

Codificação

adicione clareza ao seu código

LCMS

conheça alguns CMS para ambientes de aprendizagem

PHP-GTK

desenvolvendo uma agenda

Segurança

conheça 6 requisitos mínimos

Vetores e Matrizes

abordando mais funcionalidades

IDEs

Um tour através de grandes IDEs para PHP

- Delphi
- DreamWeaver
- Eclipse PDT

Eventos

fisl 8.0
I PHP Road Show

ISSN 1981-044X



Guará

Conheça uma solução em software livre para obter redução de custos no desenvolvimento de aplicações Web.



Caro leitor,

Prosseguindo no árduo trabalho de lhe proporcionar uma boa leitura, estamos lançando a terceira edição da única revista sobre linguagem PHP em português.

A partir deste momento a revista passará para uma nova fase em que os novos voluntários começam a trabalhar efetivamente. Fruto da organização interna da revista, da definição de funções e dos objetivos traçados.

Em maio, ministramos uma palestra sobre a PHP Magazine no PHP Road Show, primeiro evento de PHP no Distrito Federal. Percebemos que a maioria dos participantes já conhecia a revista, o que nos deixa muito satisfeitos com nosso trabalho. E, melhor ainda, ao final da palestra várias pessoas se ofereceram para integrar nossa equipe de colaboradores. Assim, inicialmente damos as boas vindas ao Thiago Inácio, nosso novo design gráfico, que já está com a mão na massa.

Não podemos deixar de agradecer aos autores que nos prestigiam com seus artigos altamente qualificados. Agradecemos, principalmente, a todos que já escreveram em outras edições. Aos novos autores, damos as nossas boas-vindas e já aguardamos os próximos trabalhos.

Estamos também comemorando o marco de 5.000 assinantes que nos prestigiam e nos motivam a seguir com este trabalho. Agradecemos, portanto, aos moderadores de listas famosas de tecnologia que, de maneira bastante amigável, contribuem na divulgação da revista e da linguagem PHP em si.

Nesta edição você encontrará matérias de temas variados e importantes. A matéria principal traz um pequeno demonstrativo das IDE's mais novas para PHP, assunto que tem sido muito discutido nas listas de PHP Brasil a fora. Há também um artigo sobre segurança, que apresenta alguns passos para programar de maneira segura, e outro artigo sobre codificação limpa, com dicas para os iniciantes deixarem seus códigos mais limpos e agradáveis. Você lerá também outros artigos interessantes sobre PHP-GTK, Guará, Arrays e gerenciamento de conteúdo de aprendizagem. Além disso, preparamos um resumo sobre o 8º Fórum Internacional de Software Livre (FISL 8.0) e do 1º PHP Road Show.

Mais uma vez a administração da PHP Magazine agradece a você pelo apoio. Lembramos que o sucesso da revista é fruto da sua leitura e da sua divulgação entre amigos, colegas de trabalho e em aulas.

Boa leitura.

Equipe PHP Magazine

Editores

Flávio Z. Fagundes, zach@phpmagazine.com.br

Ricardo Aragão, ricardoaragao@phpmagazine.com.br

Administração

Flávio Z. Fagundes, zach@phpmagazine.com.br

Norberto Augusto, augusto@phpmagazine.com.br

Ricardo Aragão, ricardoaragao@phpmagazine.com.br

Comercial

Norberto Augusto

Projeto gráfico

Ricardo Aragão

Flávio Z. Fagundes

Thiago Inácio

Revisão técnica

Ricardo Aragão da Silva

Flávio Zacharias Fagundes

Revisão português

Camilla Carvalho

Correspondentes e colaboradores

Adler Brediks Medrado

Cleber Sousa

Er Galvão Abbott

Josiane Lima de Oliveira

Leandro Schwarz

Pablo Dall'Oglio

Pascoal Vernieri

SUMÁRIO

3, Criando uma Agenda com PHP-GTK e SQLite

7, Tratamento de Vetores e Matrizes em PHP – Parte II

11, Escrevendo código PHP de fácil manutenção

14, Ambientes de livre distribuição para gerenciamento de conteúdo de aprendizagem

19, Um pequeno tour pelas IDEs: PDT, Delphi e Dreamweaver CS3

27, Guará: Uma solução em software livre para redução de custos no desenvolvimento de aplicações Web

31, Segurança no PHP: Os 6 requisitos mínimos

36, 8º Fórum Internacional de Software Livre

38, 1º. PHP Road Show

Criando uma Agenda com PHP-GTK e SQLite

Neste artigo iremos construir uma ferramenta para gestão de compromissos em PHP-GTK utilizando o banco de dados SQLite para armazenar os dados.

Como a maioria de vocês já sabe, o PHP conecta em todos os bancos de dados importantes existentes (Postgres, Mysql, Oracle, SqlServer, Firebird, DB2, dentre outros). Na biblioteca GTK programamos o acesso ao banco de dados da mesma forma que uma aplicação web, com a diferença de que o acesso aos dados será remoto (cliente-servidor).

Já há algum tempo eu pensava em um exemplo bem simples de aplicação em PHP-GTK, em que as pessoas pudessem copiar e colar e já sair utilizando conjuntamente com um banco de dados, mas, para isso, o banco de dados teria de ser de fácil instalação.

Foi aí que me veio a idéia de desenvolver uma Agenda, afinal, todos nós precisamos controlar nossos compromissos. E resolvi utilizar o SQLite, um banco de dados em sistema de arquivos que poderia ser compactado junto com a aplicação, dispensando a instalação de um servidor de banco de dados, tornando sua instalação infinitamente simples.

1. SQLite

O SQLite é um banco de dados relacional cuja estrutura (tabelas, índices, dados) está contida em um único arquivo no sistema. O acesso aos dados é implementado por uma biblioteca de funções escritas em C por Richard Hipp e a manipulação dos dados é realizada por meio da linguagem SQL.

Você deve estar acostumado com bancos de dados relacionais cuja estrutura cliente-servidor exige a instalação do servidor de banco de dados, que irá se comunicar com a aplicação, geralmente através de uma porta específica, via protocolo TCP/IP. Como o SQLite não tem nada disto, ele pode ser compactado juntamente com a aplicação. Imagine um arquivo chamado "meusistema.db" contendo todas as tabelas do seu sistema! Você só terá de compactar este arquivo junto com sua aplicação pa-

ra distribuir seu programa! Lembra do formato .DBF, comum entre as aplicações clipper, ou os arquivos .MDB do MS Access ? Pois é, o SQLite tem o mesmo foco: proporcionar uma estrutura de banco de dados simples em arquivo para ser distribuído juntamente com aplicações *standalone*, mas é muito melhor que seus antecessores, tendo em vista que implementa o padrão SQL92, permite transações, *triggers* e permite bancos de dados de até 2 terabytes de tamanho (2 na 41), ou seja, é muito byte. O banco de dados pode servir várias requisições de leitura (SELECT) ao mesmo tempo, entretanto, um lock do arquivo é realizado em operações de escrita (INSERT, UPDATE, DELETE).

A biblioteca de acesso ao SQLite é parte integrante do PHP5. Isto quer dizer, que as funções de criação e acesso ao banco de dados são nativas da linguagem, o que torna seu uso extremamente simples.

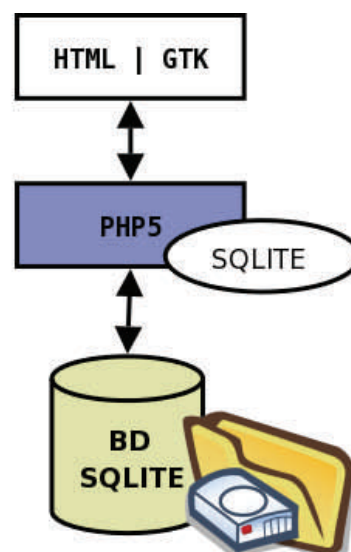


Figura 1 – Estrutura do SQLite

2. Classe Agenda

Para começar nosso programa, criaremos uma classe chamada Agenda. Iremos criá-la utilizando o mecanismo de herança. Agenda será classe filha de GtkWindow, logo, ela também será uma janela e irá possuir todos os métodos que esta possui. No método construtor, além de instanciar e centralizar a janela, iremos criar um componente calendário e conectar dois de seus sinais: *day-selected*, que é disparado sempre que é selecionada uma data, e *month-changed*, que é disparado sempre que o usuário visualiza outro mês. Iremos criar campos para digitação de data, hora, título e descrição da tarefa. Também iremos criar dois botões de ação: um para salvar o compromisso digitado e outro para excluir um compromisso da listagem de compromissos. Esta listagem será criada pelo método `createTaskList()`, que irá criar toda estrutura do GtkTreeView para visualizar as tarefas.

```
<?php
/*
 * classe Agenda
 * Pablo Dall'Oglio
 */
class Agenda extends GtkWindow
{
private $calendar;
/*
 * método construtor
 * instancia a janela e posiciona elementos
 */
public function __construct()
{
    // instancia janela
    parent::__construct();
    // define posicionamento
    parent::set_default_size(700,500);
    parent::set_position(GTK::WIN_POS_CENTER)
    // quando for fechada, aborda app
    parent::connect_simple('destroy', array
(Gtk, 'main_quit'));

    // cria Fixed, para posicionar widgets
    $fixed = new GtkFixed;
    // adiciona GtkFixed dentro da janela
    parent::add($fixed);

    // cria objeto calendário
    $this->calendar = new GtkCalendar;
    // conecta sinais de mudança de dia e mês
    $this->calendar->connect_simple('day-
selected', array($this, 'onDaySelected'));

    $this->calendar->connect_simple('month-
changed', array($this, 'onMonthChanged'));
    $fixed->put($this->calendar, 10,10);

    // cria campos de digitação para
    // data, hora, título e descrição
    $this->data = new GtkEntry;
    $this->hora = new GtkEntry;
    $this->título = new GtkEntry;
    $this->descr = new GtkTextView;

    // cria janela de rolagem para descrição
    $scroll = new GtkScrolledWindow;
```

```
$scroll->set_size_request(240,100);
$scroll->add($this->descr);

// cria e posiciona rótulos de texto
$fixed->put(new GtkLabel('Data'),10,200);
$fixed->put(new GtkLabel('Hora'),140...);
$fixed->put(new GtkLabel('Título'),...);
$fixed->put(new GtkLabel('Descrição')..);

// posiciona campos de digitação
$fixed->put($this->data, 10, 220);
$fixed->put($this->hora, 140, 220);
$fixed->put($this->título,10, 270);
$fixed->put($scroll, 10, 320);

// cria botão de salvar
$botao = GtkButton::new_from_stock
(GTK::STOCK_SAVE);
// conecta o botão ao método onSave
$botao->connect_simple('clicked', array
($this, 'onSave'));
$fixed->put($botao, 160, 440);

// cria botão de deletar
$botao = GtkButton::new_from_stock
(GTK::STOCK_DELETE);
// conecta o botão ao método onDelete
$botao->connect_simple('clicked', array
($this, 'onDelete'));
$fixed->put($botao, 560, 440);

// cria listagem de tarefas
$this->createTaskList();

// coloca listagem dentro de um scroll
$scroll = new GtkScrolledWindow;
$scroll->add($this->list);
$fixed->put($scroll, 280, 10);

// cria banco de dados
$this->createDatabase();

// atualiza lista de tarefas
$this->onMonthChanged();
}

Agora vamos ao método createDatabase(). Este método é executado toda vez que a aplicação inicia e é responsável por conectar ao banco de dados. Caso o banco de dados não exista, ele também irá criá-lo.
/*
 * método createDatabase
 * cria banco de dados, se ele não existir
 */
function createDatabase()
{
    // verifica se existe o banco
    if (!file_exists('agenda.db'))
    {
        // conecta ao banco
        // (cria se não existir)
        $this->conn = new PDO
('sqlite:agenda.db');
        // cria a tabela
        $this->conn->Query('create table agen-
da (data, hora, título, descricao)');
    }
    else
    {
        // conecta ao banco
        $this->conn = new PDO
```

```
( 'sqlite:agenda.db' );
}
```

```
}
```

O próximo método que iremos criar é o `createTaskList()`. Este método irá criar a listagem de tarefas que fica ao lado do calendário.

```
/*
 * método createTaskList
 * cria a listagem de tarefas
 */
function createTaskList()
{
    // cria objeto treeview
    $this->list = new GtkTreeView();
    // cria modelo de dados com 4 posições
    $this->model = new GtkListStore(...);
    $this->list->set_model($this->model);

    // cria 4 colunas
    $column1 = new GtkTreeViewColumn('Data');
    $column2 = new GtkTreeViewColumn('Hora');
    ...

    // cria 4 renderizadores de texto
    $render1 = new GtkCellRendererText();
    $render2 = new GtkCellRendererText();
    ...

    // empacota os renderizadores
    $column1->pack_start($render1, true);
    $column2->pack_start($render2, true);
    ...

    // adiciona as colunas na treeview
    $this->list->append_column($column1);
    $this->list->append_column($column2);
    ...

    // define dimensões da treeview
    $this->list->set_size_request(400,400);
}
```

Após o método `createTaskList()`, temos o método `onDaySelected()`. Ele é executado sempre que o usuário clica em alguma data do calendário. Seu objetivo é preparar os campos para digitação de um novo compromisso. Ele limpa todos os campos de digitação, preenchendo o campo “data” justamente com a data selecionada.

```
/*
 * método onDaySelected
 * executado sempre que o usuário
 * seleciona outra data
 */
function onDaySelected()
{
    // obtém a data selecionada
    $date = $this->calendar->get_date();
    $ano = $date[0];
    $mes = $date[1] + 1;
    $dia = $date[2];

    // limpa os campos de digitação
    $this->data->set_text("$dia/$mes/$ano");
    $this->hora->set_text('');
    $this->titulo->set_text('');
    $buffer = $this->descr->get_buffer();
    $buffer->delete($buffer->get_start_iter(),
```

```
$buffer->get_end_iter());
}
```

Nosso próximo método é o `onMonthChanged()`. Ele será executado sempre que o usuário navegar no calendário, alterando seu mês. Seu objetivo é recarregar todos os compromissos do mês corrente. Desta forma, este método será executado também em outras ocasiões, quando, por exemplo, for inserido um novo registro ou for excluído um compromisso.

```
/*
 * método onMonthChanged
 * executado sempre que o usuário
 * altera o mês do calendário
 */
function onMonthChanged()
{
    // limpa a lista de tarefas
    $this->model->clear();
    // obtém o mês exibido atualmente
    $date = $this->calendar->get_date();
    $mes = $date[1] + 1;

    // carrega todos os compromissos do mês
    // que for selecionado
    $result = $this->conn->Query("select *
from agenda where strftime('%m', data)
='{$mes}' order by data, hora");
    // percorre os compromissos, adicionando
    no modelo de dados da lista
    foreach ($result as $row)
    {
        $dados = array($row['data'],
            $row['hora'],
            $row['titulo'],
            $row['descricao']);
        $this->model->append($dados);
    }
}
```

O próximo método é o `onSave()`. Este método é executado sempre que o usuário clicar no botão de salvar e será responsável por coletar os dados digitados a respeito do compromisso e por inseri-los no banco de dados, recarregando a listagem de tarefas logo após.

```
/*
 * método onSave
 * Salva um compromisso
 */
function onSave()
{
    // obtém os dados digitados
    $data = $this->data->get_text();
    $hora = $this->hora->get_text();
    $titulo = $this->titulo->get_text();
    $buffer = $this->descr->get_buffer();
    $descricao = ...

    // insere no banco de dados
    $this->conn->Query("INSERT INTO agenda VA-
LUES ('$data', '$hora', '$titulo', '$descri-
cao')");

    // recarrega as tarefas do mês
    $this->onMonthChanged();
}
```

Quase terminando nossa Agenda, temos o mé-

todo onDelete(). Seu objetivo é obter o item selecionado na listagem de compromissos e perguntar ao usuário se ele deseja excluí-lo. Caso afirmativo, o programa irá excluir o item do banco de dados, baseado na data e hora do compromisso, recarregando a listagem de compromissos logo após.

```
/*
 * método onDelete
 * Exclui um compromisso
 */
function onDelete()
{
    // obtém o registro selecionado
    $selection =
        $this->list->get_selection();
    list ($model, $iter) =
        $selection->get_selected();
    // sel há seleção
    if ($iter)
    {
        // obtém a data e hora
        $data = $model->get_value($iter, 0);
        $hora = $model->get_value($iter, 1);

        // pergunta ao usuário
        $dialog = new GtkMessageDialog(...
            'Deseja excluir a tarefa ?');
        $resposta = $dialog->run();

        if ($resposta == Gtk::RESPONSE_YES)
        {
            // exclui registro
            $this->conn->Query("DELETE FROM
agenda WHERE data='$data' and hora='$hora'");
        }
        // fecha diálogo
        $dialog->destroy();
    }
    // recarrega as tarefas do mês
    $this->onMonthChanged();
}
```

Por último, temos o método que é executado quando a janela da aplicação é fechada, ou seja, destruída. Neste caso, estamos fechando a conexão PDO com o banco SQLite simplesmente destruindo o objeto de conexão.

```
/*
 * método __destruct
 * Executado quando janela for destruída
 */
function __destruct()
{
    // fecha conexão PDO
    unset($this->conn);
}
}
```

Agora que terminamos de escrever nossa classe, veja a seguir como ficou nosso programa. Nele, precisamos somente instanciar um objeto da classe Agenda. Como esta classe é filha de GtkWindow, utilizamos o método show_all() para exibi-la na tela.

```
$app = new Agenda;
$app->show_all();
Gtk::Main()
?>
```

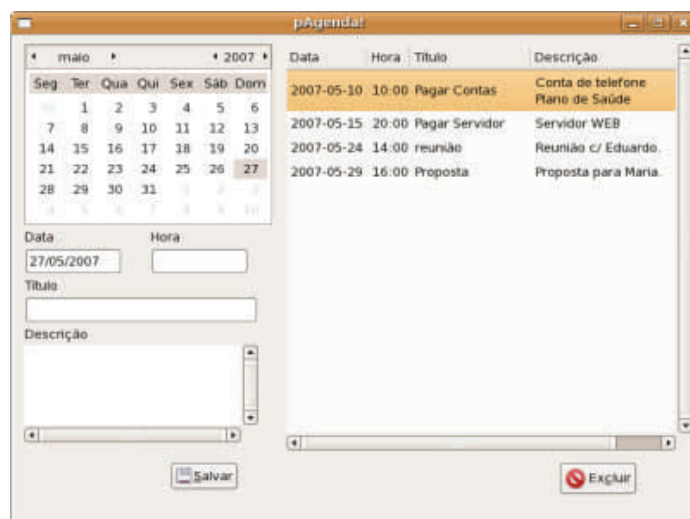


Figura 2 – Tela da Agenda criada

Considerações finais

Neste artigo escrevemos uma simples aplicação para controle de compromissos em PHP-GTK com menos de 200 linhas de código, descartando os comentários. Como o programa ficou um pouco grande para o formato da revista, optamos por cortar alguns pedaços, substituindo-os por “...” por motivos didáticos. Você pode fazer download da aplicação completa no site da Agenda, em <http://pagenda.php-gtk.com.br>

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 da Agenda] – <http://pagenda.php-gtk.com.br>

Pablo Dall'Oglio - pablo@dalloaglio.net

Pablo Dall'Oglio é formado em Análise de Sistemas pela UNISINOS. Autor do livro sobre PHP-GTK pela novatec editora, programa em PHP-GTK desde sua criação em 2001. É membro do time de documentação e criador da comunidade brasileira de PHP-GTK (www.php-gtk.com.br). Atualmente, é diretor de tecnologia e proprietário da Adianti Solutions (www.adianti.com.br), onde atua como consultor de tecnologia e engenheiro de software. Pode ser contatado pelo e-mail pablo@php.net. Pablo Dall'Oglio é formado em Análise de Sistemas pela UNISINOS. Autor do livro sobre PHP-GTK pela novatec editora, programa em PHP-GTK desde sua criação em 2001. É membro do time de documentação e criador da comunidade brasileira de PHP-GTK (www.php-gtk.com.br). Atualmente, é diretor de tecnologia e proprietário da Adianti Solutions (www.adianti.com.br), onde atua como consultor de tecnologia e engenheiro de software. Pode ser contatado pelo e-mail pablo@php.net.

