

PHP-GTK

construa um gerenciador de downloads

FTP

faça upload seguro de arquivos

Banco de Dados

mapeamento objeto-relacional

Imagens

geração de mapas

Integração de Tecnologias

Javascript, MySQL e PHP



hello world



ISSN 1981-044X



00006>



9 771981 044000





PHP-GTK

FTP

ORM

GDLib

editorial

Apresentação

Caixa de Entrada

Equipe

artigos

Um gerenciador de downloads em PHP-GTK

por Pablo Dall'Oglio

Upload seguro de arquivos usando FTP verificando vírus

por Helton Ritter

Mapeamento objeto-relacional usando PHP e MySQL

por Aleksandro Assis

Mapas temáticos gerados com PHP

por Raquel Dezidério Souto

Integrando ferramentas na construção de um calendário

por Leandro Schwarz

Amigo leitor,

É com grande alegria que lhe apresentamos a 6ª edição da PHP Magazine! Mais uma vez, a sua PHP Magazine está repleta de artigos especialmente selecionados para você.

Em nossas páginas, o leitor poderá desfrutar de cinco excelente artigos dos nossos colaboradores. Pablo Dall'Oglio, o criador do PHP-GTK Brasil, nos presentiu com um artigo sobre o gerenciamento de downloads PHP-GTK. Focando em segurança, Helton Eduardo Ritter, membro da equipe PHP Magazine, nos apresenta um artigo sobre Upload seguro de arquivos, via FTP, mediante prévia verificação do antivírus. O artigo "Mapeamento objeto-relacional usando PHP e MySQL", de Alexsandro Assis, abordará a área de banco de dados e o ajudará a manter a padronização, produtividade e portabilidade entre diferentes bancos de dados. Na sequência, Leandro Schwarz abordará todos os passos para a criação de um módulo de calendário. Por último, Raquel Dezidério Souto demonstra como programar em PHP para geração de mapas temáticos.

Novamente, aproveitamos para agradecer aos quase 11.000 leitores que nos acompanham no portal e contribuem com sugestões, artigos, e-mails de incentivo e cobranças. Vocês são nossos incentivadores e podem participar ativamente da revista. Estamos esperando o seu artigo!

Boa leitura!

Equipe da PHP Magazine

Editores

Flávio Zacharias Fagundes, zach@phpmagazine.org.br
Ricardo Aragão, ricardoaragao@phpmagazine.org.br

Administração

Flávio Zacharias Fagundes, zach@phpmagazine.org.br
Ricardo Aragão, ricardoaragao@phpmagazine.org.br

Comercial

Flávio Zacharias Fagundes
Ricardo Aragão

Projeto gráfico

Flávio Zacharias Fagundes
Ricardo Aragão

Revisão técnica

Ricardo Aragão da Silva
Flávio Zacharias Fagundes
Helton Eduardo Ritter

Revisão - Língua Portuguesa

Camilla Carvalho, camilla@phpmagazine.org.br



www.phpmagazine.org.br

Comercial

comercial@phpmagazine.org.br

Contato/Suporte

contato@phpmagazine.org.br

Marketing

mkt@phpmagazine.org.br



Novos Artigos

Acompanho a revista desde a primeira edição e é visível o salto de qualidade que vem apresentando a cada edição. Na edição de número 5, gostei especialmente do artigo "Desenvolvendo em três camadas com PHP, MVC e Ajax". Meus parabéns aos autores. Gostaria de saber se têm interesse em publicar artigos focados no Zend Framework, pois é nesta área que gostaria de compartilhar informações.

Leandro Lages

Agradecemos os elogios. Todo artigo é bem-vindo! Lembramos que você pode submeter seus artigos pelo portal www.phpmagazine.org.br.

Elogios

Parabéns pelo trabalho e pelo sucesso da revista. Aqui no Brasil, ter um trabalho destes já é motivo de orgulho.

Ricardo Chiavelli

A equipe PHP Magazine agradece o elogio. Você é a parte principal da nossa publicação, Magazine: é o leitor. Sem você a revista não sobrevive.

Opensource

Poderiam me informar o software que usam para formatar a revista? É Opensource?

Guilherme Machado - Curitiba - PR

Atualmente utilizamos o *Microsoft Publisher*, com outros softwares de apoio pra edição de imagens, que infelizmente, não são *opensource*. Em alguns casos se faz necessário aplicar realce de sintaxe, ao código fonte alguns artigos. Isso é feito através da função `show_source` do próprio PHP.

Críticas

Gosto da revista, porém os intervalos são muito grandes. Quero fazer uma crítica construtiva: algumas imagens estão simplesmente horríveis.

Bruno Merlin

Sempre procuramos incluir imagens de boa qualidade nos artigos, no entanto, alguns artigos são submetidos com as imagens em baixa qualidade e não há tempo hábil para obter outras imagens, com melhor qualidade, junto aos autores.

Edição impressa da revista

Sou desenvolvedor web e gostaria de saber se vocês tem a versão impressa da revista.

Edbraulio Vieira

Ainda não dispomos de uma versão impressa da revista. Estamos trabalhando para divulgar mais o projeto, expandindo-o sobre outras meios, principalmente através de nosso portal e das edições digitais.

Edição 6

Pessoal, gostaria de saber quando estará disponível a sexta edição da revista? Está demorando! Adorei as outras edições, muito bom o trabalho de vocês, parabéns!

Marcelo Ferreira

Sofremos algum atraso no lançamento da edição, pois estávamos buscando melhorar os serviços oferecidos em nosso portal. Pedimos a compreensão de nossos leitores e firmamos o compromisso de sempre buscar oferecer conteúdos com a qualidade de sempre.

ENVIE E-MAIL PARA

contato@phpmagazine.org.br

Recomendamos o envio de mensagens com os seguintes dados do remetente: nome completo, cidade e estado. A redação se reserva o direito de selecionar e adaptar as mensagens sem alterar o conteúdo geral.



PHP-GTK

Criando Aplicações Gráficas com PHP

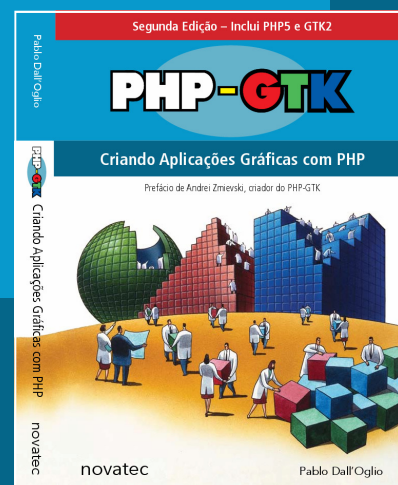
Todo programador já deve ter ouvido falar sobre PHP, uma das linguagens mais utilizadas no mundo. A popularidade do PHP deve-se à sua facilidade em criar aplicações dinâmicas para a internet, com suporte à maioria dos gerenciadores de bancos de dados existentes, além de um conjunto de funções que permitem desde a criação de simples portais até complexas aplicações de comércio eletrônico.

O que poucos entenderiam até alguns anos atrás é a possibilidade de criar aplicações-cliente, com janelas, botões, listas e painéis em PHP. Pois este é justamente o foco de PHP-GTK Criando Aplicações Gráficas com PHP. O PHP tem crescido muito nos últimos anos e se tornou uma linguagem de propósitos gerais. Este livro é prova disto. Por intermédio do PHP-GTK, você pode utilizar o PHP para desenvolver aplicações com visual atraente e nativo para várias plataformas, como Linux, Windows e Mac, utilizando a mesma linguagem que utiliza para criar aplicações para a Web.

Principais tópicos abordados no livro:

- Introdução ao PHP, arrays, strings, bancos de dados
- Tutorial completo sobre orientação a objetos
- Exemplos de utilização de janelas, árvores, listas, botões
- Exemplos de utilização de menus, diálogos, formulários
- Crie suas próprias classes, estendendo o GTK
- Exemplos com bancos de dados (inserção, listagem, navegação)
- Utilização da ferramenta Glade (geração de notas fiscais)

www.novatec.com.br



Um gerenciador de Downloads em PHP-GTK

Neste artigo iremos desenvolver um pequeno utilitário em PHP-GTK para baixar arquivos remotos. Assim como nos artigos anteriores, nosso objetivo não é desenvolver nenhuma ferramenta profissional, mas mostrar de forma didática como utilizar os recursos do PHP em conjunto com o GTK.

1. O Programa

A interface do programa será bastante simples. Como pode ser visto na Figura 1, a janela é composta de um campo onde o usuário irá digitar o endereço remoto do arquivo a ser baixado (objeto **\$file**), um botão "OK" para iniciar o download (objeto **\$inicia_btn**) e um botão "Cancelar" (objeto **\$cancel_btn**) para interromper o download. Logo a seguir, tem-se uma barra de progresso (objeto **\$progress**) que indica o percentual que já foi baixado do arquivo até o momento.

Para criar a interface do programa, a classe **GtkWindow**, que representa uma janela, será estendida. Desta forma, em seu método construtor, definiremos quais objetos estarão contidos na janela principal e também as ações do botão "OK", que executará o método **download()** e do botão "Cancelar" que executará o método **abort()**.

```
<?php
/**
 * classe para baixar arquivos
 * @author Pablo Dall'Oglio
 */
class Downloader extends GtkWindow
{
    private $file; //local do arquivo
    private $inicia_btn; //botão ok
    private $cancel_btn; //botão cancelar
    private $progress; //barra de progresso
    private $aborted; //controle de abortar

    /**
     * cria a interface
     */
    function __construct()
    {
        parent::__construct();
        parent::set_position(Gtk::WIN_POS_CENTER);
        parent::set_size_request(430,-1);
        parent::set_title('pDownloader!');

        // cria uma caixa vertical
        $vbox = new GtkVBox;
        $this->aborted = FALSE;
```

```
// cria uma caixa para o arquivo
$this->file = new GtkEntry('http://file.tgz');
// cria botões de OK e Cancelar
$this->inicia_btn = GtkButton::new_from_stock...
$this->cancel_btn = GtkButton::new_from_stock...
// define as ações dos botões
$this->inicia_btn->connect_simple('clicked',
    array($this, 'download'));
$this->cancel_btn->connect_simple('clicked',
    array($this, 'abort'));

// cria uma barra de progresso
$this->progress = new GtkProgressBar();
$this->progress->set_fraction(0);
$this->progress->set_text('Aguardando...');

// cria uma caixa horizontal
$hbox = new GtkHBox;
$hbox->pack_start($this->inicia_btn, FALSE,...
$hbox->pack_start($this->cancel_btn, FALSE,...

// organiza a interface
$vbox->pack_start($this->file, TRUE, TRUE);
$hbox->pack_start($hbox, FALSE, FALSE);
$vbox->pack_start($this->progress, FALSE,...

// adiciona a caixa vertical na janela
parent::add($vbox);
parent::show_all();
}
```

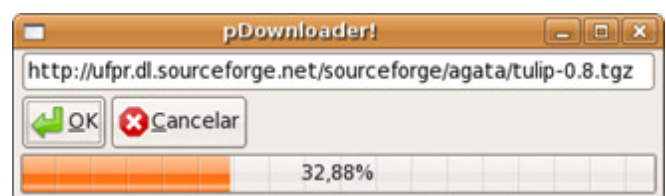


Figura 1 – Aplicação rodando

O próximo método que iremos escrever é o método **abort()**. Este método será executado sempre que o usuário clicar no botão cancelar. Seu funcionamento será bastante simples. Ele irá definir o valor da propriedade **\$aborted** como verdadeiro (TRUE) e também irá exibir uma mensagem ao usuário por meio de um diálogo de

Warning, alertando que o download foi cancelado.

```
/**
 * aborta o processamento
 */
function abort()
{
    // define o valor desta propriedade
    $this->aborted = TRUE;
    // exibe mensagem de warning
    $dialog = new GtkMessageDialog(null,
    Gtk::DIALOG_MODAL, Gtk::MESSAGE_WARNING,
    Gtk::BUTTONS_OK, 'Download Abortado !!');
    $response = $dialog->run();
    $dialog->destroy();
}
```

2. Baixando o arquivo

O método **download()** será responsável por baixar o arquivo. Para tal, ele irá abrir ambos arquivos remoto e local por meio do método **fopen()**. Em um laço de repetições, utilizaremos o método **fgets()** para ler os dados remotos e o método **fwrite()** para gravar no arquivo local. A função **events_pending()** e **main_iteration()** são utilizadas para que o GTK atualize a interface principal com o percentual de andamento do download e para que a mesma não fique congelada até o final do processo.

