Programação para a Web utilizando

Autores:
Alexandre Arroyo
Fabio Santos

Divisão de Serviços à Comunidade
Centro de Computação
Unicamp

Licenciamento de Uso

Índice

Última atualização em 08/11/2005

O que é PHP?

A abreviação PHP vem de “Hypertext PreProcessor”, que é uma linguagem de programação de código aberto muito utilizada para a criação de scripts que são executados no servidor web para a manipulação de páginas HTML. Apesar de ser mais utilizado em aplicativos para a web, o PHP também suporta programação na linha de comando e aplicações gráficas cliente para serem executadas em interfaces gráficas através do PHP-GTK.

História

O PHP foi criado por volta de 1994 por Rasmus Lerdorf, que inicialmente utilizava-o em sua home page pessoal (Personal Home Page). Em meados de 1995 ele passou a ser utilizado por outras pessoas e foi reescrito com novos recursos, sendo renomeado para Personal Home Page Tools/FI (Form Interpreter), e entre os novos recursos, passou a contar com suporte ao mSQL. Dois anos mais tarde o PHP deixou de ser um projeto pessoal de Rasmus Lerdorf e passou a ser desenvolvido por uma equipe de colaboradores, e neste período, foi lançada a versão 3 da linguagem. A partir da versão 4 o PHP passou a utilizar a engine de scripting da Zend, para melhorar a performance e suportar uma variedade maior de bibliotecas externas e extensões. Até Agosto de 2003, o PHP estava sendo utilizado em aproximadamente 13.000.000 de domínios (Pode-se acompanhar esta estatística em http://www.php.net/usage.php).

Vantagens

O PHP tem inúmeras vantagens, como veremos a seguir:

- É uma linguagem de fácil aprendizado;
- Tem performance e estabilidade excelentes;
- Seu código é aberto, não é preciso pagar por sua utilização, e é possível alterá-lo na medida da necessidade de cada usuário;
- Tem suporte nos principais servidores web do mercado, principalmente no servidor web Apache (o mais utilizado no mundo);
- Suporta conexão com os bancos de dados mais utilizados do mercado, como por exemplo, MySQL, PostgreSQL, Oracle e DB2;
- É multiplataforma, tem suporte nos sistemas operacionais mais utilizados no mercado;
- Suporta uma variedade grande de padrões e protocolos, como o XML, DOM,
- IMAP, POP3, LDAP, HTTP, entre outros;
- Não precisa ser compilado.

Sintaxe Básica

O PHP tem uma sintaxe muito simples e enxuta, o que facilita muito a organização dos scripts a serem desenvolvidos. Outro ponto interessante que veremos é que os códigos em PHP são embutidos no HTML, ao invés de gerá-lo por completo, facilitando muito a análise de possíveis erros nos scripts desenvolvidos. A seguir, exemplos da sintaxe do PHP:

1	2	3	4
<?php ... . ?>	<? .... ?>	<% .... %>	<script language="PHP"> .... </script>

Variáveis

Manipular variáveis em PHP é uma atividade simples, como veremos a seguir:

- não é necessário declarar as variáveis, isto é feito quando atribuímos algum valor para elas;
- para declará-las, é necessário apenas colocar como primeiro caracter o '$' , juntamente com a string referente ao nome da variável, e esta string deve começar com uma letra ou o caracter '_';
- PHP é case sensitive, isto é, '$a' é diferente de '$A'. É aconselhável utilizar os nomes de variáveis com letras minúsculas, por causa das variáveis pré-definidas da linguagem, que são declaradas com maiúsculas;

- PHP suporta os seguintes tipos de variáveis:

  - inteiros (integer ou long);
  - ponto flutuante (double ou float);
  - strings
  - arrays
  - objetos *

**** Como se trata de um curso básico, não entraremos em detalhes sobre este tipo***

Tipos suportados

- *Inteiros*

  Sintaxe:

```
$curso = 1000;
$curso = -1000;
$curso = 0234; (inteiro base octal)
$curso = 0x34; (inteiro na base hexadecimal)
```

- *Ponto flutuante*

  Sintaxe:

```
$curso = 1.050;
$curso = 52e3; (equivale a 52000)
```

- *Strings*

Sintaxe:

```
$curso = 'PHP';

# desta maneira, o valor da variável será exatamente o texto
contido entre as aspas

$curso= “PHP”;

# desta maneira, qualquer variável ou caracter de escape será
expandido antes de ser atribuído
```

- *Caracteres de Escape*

\n	nova linha;
\r	retorno de carro (semelhante a \n)
\t	tabulação horizontal
\\	a própria barra (\)
\$	o símbolo $
\’	aspas simples
\”	aspas duplas

- *Arrays* : Array é um tipo de variável que possui seu conteúdo agrupado por índices, como um vetor ou um dicionário. Estes índices podem ser de qualquer tipo suportado pelo PHP, como é mostrado a seguir:

Sintaxe:

```
$estilo_musical[0] = 'pagode';
$estilo_musical[1] = “drum \'n\' bass”;
$estilo_musical[“MPB”] = 'Gilberto Gil';
$estilo_musical[“Rock”] = 'Blind Guardian';
```

- *Listas* : Utilizadas em PHP para realizar atribuições múltiplas, como por exemplo, atribuir valores de um array para variáveis, como mostra a seguir:

  Sintaxe:

```
list($a,$b,$c) = array(0=>”a”, 1=>”b”, 2=>”c”);
```

O trecho de código acima atribuirá simultânea e respectivamente os valores do array às variáveis passadas como parâmetros para o comando **`list`**. É muito importante lembrar que só serão passadas ao comando **`list`** os elementos do array que possuírem os índices com valores inteiros e não negativos.

- *Booleans* : Em PHP, não existe um tipo específico para as variáveis do tipo **boolean**, ele trata este tipo com valores inteiros: `0` (zero) para false e valores diferentes deste como true.

Transformações de tipos

É possível fazer transformações de tipos de variáveis através das seguintes formas:

- *Coerções* : quando ocorrem determinadas operações matemáticas entre dois valores de tipos diferentes, como por exemplo a adição, o PHP converte um deles automaticamente. Um exemplo disso seria a conversão de uma string para um valor numérico (inteiro ou ponto flutuante), que segue as seguintes regras:
  - É analisado o ínicio da string, se contiver um número, ele será analisado, caso contrário, o valor será 0 (zero);
  - O número pode conter o sinal no início (+ ou -);

- Se a string contiver um ponto em sua parte numérica a ser analisada, ele será considerado, e o valor obtido será um ponto flutuante;

- Se a string contiver as letras "e" ou "E" em sua parte numérica a ser analisada, o valor seguinte será considerado como expoente da base 10, e o valor obtido será um ponto flutuante.

Exemplo de sintaxe:

```
$curso = 1 + “12.8”; ($curso == 13.8)
$curso = 1 + “15”; ($curso == 16)
$curso = 1 + “1.5e3”; ($curso == 1501)
$curso = 1 + “10curso”; ($curso == 11)
$curso = 1 + “  10curso”; ($curso == 11)
$curso = 1 + “+A10testes”; ($curso == 1)
```

- *Transformações explícitas de tipos* : desta forma precisaremos utilizar a sintaxe de typecast do PHP, como os exemplos a seguir:

```
$curso = 20;  (integer(20))
$curso = (double)$curso;  (double(20.0))
$curso = 3.9;  (double(3.9))
$curso = (int)$curso (o valor é truncado e fica como integer(3))
```

- *Tipos suportados nas transformações explícitas:*

```
(int), (integer) =  muda para inteiro;
(real), (double), (float) = muda para ponto flutuante;
(string) = muda para string
(array) = muda para array
(object) = muda para objeto
```

- *Função settype* : trabalha igualmente as tranformações explícitas, porém com sintaxe diferente, como o exemplo a seguir:

```
$curso = 20;  (integer)
settype($curso, double);

# o valor da variável $curso foi transformada em ponto flutuante
```

Operadores

- *Aritméticos:*

+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Módulo

- *Strings:*

.	Concatenação

- *Atribuição:*

=	Atribuição simples
+=	Atribuição com adição
-=	Atribuição com subtração
*=	Atribuição com Multiplicação
/=	Atribuição com divisão
%=	Atribuição com módulo
.=	Atribuição com concatenação

Exemplo:

```
$curso = 7;
$curso += 2;  ($curso fica com o valor 9)
```

- *Lógicos:*

and	"e" lógico
or	"ou" lógico
xor	"ou" exclusivo
!	Não (inversão)
&&	"e" lógico
||	"ou" lógico

- *Comparação:*

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

- *Incremento e decremento:*

++	incremento
--	decremento

Estes podem receber o valor antes ou depois da variável:

- *Antes:* retorna o valor da variável antes de incrementá-la ou decrementá-la:

  Exemplo:

```
$a = 1;
$b = ++a; ($b recebe 2, valor de $a já incrementado)
```

- *Depois:* retorna o valor da variável já incrementada ou decrementada:

  Exemplo:

```
$a = 1;
$b = a++; ($b recebe 1  e $a passa a ter 2)
```

Estruturas de controle

- ***If*** : O comando **if** testa a condição passada e executa o bloco de código caso o valor retornado da condição seja verdadeiro:

```
$a = 1;

if ($a == 1)
{
....
....
...
}
```

Caso a condição passada retorne um valor falso, e seja necessário executar um bloco de código diferente, utiliza-se a instrução **else*:***

```
$a = 1;
$b = 2;

if ($a > $b)
   {
     ....
     ...
   }
else
   {
    .....
    ....
   }
```

Ainda existe a instrução **elseif,** para situações onde precisa-se verificar mais que uma condição:

```
$a = 1;
$b = 2;
$c = 3;

if ($a > $b)
   {
      echo ‘$a é maior que $b ‘;
   }
elseif ($a > $c)
    {
      echo ‘ $a é maior que $c ‘;
    }
else
    {
      echo ‘ $a é menor que $b e $c ‘;
    }
```