Tratamento de Vetores e Matrizes em PHP – Parte II

As formas mais utilizadas para a criação e manipulação de vetores e matrizes já foram explicadas na segunda edição, portanto, o objetivo deste artigo é demonstrar ao leitor as funções específicas para ordenação de matrizes e vetores em PHP.

Grande parte dos vetores ou matrizes devem ser mesclados, divididos, tratados ou ordenados, para permitir que seus dados sejam trabalhados de forma mais adequada pelo sistema. Nas seções seguintes serão demonstradas algumas das funções da distribuição nativa do PHP específicas para estas tarefas.

1. Mesclagem, divisão e fatiamento de vetores

É possível juntar vetores separados em um único vetor ou ainda dividir um vetor em vários vetores diferentes. No *script* a seguir, foi utilizada a função **array_merge()** para mesclar os dois vetores **\$vetor01** e **\$vetor02** em um único vetor **\$frutas**. Esta função pode ser utilizada para mesclar vários vetores ao mesmo tempo. Para isso, basta acrescentar os vetores como argumentos na chamada da função. O vetor **\$frutas** foi, também, dividido em subvetores de três elementos, no vetor **\$chunk** através da função **array_chunk()**. O terceiro argumento desta função é opcional. Se este argumento for omitido ou **false**, serão criados novos índices numéricos no vetor de saída, ou seja, apenas os valores dos elementos serão passados. Caso seja **true**, os mesmos índices do vetor de dados serão mantidos, sendo passados para o vetor de saída o valor e o índice dos elementos. Por fim, uma fatia do vetor **\$frutas** foi copiada para o vetor **\$slice** através da função **array_slice()**.

Observe no *script* a seguir que a função foi chamada com a passagem de três argumentos. O primeiro argumento é o vetor de dados. O segundo argumento é um número correspondente ao índice do primeiro elemento a ser copiado do vetor de dados. Se o índice for positivo, o índice será contado a partir do primeiro elemento do vetor de dados, enquan-

to que se for negativo, o índice será contado a partir do final do vetor de dados. O terceiro argumento, que pode ser omitido, é o número de elementos que serão copiados para o vetor de saída. Se o número de elementos a serem copiados for positivo, então será copiado este número de elementos na direção do índice para o final do vetor de dados. Se for negativo, será copiado este número de elementos na direção do índice para o início do vetor de dados. Por fim, se este argumento for omitido, o vetor de saída conterá os elementos do índice até o final do vetor de dados.

```
<?
$vetor01 = array("maça", "laranja",
                "melancia", "pera");
$vetor02 = array("abacaxi", "pêssego",
                "limão");
$frutas = array_merge($vetor01,
                    $vetor02);
$chunk = array_chunk($frutas, 3, false);
$slice = array_slice($frutas, 2, 2);
?>
```

As saídas nas telas, mostradas neste artigo foram obtidas através das funções **show_vars()** e **print_a()** da biblioteca **debuglib**, constituindo importante recurso, tanto para programadores recém-iniciados, quanto para aqueles com muitos anos de experiência. A biblioteca não faz parte da distribuição padrão do PHP, no entanto, pode ser distribuída e utilizada gratuitamente. O *download* pode ser feito em <http://www.atomar.de/>.

DEBUG (runtime: 0.000321 sec)

global script variables	
vetor01	0 maçã
	1 laranja
	2 melancia
	3 pera
vetor02	0 abacaxi
	1 pêssego
	2 limão
frutas	0 maçã
	1 laranja
	2 melancia
	3 pera
	4 abacaxi
	5 pêssego
	6 limão
chunk	0 0 maçã
	0 1 laranja
	0 2 melancia
	1 0 pera
	1 1 abacaxi
	1 2 pêssego
slice	0 melancia
	1 pera

Figura 1 – Saída da tela para o script anterior

2. Inversão, reversão e embaralhamento

```
<?
$frutas = array("maçã", "laranja",
    "melancia", "pera", "abacaxi",
    "pêssego", "limão");
$reverse = array_reverse($frutas, false);
$flip = array_flip($frutas);
$shuffle = $frutas;
srand(1000000 * microtime());
shuffle($shuffle);
?>
```

O script acima apresenta a projeção na tela abaixo. A explicação será fornecida na sequência.

Inversão de um vetor significa tornar o primeiro elemento deste vetor o último elemento de um novo vetor, o segundo elemento, o penúltimo e assim por diante. No script seguinte, foi utilizada a função **array_reverse()** para inverter o vetor **\$frutas** em um outro vetor nomeado **\$reverse**. Além do vetor a ser invertido, é possível inverter também os índices do vetor de dados, passando-se um segundo argumento opcional com valor **true**.

A reversão de um vetor é a troca dos valores dos elementos pelos seus índices e vice-versa. A função **array_flip()** realiza esta tarefa, bastando apenas que seja passado o vetor a ser revertido, como pode ser visto no vetor **\$flip**.

O embaralhamento de vetores pode ser realiza-

do através da função **shuffle()**. Observe que esta função é baseada em um algoritmo pseudo-aleatório, significando que para que haja uma aleatoriedade confiável, deve-se “semear” uma semente para o algoritmo antes do seu uso. Isto é feito por meio da função **srand()** associada à função **microtime()**, que retorna a parte com o número de microssegundos da hora atual.

DEBUG (runtime: 0.008827 sec)

global script variables	
frutas	0 maçã
	1 laranja
	2 melancia
	3 pera
	4 abacaxi
	5 pêssego
	6 limão
reverse	0 limão
	1 pêssego
	2 abacaxi
	3 pera
	4 melancia
	5 laranja
	6 maçã
flip	maçã 0
	laranja 1
	melancia 2
	pera 3
	abacaxi 4
	pêssego 5
	limão 6
shuffle	0 limão
	1 pera
	2 abacaxi
	3 maçã
	4 pêssego
	5 melancia
	6 laranja

Figura 2 – Saída da tela para o script anterior

3. Ordenar os elementos do vetor

Existem várias funções específicas para ordenação de vetores. Nas seções seguintes serão demonstradas algumas delas. A função **sort()** ordena os elementos de um vetor em ordem crescente. Além do vetor a ser ordenado, um argumento opcional pode ser passado de forma a identificar como os elementos devem ser comparados entre si. Se for passada a constante **SORT_NUMERIC**, os elementos serão comparados numericamente. Se for passado **SORT_STRING**, será considerada a ordem alfabética. Por fim, se o argumento não for passado, será considerado o valor padrão **SORT_REGULAR**, que faz a detecção automática do tipo de comparação. A função **rsort()** ordena os elementos de um vetor em ordem decrescente e funciona de forma análoga à função **sort()**.

As funções **asort()** e **arsort()** trabalham de forma análoga às funções **sort()** e **rsort()**, respectivamente. A única diferença entre estes pares de função é que o par

asort() e **arsort()** mantêm os índices originais do vetor de dados, enquanto que as funções **sort()** e **rsort()** criam novos índices numéricos.

Para facilitar o entendimento do *script* seguinte, os vetores foram criados com o mesmo nome da função que os chama. Observe também que estas funções são funções do tipo **bool** ou do tipo **void**, retornando **true**, **false** ou não retornando valor algum. Portanto, todas as alterações são realizadas no próprio vetor. Por motivos didáticos, o vetor é clonado antes de ser passado, para evitar que seus dados sejam modificados.

```
<?
$frutas = array("maça", "laranja",
               "melancia", "pera", "abacaxi",
               "pêssego", "limão");
$sort = $frutas;
sort($sort, SORT_STRING);
$rsort = $frutas;
rsort($rsort, SORT_STRING);
$asort = $frutas;
asort($asort, SORT_STRING);
$arsort = $frutas;
arsort($arsort, SORT_STRING);
?>
```

O *script* anterior apresenta a projeção na tela abaixo.

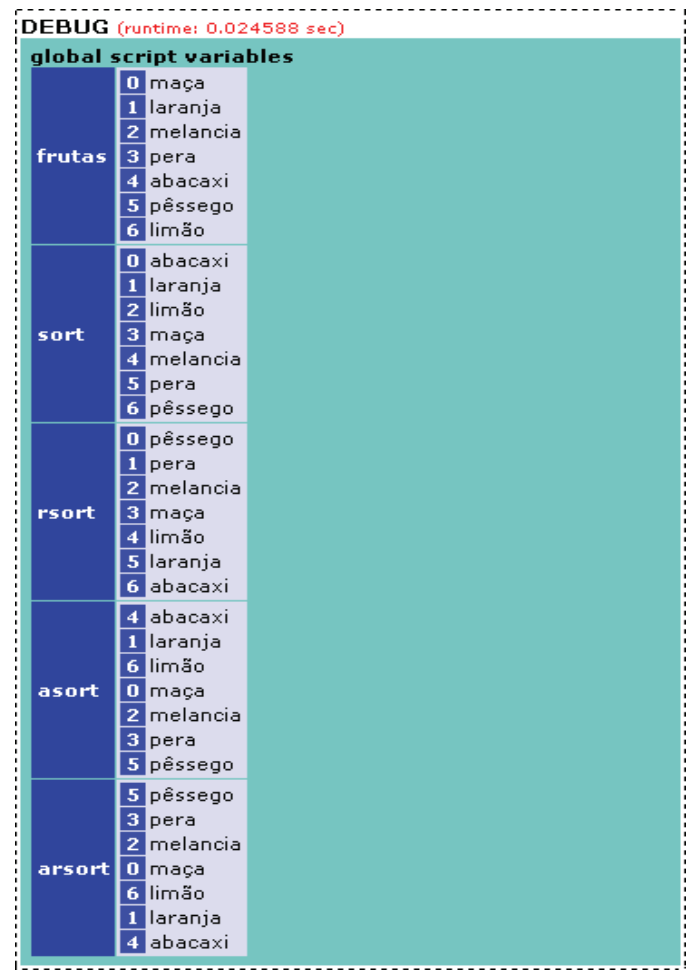


Figura 3 – Saída da tela para o *scrip* anterior

3. Ordenar os índices do vetor

As funções mostradas na sessão anterior ordenam o vetor pelo valor de seus elementos, mas também é possível ordenar pelo valor de seus índices, por meio do par de funções **ksort()** e **krsort()**. Estas funções são análogas às funções **sort()** e **rsort()**, respectivamente, exceto por ordenarem o vetor baseadas nos valores dos índices. Um exemplo de utilização destas funções está demonstrado no *script* a seguir.

```
<?
$frutas = array_flip(array("maça",
                          "laranja", "melancia", "pera",
                          "abacaxi", "pêssego", "limão"));
$ksort = $frutas;
ksort($ksort, SORT_STRING);
$krsort = $frutas;
krsort($krsort, SORT_STRING);
?>
```

Segue abaixo, a saída da tela obtida com o *script* anterior.

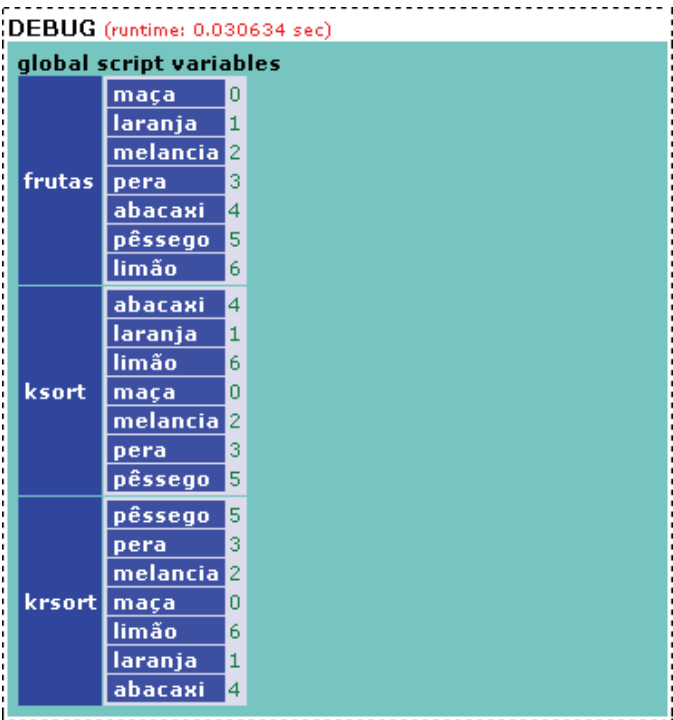


Figura 4 – Saída da tela para o *script* anterior

4. Ordenar nomes de arquivos

Se um vetor que guarda nomes de arquivos for ordenado pelas funções anteriores, erros podem ocorrer, pois a comparação alfabética de valores com letras e números nem sempre é facilmente identificada pelo computador. Por exemplo, se os valores **imagem1**, **imagem2** e **imagem10** forem ordenados alfabeticamente, a ordenação resultaria **imagem1**, **imagem10** e **imagem2**, nesta ordem. Considere o

script a seguir.

```
<?
$fotos[] = "casa3.jpg";
$fotos[] = "casa1.jpg";
$fotos[] = "apto2.jpg";
$fotos[] = "apto20.jpg";
$fotos[] = "apto10.gif";
$fotos[] = "Apto14.jpg";
$fotos[] = "apto13.jpg";
$fotos[] = "apto13.png";

$natsort = $fotos;
natsort($natsort);
$natcasesort = $fotos;
natcasesort($natcasesort);
?>
```

Para solucionar este problema, foram utilizadas funções de ordenação baseadas num algoritmo de **ordem natural**. Este algoritmo visa simular a forma de ordenação do cérebro humano, quando há caracteres alfabéticos e numéricos misturados. A função **natsort()** realiza a ordenação alfabética em ordem crescente baseada em um destes algoritmos de ordem naturais. A função **natcasesort()** funciona de forma similar à função **natsort()**, porém, ela não faz distinção entre caracteres maiúsculos e minúsculos. A projeção na tela do *script* anterior segue na figura 5.

Observe que para a função **natcase()** considera-se que caracteres maiúsculos precedem caracteres os minúsculos. Isto se deve ao fato de os caracteres maiúsculos realmente precederem os caracteres minúsculos na tabela ASCII. Já a função **natcasesort** **realmente** não faz distinção entre a caixa do caractere, significando que, por exemplo, **Apto14.jpg** e **apto14.jpg** seriam considerados idênticos, sendo, portanto mantida a ordem que foi colocada no vetor.

DEBUG (runtime: 0.057833 sec)

global script variables	
fotos	0 casa3.jpg
	1 casa1.jpg
	2 apto2.jpg
	3 apto20.jpg
	4 apto10.gif
	5 Apto14.jpg
	6 apto13.jpg
	7 apto13.png
natsort	5 Apto14.jpg
	2 apto2.jpg
	4 apto10.gif
	6 apto13.jpg
	7 apto13.png
	3 apto20.jpg
	1 casa1.jpg
	0 casa3.jpg
natcasesort	2 apto2.jpg
	4 apto10.gif
	6 apto13.jpg
	7 apto13.png
	5 Apto14.jpg
	3 apto20.jpg
	1 casa1.jpg
	0 casa3.jpg

Figura 5 – Saída da tela para o *script* anterior

Considerações finais

Este artigo apontou várias funções que podem ser utilizadas no tratamento de vetores. Espera-se que este documento tenha elucidado o funcionamento das funções mais utilizadas para manipulação de vetores e matrizes. No entanto, a seção do manual do PHP que trata de vetores é muito vasta, o que torna inviável a menção a todas elas. Se ainda assim, o leitor não encontrar no manual do PHP uma função que realize a tarefa desejada, sugere-se que um algoritmo seja criado com base nas funções aqui demonstradas.

Referências e links sugeridos

[PHP Magazine] – <http://www.phpmagazine.com.br>
[Manual PHP] – <http://www.php.net>
[PHP Brasil] – <http://www.phpbrasil.com>

Leandro Schwarz - leandroschwarz@gmail.com

Engenheiro eletricitista pela Universidade Federal de Santa Catarina (UFSC). Atuando desde 2000 com desenvolvimento WEB, possui sólidos conhecimentos em PHP e MySQL.

Atualmente é mestrando em Engenharia Elétrica no Instituto de Engenharia Biomédica da UFSC e projeta websites e lojas virtuais como autônomo. Engenheiro eletricitista pela Universidade Federal de Santa Catarina (UFSC). Atuando desde 2000 com desenvolvimento WEB, possui sólidos conhecimentos em PHP e MySQL.

Atualmente é mestrando em Engenharia Elétrica no Instituto de Engenharia Biomédica da UFSC e projeta websites e lojas virtuais como autônomo.

Escrevendo código PHP de fácil manutenção

Este artigo, voltado para desenvolvedores PHP iniciantes, mostrará algumas técnicas que permitem a codificação de scripts PHP de fácil manutenção.

Introdução

Uma das maiores críticas ao PHP é que a linguagem facilita os desenvolvedores a programarem códigos sujos e de difícil manutenção.

Normalmente, quem faz estas críticas conheceu PHP a alguns anos atrás, numa época em que os desenvolvedores não se preocupavam com a qualidade do código e, tendo em vista a facilidade que a linguagem nos permite desenvolver, os desenvolvedores acabaram por criar códigos do tipo 'macarronada', onde se misturavam HTML, PHP, Conexão com bancos de dados e etc. Tudo em um único arquivo.

Realmente este tipo de código existiu e em alguns locais ainda existem, porém, algumas técnicas simples nos permitem criar um código limpo e de fácil manutenção.

