
<td valign = "top">
<table border="0" width="100%">
<tr>
<td align = "center" bgcolor = "#C0C0FF">
<font style = "tahoma" size = 4>
    Destaques
</font>
</td>
</tr>
<?>
```

```

$result = mysql_query('select titulo,
conteudo from noticias;',$conexao);
$Quantos = mysql_num_rows($result);
$i=0;
for ($i==0; $i<$Quantos; $i++)
{
    echo '<tr><td bgcolor="#C0FFFF">';
    echo '<span style="font-size:8.0pt;
        font-family:Verdana;color:#000080">';
    echo '<a target = "Display" href=';
    if (session_is_registered($nome))

    {
        echo '"Conteudo.php?Texto='.
            mysql_result($result,$i,Conteudo).
        '"';
        echo mysql_result($result,$i,titulo
);
        echo '</a></td></tr>';
    }
    else
    {
        echo '"Notuser.php"';
        echo mysql_result($result,$i,titulo
);
        echo '</td></tr>';
    }
}
?>
</table>
</td>
</tr>
</table>
</body>
</html>

```

- if (session_is_registered(\$nome)): esta função testa se a seção foi registrada pelo formulário de login. Se foi registrada, executará o arquivo Conteudo.php normalmente. Caso contrário, chamará Notuser.php (código a seguir):

```

<?
echo '<span style="font-size:12.0pt;
        font-family:Verdana;color:#000080">';
echo '<p align="center">';
echo '<O usuário deve ser logado!</p>';
?>

```

Este Notuser.php tem a função de jogar na tela uma mensagem avisando que o usuário

precisa ser logado.

Agora você já sabe como restringir conteúdos do site para que apenas usuários logados possam ter acesso. Teste todos os exemplos anteriores e procure entender cada comando. Dessa forma, ficará fácil editar qualquer código para incluir scripts PHP.

Existem softwares que permitem que se trabalhe com o HTML e o PHP juntos, fornecendo recursos de formatação de páginas etc. Isto é interessante, pois, a formatação da página será muito avaliada pelos visitantes de seu site.

Apresentar essas ferramentas está fora do escopo deste livro, mas tenha certeza de que os recursos do PHP apresentados aqui serão muito utilizados com elas.

Imagens automáticas

Não é difícil encontrar situações em que precisamos colocar imagens para que sejam vistas pelo internauta. Uma situação clássica é uma seqüência de fotos de um imóvel.

Para que fique fácil que estas imagens sejam vistas no site, criaremos agora uma rotina na qual todas as imagens de um determinado diretório sejam automaticamente apresentadas ao carregar a página. Esta rotina tem objetivo de evitar que o código do site seja editado para inclusão de novas imagens.

Aproveitaremos este momento para mostrar o mecanismo de uma string, como array de caracteres. Crie um arquivo chamado ListarImg.php e salve-o no mesmo diretório dos demais (código a seguir):

```
<?
$dir = dir("Imagens/");
$files = "Imagens/";
while ($entrada_arq = $dir->read())
{
    if ($entrada_arq <> '.' and $entrada_arq <>
'..')
    {
        $Quantos = strlen($entrada_arq);

        $item = $Quantos - 3;
        $Ext = $entrada_arq[$item];

        $item = $Quantos - 2;
        $Ext .= $Ext.$entrada_arq[$item];

        $item = $Quantos - 1;
        $Ext .= $Ext.$entrada_arq[$item];
```

```

if ($Ext=='jpg')
{
    echo '<img src=""';
    echo $files.$entrada_arq.'"><BR>';
}
}

$d->close();
?>

```

O arquivo que acabamos de criar abrirá na tela todas as imagens de extensão jpg que estiverem no diretório definido pela variável \$files. Lembre-se de que o PHP diferencia letras maiúsculas de minúsculas. Verifique se a extensão dos arquivos estão com letras minúsculas, conforme o código anterior.

- dir("Imagens/"): retorna o componente diretório do caminho especificado. Em nosso caso é 'Imagens/';
- \$files = "Imagens/": define o diretório onde estão as imagens a serem mostradas no site;
- while (\$entrada_arq = \$d->readO): gera um ciclo de leitura as entradas do diretório de imagens. Read() é um método do objeto \$d que representa este diretório e tem a função de ler o nome dos arquivos contidos nele;
- \$Quantos = strlen(\$entrada_arq);: armazena na variável \$Quantos a quantidade de caracteres do nome do arquivo atualmente lido pelo método read();
- \$entrada_arq[\$item];: representa a posição \$item da String \$entrada_arq;
- if (\$Ext==jpg'): cria a condição para mostrar somente arquivos de imagem com extensão jpg;
- echo \$files.\$entrada_arq: envia ao browser o arquivo com seu respectivo caminho para ser apresentado pela página.

Para que as imagens sejam mostradas no site, basta enviar os arquivos para o diretório definido pela variável \$d. Em nosso index.php, abaixo do formulário de login, substitua o código do menu a seguir:

```
<font style = "Tahoma" size = 2>
    Imagens
</font>
```

Por este:

```
<font style = "Tahoma" size = 2>
    <a href="ListarImg.php" target="Display">
        Imagens
    </a>
</font>
```

Ao rodar o index.php, é só clicar no link Imagens para ver na página as imagens do diretório "Imagens/".

Enviando e-mail

Enviar e-mail utilizando o PHP é muito simples e fácil de fazer. Utilizaremos uma função chamada mail(), que é a responsável pela realização desta operação.

Para enviar e-mail por nosso site, criaremos um formulário que permitirá enviar alguns dados, em nosso caso, nome, email e uma mensagem do usuário. Criaremos, então, um arquivo chamado FaleConosco.php e colocaremos nele o código a seguir:

```
<html>
    <head><title>Fale Conosco</title></head>
    <body>
        <form method="post" name="Contato" action="">
            <table border="0" width="400" cellspacing="0">
                <tr>
                    <td width="100%" colspan=2>
                        <FONT face="Verdana" size=1>
                            Preencha os dados abaixo e em breve
                            , entraremos em contato.<BR><BR>
                        </FONT>
                    </td>
                <tr>
                    <td width="18%">
                        <FONT face="Verdana" size=1>
                            Nome
                        </FONT>
                    </td>
                    <td>
                        <input type="submit" value="Enviar" />
                    </td>
                </tr>
            </table>
        </form>
    </body>
</html>
```

```
</FONT>
</td>
<td>
  <input type="text" name="nome" value=
    "" size = "51">
</td>
</tr>
<tr>
  <td width="18%">
    <FONT face="Verdana" size=1>
      E-mail
    </FONT>
  </td>
  <td>
    <input type="text" name="e-mail" value=
      "" size = "51">
  </td>
</tr>
<tr>
  <td width="18%">
    <FONT face="Verdana" size=1>
      Mensagem
    </FONT>
  </td>
  <td>
    <textarea name="conteudo" rows=
      "7" cols="38">
    </textarea>
  </td>
</tr>
<tr>
  <td></td>
```

```

<td>
    <BR>
    <input onmouseover=this.style.cursor=
        "hand" type="submit" name="sub" value=
        "Enviar!" STYLE="border:0;height:18;
        width:50;background:#E8F3FD">
</td>
</tr>
</table>
</form>

<?
$acao = $HTTP_POST_VARS['sub'];

$Assunto = $HTTP_POST_VARS['nome'];
$e_mail = $HTTP_POST_VARS['e-mail'];
$Conteudo = $HTTP_POST_VARS['conteudo'];

$Destino = 'seuemail@provedor.com';

if ($acao == 'Enviar!')
{
    $success = mail($Destino,$Assunto,
        $Conteudo,$e_mail);
}
?>
</body>
</html>

```

Como esta página é composta primeiramente por um formulário HTML simples, dedicaremos a explicação ao código PHP, que vem logo após este formulário.

- \$acao=\$HTTP POST VARS['suá']: a variável \$acao recebe o valor atual do botão de envio de formulário;
- \$Assunto=\$HTTP POST VARS['nome']: a variável \$Assunto recebe o valor da caixa de texto nome. Esta variável será utilizada no envio do e-mail assim como as variáveis \$e_mail e \$Conteúdo;
- \$Destino='seuemail@provedor.com': é na variável \$Destino que armazenaremos o e-mail para onde serão enviados os dados do formulário;
- if(\$acao=='Enviar'): verifica se o load da página foi executado pelo formulário;
- \$success=mail(\$Destino,\$Assunto,\$Conteudo,\$e_mail): a função mail(do PHP é utilizada neste momento para enviar o e-mail ao endereço de destino armazenado na variável \$Destino;

Agora vamos substituir um trecho do arquivo index.php, que nos permitirá enviar e-mail por meio do menu Fale Conosco. Substitua o código do menu a seguir:

```
<font style = "Tahoma" size = 2>
    Fale Conosco
</font>
```

Por este:

```
<font style = "Tahoma" size = 2>
<a href="FaleConosco.php" target="Display">
    Fale Conosco
</a>
</font>
```

Conclusão e reaproveitamento de recursos

O aproveitamento de código é bastante útil e muito utilizado nos sistemas de informação. Na técnica de orientação a objetos, temos uma estrutura abstrata chamada classe, que permite a criação de objetos com características idênticas a ela.

O objetivo de se criar classes é a utilização de seus recursos em várias partes do sistema. Isto facilita o uso e a manutenção destes recursos, pois, caso seja necessária qualquer alteração no código, será feita apenas na classe e seus respectivos objetos serão automaticamente modificados.

Para exemplificar esse conceito, criaremos um exemplo que deverá ser incluído em nosso arquivo Base.php. Abaixo dos comandos de conexão com o banco de dados, inclua o código a seguir:

```
class Fonte
{
    function Estilo()
    {
        echo '<font style="Tahoma" color=
            "#0000FF" size = 2>';
    }
}

$Fonte = new Fonte;
```

Para utilizar o código da classe anterior, basta criar um objeto e executar a função Estilo. Veremos como funciona:

- class Fonte: cria uma classe chamada Fonte;
- function Estilo(): cria uma função chamada Estilo, a qual será executada pelos objetos da

classe;

- echo '', -: é o código da função Estilo;
- \$Fonte = new Fonte; : cria um objeto da classe Fonte, o qual será utilizado para executar os códigos contidos nesta classe.

Para utilizar o código do objeto \$Fonte criado, utilizaremos o comando seguinte:

```
<? $Fonte->Estilo(); ?>
```

Isso poderá substituir as formatações de fonte do nosso site. Quando trocar por <? \$Fonte->Estilo(); ?>, basta mudar na classe quando for necessário alterar a fonte do sistema.

Criamos um exemplo bastante simples, mas lembre-se: cada classe pode ter várias funções com tamanhos bem variados. Quanto mais classes você utilizar, mais rápida será a manutenção de seu sistema.

Agora chegamos ao final de nosso livro. Esperamos ter contribuído para seu desenvolvimento em PHP, para um sistema de qualidade, rápido e eficiente.

Utilize à vontade os conceitos apresentados, e bom trabalho.