- ***Switch*** : Comando utilizado para fazer múltiplos testes de condição. A idéia deste comando é igual ao do **elseif** , porém com algumas diferenças:

```
$a = 1;

switch ($a)
{
    case 0:
         echo ‘ $a é igual a 0 ‘;
         break;
    case 1:
         echo ‘ $a é igual a 1 ‘;
         break;
    case 2:
         echo ‘ $a é igual a 2 ‘;
         break;
    default:
         echo “ \$a é igual a $a “;
}
```

A idéia do comando **switch** é achar a condição verdadeira e executar qualquer bloco de código que esteja abaixo dela, inclusive os que não forem do seu trecho, por esse motivo, utilizamos o comando **break** logo abaixo da última linha do bloco de código, como o exemplo anterior. O comando **switch** também aceita testes de condição em qualquer tipo de variável suportado pelo PHP:

```
$a = “curso“;

switch ($a)
{
    case “PHP“:
         echo ‘ $a é igual a PHP ‘;
         break;
    case “curso“:
         echo ‘ $a é igual a Curso ‘;
         break;
    case “CCUEC“:
         echo “ \$a é igual a CCUEC “;
         break;
}
```

- ***While*** : Este comando é utilizado para realizar laços condicionais. Ele executa o bloco de código enquanto a condição passada for verdadeira, e caso a condição inicial que foi passada se torne falsa, o bloco não será executado:

```
$a = 1;

while ($a <= 10)
{
  echo “Número“.$a++.“<br>“;
}
```

- ***Do..while*** : Este comando tem a mesma idéia que o comando **while**, porém, seu teste de condição é feito no final do bloco de código:

```
$c = 0;
do
{
    echo “Número“.++$c.“<br>“;
} while ($c < 10);
```

- ***For*** : Como nos outros comando que realizam laços condicionais, o comando **for** também precisa de uma condição para ser testada a cada laço realizado, porém, este comando necessita de mais dois parâmetros, que seriam a declaração da variável contadora e a instrução de incremento:

```
for ($a=0; $a<=10; $a++)
    {
        echo “Número”.$a.”<br>”;
    }
```

Quebra de fluxo

- ***Break*** : O comando **break** pode ser utilizado em comandos de laços condicionais e no comando **switch**, e sua função é parar imediatamente a execução do laço condicional, prosseguindo normalmente com a execução do script:

```
$a = 20;

while ($a > 0)
    {
        if ($a == 3)
             {
           echo “Número inválido!”;
     break;
             }
        echo “Número  “.$a.”<br>”;
        $a--;
    }
```

- ***Continue*** : O comando **continue** também funciona dentro dos laços condicionais, porém, não pára o fluxo do bloco de código, e sim, volta para o início dele:

```
for ($a=0;$a<=10;$a++)
{
  if ($a == 5)
    {
      echo "<p>Pulou o Numero ==> $a</p>";
      continue;
    }

  echo "Numero ==> $a<br>";

}
```

Saída: echo()

A função **echo** faz a impressão de um ou mais argumentos na janela do navegador.
Exemplos:
Echo “Essa instrução irá imprimir no navegador.”;
Echo (“Também pode-se usar parênteses.”);

Para a construção sem o uso de parênteses, é possível passar mais de um argumento para a função:
Echo “Primeiro argumento”, “Segundo argumento” ;

Também existe uma construção abreviada do **echo** que possibilita alternar entre PHP e HTML rapidamente.
<P>Aqui é HTML **<?=**”Aqui é PHP”**?>**</P>
Essa forma é mais utilizada em formulários.

Saída: print()

A função **print** é bem semelhante a **echo**, com duas diferenças: print aceita apenas 1 argumento e além da impressão no navegador, print retorna 1 em caso de sucesso e 0 em caso de falha na tentativa de impressão.
Exemplos:
Print “Funciona igual a função echo.” ;
Print (“Também pode-se usar parênteses.”) ;
Print (‘Também pode-se usar aspas simples.’) ;

Funções

Funções são pequenas seções independentes de código que podem ser chamadas a qualquer momento e em qualquer ordem, que servem para desempenhar tarefas específicas dentro dos scripts. O exemplo a seguir mostra a sua sintaxe básica:

```
function soma ($a,$b)
{
     $c = $a + $b;
     return $c;
}

echo "A funcao soma() retornou ==> ".soma(5,10);
```

A instrução **`return`** é opcional, já que não é obrigatório retornar algum valor em funções no PHP, outra regra é a de não permitir que sejam retornados múltiplos valores através desta instrução. Para resolver essa necessidade, pode-se retornar listas e arrays, como mostra o exemplo a seguir:

```
function soma ($a, $b)
{
     $c = $a + $b;
     $d = $c - 5;
     return array($c,$b,$d)
}

list ($f,$g,$h) = soma(10,10);

echo $f.”<br>”;
echo $g.”<br>”;
echo $h.”<br>”;
```

- *Passagem de parâmetros por referência* : Normalmente, a passagem de parâmetros em PHP é feita através dos valores das variáveis, não permitindo assim, a alteração do valor na variável original, como mostra o exemplo a seguir:

```
function contador($a)
{
  ++$a;
}

$cont = 10;
contador($cont);

echo "A variavel <b>$cont</b> contem ==> ".$cont;
```

No exemplo acima, a variável original permanecerá com o mesmo valor porque não foi definida a passagem de parâmetros por referência, o que alteraria também o valor da variável original. Uma das maneiras de se utilizar esse recurso é colocar o caracter "&" antes do nome da variável na declaração da função, como mostra o exemplo a seguir:

```
function contador(&$a)
{
  ++$a;
}

$cont = 10;
contador($cont);

echo "A variavel <b>$cont</b> contem ==> ".$cont;
```

Poderíamos também utilizar a passagem de parâmetros por referência apenas quando fossemos chamar a função, e não em sua declaração:

```
contador(&$cont);
echo $cont;
```

Escopo das variáveis

Discutimos anteriormente sobre variáveis e os tipos suportados pelo PHP. Agora, discutiremos sobre os escopos destas variáveis, que podem ser dos seguintes tipos:

- globais;
- locais;
- estáticas;
- constantes.

- *Globais*: As variáveis globais são por definição, as variáveis que podem ser acessadas dentro de todo o script. Porém, quando cria-se escopos locais como nas funções, precisaremos utilizar um tipo de chamada especial, como no exemplo a seguir:

```
$curso = 'PHP';

function mostra()
{
     global $curso;
     echo $curso;
}

mostra();
```

O mesmo recurso pode ser acessado através da array **GLOBALS**, que nos permite acessar todas as variáveis globais do script. O exemplo acima pode ser reescrito da seguinte maneira:

```
$curso = 'PHP';

function mostra()
{
     echo $GLOBALS[“curso“];
     echo $curso;
}

mostra();
```

- *Locais*: As variáveis locais são o tipo mais restrito dentro do PHP. Elas funcionam apenas dentro deste escopo, como mostra o exemplo a seguir:

```
$curso = 'PHP';

function mostra()
{
     $var_local = 'variável local';
     echo $var_local;
}

echo “<b>$var_local</b>”;
```

- *Estáticas*: As variáveis estáticas são variáveis que possuem o mesmo tempo de vida das variáveis globais, com a diferença de funcionarem apenas em escopos locais e serem inicializadas uma só vez. A seguir, um exemplo deste recurso:

```
function contador()
{
    static $i = 0;
   echo $i++.“<br>“;
}
```

```
for ($a=0; $a<=5; $a++)
{
    contador();
}
```

Sessões

Este recurso, que foi implementado na versão 4 do PHP, é muito útil para quem trabalha com scripts que necessitam passar dados em acessos subsequentes para outros scripts. Sessões também são utilizadas para:

- Customização de elementos de uma página, como cores, fontes, textos, etc;
- Gerenciamento de autenticação em sistemas para a web;
- Armazenamento de informações sigilosas dentro do servidor, evitando a passagem destas informações por meio de campos do tipo **`hidden`** do HTML ou cookies, aumentando assim a segurança destes dados.

Esse recurso já vem habilitado na instalação padrão do PHP, não havendo a necessidade de nenhuma configuração adicional, e antes de inicializar uma sessão, devemos lembrar das seguintes regras básicas de utilização:

- Nenhum conteúdo deve ser exibido antes de inicializar uma sessão;
- Em todas as páginas que forem utilizar este recurso, a sessão deve ser inicializada;

Para inicializar uma sessão, basta executar o seguinte comando:

```
<?php

session_start();

?>
```

Quando esta página for carregada, a sessão será inicializada e a ID da sessão ficará gravada em um cookie chamado PHPSESSID dentro do navegador. Esta sessão será válida enquanto o navegador estiver aberto ou enquanto a função **`session_destroy()`** não for executada.

Para visualizarmos a ID da sessão corrente, utilizamos a função **`session_id()`**, como mostra o exemplo abaixo:

```
<?php

session_start();

$id_sess = session_id();

echo "A ID da sessão corrente é ====> <b>$id_sess</b>";

?>
```

Superglobal `$_SESSION`

Adicionar valores em uma sessão é uma tarefa muito simples, e para isso, utilizaremos a superglobal **`$_SESSION`**, como mostra o exemplo abaixo:

```
<?php

session_start();

$_SESSION["curso"] = "PHP Intermediário";
$teste = "Teste de Sessões!"
$_SESSION["teste"] = $teste;

?>
```

OBS: Em versões anteriores ao PHP 4.2.x, ou se a flag `register_globals` estiver habilitada (desabilitada por padrão), é necessário adicionar variáveis em versões através da função `session_register()`, mas por questões de segurança, é indicado permanecer com a configuração padrão, que utiliza a superglobal `$_SESSION`.

Atribuir os valores de uma sessão para variáveis globais ou locais também é uma tarefa simples:

```
<?php

session_start();

$curso = $_SESSION["curso"];
$teste = $_SESSION["teste"];

?>
```

Os últimos dois recursos que serão apresentados para manipular sessões são:

`session_unset()` - Limpa todas as variáveis da sessão corrente.

`session_destroy()` - Finaliza a sessão corrente.