O que é código de fácil manutenção?

O código é de fácil extensão a ponto de novas funcionalidades serem adicionadas a ele?

Depois de alguns meses, você consegue entender o código facilmente a ponto de adicionar funcionalidades nele?

Outro desenvolvedor pode entender o código facilmente a ponto de adicionar funcionalidades nele?

Se a resposta for sim para todas as perguntas acima, seu código é de fácil manutenção e provavelmente este artigo não lhe será muito útil. Caso a resposta tenha sido não, sugiro que continue lendo este artigo, pois alguma coisa poderá lhe servir.

Quem vai trabalhar com esse código?

Neste artigo vamos pressupor que o código será manipulado por mais de um desenvolvedor. Mesmo que seja manipulado por apenas um desenvolvedor, é interessante utilizar as técnicas e ferramentas apre-

sentadas.

Codificação inline: um grande problema

O que é mais fácil de entender?

```
if ($a>50){ echo 'Código inválido'; } else  
{ echo 'Código válido'; }
```

ou

```
if ($a > 50)  
{  
    echo 'Código inválido';  
}  
else  
{  
    echo 'Código válido';  
}
```

Muitos desenvolvedores criam códigos ilegíveis por medo de ter um código com mais linhas. Ora, é melhor ter um código maior, porém mais fácil de ler, do que um código menor e mais difícil de ler.

Outra forma de codificação bem comum é:

```
function somaDoisNumeros($num1,$num2) { return  
$num1+$num2; }
```

Não seria mais legível se fosse assim?

```
function somaDoisNumeros($num1,$num2)  
{  
    return $num1+$num2;  
}
```

Reescrever funcionalidades já existentes

É muito comum encontrar neste tipo de código funções definidas pelo usuário que fazem a mesma coisa que uma função interna do PHP. Isto além de tornar o código mais difícil de manter, ainda faz com que ele perca em performance, porque ele vai executar um procedimento em um nível mais alto sendo que já existe esta funcionalidade pronta para ser executada em um nível mais baixo.

Ex:

```
function converter_para_inteiro($numero)
{
    return (int)$numero;
}
```

Evitar reescrever funções internas também ajuda na performance de sua aplicação, porque as funções internas são escritas em C, ou seja, já estão no core do PHP e não precisam passar pelo processo de compilação do PHP antes de serem executadas.

Uso indiscriminado de constantes

Em muitos códigos são usadas constantes indiscriminadamente, isso favorece a difícil manutenção do código, porque uma pessoa que nunca o viu terá que ficar pesquisando dentro do seu código pela definição das constantes para saber o que se passa.

Ex:

```
define("QUEBRA_LINHA_HTML", "<br>");
define("NEGRITO_INICIO", "<b>");
define("NEGRITO_FIM", "</b>");

echo "Você deve fazer o ".NEGRITO_INICIO.
"login".NEGRITO_FIM." no sistema antes de usa-
lo".QUEBRA_LINHA_HTML;
```

Este código poderia ser simplesmente assim:

```
echo "Você deve fazer <b>login</b> no sistema
antes de usa-lo.";
```

Estas constantes que eu coloquei ainda são bem intuitivas, eu já encontrei algo assim:

QBR_LN_HTML É difícil ou não?

Muitos desenvolvedores utilizam este mecanismo das constantes para não misturar HTML com PHP. Não seria mais fácil resolver usando templates?

Concatenando variáveis

Ao concatenar variáveis, procure sempre, fazer da seguinte forma:

```
$variavel = "Este artigo fala sobre código
de fácil manutenção com PHP";
$variavel .= " Desta forma é mais legível";
$variavel .= " Não acha?";
```

Não deixe a variável grudada com os operador de atribuição e concatenação.

Utilizando template engine

Para não misturar HTML com PHP você pode utilizar templates engine. O mais utilizado atualmente é Smarty. Utilizando uma template engine, você separa o código PHP do código HTML, tornando assim a integração entre os dois mais limpa e

permitindo que o web designer de sua equipe trabalhe mais confortavelmente, sem ter medo de alterar nenhum código PHP por engano. Lembre-se: Nem sempre o webdesigner programa em PHP.

Padrões de codificação

Tenha um padrão de codificação, pois assim sempre é mais fácil entender o código.

Você não precisa criar um padrão de codificação do zero. Você pode incorporar nos seus códigos o padrão de codificação de frameworks, que já estão consolidados como o padrão utilizado pelo PEAR ou o padrão utilizado pelo Zend Framework. Se você não quiser utilizá-los como o seu padrão oficial, você pode tê-los como exemplo e adaptá-los as suas necessidades.

É muito mais fácil ter um código onde todas as variáveis tenham o padrão 'camelCaps', onde variáveis iniciam com a primeira letra em minúscula e as demais palavras com maiúsculas, exemplo:

```
$colecacaoUsuarios, $objConexao
```

ao invés de usar assim:

```
$COLECAOUSUARIOS, $Colecacaousuarios.
```

Viu a diferença?

Comentários

Inicie todo o seu código com um bloco de comentário com informações sobre aquele respectivo arquivo, data de criação, o que falta fazer (TODO), bugs a serem corrigidos (FIXME), etc.

Ex:

```
/*
 * CLASSE DE CONEXÃO COM BANCO DE DADOS
 *
 * Esta classe implementa uma camada de conexão
ao banco de dados MySQL
 * utilizando PDO e etc, etc, etc,
 *
 * author: Adler Medrado
 * date: adler@neshertech.net
 *
 * TODO: Implementar método de controle de
transações
 * FIXME: método lastInsertID não retorna a in-
formação correta
 */
```

Não use comentários de uma linha que sejam óbvio, por exemplo:

```
// Este método grava usuário no banco
public function gravaUsuarioNoBanco()
{
    // Aqui ele executa o método inserir
    $this->inserir();
}
```


Você pode usar também o PHPDocumentor que, por meio dos blocos de comentários, gera uma documentação parecida com o Javadoc. Ele também é bem interessante.

Controle de versão

Para uma aplicação que é modificada constantemente, é interessante utilizar softwares de controle de versão como o CVS, CVSNT ou SVN, mesmo que você seja o único desenvolvedor da aplicação.

Ferramentas de trac são bem-vindas mesmo que seja para mostrar as mensagens de commit detalhadamente.

Abstração de banco de dados

Escrever suas classes de abstração é uma boa, pois permitirá que qualquer mudança em relação a banco de dados na sua aplicação, seja feita apenas nestas classes e não na aplicação como um todo.

Usar PDO é uma boa idéia, pois esta extensão está se tornando um padrão para o desenvolvimento PHP, mas existem outras classes de abstração prontas, como a ADOdb por exemplo.

Utilizando frameworks

Você pode utilizar um framework porque eles já implementam os padrões descritos anteriormente e já possuem abstração para banco de dados, por exemplo, além de implementar a maioria dos padrões de projetos (design patterns). Outro ponto em utilizar frameworks já existentes é que muitos possuem as camadas bem definidas, e

pelo fato de serem código aberto (open-source), a correção de bugs é frequente.

Existem ótimos frameworks em PHP - eu diria até que existem frameworks demais - como o Cake, o Symfony e o Zend Framework.

Usando o framework, qualquer pessoa que for dar manutenção no código terá de conhecer os padrões do framework. Sendo assim, uma boa parte na hora de entender o código por parte do novo desenvolvedor será poupada.

Conclusão

Este artigo visou demonstrar bem simples e diretamente algumas técnicas para que seu código seja de fácil manutenção. Existem outras técnicas que infelizmente não foram incluídas neste, porém, partindo destas dicas seu código será mais legível, mais limpo e até mesmo mais ágil.

Qualquer dúvida, por favor entrem em contato comigo por e-mail.

Referências e links sugeridos

[Subversion] - <http://subversion.tigris.org/>

[CVSNT] - www.cvsnt.org

[Symfony] - www.symfony-project.com/

[Cake] - <http://www.cakephp.org>

[Zend Framework] - <http://framework.zend.com>

[Smarty] - <http://smarty.php.net>

[PDO] - <http://www.php.net/pdo>

[Site Pessoal] - www.neshertech.net/adler

[Nesher Technologies] - <http://www.neshertech.net>

[PHPDF] - <http://www.phppdf.org.br>

Adler Brediks Medrado - adler@neshertech.net

Analista e desenvolvedor, utiliza PHP na maioria dos seus projetos.

Atualmente presta serviços em um órgão federal em Brasília e é instrutor de cursos PHP na X25 Treinamento e Consultoria. Além disso, também é proprietário da empresa Nesher Technologies que presta serviços de consultoria para empresas do Brasil e do exterior.

Membro da equipe de coordenação do Grupo de usuários PHP do Distrito Federal (PHPDF).



Amplie os horizontes da sua empresa

Divulgue sua empresa e produtos para a comunidade PHP do Brasil.
Conheça os planos para anunciar no portal e na revista.
Solicite um orçamento através de comercial@phpmagazine.com.br.



Ambientes de livre distribuição para gerenciamento de conteúdo de aprendizagem

Neste artigo será feita uma breve introdução aos Ambientes Virtuais de Gerenciamento de Conteúdo de Aprendizagem, dando destaque para os de distribuição livre, bem como suas características básicas, uma pequena análise e as expectativas para o futuro desses ambientes em relação à educação a distância.

Até pouco tempo estudar era visto meramente como um simples ato de ir a uma instituição de ensino, assistir as aulas e tentar entender os assuntos lecionados. Hoje, em muitos lugares, essa visão continua a mesma, porém, já existe uma vertente em prol de uma nova visão: a de que o professor é apenas o mediador no processo de aprendizagem.

Mediador, nesse contexto é aquele que não é o dono da verdade, mas que tenta traçar um caminho para que seus alunos possam descobrir suas próprias verdades por meio de bases sólidas de experiências e vivência auxiliadas por ele, e a partir daí construir seu próprio conhecimento.

O importante é fazer com que o aluno evolua em seu próprio ritmo sem, no entanto, desprendê-lo de um determinado grupo. Para que isso aconteça são necessárias ferramentas que mantenham a atenção desse indivíduo de tal forma que ele seja induzido a fazer atividades necessárias para seu aprendizado de maneira espontânea e até, por que não, prazerosa.

A esse tipo de pensamento denomina-se construtivismo, onde o estudante constrói seu conhecimento através de pequenas atividades individuais ou coletivas [1]. Teorias como o construtivismo, behaviorismo e o cognitivismo tentam descrever como o processo de aprendizagem acontece.

Hoje, existem vários ambientes instrucionais que acolhem a essas teorias pedagógicas, dando enfoque mais para uma do que para a outra. No entanto, é necessário saber qual utilizar em cada situação.

Na modalidade de ensino a distância existem os Ambientes Virtuais de Aprendizagem, também co-

nhecidos por várias outras denominações, dentre elas a LCMS (*Learning Content Management System*), que será abordado neste texto. No geral, porém esses ambientes cumprem com o mesmo objetivo: disponibilizar conteúdos de aprendizagem, e em alguns casos gerenciá-los, sendo que geralmente utilizam uma ou várias dessas teorias pedagógicas em suas estruturas.

Nos próximos tópicos serão descritos de forma sucinta esses ambientes, suas características básicas, uma pequena análise e as expectativas referentes às possíveis melhoras em seu desenvolvimento.

O que são LCMS

Antes de entrar no conceito de LCMS (*Learning Content Management System*), é interessante entender o significado de três outras siglas: CMS, LMS e RLO.

O CMS (*Content Management System*) ou Sistema de Gestão de Conteúdo é um software que se destina à agregação de vários conteúdos de forma a gerenciá-los. Os *blogs* são exemplos de CMS, ou seja, você envia um texto e o software mantém a organização através da criação de categorias (dias, mês, ano, outros), mantendo para isso mecanismos de publicação em que o usuário só tem que se preocupar com o seu texto.

Um CMS, contudo, possui uma utilização muito mais ampla que apenas o gerenciamento de conteúdos de *blogs*, possui várias funcionalidades como, por exemplo, a de administração de artigos, mídias, sites, relatórios, lojas e cursos. O seu principal foco é possibilitar a agilização do processo de publi-

cação, por meio de ferramentas WYSIWYG (*What You See Is What You Get*), ou seja, o que você vê é o que você tem, e claro o gerenciamento de tais conteúdos [2]. Exemplos de CMS muito utilizados são o Joomla, Drupal, Mambo, Wordpress e PHPNuke. No site cmsmatrix.org podem ser feitas algumas comparações entre esses sistemas.

O LMS (*Learning Management System*) ou Sistema de Gestão de Aprendizagem é um CMS, porém, seu conteúdo é voltado para aprendizagem. Ele é responsável pelo gerenciamento de cursos, que por sua vez são criados por outras ferramentas, uma vez que, este não possui capacidade de geração de conteúdo educacional. Por essa razão, não há como personalizar o ambiente. Após o curso ser criado, não é possível a reutilização de partes, somente do todo, ou seja, os usuários sempre têm o mesmo curso disponível.

Os RLO (*Reusable Learning Objects*) ou Objetos de Aprendizagem Reutilizáveis possibilitam uma melhor padronização no desenvolvimento de ambientes virtuais de ensino, garantindo assim uma gestão mais completa de seus conteúdos. Um padrão muito conhecido que utiliza essa tecnologia é o SCORM (*Sharable Courseware Object Reference Model*) ou Modelo de Referência para Objetos de Conteúdos Compartilháveis.

Os RLO possibilitam desmembrar um determinado conteúdo em partes menores, como módulos, que podem ser reutilizados em diferentes ambientes. Esses pequenos módulos possuem a menor informação possível integrando objetivo, atividade e avaliação da aprendizagem [3].

Além de possibilitarem a reutilização de partes do conteúdo em outros cursos ou ambientes, os RLO também possibilitam uma melhor personalização do ambiente, uma vez que o usuário, mais especificamente o aluno poderá selecionar quais módulos pretende fazer e quais atividades. Em contra partida, os instrutores e administradores poderão disponibilizar aos poucos seus materiais e ferramentas de comunicação, bem como melhorá-los através do acúmulo de conhecimento adquirido com a sua utilização.

Por fim, o Sistema Gestor de Conteúdo de Aprendizagem (LCMS), é o ambiente que agrega todos os conceitos vistos até aqui: CMS + LMS + RLO. Um LCMS tem como objetivo central gerenciar, elaborar e postar conteúdos instrucionais ou RLO [2]. Alguns autores também o denominam de Ambiente Virtual de Aprendizagem, ou VLE (*Virtual Learning Environment*) [4].

Requisitos para um Sistema de Gerenciamento de Aprendizagem

Um Sistema Gestor de Conteúdo de Aprendizagem deve ser capaz de no mínimo prover ferramentas de comunicação síncronas e assíncronas.

As ferramentas de comunicação síncronas são aquelas que tentam simular de forma on-line um ambiente de ensino presencial, através de mensagens instantâneas, *chat*, áudio e vídeo, por exemplo. Podemos considerá-las como sendo em tempo real. Enquanto que as ferramentas assíncronas são aquelas que não precisam de comunicação em tempo real como, por exemplo: e-mail, fórum, glossário, etc. [5].

É conveniente que um LCMS se apoie em alguma teoria pedagógica como o construtivismo, behaviorismo e o cognitivismo, ou em várias, de modo que possa ser direcionado para determinado grupo, o qual possa fazer melhor uso do sistema através do conhecimento prévio dos objetivos a serem alcançados.

Outras características interessantes de se fazerem presente nesse tipo de ambiente estão listadas a seguir:

- Disponibilidade, possibilidade de acesso em qualquer lugar e a qualquer hora;
- Versatilidade, articulação de conteúdos multimídia no processo de ensino-aprendizagem;
- Espaço para troca de idéias, *feedback* e pesquisas (fóruns, enquetes, diálogos, *chat*);
- Ferramentas de trabalho colaborativo (wikis, glossários);
- Ferramentas de avaliação (testes do tipo *quiz*, questionários, provas);
- Ferramentas de gestão (entrega de trabalhos, notas, grupos, agendas);
- Possibilidade de apresentação dos conteúdos em vários idiomas;
- Acessibilidade, possibilitando que pessoas com deficiências possam ter acesso ao material.
- Portabilidade, permitindo que o LCMS possa ser instalado em Sistemas Operacionais diferentes e que possa ser acessado em diferentes tipos de computadores;
- Distribuição livre, inclusive com o código fonte, garantindo maior conhecimento da ferramenta e conseqüente desenvolvimento e melhoramento de funções.

Atualmente muitos LCMS de distribuição livre já possuem a maioria dessas características, possibilitando a um maior número de pessoas o acesso a essa tecnologia. Isso proporciona o aumento de comunidades de desenvolvimento e distribuição de material referente às mesmas, o que só vem a contri-

buir com a sua melhoria e divulgação.

Vantagem na utilização desses Sistemas

Os LCMS podem ser utilizados tanto para cursos presenciais quanto a distância. O seu principal objetivo é servir de apoio ao aprendizado e, portanto, integrasse muito bem em qualquer um desses modelos de ensino.

Analisando os principais atores do LCMS: administrador, tutor e aluno, podemos destacar as seguintes vantagens mais evidentes, respectivamente:

- Administrador – como o conteúdo é centralizado através da utilização de bancos de dados, o administrador possui maior domínio sobre o que está sendo feito em cada curso, podendo excluir, editar e adicionar ferramentas aos mesmos e gerenciar contas de usuários;
- Tutor – esse elemento é geralmente representado pelo professor, que através do domínio do ambiente gerencia os conteúdos de suas aulas, o andamento dos seus alunos e determina o ritmo do curso;
- Aluno – existem diversas ferramentas de apoio ao aluno, como fórum, *chat*, *e-mail* e outras que podem ser utilizadas para tirar dúvidas, enviar trabalhos, enviar sugestões, etc. O aluno também pode ter acesso, em alguns casos, ao seu desempenho através de notas e testes de avaliação no estilo *quiz*, com o resultado on-line.

O LCMS pode ser utilizado tanto como ferramenta de apoio, como ambiente de aula, ou seja, como um elo entre aluno e professor sem a necessidade da presença física.

A maioria dos LCMS já suporta padrões como o SCORM, o que garante que o curso seja reutilizado e reformulado várias vezes.

Análise de Sistemas de Gerenciamento de Aprendizagem

O objetivo dessa análise será selecionar LCMSs que tenham recursos como: ferramentas de comunicação, produtividade e administrativas, além de suporte a língua portuguesa e licença GNU. Não será feita uma exploração mais detalhada de cada sistema porque o foco principal é somente indicar quais possuem uma melhor adaptação ao público que busca por educação a distância de forma rápida e com bastante material para tal, sem, no entanto, excluir as demais.