Neste método utilizamos a função **get_headers()** para obter a informação do tamanho do arquivo, antes de iniciar o download, para poder calcular o percentual concluído, e a função **pathinfo()**, para extrair somente o nome do arquivo da URL. Assim, o arquivo remoto será salvo na pasta **/tmp**, mantendo o mesmo nome.

O índice de conclusão do download (variável **\$indice**) será obtido acumulando a quantidade de bytes já baixados, por meio da variável **\$downloaded** e fazendo a sua proporção em relação ao tamanho total do arquivo (variável **\$filesize**).

```
/**
 * baixa um arquivo e salva na pasta /tmp
 */
function download()
{
    // obtém o nome do arquivo
    $url = $this->file->get_text();

    // obtém o tamanho do arquivo
    $header = get_headers($url, 1);
    $filesize = $header['Content-Length'];
    $remote = fopen($url, "r"); // abre o arquivo
    if (!$remote) // se arquivo não existe
    {
        return;
    }
    // obtém somente o nome do arquivo
    $pathinfo = pathinfo($url);
    $basename = $pathinfo['basename'];

    // cria um arquivo localmente
    $local = fopen ("/tmp/{$basename}", 'w');
    $downloaded = 0;

    // laço de repetições para baixar o arquivo
```

```
while(!feof($remote) and !$this->aborted)
{
    // atualiza a interface
    while (gtk::events_pending())
    {
        gtk::main_iteration();
    }
    // baixa até 1kb
    $var = fgets($remote, 1024);
    // grava o conteúdo localmente
    fwrite($local, $var);
    // total de bytes baixados
    $downloaded += strlen($var);
    // calcula índice para barra de progresso
    $indice = $downloaded/$filesize;
    $percent = ((int) ($indice * 10000))/100;
    // atualiza barra de progresso
    $this->progress->set_fraction($indice);
    $this->progress->set_text("{ $percent}%");
}

$this->aborted=FALSE;
// fecha arquivos
fclose($remote);
fclose($local);
}
```

3. Instanciando a classe principal

Para finalizar o programa, basta instanciarmos a classe principal para que a janela seja exibida na tela.

```
// instancia classe
new Downloader;
Gtk::main();
?>
```

Considerações finais

Neste artigo, demos continuidade à nossa série de artigos sobre PHP-GTK, sempre demonstrando uma funcionalidade específica da linguagem PHP útil no dia-a-dia. Devido ao espaço restrito aqui, algumas linhas foram truncadas. Logo a seguir, na seção referências, encontra-se o [Site do Programa], onde o mesmo poderá ser baixado integralmente.

Referências e links sugeridos

[PHP-GTK Brasil] – <http://www.php-gtk.com.br>
[Livro PHP-GTK] – <http://www.php-gtk.com.br/book>
[Site do Autor] – <http://www.pablo.blog.br>
[Site do Programa] – <http://downloader.php-gtk.com.br>

Pablo Dall'Oglio - pablo@dalloaglio.net

Pablo Dall'Oglio é bacharel em Informática pela UNISINOS. Autor dos livros "PHP Programando com Orientação a Objetos" e "PHP-GTK - Criando Aplicações Gráficas com PHP", pela editora Novatec. Também foi o criador do PHP-GTK Brasil. Atualmente, é diretor de tecnologia e proprietário da Adianti Solutions (www.adianti.com.br).

Upload seguro de arquivos usando FTP verificando vírus

Este artigo apresenta uma técnica não usual de fazer upload de arquivos para o servidor. Essa forma não garante, mas auxilia a segurança do servidor. Da maneira proposta, o daemon do apache não precisa ter permissão de gravação em um diretório ou arquivo em uma URL pública para fazer a transferência.

Segurança não é brincadeira, não é de simples implementação, mas é necessária e, em algumas situações, é crítica. Vale ressaltar que segurança total não existe. O máximo que podemos fazer é aplicar boas técnicas e monitorar sempre os sites que gerenciamos.

O que define tudo é o conhecimento. Se o invasor, ou quem quer que seja, que esteja tentando burlar o sistema souber mais do quem o projetou/desenvolveu, então ele conseguirá.

Da tentativa de dificultar a concretização da má intenção é que surgem as práticas. Uma delas é o upload usando FTP e a verificação de vírus no arquivo carregado.

1. Módulo `php_clamavlib`

ClamAV é um anti-vírus conhecido do mundo Linux, principalmente por integrar-se com diversos outros serviços como o Postfix, SendMail, Samba e, dentre outros, o Apache.

Para PHP também há um módulo que pode ser instalado através do gerenciador de pacotes de sua distribuição.

Com o módulo do ClamAV disponível em PHP podemos verificar vírus em um arquivo antes de servi-lo ao cliente ou carregá-lo do cliente para a aplicação.

O comportamento de verificação de vírus pode ser habilitado no `php.ini` (se o PHP estiver compilado com suporte) para fazer a análise sempre que um arquivo for carregado, pois esta opção não está ativa por padrão. Caso detecte um vírus, o arquivo do upload é apagado e em `$_FILES['file_field']['error']` o erro é registrado.

2. Upload por FTP

Esse método soa estranho aos ouvidos de muitos, pois afinal o script PHP que está em um servidor não

consegue se conectar a máquina do cliente e carregar de lá o arquivo necessário.

O que acontece não é propriamente um upload. É basicamente uma cópia, pois o upload é feito da maneira tradicional (por http). Esse método, entretanto, é inseguro, pois precisamos dar permissão de gravação para o usuário que roda o *daemon* do Apache. O problema não é propriamente a permissão de gravação, mas o fato de costumarmos fazê-la em um diretório público que, se não for monitorado, pode virar um "servidor gratuito" de tudo quanto é arquivo, exceto os esperados pela aplicação.

Uma forma de melhorar isso sem usar-se do FTP é nunca fazer upload para um diretório público e uma página PHP (que tem acesso a esse diretório) fazer o streaming desse arquivo (com `file_get_contents()` por exemplo).

O método proposto consiste em colocar o arquivo em uma local temporário no servidor durante o upload, fazer as verificações e, se passar, transferi-lo por FTP para o diretório definitivo.

3. Vantagens do método

A primeira e mais importante delas é que não precisamos ter nenhum arquivo público com permissão de gravação.

Uma segunda vantagem: podemos colocar o arquivo em um diretório público diretamente e depois não precisamos usar funções para ler o arquivo de um local não público ao exibi-lo na Web. Usa-se menos processamento, uma vez que o carregamento (que agora tem uma carga maior) é feito uma vez e a exibição ocorre em vários momentos.

Podemos criar uma organização maior em nossos servidores. Pode-se ter um servidor que receba o arquivo via upload *http*, faça as verificações e se estiver limpo, passe o arquivo para outro servidor (por FTP). Assim

estamos expondo apenas uma máquina e protegendo as informações já consolidadas.

No caso de termos dois servidores ou mais, o método pode ser ainda otimizado, fazendo a transferência do arquivo por SSH usando `ssh2_scp_send()`. Uma regra interna de *firewall* poderia liberar a conexão SSH apenas para o servidor que enviará o arquivo. Essa mesma regra poderia ser aplicada ao caso do FTP.

FTP não é um protocolo seguro. Assim, se na rede (dos servidores) estiver rodando algum *sniffer*, ainda existe a possibilidade de interceptação/adulteração do arquivo quando transferido de um para outro.

Com apenas um servidor e fazendo conexão FTP para "127.0.0.1" ou "*localhost*" não saímos para a rede, logo, um possível *sniffer* neste ambiente não tem ação.

4. Implementação

O objetivo desse artigo é mais didático do que prático, visto que ele pode não ser totalmente implementável em seu ambiente, caso não se tenha acesso administrativo ao servidor para instalar/configurar SSH, ClamAV... Abaixo uma função que é analisada rapidamente na sequência:

```
function upload($form_field,
               $tam_max,
               $tam_min,
               $ftp_server,
               $ftp_user,
               $ftp_passwd,
               $accept_files,
               $ftp_folder)
{
    $conn = @ftp_connect($ftp_server);
    if(!$conn)
        return "ERRO: Falhou ao conectar com o servidor FTP";

    $login = @ftp_login($conn,$ftp_user,$ftp_passwd);
    if(!$login)
        return "ERRO: Usuario ou senha invalida, nao conseguiu logar";

    $original_name_file = $_FILES[$form_field]['name'];

    $e = explode('.', $original_name_file);
    $cnt = count($e);
    $ext = $e[$cnt-1];
    if(!in_array($ext, $accept_files))
        return "ERRO: O arquivo fornecido não é de um tipo válido!";

    // *1024 porque o size é em bytes por padrão
    if($_FILES[$form_field]['size'] < ($tam_min*1024) or
       $_FILES[$form_field]['size'] > ($tam_max*1024))
    {
        @ftp_close($conn);
        return "ERRO: O tamanho do arquivo não é válido";
    }

    // move do diretório temporário de upload para outro diretório temporário
    if(!@move_uploaded_file($_FILES[$form_field]['tmp_name'], '/tmp/'. $_FILES[$form_field]['name']))
    {
        @ftp_close($conn);
        return "ERRO: Problema ao gravar o arquivo no diretório temporário";
    }

    // verificação de vírus
    $virus = cl_scanfile('/tmp/'. $_FILES[$form_field]['name']);
    if($virus != '')
    {
        @ftp_close($conn);
        @unlink("/tmp/{$_FILES[$form_field]['name']}");
        return "ERRO: Virus ".$virus." encontrado no arquivo! Upload apagado do servidor!";
    }

    // copia por ftp para o diretorio definitivo
    $put_file = ftp_put($conn,$ftp_folder.'/'. $_FILES[$form_field]['name'],
                       '/tmp/'. $_FILES[$form_field]['name'], FTP_ASCII);

    if(!$put_file)
```

```

    return "Erro ao enviar para o servidor definitivo";

@unlink("/tmp/{$_FILES[$form_field]['name']}");
@ftp_close($conn); // suprimir o erro com @ não é performático

return 'ok';

} // fim da função de upload

```

Os parâmetros da função são os seguintes:

- Nome do *form field* do formulário que submete a página ao servidor;
- Tamanho máximo em Kb que o arquivo a ser carregado pode ter;
- Tamanho mínimo, em Kb;
- IP ou nome do servidor de FTP, "*localhost*" ou o nome do seu servidor, lembrando a questão da transferência não segura;
- Usuário de FTP. Apenas o usuário, sem "*@server*";
- Senha de FTP;
- *Array* no formato **\$a = array('gif', 'jpg', 'png')** das extensões de arquivos que o upload deve ser permitido;
- Diretório em que os arquivos depois de passarem pela verificação deverão ser colocados;

A função `move_uploaded_file()` recebe o arquivo carregado e o deixa em um diretório temporário não público em que há permissão de gravação, ao contrário da prática comum que é movê-lo para o diretório definitivo e público do site ("gravável" pelo usuário que roda o apache).