Utiliza-se estas duas funções no final do uso da sessão, que consequentemente expira o cookie PHPSESSID no navegador, finalizando a sessão corrente. O código PHP

para este fim é apresentado no exemplo abaixo:

```
session_unset();
session_destroy();
```

Upload de Arquivos

O PHP é capaz de receber o upload de qualquer navegador que siga a norma RFC-1867. Isto permite que se faça upload de arquivos de texto de binários.
Com as funções de autenticação e manipulação de arquivos do PHP, você tem o controle completo de quem pode fazer o upload de arquivo e o que fazer com o arquivo após seu upload. Abaixo, um exemplo de um formulário HTML para realizar esta tarefa:

```
<form enctype="multipart/form-data" action="_URL_" method="POST">
<input type="hidden" name="MAX_FILE_SIZE" value="30000">
Send this file: <input name="arquivo" type="file">
<input type="submit" value="Enviar Arquivo">
</form>
```

O atributo **enctype** da tag **<form>** é o mais importante para a realização de um upload, sem ele, este recurso não irá funcionar. O campo do tipo **file**, **MAX_FILE_SIZE**, indica ao navegador o tamanho máximo do arquivo a ser enviado.
Agora veremos mais sobre como o PHP processa o upload de arquivo dentro do servidor:

A variável $_FILES

Este array nos fornece as informações sobre o arquivo que o navegador enviou ao servidor. Abaixo, a lista dos valores que este array nos disponibiliza.

`$_FILES['arquivo']['name']`	o nome original do arquivo no computador do usuário.
`$_FILES['arquivo']['type']`	o MIME type do arquivo, se o navegador deu esta informação: ex. `image/gif`
`$_FILES['arquivo']['size']`	o tamanho em bytes do arquivo
`$_FILES['arquivo']['tmp_name']`	o nome temporário do arquivo, como foi guardado no servidor.
`$_FILES['arquivo']['error']`	o código de erro associado a este upload de arquivo. Adicionado no PHP 4.2.0

Os códigos de erro que o item `$_FILES['arquivo']['error']` pode retornar são:

UPLOAD_ERR_OK	0	não ouve erro, o upload foi bem sucedido
UPLAOD_ERR_INI_SIZE	1	O arquivo no upload é maior do que o limite definido em `upload_max_filesize` no `php.ini`
UPLOAD_ERR_FORM_SIZE	2	O arquivo ultrapassa o limite de `MAX_FILE_SIZE` que foi especificado no formulário HTML
UPLOAD_ERR_PARTIAL	3	O upload do arquivo foi feito parcialmente
UPLOAD_ERR_NO_FILE	4	Não foi feito upload do arquivo

A função `move_uploaded_file()`

Quando é realizado um upload de arquivo para o servidor web, este arquivo fica em um diretório temporário, normalmente em `/tmp` (podemos alterar este diretório no `php.ini`), e caso não seja inicializada nenhuma ação para manipular este arquivo, ele será apagado deste diretório ao fim da execução do script. Para esta tarefa, o PHP disponibiliza a função `move_uploaded_file`, a qual descreveremos a seguir:

`move_uploaded_file()` - Esta função irá mover um arquivo carregado pelo mecanismo do PHP de HTTP POST para uma nova localização. Sua sintaxe é:

```
move_uploaded_file(string nome do arquivo, string destino);
```

OBS: O diretório de destino deve ter permissão de escrita para o PHP.

Abaixo, um exemplo de um script PHP que receberá os dados do método POST para realizar o upload:

```
<?php

$uploaddir = '/usr/local/apache/htdocs/curso_php/upload_arq/';
$uploadfile = $uploaddir. $_FILES['userfile']['name'];
print "<pre>";

if ($_FILES['userfile']['size'] != 0)
{
   if (move_uploaded_file($_FILES['userfile']['tmp_name'], $uploaddir .
$_FILES['userfile']['name']))
        {
        print "O arquivo é valido e foi carregado com sucesso. Aqui esta
alguma informação:\n";
        print_r($_FILES);
        }
        else
        {
        print "Possivel ataque de upload! Aqui esta alguma informação:\n";
        print_r($_FILES);
        }
}
else
```

```
{
        print "Possivel ataque de upload! Aqui esta alguma informação:\n";
        print_r($_FILES);
}

print "</pre>";
?>
```

Manipulação de Data e Hora

Neste capítulo iremos mostrar algumas funções interessantes que o PHP nos disponibiliza para manipulação de data e hora. As mais utilizadas e importantes são as seguintes:

- mktime()
- strftime()
- date()
- checkdate()
- getdate()

`mktime()` - retorna uma data com o formato Unix/timestamp:

Sintaxe:

```
mktime(int hora, int minuto, int segundo, int mês, int dia, int ano);
```

O valor de retorno precisa passar por uma formatação para poder ser apresentado, para isso, usamos a função `strftime()`:

```
$e = mktime(13,23,32,12,21,2003);
echo strftime("Data de Hoje: %d/%m/%Y - %T",$e);
```

Saída:

```
Data de Hoje: 21/12/2003 - 13:23:22
```

strftime() - formata uma data do tipo Unix/timestamp para uma string de apresentação:

Sintaxe:

```
strftime(string formatação, int timestamp);
```

```
$data = mktime(0,0,0,12,21,2003);
echo strftime("Data: %d-%m-%Y",$data);
```

Uma lista das principais opções para esta função encontra-se na listagem abaixo:

%d	dia do mês (01 a 31)
%m	mês (01 a 12)
%y	ano com 2 dígitos (ex. 80, 03)
%Y	ano com 4 dígitos (ex. 1980, 2003)
%H	hora atual no formato 24 horas (00 a 23)
%I	hora atual no formato 12 horas (01 a 12)
%M	minuto atual (00 a 59)
%S	segundo atual (00 a 59)
%R	hora no formato 24 horas (22:45)
%x	data sem mostrar a hora do formato Unix/timestamp (ex. 21/12/2003)
%X	horário sem mostrar a data do formato Unix/timestamp (ex. 19:33)

`date()` - retorna uma data formatada como uma string ou inteiro de acordo com as opções passadas como parâmetro. Uma lista das principais opções para esta função encontra-se abaixo:

D	representação numérica do dia do mês, incluindo o 0 (zero), nos dias de 0 a 9: (01 a 31)
J	representação numérica do dia do mês, sem o 0 (zero) nos dias de 1 a 9: (1 a 31)
M	representação numérica do mês, incluindo o 0 (zero) nos meses de 1 a 9: (01 a 12)
N	representação numérica do mês, sem o 0 (zero) nos meses de 1 a 9: (1 a 12)
F	representação textual do mês: (Janeiro a Dezembro)
T (min)	retorna o número de dias que o mês contém: (ex. 28, 29, 30 ou 31)
y (min)	representação numérica do ano, com 4 dígitos (ex. 1980, 2003)
Y (mai)	representação numérica do ano, com 2 dígitos; (ex. 80, 03)
I	representação textual do dia da semana: (ex. Domingo a Sábado)
w (min)	representação numérica do dia da semana: (0 para Domingo e 6 para Sábado)
W (mai)	representação numérica da semana no ano atual: (ex. 40)
Z	representação numérica do dia do ano: (0 a 365)
L	retorna verdadeiro se o ano é bissexto, caso contrário, retorna falso: 0 - não é bissexto 1 - é bissexto

G	retorna a hora atual no formato 12 horas: (1 a 12)
H (min)	retorna a hora atual no formato 12 horas, incluindo o 0 (zero) nas horas de 1 a 9: (01 a 12)
G	retorna a hora atual no formato 24 horas: (0 a 23)
H	retorna a hora atual no formato 24 horas, incluindo o 0 (zero) nas horas de 1 a 9: (00 a 23)
I (min)	retorna o minuto atual: (00 a 59)
S (min)	retorna o segundo atual: (00 a 59)

checkdate() - verifica se a data passada é válida, retorna verdadeiro ou falso:

Sintaxe:

```
checkdate(int dia, int mes, int ano);
```

```
$a = checkdate(11,02,1980);

if ($a)
{
   echo "Data Correta<p>";
}
else
{
   echo "<font color=\"red\">Data Incorreta</font><p>";
}
```

getdate() - retorna um array com as informações de uma data no formato Unix/timestamp

Sintaxe:

```
array getdate(int timestamp);
```

```
$data_array = getdate();
print_r($data_array);
```

A lista dos itens mais importantes deste array retornado encontra-se abaixo:

"seconds"	retorna os segundos (0 a 59)
"minutes"	retorna os minutos (0 a 59)
"hour"	retorna as horas (0 a 23)
"mday"	retorna o dia do mês (1 a 31)
"wday"	retorna a representação numérica do dia da semana (0 para Domingo e 6 para Sábado)
"mon"	retorna a representação numérica do mês atual (1 a 12)
"year"	retorna a representação numérica do anual atual (2003)
"yday"	retorna a representação numérica do dia do ano (1 a 365)
"weekday"	representação textual do dia da semana (Domingo a Sábado)
"month"	representação textual do mês atual (Janeiro a Dezembro)

Enviando e-mails

Para enviar e-mails a partir de scripts PHP, é necessário que o servidor tenha instalado um aplicativo servidor de correio eletrônico, que normalmente pode ser o SendMail, Postix, QMail, ou qualquer outro que possa realizar esta função. No arquivo de configuração php.ini é necessário alterar o valor da diretiva chamado sendmail_path, onde será colocado o PATH do executável do aplicativo servidor de correio eletrônico. No nosso exemplo, vamos utilizar o SendMail, e o valor da diretiva ficaria como está apresentado abaixo:

```
sendmail_path = /usr/lib/sendmail -t -i
```

A seguir, temos dois exemplos de como podemos enviar emails a partir do PHP:

Ex 1:

```
mail("santos@ccuec.unicamp.br", "My Subject", "Line 1\nLine 2\nLine 3");
```

Ex. 2:

```
$assunto = "#ID do Usuário#";
$para = "Fabio Santos <santos@ccuec.unicamp.br>, Fabio Yahoo!
<santos@yahoo.com.br>";
$remetente = "Webmaster Mangue <santos@ccuec.unicamp.br>";

$mensagem = "Sua socilitação foi encaminhada para os responsáveis, por
favor, aguarde o retorno dos técnicos!!!";

$headers  = "To: ".$para."\r\n";
$headers .= "From: ".$remetente."\r\n";
$headers .= "Cc: ".$remetente."\r\n";
$headers .= "Bcc: ".$remetente."\r\n";

mail($para,$assunto,$mensagem,$headers);
```

Projeto

O projeto consiste no desenvolvimento de um sistema (SRC – Sistema de Registro de Chamados) cujo objetivo é registrar chamados de usuários para uma equipe de técnicos de plantão e o conseqüente atendimento desses chamados. Também vai disponibilizar uma consulta geral a partir dos dados armazenados.