Os recursos referentes às ferramentas de comunicação devem incluir mecanismos síncronos e assíncronos. As ferramentas de produtividade devem possibilitar melhor ajuda de contexto, consulta *off-*

line do material, calendário e busca, por exemplo. Já as administrativas englobam os aspectos de gerenciamento de cursos, material, alunos e das próprias ferramentas de comunicação.

A licença GNU se deve ao fato do código ser aberto, o que gera um maior número de pessoas empenhadas em seu projeto, maior número de comunidades e conseqüentemente de material e suporte.

A questão do suporte em língua portuguesa é outro ponto importante, pois permite um melhor uso das ferramentas tanto por administradores, que não terão que ficar traduzindo, quanto para professores e alunos que poderão com uma simples leitura entender qual a função de determinadas ferramentas.

Como o objetivo dessa análise é somente eliminar ferramentas que possuam poucos recursos, foi utilizado para um primeiro nivelamento um *site* especializado em fazer comparações entre diferentes Sistemas de Gestão de Conteúdo de Aprendizagem: o Edutools [6]. Outro *site* também muito interessante para fazer esse tipo de análise é o Open Source CMS [7].

A seleção foi feita da seguinte forma: primeiro foi verificado quais possuíam licença GNU, em segundo a variedade de recursos úteis, e por último a disponibilidade de material na Internet e suporte em língua portuguesa.

Na primeira seleção muitas ferramentas possuíam licença GNU, mas algumas não tinham recursos essenciais como *Chat* ou possuíam páginas pouco intuitivas e na sua grande maioria, não tinham comunidades em português difundidas.

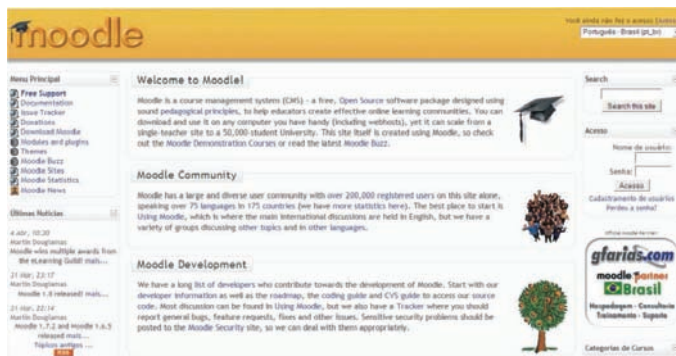
Restaram para uma segunda seleção quatro LCMS muito interessantes: Claroline, Sakai, ATutor e o Moodle. Todos eles com muitos recursos e interface clara e agradável. Porém, o Projeto Sakai ainda não é muito utilizado no Brasil e, devido a esse fato, pouco material sobre o mesmo foi encontrado em português, sendo excluído desta forma da próxima etapa da seleção. O Claroline também foi excluído devido à falta de segurança em torno de sua continuação, uma vez que já possui inclusive outro projeto baseado no mesmo código, o Dokeos [4].

Por fim, restaram apenas dois projetos: Atutor e o Moodle. Destes dois, somente o Moodle possui diversidade de material em português. Contudo, será feita uma pequena descrição do Atutor, por possuir características que o diferenciam, como o seu significativo foco em acessibilidade.

Moodle

O Moodle (acrônimo de *Modular Object-Oriented Dynamic Learning Environment*) é sem

dúvida o mais completo LCMS GNU atualmente. Isso se deve ao fato dele ser gratuito, ter tradução para o português, ampla documentação e diversas comunidades e, além de tudo, tem um forte apelo pedagógico baseado no construtivismo social.



Site do Moodle

Assim como a grande maioria dos LCMS, o Moodle é desenvolvido em PHP e suporta diversos bancos de dados, com destaque para o Mysql. Devido a essas características, pode ser instalado facilmente em servidores web com suporte a PHP e a base de dados, sendo leve e compatível com a maioria dos navegadores, inclusive os de tecnologia mais antiga. [8]

O Moodle está dividido em vários módulos e possui três níveis de visão: do administrador, do tutor e do aluno. O administrador é o responsável pela manutenção do site, através da inclusão de extensões e até a modificação do código PHP, se necessário. O tutor tem pleno poder de modificar o curso, inserindo ou excluindo módulos, bem como escolhendo seu tipo, que pode ser semanal, por tópico ou por discussão. O aluno além de ter a sua disposição ferramentas para sua auto-avaliação, ainda conta com outras para comunicação, como os *chats*, e-mail, fóruns, etc.

Para complementar, o Moodle é compatível com o SCORM e possui os seguintes módulos: Tarefa, Chat, Pesquisa de Opinião, Fórum, Questionário, Recursos, Pesquisa de Avaliação e Laboratório de Avaliação.

ATutor

O ATutor, assim como o Moodle, é uma ferramenta voltada para a gestão de cursos de aprendizagem, porém, seu principal foco está na acessibilidade e facilidade de uso de seus recursos. Ele possui licença GNU e é um ambiente totalmente desenvolvido em PHP, estando de acordo com o padrão SCORM e com as especificações de acessibilidade W3C WCAG 1.0.



Site do ATutor

A acessibilidade possibilita que pessoas com determinadas deficiências ou que possuam equipamentos mais antigos e conexão lenta possam ter acesso ao conteúdo do site, assegurando desta forma uma maior aceitação e um público mais diversificado. No ATutor podemos verificar várias características referentes a essa função. Dentre elas, temos: telas de atalho e de acesso, configuração de preferência de acessibilidade a priori (que oculta imagens e controles desnecessários), textos alternativos para imagens, rótulos em formulários, foco para campos de formulário, ampliação de imagens e possibilidade de alteração da interface através de folhas de estilo [9].

O padrão SCORM permite que o conteúdo dos cursos seja exportado ou importado de/para outras ferramentas com o mesmo padrão. Outra característica interessante é a possibilidade de exportar o conteúdo do ATutor em pacotes de conteúdo, podendo acessá-los *off-line*.

Expectativas para o futuro desses Sistemas

Atualmente, combinações entre diferentes tecnologias têm sido utilizadas para se chegar a uma comunicação mais efetiva nas aulas a distância. Um exemplo disso é o uso do LCMS em conjunto com o Skype. Sendo que o LCMS é utilizado efetivamente como o ambiente de estudo, onde se encontram todos os roteiros das aulas, avisos, materiais, *chats*, e-mail e fóruns, e o Skype servindo como um apoio à aula, propiciando comunicação através de áudio e vídeo. Grande maioria dos LCMS gratuitos não oferece esse recurso. Junto com o Skype, muitas vezes também utiliza-se um *plugin* chamado Festoon, que dentre outras funções possibilita compartilhar o *desktop* do computador, outra característica ausente em LCMS atualmente.

Além do Skype, muitos outros softwares são utilizados, como o Windows Live Messenger, Yahoo Messenger e o Talk. Porém, a necessidade de utilização está mais evidenciada quando a comuni-

cação visual for indispensável.

Outra tecnologia que já vem contribuindo com a educação a distância é a televisão. Aqui no Brasil, um bom exemplo é o Telecurso da Fundação Roberto Marinho, em que o aluno tem a possibilidade de assistir as aulas e praticar os exercícios em sua própria casa.

Com o surgimento da televisão digital, criou-se uma grande expectativa com relação a seu uso em EAD, pois a mesma possibilita a integração entre vários meios de comunicação, como internet, rádio e televisão. Tal integração possibilitará o desenvolvimento de programas de EAD que envolvam todos os benefícios já citados, como comunicação por vídeo e áudio, utilização de LCMS. Tudo ao mesmo tempo em apenas um equipamento (e acessórios necessários). Porém, o uso dessa tecnologia ainda vai tardar, pois o seu custo ainda é muito alto e sua implantação ainda está em andamento no Brasil.

Voltando para a realidade atual, mais especificamente ao que se possui de mais palpável, em se tratando de EAD, a Internet oferece, como foi analisado anteriormente, meios de comunicação entre grupos de pessoas que estejam participando de cursos a distância. Porém, muito ainda tem que ser melhorado, principalmente quando o tema em questão é a universalização da educação, ou seja, garantir que o conhecimento seja passado para qualquer pessoa, independente de deficiências físicas ou da plataforma de acesso.



Garantir que pessoas com deficiência física tenham acesso ao conteúdo de um LCMS deveria ser priorizado, uma vez que a tecnologia para isso tem evoluído a cada dia e facilitado tal integração. Exemplos disso são softwares que permitem a leitura de textos: o Dosvox [14] e o Webbie (*browser* para deficientes visuais) [15]; linguagem de sinais como o *player* da Rybená [13] (imagem ao lado); e, claro, os padrões de acessibilidade para web.

Os padrões de acessibilidade para web definem regras de validação para que uma página seja considerada acessível, dentre elas podemos citar: sintaxe

do HTML, sintaxe de CSS, uso correto de tabelas, descrição de imagens, etc. Para tal, existem diversas ferramentas de validação como: o DaSilva [10], o Hera [16] e o Examinator [17] [11].

A garantia de acessibilidade da web não é tudo. Para que um site possa ser realmente acessível ele deve ter usabilidade, ou seja, deve ser fácil de usar e prover diversas formas para que isso aconteça. De que adianta um LCMS, com design sofisticado, muitas ferramentas de comunicação, mas péssima navegabilidade pelo teclado e incompatibilidade com navegadores mais antigos ou pouco usados? Para que isso não aconteça é necessário, não somente usar essas ferramentas de validação, mas também testar com vários usuários e com as mais diversas dificuldades e plataformas, para só assim ter alguma garantia de que o software é acessível a todos ou pelo menos para a grande maioria [12].

Referências

- [1] <http://fgsnet.nova.edu/cread2/pdf/Andrade.pdf>
- [2] <http://www.elearningbrasil.com.br/>
- [3] http://www.abed.org.br/publicue/cgi/cgilua.exe/s_y_s/_s_t_a_r_t_.h_t_m?UserActiveTemplate=1por&inoid=883&sid=135
- [4] <http://www.ginux.ufla.br/documentacao/monografias/mono-HudsonPires.pdf>
- [5] www.iscap.ipp.pt/paol/docs/repositorio/introd_elearning.pdf
- [6] <http://www.edutools.info/>
- [7] <http://www.opensourcecms.com/>
- [8] <http://docs.moodle.org/pt/>
- [9] <http://www.ttissa.org/e-learning/help/accessibility.php?lang=pt-br>
- [10] <http://www.dasilva.org.br/>
- [11] <http://www.bengalalegal.com/validacao.php>
- [12] <http://www.bengalalegal.com/teoriaepratica.php>
- [13] <http://www.rybena.com.br/>
- [14] <http://intervox.nce.ufrj.br/dosvox/>
- [15] <http://www.webbie.org.uk/>
- [16] <http://www.sidar.org/hera/>
- [17] <http://www.acesso.unic.pt/webax/examinator.php>

Josiane Lima de Oliveira - josianeolive@hotmail.com - www.josianeolive.blogspot.com

Formada em Tecnologia em Processamento de Dados pela Universidade Federal do Pará. Atua desde 2004 como Analista de Suporte na Empresa de Processamento de Dados do Estado do Pará (PRODEPA) e no desenvolvimento de páginas web de forma autônoma.

Formada em Tecnologia em Processamento de Dados pela Universidade Federal do Pará. Atua desde 2004 como Analista de Suporte na Empresa de Processamento de Dados do Estado do Pará (PRODEPA) e no desenvolvimento de páginas web de forma autônoma.

Um pequeno tour pelas IDEs: PDT, Delphi e Dreamweaver CS3

Nesta edição optamos por apresentar aos nossos leitores um pequeno tour por algumas das IDEs mais conhecidas pela comunidade. Não foi possível explorar ao máximo cada uma delas, devido a questões de espaço, mas esperamos que o pouco aqui apresentado, permita ao leitor ter uma visão melhor sobre cada uma delas.

Um pouco sobre o PDT

O PDT (PHP Development Tools), que inicialmente se chamava PHP IDE Project, é um projeto que fornece uma ferramenta de desenvolvimento baseada na plataforma Eclipse. O projeto abrange o desenvolvimento de componentes necessários para se desenvolver em PHP e facilitar a sua extensibilidade.

Os princípios do projeto são:

- aprendizado fácil e intuitivo;
- integração sem remendos com o WTP;
- aderência ao padrão Eclipse;
- capacidade de extensão;
- apoio contínuo de desenvolvedores PHP.

O Eclipse Web Tools Platform Project (WTP) provê APIs para desenvolvimento J2EE e centrados em web. Você pode baixar o PDT no seu site do portal da Eclipse Foundation (<http://download.eclipse.org/tools/pdt/downloads/>). O *release* que vamos utilizar aqui é o 0.7 RC3 estável, que foi gerado em cinco de abril deste ano. Os testes foram feitos em ambiente Windows com o pacote completo do PDT (PDT *All-in-One*).

Após baixarmos o zip, o descompactamos em um diretório qualquer do disco, abrimos a pasta Eclipse e executamos o arquivo “eclipse.exe”. Esta foi a instalação. É simples e breve. Não se esqueça que o Eclipse é feito em Java, portanto necessita de uma Virtual Machine Java instalada.

Na primeira tela que é exibida você deve definir o diretório de trabalho do Eclipse. Você pode ter

mais de um na mesma máquina de modo a melhor organizar seus projetos.

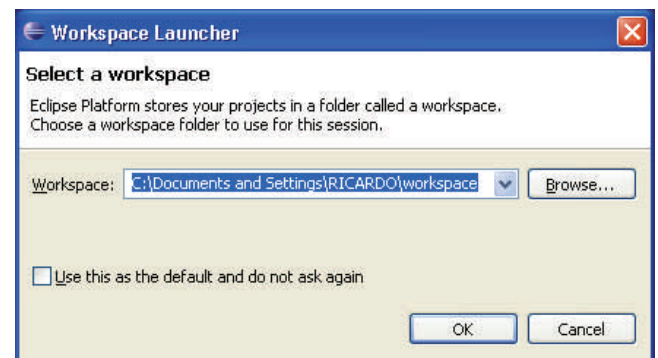


Figura 1 – Definindo Workspace no PDT

Após escolhida a área de trabalho, o Eclipse termina de carregar seus módulos e está pronto para ser usado em seus projetos. Lembramos que esse mesmo Eclipse baixado pode ser utilizado não só com PHP, mas também com qualquer outra linguagem suportada pelo mesmo. Essa é uma das vantagens em ambientes de desenvolvimento que utilizam mais de uma tecnologia: ter apenas um ambiente de desenvolvimento padrão.

Antes de criarmos um projeto, vamos falar um pouco das configurações do PDT. Acessando o menu *Window*, item *Preferences...*, iremos visualizar o item PHP contendo as principais configurações de projetos PHP. Nesta parte você pode configurar detalhes de execução, como se o padrão é utilizar PHP 4 ou 5, detalhes de debug, detalhes do editor, *templates* utilizados na digitação ou na criação de novos

arquivos, e detalhes do servidor padrão em que as fontes ficarão, o que não necessita ser no diretório da área de trabalho.

Fora da parte específica do PHP nas configurações, é importante verificar alguns detalhes que possam atrapalhar o seu projeto. No item *General/Workspace*, você deve definir qual a codificação de caracteres deseja utilizar para arquivos texto. Você também pode fazer essa definição por extensão de arquivo. Basta selecionar o *General/Content Types* e expandir a árvore à direita e ir até o item PHP. Nele você pode modificar a codificação para todas as extensões de PHP. Isto é importante principalmente se você vai importar projetos já existentes para o PDT. Na maioria das vezes os editores utilizam a codificação LATIN-1 (ISSO-8859-1).

Estas configurações podem ser feitas de modo que todos em um ambiente de desenvolvimento tenham o mesmo padrão, bastando modificá-las e depois compactar novamente o Eclipse e redistribuí-lo aos desenvolvedores. Tudo de uma forma muito simples.

Até o momento, não vimos o potencial do PDT no desenvolvimento. Então, vamos criar o nosso primeiro projeto PHP, através do menu *File/New/Project*. Desse modo teremos uma janela de *wizard* com o tipo de projeto que desejamos, no caso um PHP Project. Avançando teremos uma janela onde preencheremos o nome do projeto, o local de armazenamento dos fontes, que pode ser diferente do *workspace*, e qual a versão do PHP utilizar.

Após criado o projeto, você pode criar seus arquivos PHP, HTML, JavaScript, CSS, ou qualquer outro que seja necessário. A janela abaixo mostra um projeto aberto no PDT.

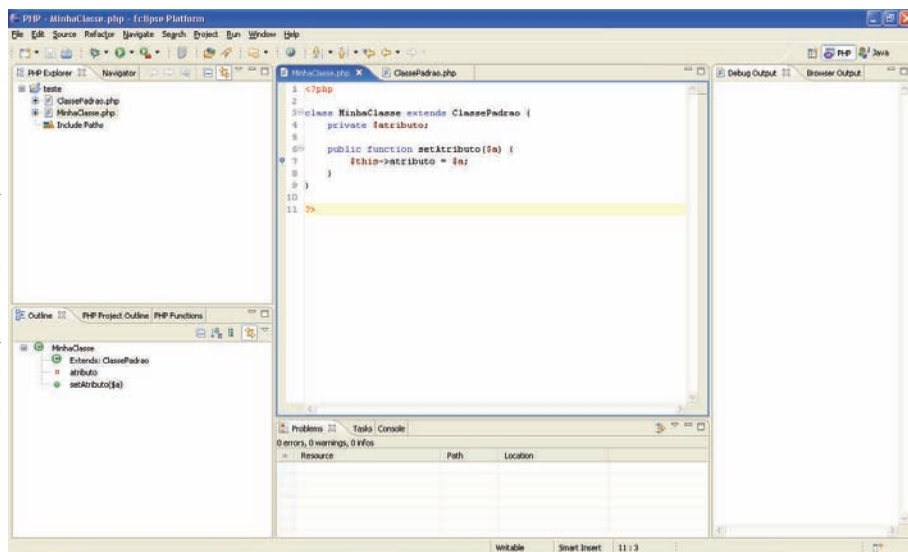


Figura 3 – Interface do PDT

Nesta janela podemos perceber a quantidade de funcionalidades que o Eclipse fornece, como a árvore do projeto, o *outline* com as classes, funções e dentre outras, a saída do debug, o *browser output*, erros direto no código fonte, saída no console, etc. Cada desenvolvedor pode organizar as subjanelas do seu melhor modo. Lembramos que para rodar o *debug* e a execução dentro do Eclipse, é necessária a configuração de mais alguns itens que não será mostrado nessa matéria.

Um ponto importante é o que mais o Eclipse pode nos fornecer como IDE, além do PDT. Um bom exemplo é a integração com sistemas de controle de versão, como o CVS, nativamente, e com o *Subversion*, através do *plugin subclipse*. Há diversos outros *plugins* que podem ser utilizados em conjunto, como para UML, para acesso e gerenciamento de banco de dados, implantação, etc.