Para fazer a verificação de vírus, usa-se `cl_scanfile()` que, ao encontrar um vírus, devolve uma string com o nome dele. Caso contrário, retornará vazio. Use o Eicar para testes ou outro vírus que você saiba anular sua ação e talvez limpar sua máquina.

Passando por todas as verificações, o arquivo então é transferido `ftp_put()`, para o seu local definitivo. A função `upload()` retorna uma string informando o erro, ou retorna "ok" no caso de nenhum ter havido.

Considerações finais

A metodologia discutida neste artigo é melhor que a tradicional, mas ainda oferece riscos, pois poderemos ter alterações no arquivo ou na leitura destas informações entre a máquina do cliente e o servidor. Neste caso o uso de HTTPS possivelmente ajudaria.

A partir do momento em que o arquivo sai da sua rede até ele chegar ao servidor, ele está em um ambiente fora de seu controle. **O ideal seria comparar se o arquivo que chegou no servidor é exatamente igual àquele que saiu do cliente. Mas de uma maneira transparente, sem que o usuário tenha que gerar uma MD5 do arquivo e depois compará-la.**

As dependências de softwares e configurações no servidor são maiores. Também haverá maior processa-

mento e, conseqüentemente, o processo ficará mais lento como um todo. Tal fato é agravado quando há transferência do arquivo de um servidor para outro (FTP ou SSH). Entretanto ocorre somente uma vez e, no futuro, temos mais garantias. Prós e contras pesados. Se for iniciar um projeto, sugiro que pense bastante em segurança.

Referências e links sugeridos

PESSOA, Márcio. **Segurança em PHP**. São Paulo: Novatec, 2007.

[Eicar] – <http://www.eicar.org/download/>

[Mod_clamav] - http://software.othello.ch/mod_clamav/

[ClamAV] - <http://www.clamav.net/download/third-party-tools/3rdparty-library/>

Helton Eduardo Ritter - heltonritter@gmail.com

Bacharelado em Sistemas de Informação. Técnico em Informática, professor adjunto e funcionário da SETREM, atuando na mesma como desenvolvedor Web desde 2006. Membro da equipe da PHP Magazine deste março de 2008. Mantém um site portfólio: www.sourcedreams.eti.br

Mapeamento objeto-relacional usando PHP e MySQL

Conceituar e demonstrar o desenvolvimento e uso de classes em PHP para efetuar o mapeamento objeto-relacional, visando padronização, produtividade e portabilidade entre diferentes bancos de dados. Tornando, dessa forma, o uso de programação orientada a objetos transparente nas tarefas de manutenção de registros em banco de dados.

Mapeamento objeto-relacional é uma técnica que consiste em tratar os registros de uma tabela do banco de dados como objeto, visando tornar mais amigável o uso de um banco de dados relacional com linguagens ou projetos que utilizam a orientação a objetos. No mapeamento objeto-relacional não será utilizada a linguagem SQL para fazer manutenção dos registros do banco, mas chamadas a métodos que estarão encapsulando os comandos SQL que farão a manutenção nos registros.

O funcionamento do mapeamento objeto-relacional é mais complexo do que o descrito neste artigo. A idéia é construir classes em PHP que nos permita inserir, alterar e apagar registros do banco de dados MySQL, utilizando a idéia de mapeamento objeto-relacional.

1. Classes

Serão usadas três classes para executar o mapeamento objeto-relacional:

Classe db – classeDados.php – tem como objetivo a conexão e execução dos comandos no banco MySQL.

Classe tabela – classeTabela.php – tem como objetivo a montagem dos comandos de insert, update e delete que serão executados em uma determinada tabela do banco de dados.

Classe coluna – classeColuna.php – esta classe trata as informações referentes as colunas de uma determinada tabela.

Esclarecendo, este artigo tem como objetivo demonstrar a técnica de mapeamento objeto relacional. Não é o objetivo dar uma solução completamente pronta. Os exemplos são meramente demonstrativos, pois não houve tratamento de erros e os scripts foram simplificados ao máximo visando diminuir o tamanho do artigo. Fica como desafio melhorar as classes tornando-as 100% utili-

záveis.

```
<?php
```

```
/**
 * @objetivo Responsável por fazer integração
 * com o servidor MySQL
 * @author Alessandro Luiz de Assis
 * @email alexsandro.assis@gmail.com
 * @copyright www.pointdainformacao.com.br
 * @Criado em 09/08/2008
 */

define('SERVIDOR' , 'localhost');
define('USUARIO' , 'root');
define('SENHA' , '');
define('BANCO' , 'artigo');

class db {

    #Função para executar as transações no banco
    public function executar( $sql, $con ){
        @$Seleciona = mysql_query( $sql, $con );
        if (!$Seleciona )
            die('Erro.: '. $sql. '<br>'.mysql_error
            ());
        return $Seleciona;
    }

    # Conecta ao servidor e seleciona o banco de
    # dados
    public function conectar ( ){
        $Conexao = mysql_connect( SERVIDOR,
        USUARIO, SENHA);
        if (!$Conexao )
            die('Erro.: '.mysql_connect_error());
        if (!mysql_select_db (BANCO, $Conexao))
            die('Erro.: '.mysql_error());
        return $Conexao;
    }

    #Fecha a conexão com o banco de dados
    public function desconectar ( $con ){
        mysql_close($con);
    }
}
```

```
}
```

O código descrito acima pertence ao arquivo classe-Dados.php, e possui três métodos (executar, conectar e desconectar) responsáveis por fazer a integração/ utilização do banco de dados MySQL.

Esta classe pode ser expandida para tratar outros bancos (SGBD's) se tornando também uma classe de abstração de dados, o que amplia muito sua utilidade. Também pode ser usada para implementar métodos para tratar as pesquisas e as seleções do banco de dados.

```
<?php
```

```
/**
 * @config Responsável por armazenar e
 * controlar os * dados das colunas
 * @author Alessandro Luiz de Assis
 * @email alexsandro.assis@gmail.com
 * @copyright www.pointdainformacao.com.br
 * @Criado em 09/08/2008
 */
```

```
class coluna{

    # Cria os atributos privadas
    private $nome;
    private $tipo;
    private $tamanho;
    private $chave;
    private $valor;

    # Construtor da classe
    public function __construct
    ($nome, $tipo, $tamanho)
    {
        $this->nome = $nome;
        $this->tipo = $tipo;
        $this->tamanho = $tamanho;
        $this->chave = 'N';
        $this->valor = '';
    }

    # Função de atribuição de valor a coluna
    public function setValor($valor){
        $this->valor = $valor;
    }

    # Função para pegar o valor a coluna
    public function getValor(){
        return $this->valor;
    }

    # Função para pegar o tipo da coluna
    public function getTipo(){
        return $this->tipo;
    }

    # Função para pegar o tamanho da coluna
    public function getTamanho(){
        return $this->tamanho;
    }

    # Função para pegar se a coluna é chave
    public function getChave(){
        return $this->chave;
    }
}
```

```
# Função para pegar o nome da coluna
public function getNome(){
    return $this->nome;
}
```

```
# Função para tornar esta coluna chave
public function isChave(){
    return $this->chave = 'S';
}
```

```
}
?>
```

O código descrito acima pertence ao arquivo classe-Coluna.php e possui vários métodos responsáveis por fazer o mapeamento das colunas da tabela. Para isso cria-se "objetos coluna" para cada coluna da tabela. Nesta classe existem métodos para armazenar (set), pegar (get) valores das colunas e identificar quais colunas fazem parte da chave da tabela.

```
<?php
```

```
/**
 * @config Responsável por fazer o mapeamento
 * das tabelas do banco de dados
 * @author Alessandro Luiz de Assis
 * @email alexsandro.assis@gmail.com
 * @copyright www.pointdainformacao.com.br
 * @Criado em 09/08/2008
 */

# Inclui a classe banco
require_once( "classeDados.php" );

# Inclui a classe abstrata banco
require_once( "classeColuna.php" );

# Inclui a classe de Banco de Dados
class tabela extends db {

    # Cria os atributos privadas
    public $nome;
    public $coluna = array();
    public $sql = '';
    public $conexao;

    # Construtor da classe
    public function __construct($nome){
        $this->nome = $nome;
    }

    # Cria metodo para criar as colunas com seus
    # atributos
    public function coluna($nome, $tipo, $tamanho)
    {
        $this->$nome = new coluna
        ($nome, $tipo, $tamanho);
        $this->coluna[$nome] = $nome;
    }
}
```

O fragmento de código descrito acima é o início do arquivo (classeTabela.php). Neste fragmento de código, gostaria de chamar atenção para dois pontos importantes:

1 – A criação do atributo coluna para do tipo Array para armazenamento do nome das colunas.

2 – O método coluna, que indica dentro da classe

tabela a classe coluna.

Ex.: `$tabela->id_grupo->isChave();`

Os fragmentos de códigos descritos abaixo são a parte mais importante da classe tabela. Estes fragmentos são os responsáveis por executar as ações de INSERT, UPDATE e DELETE, respectivamente, no banco de dados.