Os scripts ficarão armazenados no diretório de publicação do servidor web Apache, que nesse curso encontra-se em: **`/usr/local/apache/htdocs/cursophp`**.

Se for necessário, deverão ser feitas as seguintes inicializações:

- Inicializar o servidor Apache: **`/usr/local/apache/bin/apachectl start`**
- Inicializar o MySQL: **`/etc/init.d/mysql start`**
- Inicializar o Sendmail: **`/etc/init.d/sendmail start`**

1 - Base de dados e tabelas

O sistema utilizará o servidor de banco de dados **MySQL**, no qual foi criada a base de dados **cursophp**, contendo as seguintes tabelas:

chamados

num_chamado integer primary key not null auto_increment,
data_chamado date not null,
hora_chamado time not null,
usuario varchar(40) not null,
email varchar(40) not null,
problema text not null,
equipe_acionada varchar(40) not null

atendimentos

num_chamado integer not null,
data_atendimento date not null,
responsavel varchar(40) not null,
equipe_responsavel varchar(40) not null,
solucao text not null

usuarios

login varchar(20) not null,
senha varchar(10) not null,
nome varchar(40) not null,
email varchar(40) not null,
tipo_usuario varchar(20) not null

2 – Funções utilizadas no sistema

Os scripts **funcoes.php** e **misc.php** conterão as funções utilizadas no sistema. Todos os scripts que chamarem essas funções, vão conter os comandos **include ("funcoes.php")** e **include ("misc.php")** no início do seu código.

O **include** é uma função que permite a inclusão do conteúdo de um certo arquivo em outro arquivo. Esse conteúdo pode ser qualquer tipo de código PHP, HTML ou simplesmente texto. Os arquivos a serem incluídos podem conter uma biblioteca de funções ou classes.
A função **require** tem o mesmo objetivo da função include, no entanto, em caso de erro, o include apenas mostra uma mensagem de warning e o script continuará executando, já o require causa um Fatal Error, encerrando a execução do script.

funcoes.php

```
<?php

function monta_cabecalho()
{
echo ('
<html>
<head>
<title>SRC</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#FFFFFF" text="#000000" link="#333399" vlink="#663399"
alink="#CC0000" topmargin="1">
<table width="760" border="0" cellspacing="0">
  <tr><td><img src="imagens/topo.jpg" width="760" height="57"></td></tr>
</table>
');
}


function monta_menu($login,$tip_us)
{
echo ("
<td width=\"150\" valign=\"top\" bgcolor=\"#FFFFFF\"
background=\"imagens/fundo.jpg\"> <br>
    <table width=\"135\" border=\"0\" align=\"center\" cellpadding=\"0\"
cellspacing=\"0\">
       <tr>
          <td valign=\"top\">
  <p><font size=\"1\" face=\"Verdana, Arial, Helvetica, sans-serif\">
      Login: <b>$login</b> <br><br>
             <br>
             <br>
      <a href=\"form_chamado.php\">Incluir Chamado</a><br>
             <br>
  ");

if ($tip_us == 'user_atend')
{
  echo("
             <br>
```

```
              <a href=\"form_atendimento.php\">Registrar Atendimento</a><br>
              <br>
");
}
  echo("
              <br>
              <a href=\"consulta_chamados_por_per.php\">Consultar Chamados por
Período </a><br>
              <br>
              <br>
              <a href=\"pagina_principal.php\">Página Inicial </a><br>
              <br>
              <br>
              <br>
              <a href=\"logout.php\">Logout </a><br>
              <br>
              <br>
              </font>
  </p>
  </td>
       </tr>
    </table>
</td>
");

}


function monta_rodape()
{
echo ('
<table width="760" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td><img src="imagens/rodape.jpg" width="760" height="18"></td>
  </tr>
</table>
</body>
</html>
');

}

?>
```

Misc.php

3 – Página inicial/autenticação do sistema

```
<?php

session_start();

function ver_session()
{

  if (!$_SESSION["usuario_sys"])
```

```
  {
    monta_cabecalho();
    echo "<br><br><p><font size=\"1\" face=\"Verdana, Arial, Helvetica,
    sans-serif\">Não é permitido acessar esta área antes de prévia
    autenticação!!<p><b><a href=\"index.php\">Página Principal</a></b></font>";
    monta_rodape();
    die();
  }
}

// Comando que faz a conexão com o banco de dados MySQL
   $conec = mysql_connect('localhost', 'user_curso', 'cursophp');
   mysql_select_db('cursophp', $conec);
?>
```

A página de autenticação do sistema (index.php) terá o seguinte código:

3.1) index.php

```
<?php
include ("funcoes.php");

monta_cabecalho();

?>

<table width="760" border="0" cellspacing="0">
  <tr>
   <td width="610" valign="top"><br>
     <form action="autentica_src.php" method="post">
     <table width="550" border="0" cellspacing="0" align="center">
            <tr valign="top">
         <td align="center"> <BR><font face="Verdana, Arial, Helvetica, sans-
serif" size="2"><b><font color="#264989">Autenticação</font></b></font></td>
        </tr>
        <tr valign="top" heigth="3">
          <td> <BR>   </td>
        </tr>
        <tr valign="top">
          <td align="center">
                              <?php

                                if (isset($_GET["erro_aut"]))
                                {
                                      echo ("
                                              <table align=center>
                                              <tr>
                                                <td><font face=\"Verdana, Arial,
Helvetica, sans-serif\" size=\"2\" color=\"red\">Usuário ou senha
inválidos!</font></td>
                                            </tr>
                                            <tr valign=\"top\" heigth=\"3\">
                                              <td> <BR>   </td>
                                           </tr>
                                              </table>
                                            ");
                                }

                                ?>
```

```
                        <table>
                              <tr>
                                    <td><font face="Verdana, Arial, Helvetica,
sans-serif" size="2">Usuário: </font></td><td><input type="text" name="usuario"
size="20"></td>
                              </tr>
                            <tr valign="top">
                              <td><font face="Verdana, Arial, Helvetica, sans-
serif" size="2">Senha: </font></td><td><input type="password" name="senha"
size="20"></td>
                        </tr>
                          <tr valign="top">
                              <td><input type="submit" name="submit" size="20"
value=" Autenticar "></td><td> </td>
                        </tr>
                        </table>
       </tr>
       <tr valign="top" heigth="5">
         <td> <BR>   </td>
       </tr>
       <tr valign="top">
         <td> <BR>   </td>
       </tr>
       <tr valign="top">
         <td> <BR>   </td>
       </tr>
       <tr valign="top">
         <td> <BR>   </td>
       </tr>
    </table>
      </form>
  </td>
 </tr>
</table>

<?php

monta_rodape();

?>
```

4 – Os scripts autentica_src.php e a homepage do sistema

Este script realiza a autenticação do usuário no sistema:

4.1) autentica_src.php

```
<?php

session_start();

include("misc.php");
include("funcoes.php");

$log_user = $_POST["usuario"];
$sen_user = $_POST["senha"];

// Monta a instrução sql
$instr_sql    = "select email, tipo_usuario from usuarios where login='$log_user'
and senha='$sen_user'";

// Executa a instrução sql, passando os parâmetros base de dados, instr. Sql e
// conexão
$query = mysql_query($instr_sql);

// A função mysql_fetch_assoc() busca o resultado de uma linha e o coloca numa matriz
// associativa
$reg_user_aut = mysql_fetch_assoc($query);

if ($reg_user_aut)
{
       $_SESSION["usuario_sys"] = $log_user;
       $_SESSION["email_usuario_sys"] = $reg_user_aut["email"];
       $_SESSION["tipo_usuario_sys"] = $reg_user_aut["tipo_usuario"];

       //print_r($_SESSION);

       header("Location: pagina_principal.php");
}
else
{
    header("Location: index.php?erro_aut=1");
}

?>
```

4.2) Script pagina_principal.php

Esta página aparecerá após o usuário realizar a sua autenticação no sistema:

```
<?php

session_start();

include ("misc.php");
```

```
include ("funcoes.php");

ver_session();


monta_cabecalho();

?>

<table width="760" border="0" cellspacing="0">
  <tr>

   <?php

       monta_menu($_SESSION["usuario_sys"],$_SESSION["tipo_usuario_sys"]);

   ?>

   <td width="610" valign="top"><br>
     <table width="550" border="0" cellspacing="0" align="center">
        <tr valign="top">
          <td> <BR><font face="Verdana, Arial, Helvetica, sans-serif"
size="2"><b><font color="#264989">Objetivo </font></b></font></td>
        </tr>
        <tr valign="top">
          <td><BR><font face="Verdana, Arial, Helvetica, sans-serif"
size="2">Esse sistema registra e armazena informações referentes a chamados
feitos pelos usuários e atendimentos realizados pela equipe de plantão. Também
disponibiliza a consulta dos dados armazenados.</font></td>
        </tr>
        <tr valign="top">
          <td> <BR>   </td>
        </tr>
        <tr valign="top">
          <td> <BR>   </td>
        </tr>
        <tr valign="top">
          <td> <BR>   </td>
        </tr>
        <tr valign="top">
          <td> <BR>   </td>
        </tr>
     </table>
   </td>
  </tr>
</table>

<?php

monta_rodape();

?>
```