A versão 1.0 está prevista para setembro deste ano. Abaixo algumas melhorias e novas características principais:

- interface gráfica genérica para Zend Debugger e XDebug;
- melhoria do *parser* e do modelo de infraestrutura;
- preferências na execução do PHP;
- suporte a *annotations*;
- suporte inicial a refatoração
- integrar editores de HTML, CSS e JavaScript

O único pesar do Eclipse é que como é feito em Java necessita de uma máquina razoável com um mínimo de 512MB de RAM, pois o que utilizamos,

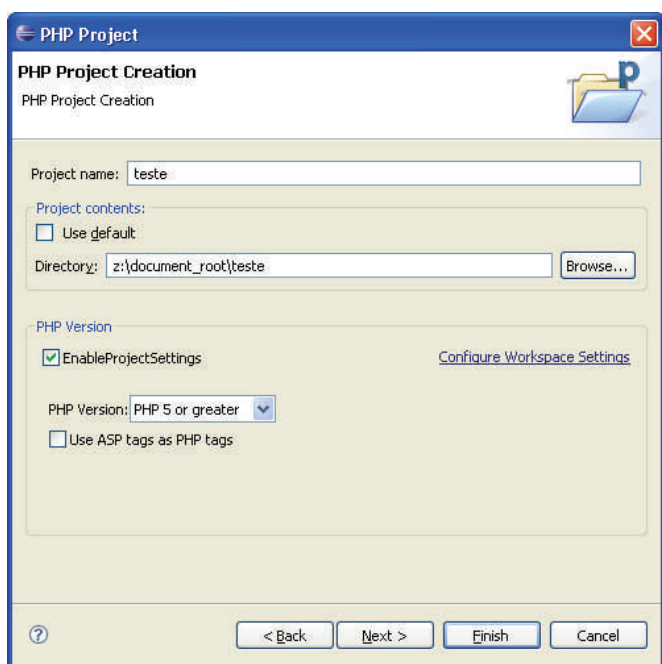


Figura 2 – Definindo um projeto

assim que ele foi iniciado, consumiu 77 MB em uma máquina com 512MB.

O PDT surge para se tornar uma grande plataforma de desenvolvimento livre para o PHP. Só o apoio da comunidade pode melhorá-lo ou destruí-lo. Como o seu desenvolvimento é uma parceria da IBM com a Zend, podemos esperar até uma boa integração com o Zend Framework.

Apresentando o Delphi for PHP

Esta IDE foi lançada em Março de 2007 após a parceria firmada entre a Qadram Software Development e Codegear/Borland atendendo à necessidade de alguns desenvolvedores que desejavam uma ferramenta que permitisse gerar código PHP manipulando os objetos visualmente. A ferramenta é praticamente baseada nos componentes denominados VCL, que são escritos em PHP5 e, em diversos casos, apenas abstrai o acesso a componentes já existentes e bem conhecidos dos desenvolvedores PHP como ADOdb, Xajax, JsCalendar, entre outros.

Apesar de o Delphi for PHP ser uma ferramenta proprietária, a biblioteca VCL é software livre e é baseada na licença LGPL. Alguns componentes suportam o uso a Ajax e tornam bem simples a maioria das tarefas.

Um pequeno exemplo

Seremos objetivos criando um exemplo com um grid usando MySQL. Primeiramente vamos criar um projeto:

Selecionamos *Application* e após clica-se em *ok*, e o projeto será criado. Na sequência, surgirá a interface abaixo. A visão de design e nenhum objeto gerado até o momento.

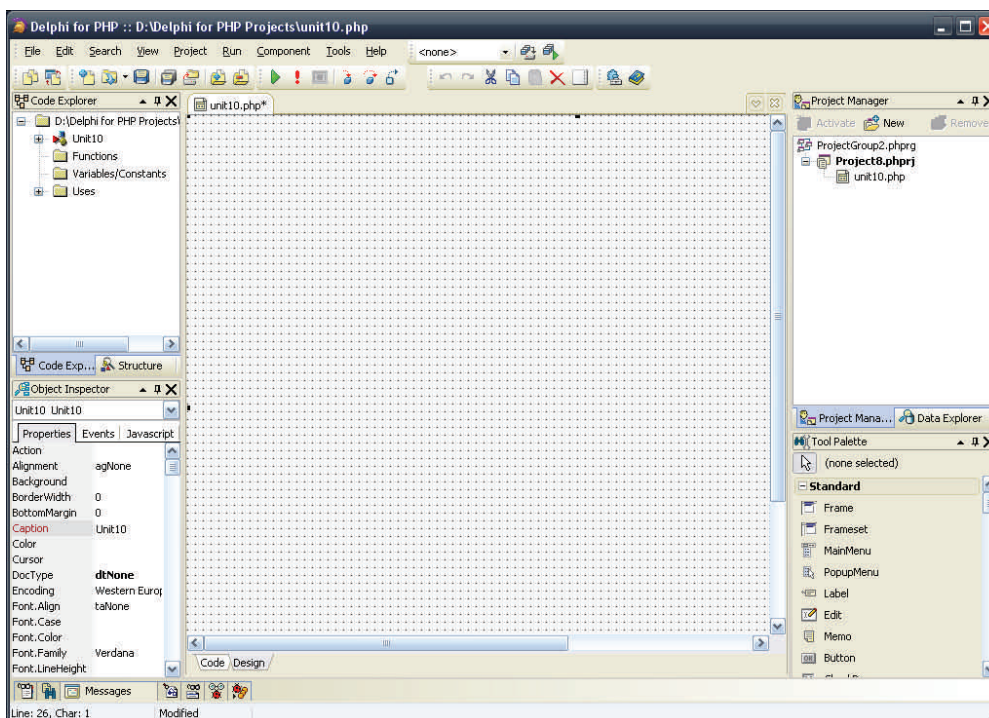


Figura 5 – Interface principal da IDE na visão design

Em seguida, clicamos na aba *Data Explorer*, no lado direito da tela e configuramos uma conexão com o banco MySQL.

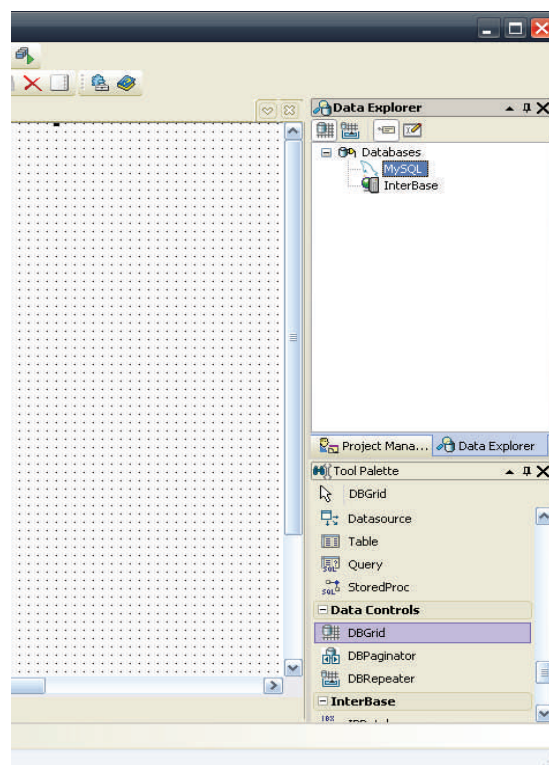


Figura 6 – Configurando uma conexão com o MySQL

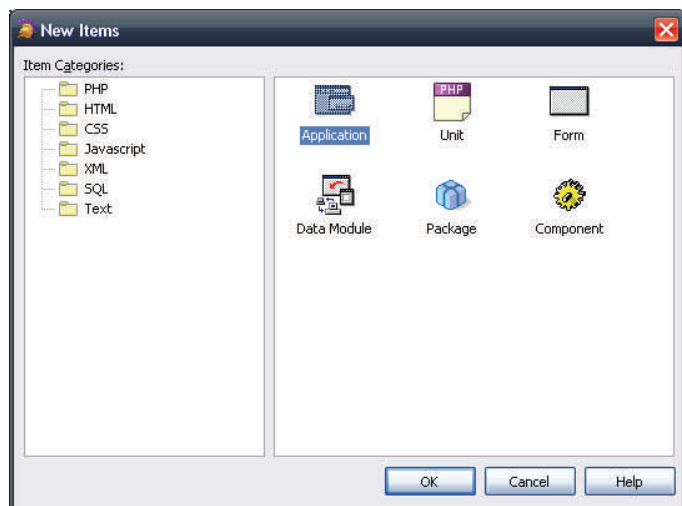


Figura 4 – Definindo um projeto PHP no Delphi

Após selecionar o banco, arrasta-se uma tabela para o *stage* e pronto. Nosso *grid* está criado, conforme a figura seguinte.

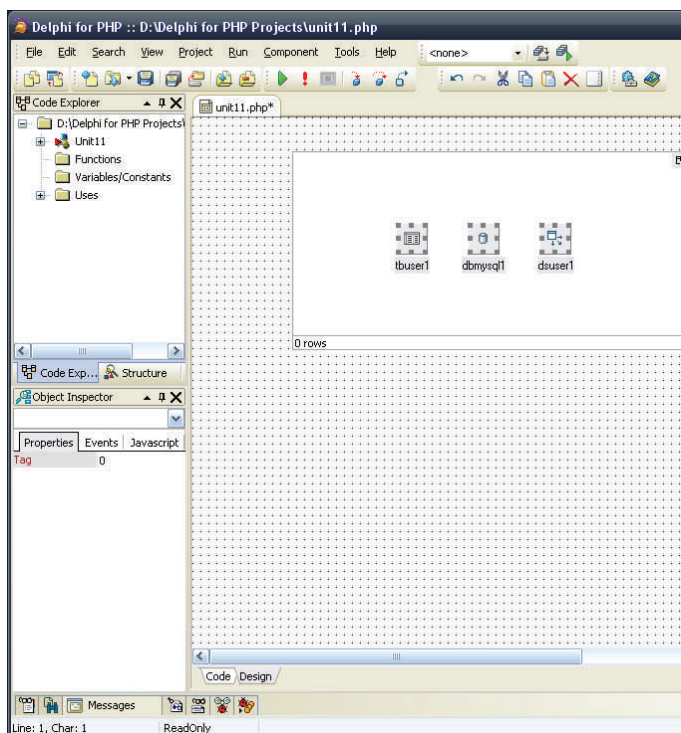


Figura 7 – Grid criado

Ao executar a aplicação definida acima em um *browser*, o resultado apresentado será semelhante ao da figura seguinte.

Host	User	Password	Select_priv
localhost	root	*6BB4837EB7432911	Y
%			N
localhost			Y

3 rows

Figura 8 – Resultado apresentado da execução da aplicação no Internet Explorer

Fazer um grid como esse, com recursos que permitem a edição sem *reload* da página e sem o intermédio de uma ferramenta dessas, é mais complexo e, por mais que você já tenha os componentes prontos, você não tem essa produtividade oferecida pelo Delphi for PHP.

O Código PHP gerado para o programa acima foi:

```
<?php
//Includes
require_once("vcl/vcl.inc.php");
use_unit("styles.inc.php");
use_unit("db.inc.php");
use_unit("dbtables.inc.php");
```

```
use_unit("dbgrids.inc.php");
use_unit("forms.inc.php");
use_unit("extctrls.inc.php");
use_unit("stdctrls.inc.php");

//Class definition
class Unit11 extends Page
{
    public $dduser1 = null;
    public $dsuser1 = null;
    public $dbmysql1 = null;
    public $tbuser1 = null;
}

global $application;

global $Unit11;

//Creates the form
$Unit11=new Unit11($application);

//Read from resource file
$Unit11->loadResource(__FILE__);

//Shows the form
$Unit11->show();
```

?>

Debugger interno e compatibilidade do código gerado

O Delphi conta com um *debugger* interno, chamado *PHP DBG Listener*, que é um *debugger* que possui versão *free* e versão comercial. Este é o *debugger* para PHP mais conhecido e tem sua qualidade comprovada pela comunidade. A versão *Free* já é suficiente para a maioria dos desenvolvedores e foi adotada pelo Delphi for PHP. Podemos controlar o DBG por meio da IDE, incluindo e retirando Breakpoints.

O código gerado pelo Delphi for PHP é compatível apenas com PHP5. Inicialmente, foi divulgado que o código seria compatível com PHP a partir da versão 5.1, porém, devido a mudanças na extensão filter do PHP, o código deve ser compatível apenas com a versão 5.2+.

Suporte a bancos de dados

Atualmente, a ferramenta gera código compatível apenas com MySQL e Interbase, porém, por utilizar como base a biblioteca ADOdb, é possível alterar isso após o build da sua aplicação ser gerado. No entanto, isso gera um trabalho manual que vai contra o princípio da utilização desta IDE, que é a produtividade. Dependendo da estrutura utilizada, pode ser complexo realizar, a cada build gerado, uma alteração na classe para atender ao banco que você deseja. Além do mais, que na base de desenvolvimento você ter de usar outro SGBD para desenvolver. Porém, na palestra do Delphi for PHP foi informado que existe a intenção de o produto oferecer

o suporte a outros bancos de dados. Assim, acredita-se que é uma questão de tempo para que outros SGBDs sejam suportados.

Gerando Build

Após finalizar o desenvolvimento da aplicação, você poderá gerar um pacote para ser utilizado em um ambiente de produção. Este processo é bem simples e consiste em alguns passos *next*, *next* e *finish*, que usuários Windows e provavelmente futuros usuários desta IDE já estão familiarizados.

Nos testes surgiram alguns problemas com o *build* gerado, devido a um *bug* relacionado à extensão *filter* que já foi corrigido.

Perspectivas para o Delphi for PHP

Acredita-se que esta ferramenta tenha um longo futuro. Tivemos contatos de algumas pessoas que já encomendaram a sua licença junto à Codegear/Borland e estão bastante ansiosas.

Acredita-se que ela será amplamente utilizada em ambientes corporativos que ainda não adotaram o uso de software livre como regra ou naqueles ambientes em que há muita preocupação com a produtividade. É uma ferramenta que deverá ser considerada no momento em que as pessoas que tomam decisões forem optar sobre qual será a sua plataforma de desenvolvimento.

O Dreamweaver CS3

O Dreamweaver foi uma das ferramentas que mais garantiu destaque para a sua empresa mantenedora, a Macromedia, recentemente foi adquirida pela Adobe. Diferencia-se em algumas características de outros ambientes de desenvolvimento mais tradicionais, como o Eclipse PDT, Zend, ou mesmo do recém lançado Delphi PHP.

Com um foco inicial em layout e design, aos poucos foram introduzidos alguns recursos relacionados às linguagens de programação para web, como o próprio ColdFusion, linguagem proprietária da Macromedia, que estendeu-se até o ASP e PHP.

Tal é o seu grau de utilização por muitos desenvolvedores, que optamos por conceder destaque para essa IDE, que há pouco tempo apresentou a sua mais recente versão: o Dreamweaver CS3.

Perfil, instalação e configuração.

O Dreamweaver iniciou sua trajetória como um ótimo editor de *Layout* para *websites*, pois apresentava recursos de uma forma melhor e mais completa, do que ferramentas similares da época como o FrontPage, Netscape Composer, dentre outros. Alguns requintes, tais como menus prontos em *JavaScript*, também faziam parte do pacote. Sua adoção dentre *webdesigners* cresceu muito, tornando esta ferramenta bastante popular.

Aos poucos, seus recursos para algumas linguagens de programação, como ASP, PHP e ColdFusion, foram melhorando. Muitos recursos foram adicionados, tais como um forte editor de CSS, utilizando-se do já conhecido auto-completar e verificação de sintaxe. Inclusive, durante a codificação HTML, as versões mais recentes procuravam definir estilos CSS para as ações do usuário (definição de tipo e tamanho de fonte, parágrafo etc), tentando melhorar o código final.

O editor de HTML também possui algumas características de realce de sintaxe, que permitem identificar erros de *tags* não fechadas ou atributos duplicados, propriedades que não são compatíveis com todos os *browsers*. Isso traz algumas facilidades para o usuário, alertando-o de erros já nesta etapa do projeto.

Agora na sua versão mais recente, o Dreamweaver CS3 incorporou o *framework* *Spry*. Seu objetivo é oferecer mais *wizards* ao usuário.

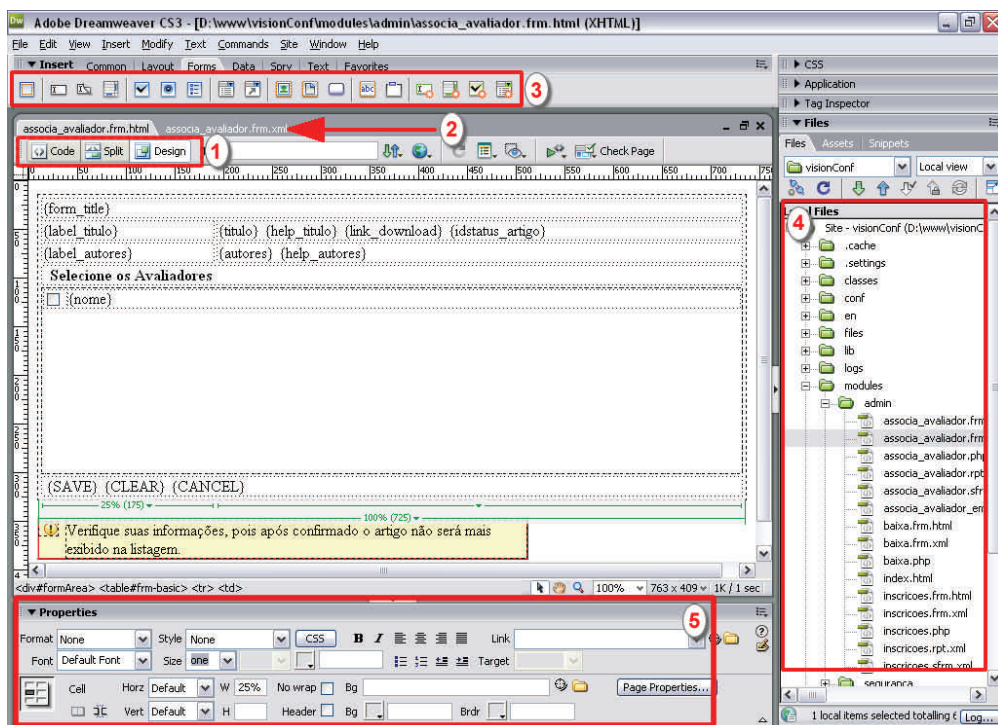


Figura 9 – Visão geral do Dreamweaver CS3

Na figura 9, pode-se observar uma visão geral

da interface da ferramenta, com a identificação das suas principais áreas. A saber:

- 1.botões de acesso rápido para alternar entre código gerado, aspecto visual dos itens e um misto de ambos;
- 2.abas para alternar entre os documentos em edição;
- 3.barra de ferramentas;
- 4.árvore de diretórios do site;
- 5.propriedades do elemento selecionado.