Os três métodos abaixo (inserir, alterar e deletar) seguem os mesmos princípios:

1 - Abre a conexão com o banco de dados;

2 - Através de um *loop* pelo array de colunas, busca os atributos em cada objeto coluna e concatena os dados, montando a string SQL que será executada no banco de dados;

3 - Executa a string SQL no banco de dados.

```
# Cria metodo para montar o comando de insert
public function inserir(){
    $this->conexao = $this->conectar();
    # Monta o comando SQL
    $campos = '';
    $valores = '';
    foreach ($this->coluna as $v){
        $campos .= $this->$v->getNome() . ',';
        $valores .= $this->formataTipo($this->$v->getValor(),$this->$v->getTipo()) . ',';
    }
    $campos = substr($campos,0,strlen($campos)-1);
    $valores = substr($valores,0,strlen($valores)-1);
    $this->sql .= 'insert into ' . $this->nome . '(' . $campos . ') values (' . $valores . ');';
    $this->executar($this->sql, $this->conexao);
}

# Cria metodo para montar o comando de update
public function alterar(){
    $this->conexao = $this->conectar();
    # Monta o comando SQL
    $valores = '';
    $clausulas = '';
    foreach ($this->coluna as $v){
        $valores .= $this->$v->getNome() . '=';
        $valores .= $this->formataTipo($this->$v->getValor(),$this->$v->getTipo()) . ',';
        if ($this->$v->getChave() == 'S'){
            $clausulas .= ' and ' . $this->$v->getNome() . '=';
            $clausulas .= $this->formataTipo($this->$v->getValor(),$this->$v->getTipo());
        }
    }
    $valores = substr($valores,0,strlen($valores)-1);
    $this->sql .= 'update ' . $this->nome . ' set ' . $valores . ' where 1=1' . $clausulas . ';';
    $this->executar($this->sql, $this->conexao);
}

# Monta a string e deleta os dados que estão no objeto na tabela
public function deletar(){
    $this->conexao = $this->conectar();
    # Monta String SQL
    $clausulas = '';
    foreach ($this->coluna as $v){
        if ($this->$v->getChave() == 'S'){
            $clausulas .= ' and ' . $this->$v->getNome() . '=';
            $clausulas .= $this->formataTipo($this->$v->getValor(),$this->$v->getTipo());
        }
    }
    $this->sql .= 'delete from ' . $this->nome . ' where 1=1' . $clausulas . ';';
    $this->executar($this->sql, $this->conexao);
}

# Formata os dados conforme o tipo de dado da tabela
public private function formataTipo($v1,$tp){
    $valor = '';
    switch ($tp){
        case 'char':
            $valor = '\\' . $v1 . '\\';
            break;
        case 'int':
            $valor = $v1;
            break;
    }
}
```



```

        case 'data':
            $data = explode('/', $v1);
            $v1 = date("Y-m-d", mktime(0, 0, 0, $data[1], $data[0], $data[2]));
            $valor = '\\' . $v1 . '\\';
            break;
    }
    return $valor;
}

# Destruidor da classe
public function __destruct(){
    $this->desconectar($this->conexao);
}

}
?>

```

No final da classe tabela existe o método **formataTipo()**, que formata os dados conforme a necessidade e o tipo de dados.

Ex.: coloca os dados em formato de data compatível com o banco de dados, como '09/25/2008', por exemplo.

Vamos ver como funciona na prática esta classe. Para exemplificar, usaremos duas tabelas conforme (Figura 1).

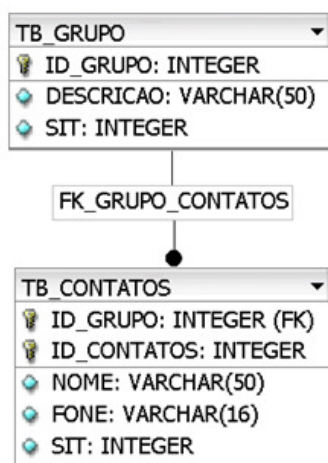


Figura 1 – Diagrama de entidade relacionamento

```

1<?php
2
3 #Inclui a classe tabela usada para testes
4 require_once( "classeTabela.php" );
5
6 /**
7 Exemplo de utilizacao das classes de dados
8 */
9 $grupo = new tabela('tb_grupo');
10 $grupo->coluna('id_grupo','int','10');
11 $grupo->coluna('descricao','char','50');
12 $grupo->coluna('sit','int','1');
13
14 $grupo->id_grupo->isChave();
15 $grupo->id_grupo->setValor('2');
16 $grupo->descricao->setValor('Trabalho');
17 $grupo->sit->setValor('1');
18
19 $grupo->inserir();
20 $grupo->deletar();

```

```

21 $grupo->alterar();
22
23?>

```

O código acima descreve a simplicidade para executar manutenção nos registros do banco de dados.

Na linha 4 é realizada a inclusão do script da classe. Na linha 9 é instanciado o objeto tabela e passado via parâmetro o nome da tabela. Das linhas 10 a 12 são criadas as colunas (objeto) da tabela conforme a estrutura da tabela.

Na linha 14, indica-se o campo que faz parte da chave primaria da tabela. Esta informação é essencial para os métodos de alterar e deletar, pois estas ações serão efetuadas com base nesta informação. Ex.: alterar o registro quando o id_grupo = 1.

Nas linhas 15 a 17, os valores de cada coluna serão atribuídos e, nas linhas de 19 a 21, são chamados os métodos para o tipo de manutenção que será efetuada no registro.

Esta é a idéia de uso de mapeamento objeto-relacional usando PHP e MySql para alterar sua aplicação para o uso de um banco de dados diferente, como o Oracle, basta alterar a classe db para usar Oracle em vez de MySql.

Referências e links sugeridos

[PHP Magazine] – <http://www.phpmagazine.org.br>
 [Point da Informação] – <http://www.pointdainformacao.com.br>

Alexsandro Assis - alexsandro.assis@gmail.com

Analista de banco de dados, possui 8 anos de experiência com banco de dados SQL Server e 6 anos de experiência em desenvolvimento de aplicações Web. Cursando o sexto período de Sistema de Informação pela Uniaraxá - Centro Universitário do Planalto de Araxá.

Mapas temáticos gerados com PHP

Neste artigo, por meio de um exemplo prático, demonstraremos como programar em PHP para geração de mapas temáticos. Diversos sistemas de informações geográficas já foram desenvolvidos e oferecem o recurso de geração de mapas temáticos. Entretanto, o tempo gasto na aprendizagem do funcionamento de tais sistemas pode ser economizado quando seguimos os passos simples deste breve tutorial.

A linguagem PHP tem uma biblioteca de funções muito útil, mas ainda pouca usada: a GDLib, uma biblioteca de exibição gráfica (em inglês: *Graphics Display Library*). A GDLib tem sido utilizada principalmente em aplicações de geração de gráficos, mas seu potencial para desenvolvimento de mapas digitais é grande e não deve ser negligenciado. Leia a apresentação da biblioteca GDLib contida no site oficial do PHP [1]:

"O PHP não está limitado a criar apenas saída em HTML. Ele também pode ser usado para criar e manipular arquivos de imagem em uma diversidade de formatos, incluindo gif, png, jpg, wbmp, e xpm. E ainda mais convenientemente, o PHP pode enviar streams de saída de imagem diretamente para o browser. Você precisará compilar o PHP com a biblioteca GD de imagens de funções para isto funcionar. O PHP e GD também podem requerer outras bibliotecas, dependendo dos formatos de imagens com os quais você queira trabalhar."

Todas as funções gráficas da biblioteca GDLib possuem descrição detalhada no site oficial do PHP [2] e o leitor pode fazer uso dele para consultas eventuais durante o desenvolvimento do projeto.

1. Algumas funcionalidades disponíveis na GDLib

Abaixo estão listadas as funções utilizadas nesse tutorial e a respectiva descrição da sintaxe.

1) **ImageCreate** – Cria uma nova imagem;

ImageCreate (int \$width , int \$height)

Onde: \$width = largura em pixels; \$height = altura em pixels.

2) **ImageColorAllocate** - Aloca uma cor em uma variável;

imagecolorallocate (resource \$image , int \$red , int \$green , int \$blue)

Onde: \$image = imagem criada com imagecreate;

\$red, \$green e \$blue = códigos RGB para vermelho (R), verde (G) e azul (B).

3) **ImageRectangle** – Desenha um retângulo sem preenchimento;

ImageRectangle (resource \$image , int \$x1 , int \$y1 , int \$x2 , int \$y2 , int \$color)

Onde: \$image = imagem criada com imagecreate; \$x1,\$y1 = coordenada superior esquerda; \$x2,\$y2 = coordenada inferior direita; \$color = identificador da cor da linha.

4) **ImageFilledRectangle** – Desenha um retângulo preenchido;

imagefilledrectangle (resource \$image , int \$x1 , int \$y1 , int \$x2 , int \$y2 , int \$color)

Onde: \$image = imagem criada com imagecreate; \$x1,\$y1 = coordenada superior esquerda; \$x2,\$y2 = coordenada inferior direita; \$color = identificador da cor do preenchimento.

5) **ImagePolygon** – Desenha um polígono sem preenchimento;

ImagePolygon (resource \$image , array \$points , int \$num_points , int \$color)

Onde: \$image = imagem criada com imagecreate; \$points = matriz de vértices do polígono; \$num_points = número de vértices do polígono; \$color = identificador da cor de preenchimento.

Exemplo de matriz: points[0] = x0, points[1] = y0, points[2] = x1, points[3] = y1, ...

6) **ImageFilledPolygon** – Desenha um polígono preenchido;

ImageFilledPolygon (resource \$image , array \$points , int \$num_points , int \$color)

Onde: \$image = imagem criada com imagecreate; \$points = matriz de vértices do polígono; \$num_points = número de vértices do polígono; \$color = identificador da cor da linha.

7) **Imagestring** – Desenha uma string horizontalmente;

Imagestring (resource \$image , int \$font , int \$x , int \$y , string \$string , int \$color)

Onde: \$image = imagem criada com imagecreate; \$font = valores que podem variar de 1 a 5, dependendo do tamanho da fonte (por *default*, a codificação é latin2), onde números maiores correspondem a fontes mais largas; \$x = coordenada x superior à esquerda; \$y = coordenada y superior à esquerda; \$string = string a ser desenhada; \$color = identificador da cor da letra.

8) **Imageline** – Desenha uma linha;

Imageline (resource \$image , int \$x1 , int \$y1 , int \$x2 , int \$y2 , int \$color)

Onde: \$image = imagem criada com imagecreate; \$x1,\$y1 = coordenadas de início da linha; \$x2,\$y2 = coordenadas de fim da linha; \$color = identificador da cor da linha.

9) **ImagePNG** - Envia uma imagem PNG para o browser ou para um arquivo;

ImagePNG (resource \$image [, string \$filename])

Onde: \$image = imagem criada com imagecreate; \$filename = se indicado o caminho, será enviada a imagem para um arquivo.

10) **Imagedestroy** – Destrói uma imagem, liberando a memória associada a ela.

Imagedestroy (resource \$image)

Onde: \$image = imagem criada com imagecreate.

O tutorial de programação de scripts em PHP para geração de mapas temáticos segue os passos básicos listados a seguir. A execução demonstrada nesse artigo fundamenta-se no trabalho de classificação de municípios do Rio de Janeiro, onde foram gerados mapas temáticos que são visualizados diretamente no navegador [3].

2 . Preparação do PHP para execução das funções gráficas GD

Para começar a desenvolver projetos com GD, o programador deve estar atento à compilação do PHP com a GDLib para que as funções possam ser interpretadas corretamente. Além disso, se precisar expandir os recursos da GD, o programador deve estar atento à instalação de outras bibliotecas. A lista básica de bibliotecas a serem instaladas para que tenha mínimo de independência na programação segue adiante:

- **GDLib** - <http://www.libgd.org/releases/>

- **Lib JPEG** – Para gerar imagens JPEG a partir de scripts PHP com GD - <ftp://ftp.uu.net/graphics/jpeg>.

- **FreeType 2** - <http://www.freetype.org>.