5 - Módulo de inclusão de chamados

5.1) Script `form_chamado.php`

Esse script cria o formulário que receberá os dados do chamado e terá o seguinte código:

```
<?php

session_start();

include ("funcoes.php");
include ("misc.php");

ver_session();

monta_cabecalho();

?>

<table width="760" border="0" cellspacing="0">
 <tr>

<?php

monta_menu($_SESSION["usuario_sys"],$_SESSION["tipo_usuario_sys"]);

?>

 <td width="610" valign="top"><br>
 <table width="550" border="0" cellspacing="0" align="center">
 <tr valign="top">
 <td> <font face="Verdana, Arial, Helvetica, sans-serif" size="2"><b><font
color="#264989">Incluir Chamado </font></b></font>

 <?php

 if ($_SESSION["erro"] == 1)
 {
            echo ('<BR><BR><font size="2" face="Verdana, Arial, Helvetica, sans-
serif" color="#CC0033"><b>Campos Obrigatórios não Preenchidos</b></font><BR>');
              $problema = trim($_SESSION["problema"]);
              $equipe_acionada = trim($_SESSION["equipe_acionada"]);
              $_SESSION["erro"] = 0;
 }

 ?>

 <FORM method=post action=incluir_chamado.php>
 <P><FONT FACE="Arial,Helvetica"><font size="2" face="Verdana, Arial, Helvetica,
sans-serif"><b>Usu&aacute;rio:</b></font><FONT SIZE=-1> <?php echo
$_SESSION["usuario_sys"]; ?><font face="Verdana, Arial, Helvetica, sans-serif">
<font size="3">  </font></font><font face="Verdana, Arial, Helvetica,
sans-serif" size="2"><br><br>
 <b>E-mail :</b></font>  <font face="Verdana, Arial, Helvetica, sans-
serif"><font size="2"><?php echo $_SESSION["email_usuario_sys"];
?></font></font></FONT></FONT>


<?php
 if (empty($problema))
 {
```

```
 echo ('<P><font size="2" face="Verdana, Arial, Helvetica, sans-
serif"><b>Descrição do Problema:
 </b></font> <font face="Verdana, Arial, Helvetica, sans-serif" size="3"><BR>
 <TEXTAREA name="problema" type=text rows=3 cols=40></TEXTAREA>
 </font><BR><BR> ');
 }
 else
 {
 echo ("<P><font size=\"2\" face=\"Verdana, Arial, Helvetica, sans-
serif\"><b>Descrição do Problema:
 </b>$problema</font>
<input type=\"hidden\" name=\"problema\" value=\"$problema\"> ");
 }

 if (empty($equipe_acionada))
 {
 echo ('<P><font size="2" face="Verdana, Arial, Helvetica, sans-serif"><b>Equipe
Acionada :</b> <font face="Verdana, Arial, Helvetica, sans-serif" size="2">
 <select name="equipe_acionada">
 <option>Suporte</option>
 <option>Conectividade</option>
 <option>Desenvolvimento</option>
 <option>Produ&ccedil;&atilde;o</option>
 </select>
 </font></FONT><BR> ');
 }
 else
 {
 echo ("<P><font size=\"2\" face=\"Verdana, Arial, Helvetica, sans-
serif\"><b>Equipe Acionada:
 </b>$equipe_acionada</font>
 <input type=\"hidden\" name=\"equipe_acionada\" value=\"$equipe_acionada\"><BR>
");
 }

 ?>

  <BR>   <font face="Verdana, Arial, Helvetica, sans-serif" size="2">
 <INPUT name="sub" type="SUBMIT" value="Enviar Dados"></font> </form>
 </td>
 </tr>
 </table></td>
 </tr>
</table>

<?php

monta_rodape();

?>
```

5.2) Script `incluir_chamado.php`

Esse script vai receber os dados enviados pelo formulário, vai fazer a consistência e gravar esses dados na tabela 'chamados'.
Esse script utiliza a função mail, passando como parâmetros destinatário, assunto, corpo da mensagem e cabeçalhos adicionais.

Saiba mais sobre algumas funções utilizadas no script a seguir

`trim`: tira espaços em branco de uma variável.

`header`: chama outro script, passando parâmetros e não retorna ao script chamador. Obs: Nenhum comando de exibição (echo, include, tags html) pode ser usado antes dessa rotina.

`mail`: envia mensagens por e-mail de acordo com os parâmetros utilizados.

`rawurlencode`: trata caracteres especiais de uma variável que vai ser enviada pela URL.

`or die:` expressão que pode ser usada como uma alternativa para o **`if/else`**.

```
<?php<?php

session_start();

include ("misc.php");

ver_session();

// Recebe variáveis globais

$usuario = $_SESSION["usuario_sys"];
$email = $_SESSION["email_usuario_sys"];
$problema = trim($_POST["problema"]);
$equipe_acionada = trim($_POST["equipe_acionada"]);

// Consiste campos
if (!$problema)
{
      $_SESSION["problema"] = $problema;
      $_SESSION["equipe_acionada"] = $equipe_acionada;
      $_SESSION["erro"] = 1;
      header("location: form_chamado.php");
}
else
{
   include ("funcoes.php");

 // Obtém data
 // data para exibição no formato dd/mm/aaaa
```

```
    $data_exib = date("d/m/Y");
 // data para ser gravada no banco no formato aaaa/mm/dd
    $data_chamado = date("Y").'-'.date("m").'-'.date("d");

 // Obtém data e hora da ocorrência
    $hora_chamado = date("H:i:s");

 // Inclui os dados na tabela acionamentos

 // Declaração SQL
 $sql = "INSERT into chamados values (' ','$data_chamado',
'$hora_chamado', '$usuario', '$email', '$problema', '$equipe_acionada')";

 // Roda a query e trata o resultado
 if (mysql_query ($sql))
 {

      $sql2 = "SELECT num_chamado from chamados where email = '$email'
and data_chamado = '$data_chamado' and hora_chamado = '$hora_chamado'";
      $query2 = mysql_query ($sql2) or die ("Erro no acesso ao banco");
      $row = mysql_fetch_assoc($query2);
      $num_chamado = $row["num_chamado"];

      mail ("curso1@ccuec.unicamp.br", "SRC - Um Novo Chamado foi
Inclusão",
              "Mais Informações:
               Número do Chamado: $num_chamado
               Data do Chamado: $data_exib
               Hora do Chamado: $hora_chamado
               Usuáio: $usuario
               E-mail: $email
               Problema: $problema
               Equipe Acionada: $equipe_acionada","SRC - Sistema de
Registro de Chamados
              ");

        $_SESSION["msg_exib"] = "incluir_chamado_ok";
        $_SESSION["data_exib"] = $data_exib;
        $_SESSION["hora_chamado"] = $hora_chamado;
        $_SESSION["num_chamado"]  = $num_chamado;

            header("Location: exibe_mensagem.php");

 }
 else
 {
            $_SESSION["msg_exib"] = "incluir_chamado_erro";
            header("Location: exibe_mensagem.php");
 }

}

?>
```

5.3) Criando o script que exibe mensagens: `exibe_mensagem.php`

Não devemos exibir as mensagens finais em um script que acessa banco de dados, pois, se o usuário clicar no botão “atualizar” do navegador, o script será processado novamente. Por isso, criaremos um script só para exibir as mensagens finais. Esse script será chamado por meio da função **`header`**.

```
<?php

session_start();

include ("funcoes.php");
include ("misc.php");

ver_session();

monta_cabecalho();

?>

<table width="760" border="0" cellspacing="0">
 <tr>

<?php

monta_menu($_SESSION["usuario_sys"],$_SESSION["tipo_usuario_sys"]);

?>

 <td width="610" valign="top">

 <?php

/*******************

 Incluir Chamado

 *******************/

 if ($_SESSION["msg_exib"] == "incluir_chamado_ok")
 {

      $data_exib = trim($_SESSION["data_exib"]);
      $hora_chamado = trim($_SESSION["hora_chamado"]);
      $num_chamado = trim($_SESSION["num_chamado"]);

      echo "<BR><BR>
            <font face=\"Verdana, Arial, Helvetica, sans-serif\"
size=\"2\">
            <center> <b><font color=\"#CC0033\"> Processado com Sucesso
</font></b></center><BR>
          <center>Data: $data_exib</center><BR>
          <center>Hora: $hora_chamado</center><BR>
          <center>Número do Chamado: $num_chamado </center><br><br>
```

```
          <center> <b> <a href=\"form_chamado.php\">Voltar</a> </b>
</center></font>";

    unset($_SESSION["data_exib"]);
    unset($_SESSION["hora_chamado"]);
    unset($_SESSION["num_chamado"]);
    unset($_SESSION["msg_exib"]);

 }


if ($_SESSION["msg_exib"] == "incluir_chamado_erro")
{

      echo "<BR><BR><font face=\"Verdana, Arial, Helvetica, sans-serif\"
size=\"2\">
              <center><b><font color=\"#CC0033\"> Erro no Processamento
</font></b></b> </center>
              <BR><BR><center> <b> <a
href=\"form_chamado.php\">Voltar</a> </b> </center></font>";

    unset($_SESSION["msg_exib"]);

}


/*******************

Incluir Atendimento

 *******************/

if ($_SESSION["msg_exib"] == "incluir_atendimento_ok")
{

      echo "<BR><BR><font face=\"Verdana, Arial, Helvetica, sans-serif\"
size=\"2\">
              <center><b><font color=\"#CC0033\"> Processado com Sucesso
</font></b></b> </center>
              <BR><BR><center> <b> <a
href=\"form_atendimento.php\">Voltar</a> </b> </center></font>";

    unset($_SESSION["msg_exib"]);

}


if ($_SESSION["msg_exib"] == "incluir_atendimento_erro")
{

      echo "<BR><BR><font face=\"Verdana, Arial, Helvetica, sans-serif\"
size=\"2\">
              <center><b><font color=\"#CC0033\"> Erro no
Processamento</font></b></b> </center>
              <BR><BR><center> <b> <a
href=\"form_atendimento.php\">Voltar</a> </b> </center></font>";
```

```
    unset($_SESSION["msg_exib"]);
}

 echo '</td></tr></table>';
 monta_rodape();
 ?>
```

5.4) Testando o módulo de inclusão de chamados

Deixe os campos do formulário em branco. Clique em enviar. Deverá mostrar uma mensagem de erro.
Preencha os campos do e clique em *enviar*. Deverá mostrar a mensagem “Processado com Sucesso”. Exibirá também a data/hora da inclusão e o número do chamado.