O download pode ser feito no site da Adobe. O link está citado nas referências. Trata-se de um produto pago, mas a empresa permite que você faça uma avaliação por trinta dias.

A instalação é bastante simples, sendo apenas necessário seguir os passos descritos no instalador que o software já estará apto para uso.

Definindo um projeto

O Dreamweaver não possui diretamente o conceito de projeto. Para a IDE, o usuário está definindo um site. Um site, nada mais é do que a definição de algumas características básicas como diretório das páginas, host onde serão alocadas as páginas, usuário e senha para conexão. Abaixo é apresentada apenas a área de definição remota, pois contém algumas informações mais relevantes aos nossos comentários.

alguns itens, como FTP Seguro e conexão por Firewall.

Os itens a partir de “Enable file check in and check out”, são de uso opcionais por parte do desenvolvedor. Este deve habilitá-los se desejar trabalhar em uma equipe com mais desenvolvedores em um mesmo site. Estas configurações adicionais são para controle de edição concorrente de arquivos do site.

Desenvolvimento em equipe

Muitos podem desconhecer, mas o Dreamweaver possui algumas funcionalidades que auxiliam no desenvolvimento em equipe. Claro que pode parecer pouco se comparado a um recurso de “versionamento” de arquivos e times de desenvolvimento, mas, em alguns casos, pode contribuir para o seu trabalho.

Trabalhando com o conceito de check in e check out, é possível que um usuário detenha para si o controle de edição de um arquivo. Dessa forma, inconsistências geradas pelo trabalho da equipe em um mesmo projeto são reduzidas.

Você encontrará mais informações sobre sua utilização na nota técnica da Adobe citada nas referências deste artigo. Basicamente quando o usuário realiza um check out em um arquivo, está obtendo a versão mais recente para o diretório local de trabalho. O arquivo ficará bloqueado para uso deste usuário, até que o mesmo faça check in no arquivo. Os outros usuários visualizarão o arquivo com o ícone de um pequeno cadeado na árvore de diretórios, indicando que o arquivo está bloqueado para edição. Na figura abaixo, podemos observar a visão de arquivos do servidor com o diretório local. Os arquivos bloqueados para edição, serão listados referentes ao item 1, destacado na imagem.

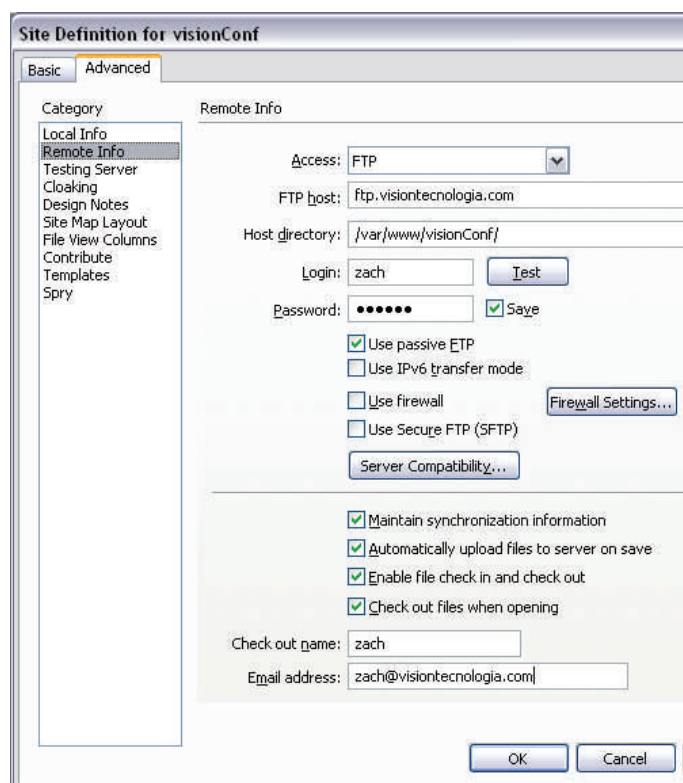


Figura 10 – Características remotas da definição de um site

Como se pode observar, já estão disponíveis

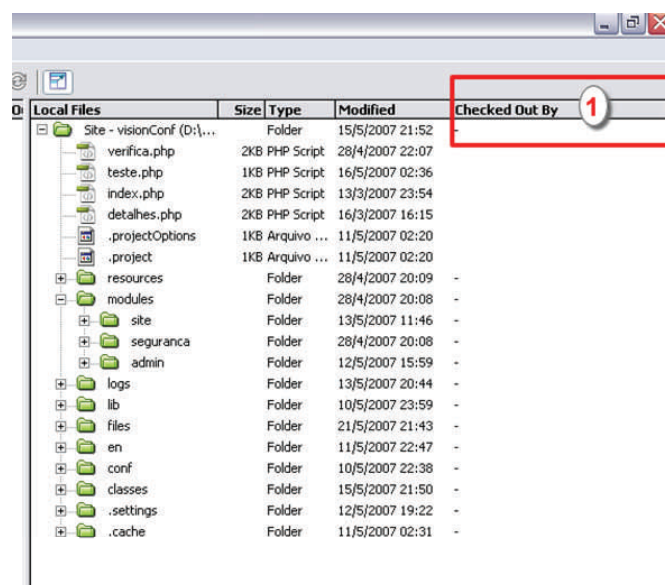


Figura 11 – Visão dos arquivos remotos

Sempre que se for iniciar um turno de trabalho, recomenda-se fazer a operação de sincronização para obter a versão mais recente dos arquivos do servidor para o diretório local. Antes de concluir esta operação, o Dreamweaver exibe uma listagem dos arquivos que serão sobrescritos. A operação pode ser feita em ambos sentidos, tanto do diretório local para o servidor ou, como seria o mais recomendado, do servidor para o diretório local.

Não abordamos neste artigo a possibilidade da utilização do Visual SourceSafe(VSS) integrado ao Dreamweaver. Para mais informações sobre isto, recomendamos a leitura de algumas notas técnicas no *site* da própria Adobe.

Novos recursos

Na versão CS3, o Dreamweaver incorporou o Spry Framework, desenvolvido pela própria Adobe para oferecer alguns recursos de Ajax para suas aplicações. Por meio da interface da própria ferramenta, o usuário poderá definir componentes visuais que se utilizam desse *framework*, que deverá ser exportado junto com a aplicação. O Spry se divide em três grupos de funcionalidades: Spry Data, Spry Widgets e Spry Effects.

Com o Spry Data, é possível integrar facilmente o banco de dados ou os arquivos XML à aplicação, já com opções de filtro e ordenação.

Por meio do Spry Widgets, pode-se facilmente adicionar elementos como listas, abas, validação de formulários etc. Estes elementos podem mudar a apresentação da sua aplicação.

Já com o Spry Effects, é possível adicionar elementos de efeitos visuais em suas páginas. Os elementos poderão crescer (*grow*), encolher (*shrink*), realçar (*highlight*), esmaecer (*fade*), dentre outros efeitos. Na figura 5, apresenta-se a paleta de componentes Spry, na barra de ferramentas.

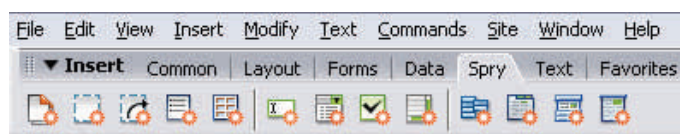


Figura 12 - Paleta dos componentes Spry

O diferencial da ferramenta é oferecer ao usuário todos esses recursos apenas a um clique, arrastando e definindo propriedades dos elementos, para alterá-los de acordo com os seus propósitos.

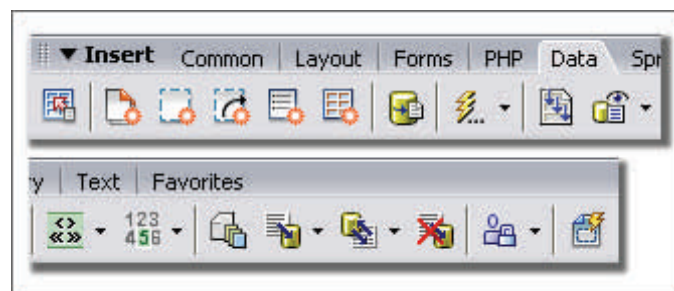


Figura 13 - Data Components integrados com o Spry

Não poderíamos deixar de citar a paleta de *Data Components*. Com ela, é possível conectar com o banco de dados e criar listagens com paginação sem editar praticamente código algum. Pode-se conectar diretamente a bancos de dados MySQL, definindo uma conexão. Criada uma conexão, você poderá utilizá-la para definir um *RecordSet*.

Com os *RecordSets*, pode-se criar tabelas dinâmicas e adicionar paginação às mesmas. Tudo isto com o auxílio e a facilidade da interface visual da IDE. Na figura abaixo, a definição de um *RecordSet* e o conjunto de dados retornado pelo mesmo.

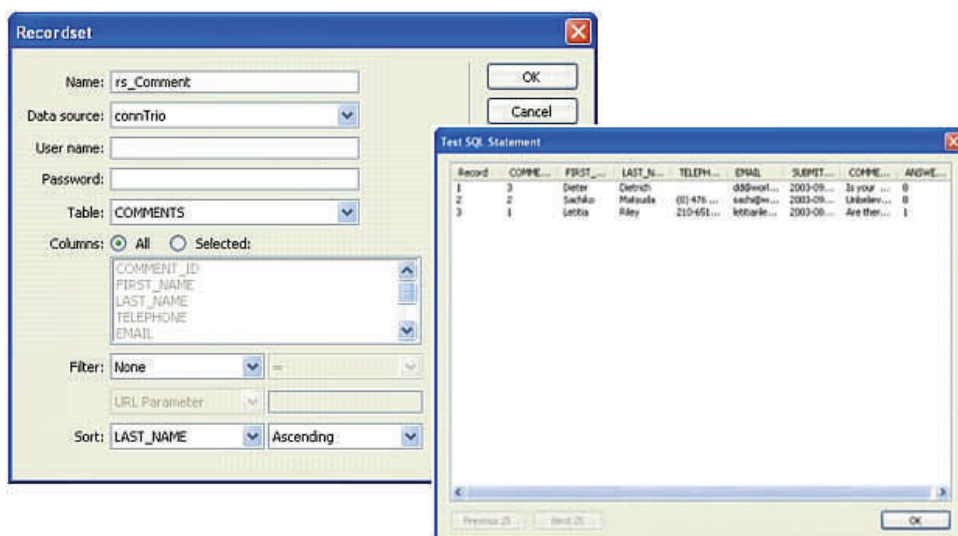


Figura 14 - *RecordSet* e o seu conjunto de dados retornado

Algumas considerações sobre a IDE

O Dreamweaver pode ser uma ótima opção para iniciantes. Sua facilidade de configuração e uso pode tornar menos frustrante o processo de aprendizagem. A inclusão de um *framework* como o Spry é ótima para ajudar desenvolvedores iniciantes a contextualizarem-se com a utilização de bibliotecas. Os recursos advindos com o *Spry* reduzem o esforço de buscar bibliotecas para necessidades de abas, menus e algumas animações em sua aplicação.

Concluindo

Procuramos expor alguns pontos importantes

destas conhecidas IDEs, para que o leitor possa tomar conhecimento e optar pela que mais lhe convir, ou simplesmente se interesse em realizar uma avaliação sobre cada uma delas.

Apesar de apresentarmos as três IDEs, não seria adequado compará-las, pois são projetos diferentes entre si. O PDT é um ótimo produto da consagrada ferramenta Eclipse. O Delphi estabeleceu fortes ali- cerces no desenvolvimento *desktop* e agora está buscando participação no desenvolvimento Web. E o Dreamweaver que, de uma ferramenta de *design* e *layout*, está cada vez mais apresentando característi- cas voltadas à programação também.

De qualquer forma, a melhor IDE para você, será a que melhor adaptar-se à sua forma de traba- lho e aos recursos que você dispõe para investimen- to.

Caso nossos leitores demonstrem interesse em

matérias que explorem detalhadamente as ferramen- tas apresentadas ou surjam interessados em continu- ar abordando o tema, nossa equipe concederá conti- nuidade ao tema.

Referências e links sugeridos:

PDT

http://www.zend.com/pdt#all_in_one

Delphi for PHP

<http://www.codegear.com/products/delphi/php>

<http://www.qadram.com/news.php#6>

<http://dd.cron.ru/dbg/>

<http://neshertech.net/adler/2007/04/09/bug-no-vcl-delphi-for-php-corrigido/>

Dreamweaver

<http://www.adobe.com/products/dreamweaver/>

http://www.adobe.com/cfusion/knowledgebase/index.cfm?id=tn_15336

Adler Brediks Medrado - adler@neshertech.net

Analista e desenvolvedor, utiliza PHP na maioria dos seus projetos.

Atualmente presta serviços em um órgão federal em Brasília e é instrutor de cursos PHP na X25 Treinamento e Con- sultoria. Além disso, também é proprietário da empresa Nesher Technologies que presta serviços de consultoria pa- ra empresas do Brasil e do exterior.

Membro da equipe de coordenação do Grupo de usuários PHP do Distrito Federal (PHPDF).

Flávio Zacharias Fagundes - zach@phpmagazine.com.br

Flávio Fagundes é formado em Ciência da Computação pela PUCRS em 2003 e atualmente trabalha como analista de sistemas em uma empresa de captura de transações eletrônicas.

Utiliza e fomenta a tecnologia PHP desde 2002 em vários projetos. Presta consultoria em desenvolvimento de siste- mas e também é um dos fundadores da revista PHP Magazine, participando ativamente de sua gestão.

Ricardo Aragão da Silva - ricardoaragao@phpmagazine.com.br - www.raragao.eng.br

Engenheiro de computação e entusiasta da linguagem PHP desde 2002.

Presta consultoria em desenvolvimento de sistemas e leciona em cursos de graduação e pós-graduação em faculda- des de Brasília no Distrito Federal.

Atualmente é chefe da Célula de Desenvolvimento do Centro de Desenvolvimento de Sistemas (CDS) do Exército Brasileiro, onde desenvolve sistemas com as mais diversas tecnologias de mercado.

É também fundador e administrador da revista PHP Magazine.



Guará

Uma solução em software livre para redução de custos no desenvolvimento de aplicações Web

Este artigo tem por objetivo demonstrar o Gerenciador Guará como uma alternativa desenvolvida em software livre para redução de custos no desenvolvimento de aplicações Web, bem como, comparar as vantagens de sua tecnologia sob as formas tradicionais de desenvolvimento de aplicações Web.

1. Introdução

Devido ao alto custo da aquisição de ferramentas de aplicações de gerenciamento e da necessidade de aquisições de outras ferramentas auxiliares tais como gerenciador de relatórios e Banco de Dados, aliados ao alto custo da mão de obra dos profissionais da Tecnologia da Informação, viu-se então a necessidade de criar uma solução, a qual atendesse todos esses requisitos.

Com a constante evolução das ferramentas para desenvolvimento de aplicações via Web, que são programas de computador aos quais usuários acessam com um *Web Browser* na Internet, também são *Websites* dinâmicos que oferecem não somente informações estáticas para os usuários, mas os deixa usar ferramentas dinâmicas para trabalhar com uma aplicação, disponibilizando informações gerenciais em qualquer lugar do mundo. Entende-se que com essa afirmação um administrador de empresas, com filiais espalhadas em localidades distantes geograficamente, pode acompanhar toda movimentação da empresa à distância. Devido a esta evolução, as informações contidas nestas aplicações tornam-se cada vez mais de difícil manuseio. Com isso, foi definido que a solução seria uma aplicação Servidor a qual rodaria em ambiente *Web*.

A filosofia da tecnologia Web sob Demanda possibilita a criação de um Gerenciador de Aplicações na qual o usuário pode desenvolver sistemas em grande escala e com uma melhor produtividade,

a fim de otimizar o tempo de desenvolvimento dos mesmos. A necessidade do desenvolvimento de aplicações via Web agregadas a esta filosofia permitiu a idealização do Gerenciador Guará.

A linguagem escolhida para o desenvolvimento do Guará foi o PHP, que não é compilado, mas sim interpretado por um servidor Web no provedor que hospeda o CGI. Por esse motivo o torna mais rápido do que os CGI's escritos em outras linguagens como C e Perl. Um segundo motivo que levou o desenvolvimento em PHP, foi a vasta bibliografia encontrada e a grande gama de profissionais com conhecimento e utilização desta linguagem.

Como terceiro motivo, pode-se considerar também a portabilidade a conexões do PHP, pois com esse diferencial, o Banco de Dados utilizado é de escolha do cliente, podendo ser um Banco de Dados Livre ou Proprietário.

2. O Gerenciador Guará

O Gerenciador Guará é o conjunto da necessidade da evolução para desenvolvimento de aplicações via Web agregado à filosofia da tecnologia Web sob Demanda juntamente com a linguagem PHP e a alternativa de escolha do Banco de Dados por parte do cliente.

A proposta acima permite que o cliente, ao adquirir o Gerenciador Guará, não mais fique preso as ferramentas pagas, ou seja, o Guará pode rodar em um Sistema Operacional, Servidor Web e Banco de Dados Livres.

2.1. Funcionamento

São as facilidades oferecidas pelo Gerenciador Guará, que proporcionam maior velocidade e flexibilidade no seu manuseio, visto que, o Guará monta suas páginas finais escrevendo os códigos HTML e JavaScript de forma totalmente automática.

Diferente da forma tradicional de programação, o Guará não segue os passos utilizados pelo método tradicional. Na criação de uma URL (Tela de Aplicação para Gestão de Dados) seu código é gerado automaticamente, bem como a implementação da tabela no Banco de Dados. As URLs criadas pelo gerenciador não existem fisicamente em um arquivo para serem publicadas, são encontradas somente suas definições armazenadas no Banco de Dados para que, quando forem solicitadas, elas sejam disponibilizadas em formato HTML e JavaScript.

Esta agilidade se dá por meio da utilização de um Dicionário de Dados on-line, advindo da montagem das telas através da leitura dinâmica de critérios e parâmetros técnicos cadastrados no Banco de Dados, juntamente com o conceito de Auto Gerenciamento e das características de Orientação a Objetos e Orientação a Eventos em todos os componentes disponibilizados para a aplicação.

Um outro aspecto que torna funcional o gerenciador é a aplicação de uma estrutura matricial de definição de sistemas em URLs e OBJETOS (unidades de apresentação dos dados), facilitando um desenvolvimento personalizado, rápido e adequado às necessidades particulares de cada aplicação.