No tutorial, a imagem a ser criada possui extensão PNG, que é suportada em versões de GD superiores à gd-1.6. Com relação a outras extensões de imagens supor-

tadas pela GD, observe que:

- **GIF** - Suportado apenas em versões do GD anteriores a gd-1.6. Suporte apenas para leitura está disponível com o PHP 4.3.0 e a biblioteca GD empacotada. Suporte para escrita está disponível desde o PHP 4.3.9 e PHP 5.0.1.

- **JPG** - Quando compilando a biblioteca jpeg-v6b (antes de compilar o PHP) você precisa usar a opção --enable-shared no passo de configuração. Se você não fizer, irá receber um erro alertando *libjpeg.(a|so) not found* quando você estiver no passo de configuração para compilar o PHP.

- **XPM** - se o sistema tiver um ambiente X instalado, é provável que a biblioteca já esteja disponível.

Descomprima e instale os pacotes no diretório de bibliotecas do seu sistema. Todos os trechos que estão entre colchetes devem ser substituídos pelas informações relativas à sua instalação.

Instale o GD:

```
# cd [diretorio do pacote gd]
# ./configure --with-jpeg-dir=[caminho do diretório da biblioteca Libjpeg] --with-png-dir=[caminho do diretório da biblioteca Libjpeg]
-- with-freetype-dir=[caminho do diretório da biblioteca freetype]
# make
# make install WITH_XPM=yes WITHOUT_X11=yes
```

Pare o servidor Apache se ele ainda estiver sendo executado:

```
# /usr/local/apache2/bin/apachectl stop
```

Recompile o PHP:

```
# ./configure - --with-apxs2 = [caminho do diretório apxs] --enable-track-vars --with-mysql
--enable-ftp --with-zlib --with-gd --with-jpeg
-dir=[caminho do diretório da biblioteca jpeg]
--with-png-dir=[caminho do diretório da biblioteca png] --with-freetype-dir=[caminho do diretório da biblioteca Freetype]
# make
# make install
```

Reinicie o Apache:

```
# /usr/local/apache2/bin/apachectl start
```

Execute a função **phpinfo()** para verificar se as bibliotecas foram instaladas corretamente.

Muitos documentos na web versam sobre como compilar o PHP com GD. Outras configurações de compilação são possíveis dependendo do servidor e/ou da versão das bibliotecas adquiridas e do PHP em uso. Aconselha-se que os pacotes de bibliotecas sejam baixados e que

se procure um tutorial específico na rede para a versão do Apache e do PHP que estiver utilizando, antes de começar a instalação.

3 . Planejamento do projeto

Muitas aplicações podem ser desenvolvidas com uso de mapas temáticos. Atualmente a informação geográfica tem lugar cada vez maior nos projetos de Internet. Este artigo apresenta o exemplo de uma classificação temática realizada em 34 municípios do Rio de Janeiro, envolvendo 22 temas. Para planejar o projeto, o programador deve ter em mente as necessidades de cada proposta. No exemplo, as principais características levadas em conta foram as seguintes:

- Temas: como são muito temas, foi criada uma codificação para facilitar a busca em banco de dados;
- Unidade geográfica de análise utilizada: Municipal.
- Meta-informações: Também armazenadas em banco de dados para exibição automática – Títulos de mapas, descrição dos temas, nomes dos municípios, nomes de fontes de dados etc.

4 . Planejamento do banco de dados

O tamanho e complexidade do banco de dados dependerão das características do projeto. Aqui, é apresentada uma estrutura de banco de dados simples, apenas para entendimento básico do processamento de informações na geração de um mapa temático com unidade geográfica municipal. Daqui, poderá ser personalizado e expandido, de acordo com cada projeto.

Tabelas:

- **mun:** tabela com meta-informações dos municípios. Campos: cod_mun (código do município), nome (nome do município);
- **temas:** tabela com meta-informações dos temas adotados. Campos: cod_tema (código de identificação do tema), desc_tema (descrição do tema);
- **fator:** tabela com os limites de intervalos de classes utilizados e com os valores mínimo e máximo de cada série de dados temáticos. No exemplo, foram usadas três classes, denominadas como "alto impacto", "médio impacto" e "baixo impacto". Assim, a tabela contém os campos: cod_tema (código de identificação do tema), L1 (valor numérico do limite entre as classes "baixo impacto" e "médio impacto"), L2 (valor numérico do limite entre as classes "médio impacto " e "alto impacto"), min (valor numérico mínimo encontrado na série de dados de determinado tema), max (valor numérico máximo encontrado na série de dados de determinado tema).

Para armazenar os dados referentes a cada tema, deve ser criada uma tabela para cada qual, com os seguintes campos: cod_mun (código de identificação do município) e valor (valor numérico do tema referente a determinado município). Esta medida torna mais ágil a consulta de dados pelo programa de geração do mapa

digital.

5 . Programando o script de geração dos mapas

O script de geração do mapa deve claro, destinado exclusivamente a esse fim. Se o programador inserir o script em uma página que já tem um cabeçalho HTML, incorrerá em erro. O uso da função include() do PHP também não é permitido.

A estrutura do programa PHP para geração de mapas digitais consiste das partes: 1) Cabeçalho do arquivo; 2) Conexão com o banco de dados; 3) Definição das cores; 4) Desenho da moldura do mapa; 5) Desenho do Título e do rodapé; 6) Desenho da legenda; 7) Desenho dos polígonos e 8) Geração, exibição e destruição da imagem.

A seguir, os trechos do código que correspondem a cada item e a explicação. Todos os trechos que estão em **negrito** devem ser substituídos com informações referentes ao seu projeto.

1) Cabeçalho do arquivo

```
header ("Content-type:image/png");  
$cod = código do tema que serve de base para esse mapa;  
$im = ImageCreate(890,650); //Criação da imagem.
```

2) Conexão com o banco de dados

```
$con = mysql_connect ("endereço do host do banco de dados", "usuário", "senha");  
$db = "nome do banco de dados";  
mysql_select_db ($db);
```

```
//Seleciona a tabela correspondente  
//ao tema em questão.  
$tab = $cod;
```

Visite nosso Portal



www.phpmagazine.org.br

3) Definição das cores

```
//cores da legenda
//cor amarela
$cor_02['R']=249;$cor_02['G']=248;$cor_02['B']=191;
//cor verde
$cor_01['R']=165;$cor_01['G']=200;$cor_01['B']=165;
//cor vermelha
$cor_03['R']=244;$cor_03['G']=150;$cor_03['B']=150;
//cor azul
$cor_04['R']=79;$cor_04['G']=191;$cor_04['B']=232;
//cor bege do fundo
$cor_05['R']=248;$cor_05['G']=244;$cor_05['B']=228;
//aloca as cores
$cor01 = ImageColorAllocate($im, $cor_01['R'],$cor_01['G'],$cor_01['B']);
$cor02 = ImageColorAllocate($im, $cor_02['R'],$cor_02['G'],$cor_02['B']);
$cor03 = ImageColorAllocate($im, $cor_03['R'],$cor_03['G'],$cor_03['B']);
$cor04 = ImageColorAllocate($im, $cor_04['R'],$cor_04['G'],$cor_04['B']);
$cor05 = ImageColorAllocate($im, $cor_05['R'],$cor_05['G'],$cor_05['B']);
$white = ImageColorAllocate($im, 255, 255, 255);
$black = ImageColorAllocate($im, 0, 0, 0);
```

4) Desenho da moldura do mapa

```
Imagerectangle($im,5,5,880,640,$black);
```

5) Desenho do título e do rodapé

```
//Escreve o titulo
//Pesquisa o nome do indicador no banco de dados
$result0 = mysql_query("SELECT desc_tema FROM temas WHERE cod_tema = '$cod'"
    or die ("Nao pode pesquisar ". mysql_error()));

while ($row0 = mysql_fetch_array($result0)){
    $desc_tema = $row0[0];
}

Imagestring($im,6,20,40,$desc_tema,$black);
//Escreve o rodapé
$string_fonte = "Fonte: Atlas de Indicadores de Sustentabilidade para os Municípios Costeiros do Es
tado do Rio de Janeiro, Brasil.";
imagestring($im,3,20,570,$string_fonte,$black);
$string_end = "http://www.ivides.org/atlas/mapas_ind.php, acessado em " . date("d") . "/" . date
("n") . "/" . date("Y");
imagestring($im,3,20,590,$string_end,$black);
$string_mail = "Contacte a autora: Raquel Dezidério (raquel.deziderio@gmail.com)";
imagestring($im,3,20,610,$string_mail,$black);
```

6) Desenho da legenda

```
//Cria o box da legenda
Imagefilledrectangle($im,712,340,872,560,$white);
//Cria a moldura do box da legenda
Imagerectangle($im,712,340,872,560,$black);
//Escreve a palavra "LEGENDA"
$string = "LEGENDA";
imagestring($im,5,765,350,$string,$black);
//Cria a linha que sublinha a legenda
ImageLine($im, 750, 370, 840, 370, $black);
//Pesquisa os intervalos das classes
$result0 = mysql_query("SELECT max, min, L1, L2 FROM fator WHERE cod_tema = '$cod'") or die
("Nao pode pesquisar ". mysql_error());
while ($row0 = mysql_fetch_array($result0)){
    $max = $row0[0];
    $max_aux = $max;
    $max = $max + 0.0100;
    $min = $row0[1];
    $min_exibe = $min;
    $min = $min - 0.0100;
    $L1 = $row0[2];
    $L1_aux = $L1 + 0.0001;
    $L2 = $row0[3];
```



```

        $L2_aux = $L2 + 0.0001;
    }
    //Desenha os retângulos dos intervalos
    //Escreve o primeiro intervalo
    $string1 = "Impacto Baixo";

    $string1c = "(" . $min_exibe . " - " . $L1 . ")";
    imagestring($im, 5, 748, 396, $string1,$black);
    imagestring($im, 3, 748, 411, $string1c,$black);
    //Escreve o segundo intervalo
    $string2 = "Impacto Médio";
    $string2c = "(" . $L1_aux . " - " . $L2 . ")";
    imagestring($im,5,748,456,$string2,$black);
    imagestring($im,3,748,471,$string2c,$black);

    //Escreve o terceiro intervalo
    $string3 = "Impacto Alto";
    $string3c = "(" . $L2_aux . " - " . $max_aux . ")";
    imagestring($im, 5, 748, 516, $string3, $black);
    imagestring($im, 3, 748, 531, $string3c, $black);
    //Desenha os quadradinhos
    imagefilledrectangle( $im, 720, 400, 740, 420, $cor01);
    imagerectangle( $im, 720, 400, 740, 420, $black);
    imagefilledrectangle( $im, 720, 460, 740, 480, $cor02);
    imagerectangle( $im, 720, 460, 740, 480, $black);
    imagefilledrectangle( $im, 720, 520, 740, 540, $cor03);
    imagerectangle( $im, 720, 520, 740, 540, $black);