6 - Módulo de registro de atendimentos

6.1) Script form_atendimento.php

Este script cria o primeiro formulário de atendimento, utilizado para obter o número do chamado.

```
<?php

session_start();

include ("funcoes.php");
include ("misc.php");

ver_session();
monta_cabecalho();

?>

<table width="760" border="0" cellspacing="0">
 <tr>

<?php

monta_menu($_SESSION["usuario_sys"],$_SESSION["tipo_usuario_sys"]);

?>


 <td width="610" valign="top"><br>
 <table width="550" border="0" cellspacing="0" align="center">
 <tr valign="top">
 <td>
 <p><font face="Verdana, Arial, Helvetica, sans-serif" size="2"><b><font
color="#264989">Registrar Atendimento</font></b><br></p>

<?php

if ($_SESSION["erro_atend"] == 1)
{
 echo ('<BR><font size="2" face="Verdana, Arial, Helvetica, sans-serif"
color="#CC0033"><b>Preencha o número do chamado</b></font><BR><BR>');
 $_SESSION["erro_atend"] = 0;
}

?>

 <form method="post" action="registrar_atendimento.php">
 <p><b>Número do Chamado</b>:
 <input type="text" name="num_chamado" size="10" maxlength="20">
 </p>
 <p>
 <input type="submit" name="Submit" value="Enviar">
 </p>
 </form>
 <center> <b> <a href="pagina_principal.php">Home</a> </b> </center></font>
 </td>
 </tr>
 </table>
```

```
 <td>
 </tr>
</table>

<?php

monta_rodape();

?>
```

6.2) Script registrar_atendimento.php

Esse script recebe o dado do formulário, faz a consistência, obtém dados do chamado e chama o script registrar_atendimento2.php, que vai exibir o formulário completo de atendimento.

Saiba mais sobre algumas funções utilizadas no script a seguir

mysql_num_rows: obtem o número de registros que retornou do select.
mysql_fetch_assoc: obtem os campos do registro que retornou do select, usando como índice o nome do campo.
mysql_fetch_row: obtem os campos do registro que retornou do select, da mesma forma que o comando anterior, mas utiliza um índice seqüencial, começando do valor zero.
substr: seleciona partes de uma variável string, recebe como parâmetros o nome da variável, a posição inicial e o número de posições que se quer obter. Começa da posição zero.

```
<?php<?php

session_start();

include ("misc.php");

ver_session();

// Recebe variáveis globais e tira espaços em branco
$num_chamado = trim($_POST["num_chamado"]);

// Consiste nmero do chamado
if (!$num_chamado)
{
      $_SESSION["erro_atend"] = 1;
      header("location: form_atendimento.php");
}
else
{
```

```
 // Declaração do SQL
 $sql = "SELECT data_chamado, hora_chamado, problema, equipe_acionada
from chamados where num_chamado = '$num_chamado'";

 // Roda a query e verifica se encontrou registro
 $query = mysql_query($sql) or die ("Erro no acesso ao banco");
 $achou = mysql_num_rows($query);

 // Se encontrou, guarda as variáveis
 if ($achou > 0)
 {
            $reg = mysql_fetch_assoc($query);
            $data_chamado = $reg["data_chamado"];
            $_SESSION["hora_chamado"] = $reg["hora_chamado"];
            $_SESSION["problema"] = $reg["problema"];
            $_SESSION["equipe_acionada"] = $reg["equipe_acionada"];

            //Coloca a data do chamado em formato de exibição (de aaaa-
mm-dd para dd/mm/aaaa)
            $dt = explode("-",$data_chamado);
            $dt2 = mktime(0,0,0,$dt[1],$dt[2],$dt[0]);
            $data_exib = date("d/m/Y",$dt2);

            $_SESSION["num_chamado"] = $num_chamado;
            $_SESSION["data_exib"] = $data_exib;

            header("location: registrar_atendimento2.php");
 }
 else
 {
            include ("funcoes.php");

            monta_cabecalho();

            echo '<table width="760" border="0" cellspacing="0"> <tr>';

      monta_menu($_SESSION["usuario_sys"],$_SESSION["tipo_usuario_sys"]);

            echo '<td width="610" valign="top"><br>
                  <font face="Arial"><font size="2" face="Verdana, Arial,
Helvetica, sans-serif"><BR><BR>
                  <center><font color="#CC0033"> <b> Número de chamado
nao cadastrado</b></font></center><BR><BR>
                  <center> <b> <a href="form_atendimento.php">Voltar</a>
</b> </center>
                  </font><BR><BR><BR><BR>
                  </td>
                  </tr>
                  </table>';

                  monta_rodape();
 }

}
?>
```

6.3) Script registrar_atendimento2.php

Esse script vai exibir dados do chamado e criar o formulário completo de atendimento.

```
<?php<?php

session_start();

include ("funcoes.php");
include ("misc.php");

ver_session();

$num_chamado = $_SESSION["num_chamado"];
$usuario = $_SESSION["usuario_sys"];
$email = $_SESSION["email_usuario_sys"];
$data_exib = $_SESSION["data_exib"];
$hora_chamado = $_SESSION["hora_chamado"];
$problema = $_SESSION["problema"];
$equipe_acionada = $_SESSION["equipe_acionada"];

monta_cabecalho();

echo '<table width="760" border="0" cellspacing="0"><tr>';

monta_menu($_SESSION["usuario_sys"],$_SESSION["tipo_usuario_sys"]);

echo '<td width="610" valign="top"><br>
      <table width="550" border="0" cellspacing="0" align="center">
      <tr valign="top">
      <td>
      <font face="Arial"><font size="2" face="Verdana, Arial, Helvetica,
sans-serif"><b><font color="#264989">Registrar
Atendimento</font></b></font></font><BR><BR>
      ';

 // essas condições são válidas para quando esse script for chamado pelo
registrar_atendimento3.php

 if ($_SESSION ["consist"] == 1 || $_SESSION ["consist"] == 2)
 {
            $dia = trim($_SESSION["dia"]);
            $mes = trim($_SESSION["mes"]);
            $ano = trim($_SESSION["ano"]);
            $responsavel = trim($_SESSION["responsavel"]);
            $equipe_responsavel = trim($_SESSION["equipe_responsavel"]);
            $solucao = trim($_SESSION["solucao"]);

      if ($_SESSION["consist"] == 1)
      {
            echo ('<BR><BR><font face="Verdana, Arial, Helvetica, sans-
serif" size="2"><b><font color="#CC0033">Campo(s) obrigatorio(s) nao
preenchido(s)</font></b></font><p>');
            unset($_SESSION["consist"]);
      }
```

```
      if ($_SESSION["consist"] == 2)
      {
            echo ('<BR><BR><font face="Verdana, Arial, Helvetica, sans-
serif" size="2"><b><font color="#CC0033">Data
invalida</font></b></font><p>');
            unset($_SESSION["consist"]);
      }
}

echo "
 <font size=\"2\" face=\"Verdana, Arial, Helvetica, sans-serif\">
 <b> Numero do chamado: </b>$num_chamado<BR>
 <b> Usuario: </b> $usuario <BR>
 <b> E-mail: </b> $email <BR>
 <b> Data: </b> $data_exib    <BR>
 <b> Hora: </b> $hora_chamado <BR>
 <b> Problema: </b> $problema <BR>
 <b> Equipe acionada: </b> $equipe_acionada <BR><BR><HR>
 </font>
<form method=post action=registrar_atendimento3.php>
 <p><font face=\"Verdana, Arial, Helvetica, sans-serif\"
size=\"2\"><b>Data do Atendimento </b> <BR>(formato dd/mm/aaaa)</font>
 <table BORDER width=\"580\" >
      <tr>
       <td><font face=\"Verdana, Arial, Helvetica, sans-serif\"
size=\"2\">Data : 
            <input name=\"dia\" type=\"TEXT\" value=\"$dia\" size=\"2\"
maxlength=2 >   
            <input name=\"mes\" type=\"TEXT\" value=\"$mes\" size=\"2\"
maxlength=2 >   
            <input name=\"ano\" type=\"TEXT\" value=\"$ano\" size=\"4\"
maxlength=4 >
            </font>
     </td></tr></table><br>";

if (!$responsavel)
{
      echo '<table width="400">
                  <tr>
                  <td><font face="Verdana, Arial, Helvetica, sans-serif"
size="2">
                  <b>Respons&aacute;vel :</b></font>
                  <select NAME="responsavel" >
                        <option value="Paulo">Paulo</option>
                        <option value="Carlos">Carlos</option>
                        <option value="Eduardo">Eduardo</option>
                        <option value="Cristina">Cristina</option>
                        <option value="Jose">Jose</option>
                        <option value="Marcos">Marcos</option>
                        <option value="Ricardo">Ricardo</option>

                  </select>
                  </td>
                  </tr>
                  </table>';
}
else
```

```
{
 echo "<table width=\"400\">
            <tr>
                  <td><font face=\"Verdana, Arial, Helvetica, sans-
serif\" size=\"2\">
                  <b>Respons&aacute;vel :</b> $responsavel
</font><br><br>
                  <input type=\"hidden\" name=\"responsavel\"
value=\"$responsavel\">
                  </td>
            </tr>
            </table>";
}

if (!$equipe_responsavel)
{
 echo '<table width="400">
            <tr>
                  <td><font face="Verdana, Arial, Helvetica, sans-serif"
size="2"><b>Equipe Respons&aacute;vel:</b></font>
                        <select NAME="equipe_responsavel">
                              <option value="suporte">Suporte</option>
                              <option
value="conectividade">Conectividade</option>
                              <option
value="desenvolvimento">Desenvolvimento</option>
                              <option value="producao">Producao</option>
                        </select>
                  </td>
            </tr>
            </table>';
}
else
{
      echo "<table width=\"400\">
                  <tr>
                        <td><font face=\"Verdana, Arial, Helvetica, sans-
serif\" size=\"2\"><b>Equipe Respons&aacute;vel:
                        </b>$equipe_responsavel</font>
                        <input type=\"hidden\" name=\"equipe_responsavel\"
value=\"$equipe_responsavel\">
                        </td>
                  </tr>
                  </table>";
}

if (!$solucao)
{
      echo '<p><b><font face="Verdana, Arial, Helvetica, sans-serif"
size="2">Solu&ccedil;&atilde;o :</font></b> <br>
               <textarea name="solucao" type=text rows=5
cols=50></textarea>';
}
else
{
      echo "<p><font face=\"Verdana, Arial, Helvetica, sans-serif\"
size=\"2\"><b>Solu&ccedil;&atilde;o :</b> $solucao<br></font>
```

```
              <input type=\"hidden\" name=\"solucao\"
value=\"$solucao\">";
}

// Passa os dados do chamado para o script seguinte por meio de campos
escondidos
echo "<input type=\"hidden\" name=\"num_chamado\" value=\"$num_chamado\">
        <input type=\"hidden\" name=\"email\" value=\"$email\">
            <center>
            <p><font face=\"Verdana, Arial, Helvetica, sans-serif\"
size=\"2\">
            <input name=\"sub\" type=\"SUBMIT\" value=\"Enviar Dados\">
            </font>
            <br></font></center>
            </form>
            </td>
            </tr>
            </table>
            </td>
            </tr>
            </table>";

monta_rodape();

?>
```