2.2. Aspectos técnicos diferenciados

Utilização de tecnologia Web sob Demanda e uma arquitetura natural Intranet / Extranet;

Acesso aos dados através de visão relacional em tabelas SQL, visão multidimensional dos dados e Orientação a Objetos;

Uso de algoritmo de armazenamento de dados desprovido de quaisquer declarações e totalmente dinâmico, ou seja, "desprovido de declarações". Significa, portanto, que não é preciso declarar o número de dimensões ou intervalos de valores e não existem limites em qualquer um dos dois;

A utilização de dicionário de dados on-line;

Implementações sem necessidade de manutenções que causem paralisações ou reprogramações de códigos, e nem nas páginas Web envolvidas;

Registro automático dos dados do operador credenciado, identificando seu nome, o tipo de atualização e a data e hora da operação em cada registro da base de dados;

Utilização de históricos (consulta aos dados em qualquer momento da vida do registro) e históricos excluídos (consulta aos dados excluídos e recuperação automática de registros excluídos);

Telas totalmente dinâmicas, utilizando-se integralmente das características de Orientação a Objetos e Orientação

a Eventos em todos os componentes disponibilizados pela aplicação;

Parâmetros de desenvolvimento totalmente escaláveis e padronizados;

Utilização extensiva do conceito de Auto Gerenciamento, ou seja, o gerenciador pode ser customizado conforme as necessidades apresentadas;

Possui um gerador de relatórios integrado;

Multiplataforma, ou seja, possibilidade de operação em até 90 plataformas de hardware e software diferentes;

Comunicação com diversos Bancos de Dados, livres ou proprietários;

Possibilidade de implementação automática de "EAI" (Enterprise Application Integration) e de "IDC" (Internet Data Center).

3. Criando uma aplicação no Gerenciador Guará e o impacto do desenvolvimento

Para compreender melhor as funcionalidades do Gerenciador Guará, será utilizado o método ilustrativo. Logo, os detalhes de manuseio da ferramenta não serão relevantes, mas sim os seus aspectos como: a criação de telas, nomeadas de URLs, e componentes, chamados de OBJETOS. Diante disto, ao decorrer desta seção, serão mostrados os procedimentos necessários para criar uma aplicação simples, como por exemplo, uma Agenda de Contatos.

3.1. Criando uma Agenda de Contatos

Esta Agenda de Contatos conterá os seguintes campos: **código**, **nome**, **data de nascimento**, **endereço**, **cidade**, **estado**, **país**, **telefone** (residencial, comercial, celular, fax e outros) e **e-mail**. O campo **estado** será relacionado com a tabela d99 já existente, mostrando na tela o seu segundo campo. O campo **país**, também será relacionado, com a tabela d98 e mostrará o segundo campo da mesma.

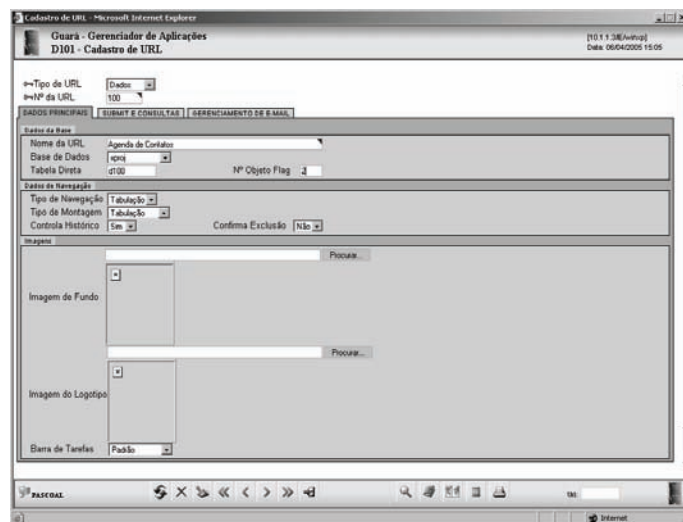


Figura 1 – URL d101 Cadastro de URL

Para criar este exemplo, é necessário acessar de dentro do Guará a URL (d101) Cadastro de URL,

onde define-se a criação da URL. Neste caso serão informados somente os parâmetros necessários para sua criação, que são: **Tipo da URL = Dados, N° da URL = 100, Nome da URL = Agenda de Contatos, Base de Dados = xproj, Tabela Direta = d100 e N° Objeto Flag = 2**, como ilustra a Figura 1.

Logo, após feita a gravação deste registro na URL (d101), o Guará automaticamente cria a tabela d100 no Banco de Dados selecionado, o qual foi definido como **xproj**, com a seguinte estrutura: CHAVE, REGISTRO e NUMERO. Com essa estrutura (veja Figura 2) é que se consegue uma melhor dinâmica na componentização do Guará.

```
mysql> desc d100;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| CHAVE | varchar(50) | YES | PRI |          |       |
| REGISTRO | text      | YES |     | NULL    |       |
| NUMERO | int(11)    | YES |     | NULL    |       |
+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

Figura 2 – Estrutura da tabela d100 no Banco de Dados

Com a definição da URL concluída, será feita a componentização desta URL, para isso, é necessário acessar a URL (d102) Cadastro de Objetos (veja Figura 3), em que serão informados os parâmetros de acordo com a Tabela 1:

Componente	Código URL	Nº Objeto	Nº Linha	Texto Prompt	Tam Caixa	Máximo	Tipo Objeto	Digitação	Consulta	Relacionamento	Intervalo Num
Código	d100	1	1	Código	5	4	TEXT	Obrigatória	Chave		Um
Nome	d100	2	2	Nome	61	60	TEXT	Obrigatória	Conteúdo		
Data Nascimento	d100	3	3	Data Nascimento			DATA	Obrigatória	Conteúdo		
Endereço	d100	4	4	Endereço	61	60	TEXT				
Cidade	d100	5	5	Cidade	41	40	TEXT				
Estado	d100	6	5	Estado	0	1	SELECT ÚNICO			d99:2	
País	d100	7	5	País	0	1	SELECT ÚNICO			d98:2	
Telefone	d100	8	8	Telefone	4	2	GRADE EXP				
E-mail	d100	9	9	E-mail	61	60	TEXT				

Tabela 1 – Definição dos componentes a serem criados

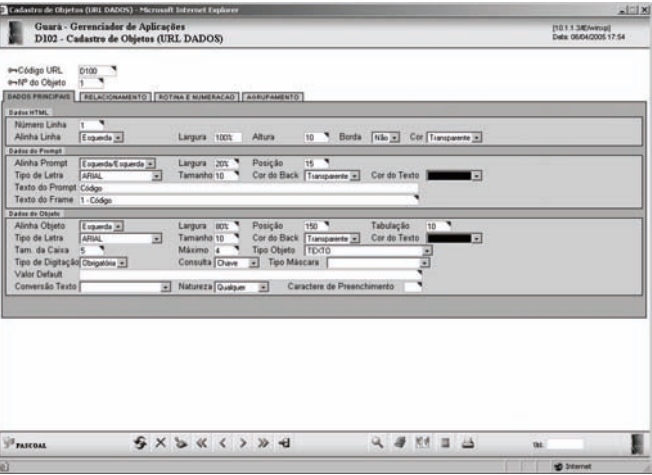


Figura 3 – URL d102 Cadastro de Objetos

Com isso, a URL de Agenda de Contatos está pronta, como mostra a Figura 4, e apta à população de dados. Note que poucos procedimentos foram necessários para implementar este exemplo, tendo inclusive sido feito relacionamentos a outras tabelas existentes (d98 e d99) como no caso dos componentes **estado** e **país**.

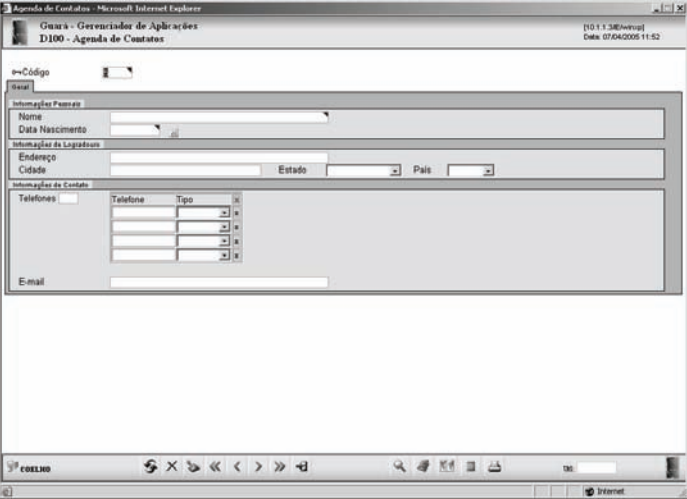


Figura 4 – URL d100 Agenda de Contatos

A Figura 5 permite observar como os dados ficam armazenados na tabela d100 no Banco de Dados, após a gravação de alguns registros na URL da Agenda de Contatos.

Cada registro armazenado fica delimitado por um caractere “^”, em que cada posição corresponde ao componente definido na URL d102, dessa forma tem-se a flexibilidade de criar mais componentes, excluir componentes existentes e dar manutenção sem provocar uma interrupção no Banco de Dados, pois não há necessidade de alterar a estrutura da tabela e sim os dados do campo REGISTRO.

Este item permitiu verificar a diferença existente na forma de criação de uma URL pelo Gerenciador Guará, comparada à forma tradicional. Visto pelo lado tradicional, os procedimentos a serem seguidos seriam: definir a estrutura da aplicação, encaminhar a estrutura para o DBA implementar, criar a tela para a interface, criar os arquivos para a comu-

nicação com o Banco de Dados e, finalmente, publicá-los em um diretório Web. Note que a forma tradicional de desenvolvimento envolve vários profissionais, diferentemente do Gerenciador Guará, em que apenas um profissional chega ao mesmo resultado em pouco tempo, sem se preocupar

permitindo assim uma redução dos profissionais envolvidos. Contudo, pode-se dizer que a utilização do Guará exerce forte influência no resultado final do desenvolvimento de Aplicações Web.

CHAVE	REGISTRO	NUMERO
1	PASCOAL^20050406183625^1^PASCOAL^25051978^EUGÊNIO R KOERICH^SÃO JOSÉ^SC^BR^48 9115137062^pasverni@terra.com.br	1
2	PASCOAL^20050406183654^2^ANA PAULA^10041980^EUGÊNIO R KOERICH^SÃO JOSÉ^SC^BR^48 9111533662^anaprobst@hotmail.com	2
3	PASCOAL^20050406183719^3^JORGE NUNES^01011960^EMÍLIO DE SOUZA^SÃO JOSÉ^SC^BR^48 346243341^nunes_jorge@yahoo.com.br	3

3 rows in set (0.00 sec)

Figura 5 – Estrutura de armazenamento matricial da Tabela d100

com a hierarquia exigida pelo modo tradicional.

3.2. O impacto da utilização do Guará nos custos de desenvolvimento

No mercado atual, é impossível sobreviver sem um mínimo de investimento em tecnologia. Das pequenas empresas até as multinacionais, a informática está presente de maneira essencial para a sobrevivência dessas organizações. Logo, todas as organizações de certo modo precisam investir algo em tecnologia. Mas por que não investirem menos sem perderem a qualidade?

A utilização de software livre pelo Gerenciador Guará proporciona benefícios como: vantagens estratégicas (trazendo maior praticidade, independentemente de posições filosóficas ou princípios morais), segurança, portabilidade, facilidade de uso, ou suporte corporativo, autonomia tecnológica, compartilhamento do conhecimento além, principalmente, das reduções de custos, que começam desde os licenciamentos até os requisitos de hardware, gerando assim impactos de escala macroeconômica.

O impacto nas reduções de custos gerados pelo Guará interfere diretamente nas aquisições de licenças de Banco de Dados, visto que não há necessidade de adquirir: (1) um Banco de Dados proprietário; (2) um Gerenciador de Relatórios, que já é integrado ao Guará; (3) um Sistemas Operacionais, dada à independência da sua plataforma; (4) Requisitos de Hardware, pois as máquinas clientes não necessitam de hardware avançado; (5) Tempo de Desenvolvimento, pois o gerenciador por meio da sua estrutura e facilidade de criação reduz em muito o tempo de desenvolvimento,

Considerações finais

Este artigo mostrou que o avanço tecnológico exige que o mercado se adapte as necessidades agregadas a uma redução de tempo e otimização do trabalho empregado pelo profissional, juntamente com a redução de custos.

Esta tendência, aliada à tecnologia Web sob demanda mais a linguagem PHP e à alternativa de escolha do Banco de Dados por parte do cliente, permitiram a idealização do Gerenciador Guará. No decorrer do trabalho foi possível perceber que sua estrutura e facilidade atendem a estas necessidades permitindo uma redução de tempo e profissionais com respectivo impacto positivo sobre o custo do desenvolvimento das Aplicações Web, quando comparado à forma tradicional de desenvolvimento.

Referências e links sugeridos

Soares, Wallace. *Programação em PHP, Conceitos e Aplicações*. Editado por Érica, Brasil: 2000.

Freire, Bruno. *Software livre no setor privado: Quer pagar quanto?*, em <http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=1553>. Brasil: março de 2004.

Ferraz, Nelson. *Vantagens Estratégicas do Software Livre para o Ambiente Corporativo*, em http://www.universia.com.br/pesquisa_bibliotecas/materia.jsp?id=3880. Brasil: março de 2004.

Amadeu, Sérgio. *Entrevista ComCiência*, em https://gestao.idbrasil.gov.br/menu_interno/docs_prog_gesac/artigos_entrevistas/Document.2004-06-16.2523. Brasil: Abril de 2007.

Pascoal Vernieri - pascoalvernieri@gmail.com

Bacharel em Informática pela URCAMP/RS. Especialista em Desenvolvimento de SW para WEB pela UNIVALI/SC e em Engenharia de Projetos de SW pela UNISUL/SC. Atuando desde 2001 com desenvolvimento WEB, possui sólidos conhecimentos em PHP, JAVA, MySQL e Oracle.

Professor da UnisulVirtual e Analista de Sistemas do Núcleo de Informática da Unisul, desenvolve também atividades de Consultor Autônomo. Bacharel em Informática pela URCAMP/RS. Especialista em Desenvolvimento de SW para WEB pela UNIVALI/SC e em Engenharia de Projetos de SW pela UNISUL/SC. Atuando desde 2001 com desenvolvimento WEB, possui sólidos conhecimentos em PHP, JAVA, MySQL e Oracle.

Professor da UnisulVirtual e Analista de Sistemas do Núcleo de Informática da Unisul, desenvolve também atividades de Consultor Autônomo.

Segurança no PHP

Os 6 requisitos mínimos

Apresentaremos neste artigo 6 requisitos que todo o desenvolvedor PHP deveria contemplar em sua aplicação. São boas práticas e hábitos simples que implementam um nível mínimo de segurança em qualquer sistema ou ferramenta desenvolvida com a linguagem.

A linguagem PHP é, sem dúvida, uma das mais populares quando o assunto é desenvolvimento de aplicações web. Existe, porém, um preconceito muito grande com a linguagem quando a questão é segurança. Neste artigo veremos 6 requisitos básicos que toda aplicação deveria possuir e que implementarão o mínimo de segurança em tudo o que você desenvolver.

Conceito: PHP é uma linguagem mais vulnerável do que as outras?

Atualmente impera no mercado de desenvolvimento um preconceito relacionado à segurança de aplicações PHP. A linguagem é frequentemente alvo de duras críticas por parte da própria comunidade de desenvolvimento, e não é raro presenciarmos aplicações extremamente vulneráveis que, com toda a certeza, dão razão a estas críticas.

Este preconceito, totalmente equivocado, tem suas origens especialmente na extrema flexibilidade de configuração e uso da linguagem e no número cada vez maior de desenvolvedores inexperientes que começam sua carreira desenvolvendo aplicações PHP, ignorando questões básicas, seja por falta de conhecimento ou porque estão procurando agilizar a sua produção pessoal.

Este artigo não é destinado apenas aos que estão começando. É surpreendente o número de desenvolvedores mais experientes que, por terem adquirido vícios, ainda ignoram as questões que apresentaremos aqui.

Requisito #1: Esqueça *register_globals*!

Register_globals é sem dúvida alguma a diretiva de configuração mais popular e polêmica já im-

plementada no PHP. É popular entre os desenvolvedores por tornar o processo de programação muito mais ágil e prático. É polêmica porque retira tanto do programador quanto do interpretador da linguagem a responsabilidade em definir a origem das informações utilizadas pela aplicação.

Esta diretiva causou tantos problemas que começou a vir desabilitada por *default* a partir da versão 4.2.0 e **será eliminada na versão 6**. Por isso, se você ainda programa com *register_globals* ligada, está mais do que na hora de mudar.

Observe o seguinte exemplo de código:

```
<?php
$url = "http://www.meusite.com.br/index.php";
$errorMsg = "Usuário não autenticado" ;
if (!$autenticado) {
    header("Location: $url?erro=$errorMsg");
} else {
    /* A aplicação age como se o usuário já
    estivesse conectado */
}
?>
```

Este código verifica por uma variável booleana chamada *\$autenticado*, que tipicamente viria de uma sessão aberta quando o usuário se autenticou em nossa aplicação ou mesmo de um cookie. Ao utilizarmos os recursos da *register_globals*, porém, não estamos dizendo ao interpretador PHP que este dado deve vir obrigatoriamente de uma destas duas fontes.

Sendo assim, o interpretador procurará por esta variável em diversas fontes das quais ele obteve informações (veja a Figura 1) até encontrá-la, despre-

zando se esta fonte era a correta.



Figura 1 – *register_globals*: PHP importa variáveis de diversas fontes

Assim sendo, torna-se extremamente simples “enganar” a sua aplicação PHP, basta que eu chame o seu arquivo passando a variável esperada por *query string*:

```
http://www.meusite.com.br/script.php?autenticado=1
```

Ao receber a *query string* contendo a informação “autenticado”, este dado será automaticamente interpretado como uma variável pelo interpretador, quer exista sessões ou cookies ou não.

Uma dúvida freqüente é “Se não tenho autorização para desligar a *register_globals* o que eu posso fazer?”. A resposta para isso é tão simples quanto tudo o mais na linguagem: **programe como se ela estivesse desligada!** Veja como fica o nosso código ao não utilizarmos a *register_globals*:

```
<?php
$url = "http://www.meusite.com.br/index.php";
$errMsg = "Usuário%20não%20autenticado" ;
if (!$_SESSION['autenticado']) {
    header("Location: $url?erro=$errMsg");
} else {
    /* A aplicação age como se o usuário já estivesse conectado */
}
?>
```

Observe a mudança: ao utilizarmos o array super global `$_SESSION`, estamos explicitando para o interpretador que a informação “autenticado” tem que, obrigatoriamente, vir de uma sessão. Isto significa que será impossível para o interpretador confundir este dado com um dado enviado pela *query string*, pois este fica armazenado em um array super global diferente:

<code>\$_SERVER</code>	Informações definidas pelo servidor web (Apache, IIS, etc)
<code>\$_GET</code>	Informações passadas por <i>query string</i> ou por um formulário que usa o método GET.
<code>\$_POST</code>	Informações enviadas, tipicamente, por um formulário web utilizando o método POST.
<code>\$_COOKIE</code>	Informações fornecidas por um cookie HTTP.
<code>\$_FILES</code>	Informações fornecidas por arquivos que foram enviados via upload HTTP.
<code>\$_ENV</code>	Informações fornecidas pelo ambiente do servidor.
<code>\$_REQUEST*</code>	Informações fornecidas por GET, POST ou COOKIE.

Tabela 1 – Os arrays super globais do PHP

*** Cuidado:** O uso do array super global `$_REQUEST` é quase tão perigoso quanto a *register_globals* em si, pois nesse caso a informação pode vir de qualquer um dos 3 métodos: GET, POST ou COOKIE.

Requisito #2: Use require e não include

O comando *include* e seus derivados – como *include_once* – é frequentemente usado em detrimento do comando *require*. Este fato curioso a princípio é consequência da similaridade do comando *include* com outras linguagens, como C e Java.

O que muita gente não sabe, porém, é que este comando carrega consigo um risco muito grande para sua aplicação: se por algum motivo a inclusão do arquivo falhar – erro de digitação, disco corrompido, etc... – será gerado um erro de nível *warning*, um nível leve de erro que não causa a parada da execução do script.

Utilizando o comando *require*, garantimos que, no caso de falha na carga do arquivo seja gerado, além do erro de nível *warning*, um erro de nível *fatal*. Em bom português isso significa que a execução do seu script será imediatamente interrompida neste caso. Por que é bom que seu script seja interrompido no caso de falha de inclusão de outro arquivo? Simples: no caso de falhas, as variáveis, as funções, as constantes e o que mais este arquivo carregava consigo, simplesmente, não estarão disponíveis.

Requisito #3: Filtre a entrada de dados!

Um dos principais conceitos de segurança é a filtrar os dados que são recebidos antes de utilizá-los. A falta de filtragem é a causa inúmeros problemas de segurança, incluindo o infame *SQL Injecti-*

Observe o seguinte código:

O problema deste código é utilizar um dado informado pela *query string* (uid) sem fazer nenhuma filtragem, ou seja, nosso script aceitará cegamente qualquer coisa que for digitada como valor de uid.

Após a conversão dos caracteres especiais desta URL o valor de uid será (sem as aspas duplas): “198 or ‘x’=’x’”. Note que, quando concatenarmos isto à nossa *query*, ela terá sofrido uma grande modificação:

É por isso que a chamamos de “Injeção de SQL”. A parte em vermelho acima é formada de SQL válido, mas completamente alienígena à nossa aplicação: **ele modifica o comportamento original da query.**

Para evitarmos este tipo de problema, devemos aplicar um tratamento à informação **antes** de utilizá-la. Caso a minha *query* – como neste exemplo – esteja esperando um número inteiro, a solução é simples: forçamos a conversão do dado recebido para número inteiro:

A partir de agora, qualquer dado enviado pela *query string* será primeiro convertido para inteiro e somente depois desta conversão é que o utilizaremos. No caso de uma tentativa de Injeção de SQL, nosso script considerará apenas a parte numérica (198), como deveria ser. Além disso, se por um acaso nosso script receber apenas texto, a função *settype* converterá isto para o número 1, ou seja, cadastre um usuário falso no registro de *id* 1 da sua base.

```
<?php
$sql = "SELECT nome, sobrenome FROM usuarios
WHERE email = '" . $_GET['email'] . "'";
?>
```

Sabemos que um endereço de e-mail é formado por um nome de usuário, uma arroba e um domínio. Para fins de simplificação do exemplo, trabalharemos como se o usuário só pudesse usar letras, números e o caractere de ponto – para quem quiser levar a validação de e-mail a sério consulte a RFC 822. Poderíamos representar isso através de uma expressão regular:

Sendo assim, só aceitaremos o dado de endereço de e-mail se ele estiver dentro dos padrões de minha expressão regular. Nosso código então ficaria assim:

```
<?php
require_once("conecta.php");
if (preg_match('/^[A-Z|a-z|0-9|\.|_|+|@|[A-Z|a-z|0-9|\.|_|+|@]]+$/', $_GET['email'])) {
    $conn = conecta();
    $sql = "SELECT nome, sobrenome FROM
usuarios WHERE email='" . $_GET['email'] .
"\"";
    $recordSet = mysql_query($sql);
    while ($record = mysql_fetch_assoc
($recordSet)) {
        echo $record['email'] . '<br />';
    }
    mysql_close();
} else {
    die("O dado informado não é um endereço de
e-mail válido.");
}
?>
```

Requisito #4: Sem erros para o usuário

Falaremos aqui de outro ponto importantíssimo na questão de segurança que é frequentemente ignorado pelos desenvolvedores: as mensagens de erro.

As mensagens de erro foram feitas para que o desenvolvedor possa trabalhar de forma mais prática e descobrir o que ele está fazendo de errado. Observe, entretanto, que, quando uma aplicação atinge maturidade suficiente para “entrar em produção”, torna-se imperativo que o usuário não visualize mensagens de erro.

A razão disso é muito simples: mensagens de erro frequentemente trazem informações sensíveis. Observe como exemplo uma típica mensagem de falha de conexão com a base de dados:

```
Warning: mysql_connect() [function.mysql-
connect]:
Access denied for user 'foo'@'localhost'
(using password: YES)
in /usr/local/apache/htdocs/script.php on line
2
```

Note, através das partes destacadas, que esta mensagem me informa:

- 1.O tipo de RDBMS: mysql
- 2.O usuário de conexão com a base: foo
- 3.Que este usuário está tentando uma conexão de dentro do servidor: localhost
- 4.O caminho absoluto da raiz web: /usr/local/apache/htdocs
- 5.O nome do script: script.php

É por isso que, quando colocamos uma aplicação em produção, ocultamos as mensagens de erro e as gravamos diretamente em um arquivo de log. Isto é muito simples de implementar:

- 1.Desative a diretiva de configuração display_errors
- 2.Ative a diretiva log_errors
- 3.Indique o caminho absoluto do arquivo de log na diretiva error_log

Exemplo:

```
display_errors = Off
log_errors = On
error_log = /log/php_errors.log
```

Requisito #5: Esconda do servidor web o que ele não precisa acessar

Quanto de nós não usamos em nossas aplicações um arquivo, tipicamente chamado de config.php ou setup.php onde guardamos, por exemplo, usuário e senha da base de dados?

Não há nada de errado nisso, mas cuidado: se este arquivo não gera saída de informação em HTML porque deixá-lo acessível via web?

Para esclarecermos o problema vamos definir 3 coisas:

- 1) minha raiz web é: /usr/local/apache/htdocs
- 2) meu arquivo de configuração fica localizado em /usr/local/apache/htdocs/config/config.php
- 3) este é um código típico que usa este arquivo:

```
<?php
require_once("/usr/local/apache/htdocs/
config/config.php");
/* blah blah blah */
?>
```

Observe que o erro neste caso é confundir interpretador PHP e servidor web. O Apache (ou IIS, Xitami, etc) não precisa saber onde está este arquivo. Isto é responsabilidade do interpretador.

Outro problema é que seu arquivo fica desnecessariamente exposto, afinal de contas, basta eu abrir um *browser* e digitar:

<http://www.meusite.com.br/config/config.php>

Não faz sentido. Por mais que você possa argumentar que não há forma de ler o código fonte deste arquivo, ainda assim, isto não muda o fato de que ele está desnecessariamente exposto.

A solução é simples: movemos o arquivo para fora da raiz web:

```
/usr/local/config/config.php
```

E depois apenas acertamos as permissões e corrigimos nosso código:

```
<?php
require_once("/usr/local/config/config.php");
/* blah blah blah */
?>
```

A partir de agora nosso arquivo de configuração só pode ser lido por quem precisa dele: o interpretador PHP.

Requisito #6: Use Criptografia

Dados sigilosos são chamados assim por um motivo. Quando tratamos especificamente de senhas é impressionante a quantidade de aplicações web que grava senhas em texto puro na base de dados.

Ora, se a senha possui a importância que tem e quem a escolhe é o usuário, por que alguém mais precisa ler essa senha?

Se a senha possui este peso em nossa aplicação, não podemos nos dar ao luxo de fazer com que ela trafegue pela aplicação totalmente exposta.

PHP implementa criptografia de várias formas, mas eu sugiro – para quem puder usar, pois exige instalação e configuração extra – a utilização da função `mcrypt`.

Os *hashes* MD5 e SHA-1 são opções válidas, mas o problema é sua fragilidade: para o MD5 já existe até dicionário de dados, enquanto o SHA-1 foi recentemente “quebrado” por um pesquisador chinês.

Conclusão

A implementação de regras básicas de segurança depende apenas da boa vontade do desenvolvedor. Você deve ter percebido como a maior parte das soluções aqui apresentadas significam simples mudanças de hábito.

É mais do que tempo de nós, desenvolvedores PHP, deixarmos de lado nossos antigos vícios e começarmos a implementar boas práticas de programação.

Ganha o profissional, ganha a sua aplicação, ganha o seu cliente, ganha o mercado. Só quem perde nessa situação é quem tenta se aproveitar de nossa aplicação.

Referências e links sugeridos

[PHP Security Consortium] – <http://phpsec.org/>

[PHP RS] – <http://www.phprs.com.br/>

Er Galvão Abbott - galvao@galvao.eti.br

Er Galvão Abbott trabalha há mais de dez anos com programação de websites e sistemas corporativos com interface web.

Autodidata, teve seu primeiro contato com a linguagem HTML em 1995, quando a internet estreava no Brasil.

Além de lecionar em cursos, escrever artigos e ministrar palestras, tem se dedicado ao desenvolvimento de aplicações web, tendo nas linguagens PHP, Perl e JavaScript suas principais paixões.

É fundador e líder do UG PHPRS, além de trabalhar como consultor e desenvolvedor para uma empresa norte-americana da área de segurança.



Amplie os horizontes da sua empresa
Divulgue sua empresa e produtos para a comunidade PHP do Brasil.
Conheça os planos para anunciar no portal e na revista.
Solicite um orçamento através de comercial@phpmagazine.com.br.

8º Fórum Internacional de Software Livre reúne 5.363 participantes

A Associação Softwarelivre.Org realizou de 12 a 14 de abril o 8º Fórum Internacional de Software Livre (FISL 8.0), em Porto Alegre. Neste ano o FISL contou



com a participação de 5.363 pessoas e, como nas edições anteriores, diversas caravanas se organizaram para o evento. Muitos grupos de usuários também se reuniram para expor suas atividades para o público participante. Neste ano o fórum contou com oficinas de alto nível com vagas limitadas. Os interessados em participar das oficinas deveriam se cadastrar antes pelo site do evento para garantir sua vaga.

A edição 2007 do FISL 8.0 integrou o Workshop de Software Livre (WSL). O WSL é um evento científico apoiado pela Sociedade Brasileira de Computação, e, desde a sua primeira edição, se caracteriza por reunir professores, pesquisadores, estudantes e profissionais para a apresentação de trabalhos de pesquisa e desenvolvimento relacionados a software livre.

Todos os anos, o FISL 8.0 cede espaço gratuitamente para que grupos de usuários (nacionais e internacionais) das mais diversas linguagens, tecnologias, sistemas operacionais e outras correntes tecnológicas relacionadas ao software livre. A organização tem como objetivo permitir que os grupos de usuários mostrem o seu trabalho, troquem experiências com outros grupos, pessoas e instituições, visando promover uma maior integração entre todos que estão ligados ao software livre.

A organização do evento tem dois objetivos desde a sua primeira edição, são eles:

- **Objetivos gerais:** Reunir a comunidade nacional e internacional em desenvolvimento e aplicação de software livre, para o compartilhamento de experiências e conhecimento, de modo a estimular o seu uso crescente, o aprimoramento de tecnologias, o ecossistema de

negócios em software livre e a difusão da filosofia do compartilhamento e da criação colaborativa.

- **Objetivos específicos:** Promover e estimular o intercâmbio de experiências entre as diversas iniciativas por meio de soluções com software livre e de código aberto, difundindo seu caráter de alternativa tecnológica viável; promover o uso e o desenvolvimento de software livre e de código aberto, como uma alternativa econômica, com custos efetivamente menores do que os ofertados tradicionalmente pelas soluções proprietárias; fortalecer as políticas públicas do Governo Federal de adoção de software livre e de código aberto; estimular o uso do software livre e de código aberto, como base de programas de inclusão digital e fomentar a adoção de negócios em tecnologia da informação baseado em software livre.

Um outro objetivo da organização é atingir o maior número de participantes e atingir acadêmicos, pesquisadores, comunidade de desenvolvedores, estudantes das diversas áreas, empresários, profissionais e técnicos do setor; diretores e técnicos da área governamental; usuários de software livre, público em geral interessado na área de informática, nos impactos sociais das tecnologias de informação e comunicação e, na discussão da liberdade do conhecimento nas suas diversas formas.

Palestras sobre PHP no evento

Como em 2006, o FISL 8.0 fez homenagens aos “pais” da Ciência da Computação colocando seus nomes como nome de cada sala de palestra. Cada um dos homenageados escolhidos tem sua participação e importância em diversas áreas. Ao todo, foram 249 palestrantes. A PHP Magazine elaborou para você uma síntese das principais palestras sobre PHP no evento. Não deixe de conferir adiante.

1ª Palestra: GeoSUAS: Uso de softwarelivre para a divulgação de informações da assistência so-

cial brasileira utilizando PHP, TERRALIB, POST-GRE

Palestrantes: Gilvan de Barros Coelho Júnior, Fabiana Fernandes Santos e Gustavo Noronha Silva

A palestra tratou do aplicativo GeoSuas do Ministério do Desenvolvimento Social e Combate a Fome, que disponibiliza na internet informações da assistência social brasileira, em um ambiente de suporte a decisão geo-referenciado, completamente desenvolvido e suportado com componentes de software livre. A palestra abordou, entre outros aspectos da aplicação, o uso de bancos de dados PostGre em ambientes de suporte a decisão e da biblioteca de geo-referenciamento Terralib.

2ª Palestra: PHP & Segurança: Uma união possível

Palestrante: Er Galvão Abbott

O palestrante expôs as práticas básicas, mas frequentemente ignoradas, para tornar sua aplicação

PHP mais segura.

3ª Palestra: Desenvolva rapidamente utilizando o framework CakePHP

Palestrante: Elton Luís Minetto

Nesta palestra, o expositor demonstrou os principais conceitos e vantagens de se programar usando o framework CakePHP. Demonstrou também a sua estrutura, configurações necessárias para instalação e uso apropriado. Para exemplificar seu uso desenvolveu uma pequena aplicação com acesso a banco de dados que também será customizada para demonstrar a produtividade da ferramenta.

Fonte: Sergio Stock (Assessoria de Imprensa Fisl)

Colaboradores: Er Galvão Abbott e Grupo PHPrs



Amplie os horizontes da sua empresa
Divulgue sua empresa e produtos para a comunidade PHP do Brasil.
Conheça os planos para anunciar no portal e na revista.
Solicite um orçamento através de comercial@phpmagazine.com.br.

1º. PHP Road Show supera expectativas

O primeiro evento da comunidade PHP de Brasília foi um sucesso. Ao todo foram 257 participantes. O evento marca o início de uma série de encontros, visando desenvolver os profissionais e estudantes da cidade.

Com o objetivo de promover o desenvolvimento e a integração entre empresas, órgãos e profissionais da área, Brasília recebeu, no dia 19 de maio, o 1º PHPDF Road Show. Em sua primeira edição, foram 257 participantes e 200 quilos de alimentos arrecadados que foram entregues à ONG Gente Nova.



Organizado pela Comunidade PHP do Distrito Federal (PHP-DF) e pelo Instituto de Desenvolvimento de Software Livre (IDSL), este evento estreou o ciclo de palestras da comunidade PHP do Distrito Federal.

1. Filas para mini-cursos e palestras

Com um total de nove temas diferentes de mini-cursos, este evento foi marcado pelo interesse dos participantes que, muitas vezes, enfrentaram filas para participar das turmas. O maior exemplo de lotação foi a turma de Introdução ao PHP. Segundo a organização, a grade contemplava somente 2 mini-cursos. Devido à demanda, abriu-se uma turma extra que foi ministrada na hora do almoço. Outros participantes também ajustaram seu horário de almoço para participar das turmas extras, como por exemplo as de PHP com AJAX e XAJAX.

A maioria das turmas permaneceu cheia, porém, as turmas avançadas não tiveram muito público, mostrando que a grande busca ainda está voltada para os passos iniciais da linguagem. “Não esperávamos essa procura”, conta Adler Medrado que faz parte da equipe de organização do evento. Já outro

integrante, Rafael Dohms, diz que “é natural que tenhamos tantos interessados no nível elementar da linguagem, uma vez que o próprio evento está ligado a instituições de ensino superior”. Pablo Sanchez, também integrante da organização, ficou muito entusiasmado com a procura e afirmou que “é necessário dar continuidade aos mini-cursos dentro da filosofia do Road Show que é disseminar e expandir o conhecimento da linguagem e fortalecer nossa comunidade”.

Para ministrar cada um dos 17 mini-cursos, foram convidados palestrantes de renome como Pablo Dall'Oglio, diretor de tecnologia e proprietário da Adianti Solutions e autor do livro *PHP-GTK*, e Manuel Lemos, gerente de desenvolvimento de sistemas de software, proprietário da PHP Classes e ícone da comunidade do PHP no Brasil.



A turma de Introdução ao PHP precisou de terceiro turno no almoço

2. Um embrião de pais importantes

O evento já nasceu com parcerias importantes, tais como a X25 Treinamento e Consultoria e a Netmake, empresa criadora do framework e ambiente

de desenvolvimento PHP ScriptCase, além da FACNET (Faculdades NDA) que, com o sucesso do evento, já pensa em desenvolver um curso de extensão envolvendo os aspectos de desenvolvimento da linguagem. Grandes empresas, como Adobe, Blue Star Technology, Hostnet, O'Reilly e Novatec, tam-

bém apoiaram e sortearam brindes durante todo o evento: Livros, camisetas, kits e até um curso foram entregues aos participantes pela organização do evento e pelos patrocinadores. A PHP Magazine também esteve presente no evento apresentando a revista e sortecendo alguns brindes.



Cleber Sousa - cleber2506@gmail.com

Analista de Sistemas e Desenvolvedor Web há 10 anos, atua desde 2004 com tecnologias livres entre elas: PHP, JSP, MySQL, Postgres e MS SQL.

Trabalhou em grandes empresas como Caixa Seguros, ANTT e Brasil Telecom. Atualmente presta serviços de consultoria para a CTIS. Desenvolve também atividades de Consultor Autônomo.