```

7) Desenho dos polígonos

```

//Cria o polígono do mar através do vetor abaixo
$values_mar = array(
    0 => 5,
    1 => 544,
    2 => 5,
    3 => 640,
    4 => 880,
    5 => 640,
    6 => 880,
    7 => 84,
    8 => 736,
    9 => 133,
    10 => 739,
    11 => 232,
    12 => 500,
    13 => 413,
    14 => 430,
    15 => 443,
    16 => 380,
    17 => 382,
    18 => 340,
    19 => 390,
    20 => 286,
    21 => 453,
    22 => 215,
    23 => 430,
    24 => 151,
    25 => 452,
    26 => 120,
    27 => 437,
    28 => 49,
    29 => 459,
    30 => 5,
    31 => 544,
);

$n_mar = 16;

//Desenha o polígono do mar
imagefilledpolygon($im,$values_mar,$n_mar,$cor04);
imagepolygon($im,$values_mar,$n_mar,$black);

//Desenha o polígono do município 31 e atribui a cor adequada
$cod_mun = 31;

```

```
//Pesquisa o valor do tema referente
//ao municipio 31
$result31 = mysql_query
("SELECT valor FROM $tab WHERE
cod_mun = '$cod_mun'")
or die ("Nao pude pesquisar " . mysql_error
());
while ($row31 = mysql_fetch_array($result31)){
$valor31 = $row31[0];
$valor31_aux = $valor31;
$valor31 = number_format($valor31,2,".",".");
}
//Define o poligono para o municipio 31
$values_31 = array(
0 => 294,
1 => 423,
2 => 296,
3 => 423,
4 => 297,
5 => 422,
6 => 298,
7 => 422,
8 => 299,
9 => 421,
10 => 306,
11 => 421,
12 => 305,
13 => 412,
14 => 301,
15 => 412,
16 => 300,
17 => 413,
18 => 297,
19 => 413,
20 => 295,
21 => 414,
22 => 295,
23 => 416,
24 => 293,
25 => 416,
26 => 292,
27 => 417,
28 => 291,
29 => 418,
30 => 291,
31 => 419,
32 => 294,
33 => 422,
34 => 294,
35 => 423,
);
$n_31 = 18;
//Testa em que intervalo está o valor
//recuperado para o município
if (($valor31 > $min)&&($valor31 < $L1)){
imagefilledpolygon
($im,$values_31,$n_31,$cor01);
}
if (($valor31 >= $L1)&&($valor31 <= $L2)){
imagefilledpolygon
($im,$values_31,$n_31,$cor02);
}
if (($valor31 <= $max)&&($valor31 > $L2)){
imagefilledpolygon
($im,$values_31,$n_31,$cor03);
}
//Desenha a borda do polígono 31
imagepolygon($im,$values_31,$n_31,$black)
```

8) Geração, exibição e destruição da imagem

```
ImagePNG($im);
Imagedestroy($im);
```

6. Considerações finais

Neste artigo, descrevemos somente o trecho de código de um dos municípios avaliados para fim de exemplificação. Para ver o código PHP utilizado na aplicação real que serviu de base para esse tutorial, visite: http://www.ivides.org/atlas/mapas_ind.php e clique no link "código-fonte do mapa". Agradecimentos especiais a toda a equipe da PHP Magazine pela oportunidade.

Referências e links sugeridos

- [1] http://br2.php.net/manual/pt_BR/intro.image.php - Apresentação da GDLib no php.net .
- [2] http://br2.php.net/manual/pt_BR/ref.image.php - Lista das funções da GDLib no php.net .
- [3] http://www.ivides.org/atlas/mapas_ind.php - Atlas de Indicadores de Sustentabilidade para os Municípios Costeiros do Estado do Rio de Janeiro.

Raquel D. Souto - raquel.deziderio@gmail.com

Bacharel em Oceanografia pela Universidade do Estado do Rio de Janeiro (2005). Especialista em instalação e configuração de sistemas operacionais *like Unix*, instalação e configuração de SGBD MySQL. Atua desde 2004 em desenvolvimento Web com PHP, de aplicações administrativo-financeiras e geocientíficas. Atualmente seus principais projetos são: portal do "Atlas de Indicadores de Sustentabilidade para os Municípios Costeiros do Estado do Rio de Janeiro" - <http://www.ivides.org/atlas> e portal do "Instituto Virtual para o Desenvolvimento Sustentável - IVIDES.org" - <http://www.ivides.org>.

Submissão de Artigos

Você também pode participar das edições da PHP Magazine. A sua participação é de interesse de toda a comunidade, portanto, visitante, efetue seu [cadastro](#) e [envie-nos](#) algum artigo sobre os tópicos disponíveis. Seu trabalho será avaliado e os melhores serão publicados na revista.

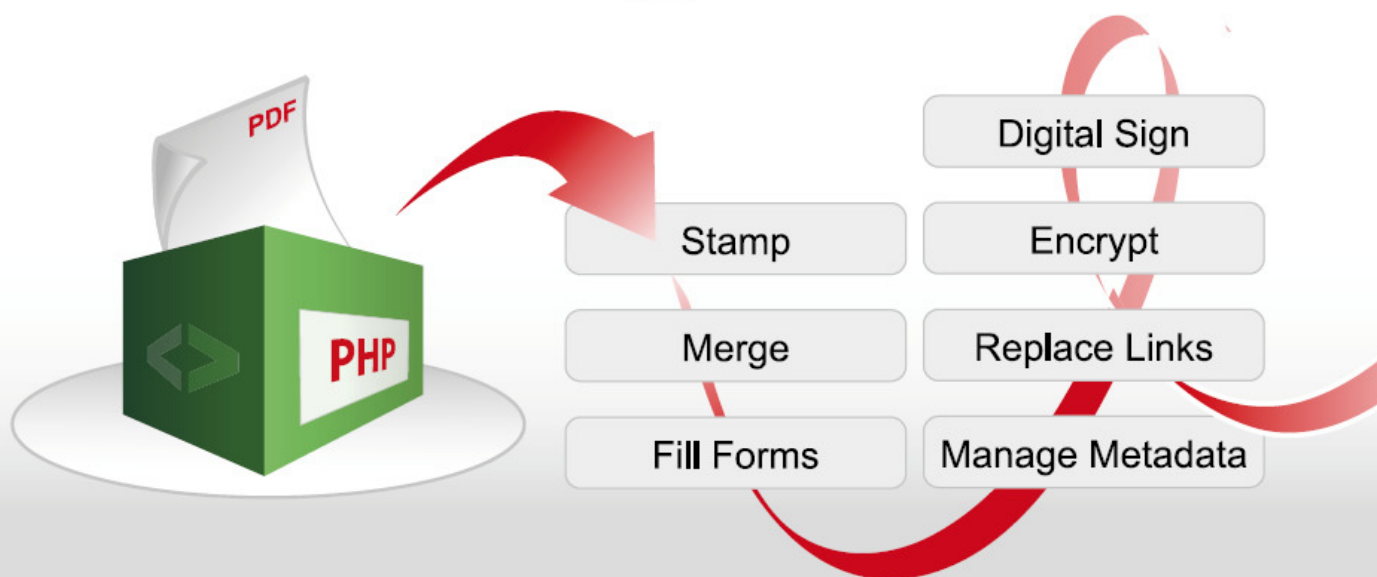
Temas propostos:

- segurança;
- banco de dados;
- configuração;
- tendências;
- frameworks;
- aplicações (cms, wiki, genéricas de projeto, erp);
- IDEs;
- compiladores;
- engines (e-acelator, ZEND);
- webservices;
- paradigmas de programação;
- miscelânea (xml, impressão, ajax, certificação, mercado);
- GTKs
- ferramentas/projetos (não apenas os opensources);

O artigo deverá ser redigido em **português**, inclusive o resumo, ou seja, não possui abstract. Estamos recebendo arquivos no formato do Microsoft Office e Open Office, para isso, sugerimos um [template](#) específico. Os autores deverão se basear nos estilos aplicados na formatação deste modelo.

Teremos o imenso prazer em analisar o seu trabalho e disponibilizá-lo em nossa revista. Mais informações sobre a submissão de artigos estão disponíveis no portal da revista.

Participe!



SetaPDF - PDF Manipulation in PHP

www.setasign.de

Integrando Ferramentas na construção de um calendário

Este artigo introdutório visa abordar todos os passos na criação de um módulo de calendário. O artigo é voltado para o público iniciante, através do desenvolvimento de um calendário simples. Leitores mais experientes poderiam continuar o desenvolvimento para tornar o script um módulo integrado a um sistema de gerenciamento comercial.

O objetivo é criar uma tabela dinâmica do mês atual, mostrando todos os dias em ordem, conforme indicado na figura 1.

<< setembro / 2008 >>						
D	S	T	Q	Q	S	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

Figura 1 – Aspecto final do calendário proposto.

1. Requisitos do sistema

O sistema a ser desenvolvido deve ser capaz de identificar a data atual e montar o calendário referente ao mês. O sistema também deve ser capaz de permitir ao usuário que mude de mês, ou seja, deve possuir dois botões: um para o mês anterior e outro para o mês posterior. A tabela deve possuir os domingos marcados na cor vermelha. Através de um banco de dados, os compromissos marcados devem ter suas datas mostradas em verde.

2. Definições do sistema

Depois da definição dos requisitos do sistema, segue-se à escolha das ferramentas e linguagens que serão utilizadas para o cumprimento dos requisitos.

Para a tabela, será utilizado HTML e, para o processamento dinâmico, evidentemente, será utilizado o PHP. O JavaScript enviará as requisições para o servidor quando se desejar alterar o mês. As diversas combinações de cores e fontes serão definidas através do CSS. O banco de dados pode ser criado com qualquer servidor compatível com o PHP. Neste artigo, foi utilizado o MySQL.

QL devido à familiaridade do autor com o mesmo.

3. HTML

A linguagem HTML (*HyperText Markup Language*) foi criada por Tim Berners-Lee. A versão atual do HTML 4.0 foi publicada em 1997 e é mantida e regulada pelo W3C (*World Wide Web Consortium*). Também está disponível a utilização do XML como linguagem de marcação padrão, também regulada por este órgão.

O código HTML é a parte mais importante de toda página de Internet e de todo script, pois, mesmo se a programação estiver correta, sem as marcações (ou tags) HTML adequadas, dificilmente obtém-se o resultado esperado.

Outro aspecto importante da marcação HTML é a compatibilidade entre os navegadores. Uma vez que o script PHP é processado no servidor, ele se torna transparente para o cliente, sendo responsabilidade do navegador a interpretação das marcações HTML. Se o HTML não for compatível com o tipo ou a versão do navegador em uso, a aparência e até mesmo os efeitos empregados podem ser diferentes do esperado.

O objetivo deste artigo não é introduzir a codificação HTML, nem mesmo as outras linguagens já citadas. Entretanto, é preciso que o leitor esteja ciente da sua importância.

O código HTML inicia pela tag **html**. Na sequência, procedem-se as tags **head** e **title**. A tag **head** e todo o conteúdo postado em seu interior dizem respeito aos dados que não fazem parte das componentes visuais do corpo da página.

O corpo da página, por sua vez, é delimitado pela tag **body**. É dentro desta tag que será definida a posição centralizada na página (tag **center**), a tabela (tag **table**) com suas linhas (tag **tr**) e colunas (tag **td**), além do formulário (tag **form**) com seus campos (tag **input**). Sugere-se ao leitor, que por ventura não esteja totalmente familiarizado com a linguagem HTML, que procure na Internet por tutoriais ou materiais para aprendiza-

do desta linguagem, de caráter fundamental para quem deseja programar em PHP para a Internet. A especificação da W3C para o HTML 4.01 está disponível em português no website do consórcio.