Saiba mais sobre algumas funções utilizadas no script a seguir

checkdate: faz consistência de datas, inclusive checa anos bissextos. Recebe como parâmetros: mês, dia e ano (nessa ordem). Os valores dos parâmetros tem que ser inteiros.

6.4) Script registrar_atendimento3.php

Esse script vai consistir os campos do formulário de atendimento, incluir os dados na tabela 'atendimentos' e enviar uma mensagem com os dados do atendimento.

```
<?php<?php

session_start();

include ("funcoes.php");
include ("misc.php");

ver_session();
```

```
// Recebe variáveis globais
$num_chamado = trim($_POST["num_chamado"]);
$email = trim($_POST["email"]);
$dia = trim($_POST["dia"]);
$mes = trim($_POST["mes"]);
$ano = trim($_POST["ano"]);
$responsavel = trim($_POST["responsavel"]);
$equipe_responsavel = trim($_POST["equipe_responsavel"]);
$solucao = trim($_POST["solucao"]);

// Consiste data
$var_data = checkdate((int)$mes, (int)$dia, (int)$ano);

// Consiste se campos obrigatórios não preenchidos
if (!$dia || !$mes || !$ano || !$responsavel || !$equipe_responsavel ||
!$solucao)
{
      $_SESSION["consist"] = 1;
      $_SESSION["num_chamado"] = $num_chamado;
      $_SESSION["email"] = $email;
      $_SESSION["dia"] = $dia;
      $_SESSION["mes"] = $mes;
      $_SESSION["ano"] = $ano;
      $_SESSION["responsavel"] = $responsavel;
      $_SESSION["equipe_responsavel"] = $equipe_responsavel;
      $_SESSION["solucao"] = $solucao;

      header("location: registrar_atendimento2.php");

}
elseif (!$var_data)
{
 $_SESSION["consist"] = 2;
 $_SESSION["num_chamado"] = $num_chamado;
 $_SESSION["email"] = $email;
 $_SESSION["dia"] = $dia;
 $_SESSION["mes"] = $mes;
 $_SESSION["ano"] = $ano;
 $_SESSION["responsavel"] = $responsavel;
 $_SESSION["equipe_responsavel"] = $equipe_responsavel;
 $_SESSION["solucao"] = $solucao;

header("location: registrar_atendimento2.php");

}
else
{

 // Formata a data para ser gravada no banco
 $data_atendimento = $ano."-".$mes."-".$dia;

 // Declaração SQL
 $sql = "INSERT into atendimentos values ('$num_chamado',
'$data_atendimento', '$responsavel', '$equipe_responsavel', '$solucao')";

 // Roda a query e trata o resultado
```

```
      if (mysql_query($sql))
      {

       // formata a data para exibição (dd/mm/aaaa)
      $data_atendimento = $dia."/".$mes."/".$ano;

            mail ("$email", "SRC - O chamado $num_chamado foi atendido",
                    "Aviso: O chamado $num_chamado foi atendido.
                       Mais Informações:

                     Data do atendimento: $data_atendimento
                     Responsáel: $responsavel
                     Equipe Responsável: $equipe_responsavel
                     Solução: $solucao","SRC - Sistema de Registro de
Chamados");

                       $_SESSION["msg_exib"] = "incluir_atendimento_ok";
                       header("Location: exibe_mensagem.php");
      }
      else
      {
            $_SESSION["msg_exib"] = "incluir_atendimento_erro";
            header("Location: exibe_mensagem.php");
      }

}

?>
```

6.5) Script exibe_mensagem.php

Para completar esse módulo é necessário o script que exibe as mensagens finais. Será utilizado também o script exibe_mensagem.php.

6.6) Testando o módulo de registro de atendimentos

Deixe o campo ‘Número do Chamado’ do formulário em branco. Deverá exibir uma mensagem de erro.
No campo ‘Número do Chamado’ do formulário, digite um número inexistente. Deverá exibir uma mensagem de erro.
No campo ‘Número do Chamado’ do formulário, digite um número válido. Deverá exibir o formulário completo para o registro de atendimentos.
No formulário completo deixe os campos obrigatórios em branco. Deverá exibir uma mensagem de erro.
No formulário completo digite datas inválidas no campo ‘Data do atendimento’. Deverá exibir uma mensagem de erro.
Preencha o formulário completo com dados válidos. Deverá exibir a mensagem ‘Processado com sucesso’.

7 - Módulo de consulta

7.1) Script consulta_chamados_por_per.php

Este script cria o formulário de consulta, e obtém um intervalo de datas.

```
<?php<?php

session_start();

include ("funcoes.php");
include ("misc.php");

ver_session();

monta_cabecalho();

?>

<table width="760" border="0" cellspacing="0">
 <tr>

<?php

monta_menu($_SESSION["usuario_sys"],$_SESSION["tipo_usuario_sys"]);

?>

 <td width="610" valign="top"><br>
 <table width="550" border="0" cellspacing="0" align="center">
 <tr valign="top">
 <td>
 <font size="2" face="Verdana, Arial, Helvetica, sans-serif"><font
color="#264989"><b>Consulta chamados ocorridos num determinado
período</b>
 </font></font>

<?php

      if ($_SESSION["consist"] == 1)
      {
            $dia_ini = trim($_SESSION["dia_ini"]);
            $mes_ini = trim($_SESSION["mes_ini"]);
            $ano_ini = trim($_SESSION["ano_ini"]);
            $dia_fim = trim($_SESSION["dia_fim"]);
            $mes_fim = trim($_SESSION["mes_fim"]);
            $ano_fim = trim($_SESSION["ano_fim"]);

            echo ('<BR><BR><font face="Verdana, Arial, Helvetica, sans-
serif" size="2"><b><font color="#CC0033">Data(s)
inválidas(s)</font></b></font>');
          unset($_SESSION["consist"]);

      }
```

```
 ?>

 <form method="post" action="consulta_chamados_periodo2.php">
 <p>
 <font size="2" face="Verdana, Arial, Helvetica, sans-serif"><b> Entre
com o período: </b><BR><BR>
 <b>De:</b>        dia

 <?php

  echo ("
 <input name=\"dia_ini\" type=\"TEXT\" size=2 maxlength=2
value=\"$dia_ini\">    mês
 <input name=\"mes_ini\" type=\"TEXT\" size=2 maxlength=2
value=\"$mes_ini\">    ano
 <input name=\"ano_ini\" type=\"TEXT\" size=4 maxlength=4
value=\"$ano_ini\"><BR><BR>
 <b>Até</b>       dia
 <input name=\"dia_fim\" type=\"TEXT\" size=2 maxlength=2
value=\"$dia_fim\">    mês
 <input name=\"mes_fim\" type=\"TEXT\" size=2 maxlength=2
value=\"$mes_fim\">    ano
 <input name=\"ano_fim\" type=\"TEXT\" size=4 maxlength=4
value=\"$ano_fim\"></font>
 ");

 ?>

 </p>
 <center>
 <p><font face="Verdana, Arial, Helvetica, sans-serif" size="2">
 <input name="sub" type="SUBMIT" value="Enviar Dados">
 </font> <font face="Verdana, Arial, Helvetica, sans-serif" size="2">
<br>
 </font>
 </center>
 </form>
 </td>
 </tr>
 </table>
 </td>
 </tr>
</table>

<?php

monta_rodape();

?>
```