O código HTML utilizado neste artigo está reproduzido na sequência. As regiões marcadas com a seta seguida de um número (➔ X) representam áreas onde serão inseridos scripts. A tabela pode parecer confusa, mas, se

o leitor analisar atentamente, perceberá que se trata de uma tabela inserida em uma célula de outra tabela.

O formulário e seus elementos possuem nomes atribuídos através da propriedade **name**. É através do nome que o JavaScript irá atuar sobre a página. Os nomes iniciam com *f* para formulário e *h* para campos tipo oculto (*hidden*) por convenção do autor.

➔ 01

```
<html>
  <head>
    <style type="text/css">
      ➔ 02
    </style>
    <script language="javascript" type="text/javascript">
      ➔ 03
    </script>
    <title>Título da Página</title>
  </head>
  <body>
    <center>
      <table cellpadding="0" cellspacing="0" border="1" width="140">
        <tr>
          <td colspan="7" width="140"><table cellpadding="0" cellspacing="0" border="0"
width="140">
            <tr>
              <td width="20" align="center"><a href="➔ 04" class="domingo">&laquo;</
a></td>
              <td width="100" colspan="5" align="center">➔ 05</td>
              <td width="20" align="center"><a href="➔ 06" class="domingo">&raquo;</
a></td>
            </tr>
          </table></td>
        </tr>
        <tr>
          <td width="20" align="center">D</td>
          <td width="20" align="center">S</td>
          <td width="20" align="center">T</td>
          <td width="20" align="center">Q</td>
          <td width="20" align="center">Q</td>
          <td width="20" align="center">S</td>
          <td width="20" align="center">S</td>
        </tr>
      </table>
    </center>
    ➔ 07
    <form name="fcalendario" action="➔ 08" method="post" enctype="multipart/form-
data">
      <input type="hidden" name="hmes" value="➔ 09">
      <input type="hidden" name="hano" value="➔ 10">
      <input type="hidden" name="hacao" value="">
      <input type="hidden" name="henviado" value="0">
    </form>
  </body></html>
```


4. JavaScript

A linguagem de programação JavaScript foi desenvolvida pela Netscape em 1995 para realizar a validação dos formulários no lado do cliente e para interagir com os elementos da página. O método AJAX (*Asynchronous Javascript And XML*) faz uso desta linguagem para produzir efeitos como os do Gmail.

Quando se deseja alterar o mês, é preciso enviar uma requisição ao servidor indicando qual o mês desejado. Isto pode ser feito através de botões HTML dentro do formulário. Porém, desejou-se uma interface mais agradável, obtida através do JavaScript.

Conforme mencionado, a identificação do elemento alvo na página é feita através da propriedade **name** do HTML. Foram criadas duas funções, **MesAnterior()** e **MesPosterior()**. Estas funções são definidas dentro da

área 03 e chamadas nas áreas 04 e 06. A definição das funções e as chamadas são reproduzidas na sequência.

A lógica empregada é a seguinte: O código PHP irá preencher os campos *hano* e *hmes*. Ao clicar nos hiperlinks **«** («) e **»** (»), são ativadas as funções específicas que modificam os campos *hacao* (para **a** ou **p**) e *henviado* (para **1**). O campo *hacao* indica se o mês desejado é o anterior ou o posterior. O campo *henviado* é um campo de controle, inserido para enviar requisições eu não tenham sido enviadas através dos hiperlinks.

Outros dispositivos de segurança podem ser inseridos no script, entretanto, neste artigo apenas será abordado este.

O processamento do formulário enviado será feito através do PHP na mesma página, na área marcada com o número 01.

➔ Área 03

```
function MesAnterior()  
{  
    document.fcalendario.hacao.value = "a";  
    document.fcalendario.henviado.value = 1;  
    document.fcalendario.submit();  
}  
  
function MesPosterior()  
{  
    document.fcalendario.hacao.value = "p";  
    document.fcalendario.henviado.value = 1;  
    document.fcalendario.submit();  
}
```

➔ Área 04

```
javascript:MesAnterior();
```

➔ Área 06

```
javascript:MesPosterior();
```



DIGITAL 3i
O seu site no domínio certo

Porque você tem que usar os planos das empresas de hospedagem ???

A Digital 3i acredita que não tem pessoa melhor para saber o quanto vai precisar de espaço do que o próprio webmaster.

Por isso, revolucionamos os planos de hospedagem, criando **setups flexíveis** de acordo com a necessidade do seu website.

Leitor da PHP Magazine tem 10% de desconto

www.DIGITAL3i.com.br **Cód: pm01**

5. MySQL

Neste exemplo puramente didático, uma base de dados bastante simples foi implementada com apenas uma tabela MySQL de quatro campos, conforme listado a seguir. Cabe salientar que o sistema pode ser melhorado através da implementação de um banco de dados mais completo. A data e a hora poderiam ter sido inseridas dentro de um mesmo campo DATETIME, porém, para facilitar a busca na base, foi preferível separar estes campos.

```
compromissos latin1_general_ci
=====
codigo      INT(10)      UNSIGNED PRIMARY
data        DATE
hora        TIME
descricao TEXT      latin1_general_ci
```

➔ Área 01

```
<?
// Arquivos necessários
require_once("config.php");
// Conexão com o banco de dados
$banco = @mysql_connect($mysql['servidor'], $mysql['usuario'], $mysql['senha'], 1);
if($banco == false)
{
    echo "Não foi possível estabelecer conexão com o banco de dados";
    exit(0);
}
$resp = @mysql_select_db($mysql['banco'], $banco);
if($resp == false)
{
    echo "Não foi possível acessar a base de dados";
    exit(0);
}
unset($mysql);
```

A primeira instrução do script é uma chamada para inclusão de outro arquivo. Neste arquivo, foi criado um vetor chamado **\$mysql** contendo os dados do servidor do banco de dados. Esta abordagem é bastante prática, pois, caso os dados do servidor mudem, é possível se alterar apenas um arquivo ao invés de todos os scripts do Website. Outra vantagem é que, com as variáveis em apenas um vetor, elas podem ser facilmente apagadas após seu uso, por meio da chamada de apenas uma instrução **unset()**. Assim, as variáveis existem por um período muito curto durante o processamento do script.

As funções de acesso ao banco de dados são chamadas precedidas pelo caractere **@** para evitar que os erros sejam ecoados na tela. O tratamento dos erros não é o mais adequado, porém, para fins didáticos, é o suficiente.

6. PHP

O código PHP pode ser incluído em praticamente todos os pontos da página. Entretanto, é uma boa prática de programação criar-se as variáveis no início do script e chamá-las quando necessário. Atenção, a utilização das boas práticas de programação facilita o entendimento do script, tanto pelo próprio programador, quanto por terceiros; isto não significa que as outras técnicas não sejam válidas. Antes de seguir quaisquer recomendações, o mais importante é que o programador possua um padrão pessoal de programação e o utilize sempre.

Normalmente, as primeiras funções a serem executadas num script são: abertura de sessões, validação do usuário e comunicação com o banco de dados. Uma vez que o script em questão não lida com sessões, nem possui usuários, deve-se conectar o banco de dados, conforme script a seguir.

Na sequência, é necessário se criar um vetor com o nome dos meses em português. Isto é necessário, pois a função **date()** do PHP retorna o nome dos meses e dos dias da semana apenas em inglês. Uma vez que a função **date()** retorna os meses em formato numérico de 1 a 12, esta é a forma utilizada para indexação do vetor, conforme indicado no script a seguir.

Conforme mencionado nas sessões anteriores, o usuário pode enviar um formulário para modificar o mês desejado. Para tal, é necessário que os dados enviados pelo formulário sejam processados. Com este intuito, pode-se utilizar a variável **\$_REQUEST**. Esta variável é bastante útil, já que agrega todos os dados das variáveis **\$_GET** e **\$_POST**, encarregadas de armazenar os dados provenientes de formulários ou requisições pelos métodos **GET** e **POST**, respectivamente.

A chamada das datas é feita através da função **date()**. Para selecionar uma data específica, utiliza-se a função **mktime()**. É preciso ser bastante cauteloso ao utilizar esta função, pois a ordem dos argumentos não é muito ortodoxa: hora, minuto, segundo, mês, dia e ano.

Há um aspecto bastante peculiar na formação da linguagem PHP. Nas linguagens fortemente tipadas, as variáveis devem ser declaradas antes de sua utilização e o tipo de dados que elas irão armazenar deve ser especi-

ficado. Este é o caso das linguagens C, C++ e suas derivadas. A linguagem PHP, entretanto, permite que o usuário utilize uma variável sem especificar o tipo. Apesar de ser bastante fácil trabalhar desta forma, pois as variáveis podem ser convertidas do tipo numérico para o tipo string com grande facilidade, a conversão reversa não é automática.

➔ Continuação da área 01

```
// Textos
$meses[1] = "janeiro";
$meses[2] = "fevereiro";
$meses[3] = "março";
$meses[4] = "abril";
$meses[5] = "maio";
$meses[6] = "junho";
$meses[7] = "julho";
$meses[8] = "agosto";
$meses[9] = "setembro";
$meses[10] = "outubro";
$meses[11] = "novembro";
$meses[12] = "dezembro";

// Recebe os dados
if(isset($_REQUEST['enviado']))
{
    if($_REQUEST['acao'] == 'a')
    {
        $data['mes'] = (int)(date('m',mktime(1,1,1,(int)($_REQUEST['hmes']) - 1,1,(int)($_REQUEST['hano']))));
        $data['ano'] = (int)(date('Y',mktime(1,1,1,(int)($_REQUEST['hmes']) - 1,1,(int)($_REQUEST['hano']))));
    }
    if($_REQUEST['acao'] == 'p')
    {
        $data['mes'] = (int)(date('m',mktime(1,1,1,(int)($_REQUEST['hmes']) + 1,1,(int)($_REQUEST['hano']))));
        $data['ano'] = (int)(date('Y',mktime(1,1,1,(int)($_REQUEST['hmes']) + 1,1,(int)($_REQUEST['hano']))));
    }
}
else
{
    // Data atual
    $data['mes'] = (int)(date('m'));
    $data['ano'] = (int)(date('Y'));
}

// Dados complementares
$data['bissexto'] = (($data['ano'] % 4) == 0)?1:0;
if(($data['mes'] == 1)||($data['mes'] == 3)||($data['mes'] == 5)||($data['mes'] == 7)||($data['mes'] == 8)||($data['mes'] == 10)||($data['mes'] == 12))
    $data['dias_no_mes'] = 31;
elseif(($data['mes'] == 2)&&($data['bissexto']))
    $data['dias_no_mes'] = 29;
elseif(($data['mes'] == 2))
```

```
$data['dias_no_mes'] = 28;
else
    $data['dias_no_mes'] = 30;
```

Os dados armazenados pela variável **\$_REQUEST** são provenientes de um campo hidden do formulário e, portanto, são do tipo string. Entretanto, a função **mktime()** requer que um dado numérico seja passado. Para realizar a conversão, a fim de poder somar ou subtrair 1 do mês, é conveniente antes **garantir** que o dado esteja convertido para o tipo inteiro. Isto é conseguido através de um typecast (**int**).

Nesta seção inicial, ainda falta definir quais as datas possuem compromissos marcados. Este procedimento pode ser realizado de diversas formas. No entanto, como para este exemplo não é necessário especificar a quantidade nem mesmo descrever o compromisso marcado para o dia. É possível pesquisar-se apenas se há com-

promissos marcados para cada um dos dias do mês.

No exemplo, foi criado um vetor cujo valor inicial foi **false**. Este vetor será o responsável por armazenar os dias com compromissos marcados. Se o banco de dados retornar um ou mais elementos registrados para a data especificada, então o elemento é criado. Utilizando como exemplo a figura 1, este pedaço do script irá criar um vetor esparsa com apenas os elementos 10, 18, 22 e 28, marcados com o valor **true**. O script em questão está reproduzido na sequência.

Todos os dados necessários já foram criados e processados, portanto, as áreas 05, 08, 09 e 10 podem ser inseridas por entre as marcações HTML.

➔ Continuação da área 01

```
// Busca as datas marcadas
$marcado = false;
for($i = 1;$i < ($data['dias_no_mes'] + 1);$i++)
{
    $sql = "SELECT COUNT(*) FROM `compromissos` WHERE `data` = '" . $data['ano'] . "-" .
    $data['mes'] . "-";
    if($i < 10)
        $sql .= "0";
    $sql .= $i . "'";
    $resp = @mysql_query($sql,$banco);
    if($resp == false)
    {
        echo "Erro no envio da requisição ao banco de dados";
        exit(0);
    }
    $linha = @mysql_fetch_array($resp,MYSQL_NUM);
    if($linha[0] > 0)
        $marcado[$i] = true;
}
?>
```

O uso da orientação a objetos juntamente com o emprego de boas práticas de programação nos possibilita manter um ritmo sustentável no desenvolvimento de aplicações. O foco deste livro é demonstrar como se dá a construção de uma aplicação totalmente orientada a objetos. Para isso, implementaremos alguns padrões de projeto (design patterns) e algumas técnicas de mapeamento objeto-relacional, além de criarmos vários componentes para que você possa criar complexas aplicações de negócio com PHP.

www.novatec.com.br



A área 05 escreve o mês e o ano. A área 08 é o nome da página atual para onde o formulário será enviado. As áreas 09 e 10 armazenam o mês e o ano nos campos do formulário.

→ Área 05

```
<? echo $meses[$data['mes']] . " / " .  
$data['ano']; ?>
```

→ Área 08

```
<? echo $_SERVER['PHP_SELF']; ?>
```

→ Área 09

```
<? echo $data['mes']; ?>
```

→ Área 10

```
<? echo $data['ano']; ?>
```

O calendário é montado na tabela através do script na área 07. Basicamente, um laço **while** é utilizado para percorrer os dias do mês e uma série de condicionais **if** posicionam os dias do mês em suas posições corretas na tabela. Para facilitar a visualização do código, a área 07 possui os textos 07a, 07b e 07c, reproduzidos na sequência.

Dentro do laço **while** é montada cada linha da tabela, onde os dias da semana são verificados com o conjunto de condições **if**. A tabela deve ser preenchida com caracteres vazios ** ** até que o primeiro dia do mês seja encontrado. As áreas 07a, 07b e 07c são muito parecidas, pois a lógica empregada é praticamente a mesma. Nas três áreas, o valor X simboliza o dia da semana: de 0 (para domingo) até 6 (para sábado).

→ Área 07

```
<? $i = 1;while($i <= $data['dias_no_mes']){ ?>  
    <tr>  
        <td width="20" align="center">07a>07c</td>  
        <td width="20" align="center">07b>07c</td>  
        <td width="20" align="center">07b>07c</td>  
        <td width="20" align="center">07b>07c</td>  
        <td width="20" align="center">07b>07c</td>  
        <td width="20" align="center">07b>07c</td>  
        <td width="20" align="center">07b>07c</td>  
    </tr>  
    <? } ?>
```

Outro aspecto importante diz respeito às marcações HTML. As tabulações e espaços são representados no código HTML, entretanto, visualmente, apenas um espaço é apresentado na tela. Isto causa a falsa impressão de que o código HTML não é sensível à endentação.

Isto se torna bastante perceptível quando formatamos a página através de tabelas. O conteúdo entre as marcações **<TABLE>** e **</TABLE>** e **<TR>** e **</TR>** não apresentam problemas com a formatação, pois elas não

A cláusula **if(\$i <= \$data['dias_no_mes'])** verifica se ainda há algum dia que não foi escrito no calendário. Na área 07c, que é onde o número dia será efetivamente escrito, caso o número de dias já tenha sido alcançado, o caractere espaço (** **) será escrito.

A cláusula **if(date(...)) == X)** verifica se o dia do mês a ser escrito está na coluna correta da tabela. Nas áreas 07a e 07b, o resultado irá guiar a formatação do texto escrito na célula, com a diferença de que a área 07a é utilizada para os domingos e a 07b para os demais dias. Dentro desta condição, verifica-se se há algum compromisso marcado. As classes CSS utilizadas serão discutidas na próxima parte.

Na área 07c, após o dia do mês ter sido impresso, o contador **\$i** é incrementado para que o próximo dia possa ser processado. Uma vez que o processamento do **while** é realizado dentro da tabela e o contador pode ser incrementado a cada célula, a cada repetição do laço **while** a variável contadora poderá ter sido incrementada até sete vezes.

Um detalhe que muitas vezes passa despercebido é a endentação do código PHP e do código HTML. O código PHP é processado no servidor, o que significa que endentar com espaços, tabulações ou escrever o código em uma linha apenas reflete meramente o perfil do programador. O autor tem preferência por formatar as áreas 01 a 07 como representado neste artigo. Entretanto, quanto aos códigos 07a, 07b e 07c, inseridos diretamente entre as marcações HTML, a preferência é escrever na mesma linha. Por motivos didáticos o código foi endentado.

guardam conteúdos. Porém, as marcações **<TD>** e **</TD>** devem ser escritas juntas, isto é, na mesma linha e sem espaços ou tabulações separando-as. Como exemplo, reproduza o código a seguir para verificar a diferença.

➔ Código exemplo

```
<table border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td width="50" height="1" bgcolor="#0000ff"><img width="1" height="1"></td>
  </tr>
  <tr>
    <td width="50">Algo escrito</td>
  </tr>
  <tr>
    <td width="50" height="1" bgcolor="#0000ff"><img width="1" height="1"></td>
  </tr>
</table>
```

```
<table border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td width="50" height="1" bgcolor="#0000ff">
      <img width="1" height="1">
    </td>
  </tr>
  <tr>
    <td width="50">
      Algo escrito
    </td>
  </tr>
  <tr>
    <td width="50" height="1" bgcolor="#0000ff"> <img width="1" height="1"> </td>
  </tr>
</table>
```

Na segunda tabela, as linhas 1 e 3 não possuirão altura de apenas um pixel, pois o navegador considera que a quebra de linha é um caractere que deve ser impresso. O resultado pode não parecer muito significativo,

porém, quando o *layout* do Website inteiro é baseado em tabelas, o resultado pode ser bem diferente do desejado.

➔ Área 07a

```
<?
if($i <= $data['dias_no_mes'])
{
  if(date("w",mktime(1,1,1,$data['mes'],$i,$data['ano'])) == X)
  {
    if(isset($marcado[$i]))
    {
      ?> class="marcado"<? }else{ ?> class="domingo"<? }}} ?>
```

➔ Área 07b

```
<?
if($i <= $data['dias_no_mes'])
{
  if(date("w",mktime(1,1,1,$data['mes'],$i,$data['ano'])) == X)
  {
    if(isset($marcado[$i]))
    {
      ?> class="marcado"<? }}} ?>
```

➔ Área 07c

```

<?
if($i <= $data['dias_no_mes'])
{
    if(date("w",mktime(1,1,1,$data['mes'],$i,$data['ano'])) == X)
    {
        echo $i;
        $i++;
    }
    else
    {
        echo "&nbsp;";
    }
}
else
{
    echo "&nbsp;";
}
?>

```

7. CSS

As folhas de estilo em cascata (CSS) foram desenvolvidas para criar uma abordagem consistente que fornecesse informação de estilos para documentos na *Web*. A idéia é formatar vários aspectos da página e de várias páginas com apenas a inserção de uma instrução, geral-

mente na forma de classe. A maneira mais prática de se utilizar uma formatação CSS é colocar toda a formatação em um documento à parte, com a extensão **css**, e anexá-lo a cada página com o comando HTML <LINK>. Neste artigo, foi preferível utilizá-lo dentro do próprio código HTML por meio da marcação <STYLE>, na área 02.

➔ Área 02

```

a.domingo:link
{
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 10px;
    font-weight: bold;
    color: #ff0000;
}
a.domingo:visited
{
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 10px;
    font-weight: bold;
    color: #ff0000;
}
a.domingo:hover
{
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 10px;
    font-weight: bold;
    color: #008000;
}
td
{

```

```

font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 10px;
font-weight: bold;
}
td.domingo
{
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 10px;
font-weight: bold;
color: #ff0000;
}
td.marcado
{
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 10px;
font-weight: bold;
color: #008000;
}

```

O código CSS trabalha através de classes e marcações HTML. As classes, quando presentes, são definidas após o ponto (**domingo**, **marcado**). As marcações, quando presentes, são definidas antes das classes (**td.marcado**, **td.domingo** e **a.domingo**) ou sem nenhuma classe associada (**td**). Neste último caso, será considerada como formatação padrão para aquela marcação HTML. Os dois pontos definem os eventos associados aos hyperlinks (**link**, **visited**, **active** e **hover**). Dentro de cada bloco são definidas diversas propriedades de cada classe ou marcação, como por exemplo, tipo, tamanho, formatação e cor da fonte. Outras propriedades dos mais diversos elementos HTML estão disponíveis, o que torna esta ferramenta bastante prática e versátil.

Considerações finais

Espera-se que este artigo tenha guiado o leitor por durante todo o processo de desenvolvimento de um *Website* dinâmico. As ferramentas mais consolidadas foram apresentadas, bem como uma forma de integrá-las em um mesmo projeto.

Salienta-se ao leitor que a formatação, as técnicas de programação e o modo de escrever o código no presente documento não devem ser considerados como únicas fontes, nem como as mais corretas. Estes quesitos são pessoais e cada programador possuirá ou desenvolverá a maneira que melhor lhe convier.

Referências e links sugeridos

[W3C] – <http://www.w3c.br>

[PHP Manual] – <http://www.php.net>

Leandro Schwarz - leandroschwarz@gmail.com

Engenheiro eletricitista pela Universidade Federal de Santa Catarina (UFSC) e Mestre em Engenharia Elétrica (área de concentração: Engenharia Biomédica) pela UFSC, Leandro atua desde 2000 com desenvolvimento Web e possui sólidos conhecimentos em PHP e MySQL.