7.2) Script consulta_chamados_por_per2.php

Este script consiste os campos recebidos do formulário, consulta as tabelas ‘chamados’ e ‘atendimentos’ e exibe os dados obtidos.

```
<?php<?php

session_start();

include ("misc.php");
include ("funcoes.php");

ver_session();

// Recebe as variáveis do formulário e tira espaços em branco
$dia_ini = trim($_POST["dia_ini"]);
$mes_ini = trim($_POST["mes_ini"]);
$ano_ini = trim($_POST["ano_ini"]);
$dia_fim = trim($_POST["dia_fim"]);
$mes_fim = trim($_POST["mes_fim"]);
$ano_fim = trim($_POST["ano_fim"]);

// formata as datas no formato Unix/timestamp para que sejam comparadas
entre si
$data_ini = mktime(0,0,0,$mes_ini,$dia_ini,$ano_ini);
$data_fim = mktime(0,0,0,$mes_fim,$dia_fim,$ano_fim);

// Consiste data de inicio e data de fim
$var_data_ini = checkdate((int)$mes_ini, (int)$dia_ini, (int)$ano_ini);
$var_data_fim = checkdate((int)$mes_fim, (int)$dia_fim, (int)$ano_fim);

// Consiste se campos obrigatorios foram preenchidos

if (!$var_data_ini || !$var_data_fim)
{
    $_SESSION["consist"] = 1;
    $_SESSION["dia_ini"] = $dia_ini;
    $_SESSION["mes_ini"] = $mes_ini;
    $_SESSION["ano_ini"] = $ano_ini;
    $_SESSION["dia_fim"] = $dia_fim;
    $_SESSION["mes_fim"] = $mes_fim;
    $_SESSION["ano_fim"] = $ano_fim;

    header("location: consulta_chamados_periodo.php");
}
elseif ($data_ini > $data_fim)
{
       $_SESSION["consist"] = 1;
       $_SESSION["dia_ini"] = $dia_ini;
       $_SESSION["mes_ini"] = $mes_ini;
       $_SESSION["ano_ini"] = $ano_ini;
       $_SESSION["dia_fim"] = $dia_fim;
       $_SESSION["mes_fim"] = $mes_fim;
       $_SESSION["ano_fim"] = $ano_fim;
```

```
       header("location: consulta_chamados_periodo.php");
}

else
{
      //transforma o dia e o mês num formato válido, por exemplo,
      //se for digitado 1 nos campos dia ou mês,transforma para 01

      $dia_ini = strftime("%d",$data_ini);
      $mes_ini = strftime("%m",$data_ini);
      $ano_ini = strftime("%Y",$data_ini);
      $dia_fim = strftime("%d",$data_fim);
      $mes_fim = strftime("%m",$data_fim);
      $ano_fim = strftime("%Y",$data_fim);

      //Formata as datas recebidas para aaaa-mm-dd
      $data_ini = $ano_ini."-".$mes_ini."-".$dia_ini;
      $data_fim = $ano_fim."-".$mes_fim."-".$dia_fim;

      //Formata as datas para exibiï¿½o (dd/mm/aaaa)
      $data_ini_exib = $dia_ini."/".$mes_ini."/".$ano_ini;
      $data_fim_exib = $dia_fim."/".$mes_fim."/".$ano_fim;

      // Declaraçãoo do SQL
      $sql = "SELECT num_chamado, data_chamado, hora_chamado, usuario,
email, problema,  equipe_acionada from chamados where data_chamado >=
'$data_ini' and
                    data_chamado <= '$data_fim' order by num_chamado";

      // Executa a query e verifica se encontrou algum registro
      $query = mysql_query($sql) or die ("Erro no acesso ao banco");
      $achou = mysql_num_rows($query);

      // Se encontrou, guarda as variï¿½eis
      if ($achou > 0)
      {

         monta_cabecalho();

         echo "<BR>
                        <table border=\"0\" width=\"760\">
                        <tr valign=\"top\">
                        <td width=\"660\">
                              <font size=\"2\" face=\"Verdana, Arial,
Helvetica, sans-serif\" color=\"#264989\"><b>Consulta chamados ocorridos
no periodo de $data_ini_exib a $data_fim_exib</b></font>
                        </td>
                              <td width=\"100\">
                        <a href=\"consulta_chamados_periodo.php\"><font
size=\"2\" face=\"Verdana, Arial, Helvetica, sans-
serif\"><b>Voltar</b></font></a>
                              </td>
                        </tr>
                  </table>
                <table border=\"0\" width=\"760\">";
```

```
         while ($reg = mysql_fetch_assoc($query))
         {
                  $num_chamado = $reg["num_chamado"];
                  $data_chamado = $reg["data_chamado"];
              $hora_chamado = $reg["hora_chamado"];
              $usuario = $reg["usuario"];
              $email = $reg["email"];
              $problema = $reg["problema"];
              $equipe_acionada = $reg["equipe_acionada"];

                    //Coloca a data do chamado em formato de exibição (de
aaaa-mm-dd para dd/mm/aaaa)
                    $dt = explode("-",$data_chamado);
                    $dt2 = mktime(0,0,0,$dt[1],$dt[2],$dt[0]);
                    $data_chamado_exib = date("d/m/Y",$dt2);

              echo "<tr>
                     <td width=\"20\">   </td>
                     <td width=\"740\"><font size=\"2\" face=\"Verdana,
Arial, Helvetica,sans-serif\"><BR>
                    <HR><BR>
                              <b><font color=\"#264989\"> Numero do
chamado: </font></b> $num_chamado <BR><BR>
                              </font>
                        </td>
                              </tr>
                  <tr>
                    <td width=\"20\">   </td>
                    <td width=\"740\"><font size=\"2\" face=\"Verdana,
Arial, Helvetica,sans-serif\">
                      <b> Data do chamado: </b> $data_chamado_exib <BR>
                      <b> Hora do chamado: </b> $hora_chamado <BR>
                      <b> Usuario: </b> $usuario <BR>
                      <b> E-mail: </b> $email <BR>
                      <b> Problema: </b> $problema <BR>
                      <b> Equipe acionada: </b> $equipe_acionada <BR>
                              </font>
                       </td>
                              </tr>";


              // Declaraçãoo do SQL
              $sql2 = "SELECT data_atendimento, responsavel,
equipe_responsavel, solucao from atendimentos where num_chamado =
'$num_chamado'";

              // Roda a query e verifica se encontrou registro
              $query2 = mysql_query($sql2) or die ("Erro no acesso ao
banco");
              $achou2 = mysql_num_rows($query2);

              // Se encontrou, guarda as variï¿½eis
              if ($achou2 > 0)
                    {
                 echo "<tr>
                   <td width=\"20\">   </td>
                   <td width=\"740\"><font size=\"2\" face=\"Verdana,
```

```
Arial, Helvetica,sans-serif\">
                     <BR> <b> <font color=\"#264989\"> Atendimento
</font> </b> <BR>
                     </font></td></tr>";

                        while ($row2 = mysql_fetch_row ($query2))
                  {
                           $data_atendimento = $row2[0];
                           $responsavel = $row2[1];
                           $equipe_responsavel = $row2[2];
                            $solucao = $row2[3];

                                        $dt_at = explode("-
",$data_atendimento);
                                        $dt_at2 =
mktime(0,0,0,$dt_at[1],$dt_at[2],$dt_at[0]);
                                        $data_atendimento_exib =
date("d/m/Y",$dt_at2);

                                        echo "<tr>
                                      <td width=\"20\">   </td>
                                      <td width=\"740\"><font size=\"2\"
face=\"Verdana, Arial, Helvetica, sans-serif\">
                                      <b> Data do Atendimento: </b>
$data_atendimento_exib <BR>
                                           <b> Responsavel: </b>
$responsavel <BR>
                                      <b> Equipe responsavel: </b>
$equipe_responsavel <BR>
                                      <b> Solucao: </b> $solucao <BR><BR>";

                                 }
                        }
                 else
                        {
                     echo '<tr>
                     <td width="20">   </td>
                     <td width="740">
                     <font size="2" face="Verdana, Arial, Helvetica, sans-
serif" color="#000000"><BR> Obs: Ate o momento, nenhum atendimento foi
registrado para esse chamado. </font>
                     </font></td></tr>';
                        }
              }

            echo "<tr>
                  <td width=\"20\">   </td>
                  <td width=\"740\"><font size=\"2\" face=\"Verdana, Arial,
Helvetica, sans-serif\">
                 <b><a href=\"consulta_chamados_periodo.php\"><BR><BR>
Voltar</a></b>
                           </td>
                           </tr>
                           </table>";
      }
      else
      {
```

```
           monta_cabecalho();
        echo '<table width="760" border="0" cellspacing="0"> <tr>';


monta_menu($_SESSION["usuario_sys"],$_SESSION["tipo_usuario_sys"]);

        echo '<td width="610" valign="top"><br>
               <font face="Arial"><font size="2" face="Verdana, Arial,
Helvetica, sans-serif"><BR><BR>
               <center> <font color="#CC0033"> <b> Nao existem chamados
para esse periodo. </b> </font> </center><BR><BR>
               <center> <b> <a
href="consulta_chamados_periodo.php">Voltar</a> </b> </center>
               </font><BR><BR><BR><BR>
             </td>
           </tr>
         </table>
        ';
       }

       monta_rodape();
}

?>
```

7.3) Testando o módulo de consulta

Deixe os campos obrigatórios do formulário em branco. Deverá exibir uma mensagem de erro.
Digite datas inválidas. Deverá exibir uma mensagem de erro.
Digite um intervalo de datas válido. Deverá exibir os dados referentes a esse período.

8 – O script logout.php

Este script realizará a expiração da sessão de usuários no sistema:

```
<?php

session_start();


session_unset();
session_destroy();

header("Location: index.php");

?>
```

Referência Bibliográfica

- ***Beginning PHP4 - Programando***
  Autores: Wankyu Choi, Allan Kent, Chris Lea, ganesh Prasad, Chris Ullman, Jon Blank e Sea Cazzell
  Editora: Makron Books

- **Documentação Oficial do PHP**
  http://www.php.net

- **Consulta a dicas e artigos do site PHP Brasil**
  http://www.phpbrasil.com

- **Para mais informações sobre o PostgreSQL, consulte o site:**
  http://www.commandprompt.com/ppbook/

Onde obter ajuda

Para ajudá-lo a solucionar dúvidas de informática, utilize o sistema Rau-Tu de perguntas e respostas, que foi desenvolvido pelo Centro de Computação da Unicamp em conjunto com o Instituto Vale do Futuro. Tem por objetivo possibilitar que um time de colaboradores possa responder a perguntas colocadas por qualquer pessoa no site, cobrindo diversas áreas de conhecimento.
Acesse: **www.rau-tu.unicamp.br**