MANUAL DE PADRONIZAÇÃO DE CODIFICAÇÃO
PHP CODING STANDARDS

versão 1.0

W7BR – Soluções e Tecnologia

contato@w7br.com

Sumário

INTRODUÇÃO

· Como escrever um código organizado, bem estruturado e documentado Melhorando visibilidade e facilitando futuras manutenções e implementações.

· Um bom padrão de codificação é importante em qualquer projeto de desenvolvimento, principalmente quando envolve vários desenvolvedores.

· Assegurar a Alta Qualidade do código, diminuir bugs.

Observação

- Esse manual é baseado no **PHP Coding Standards** do *Walker de Alencar;*
- Alguns modificações foram realizadas se adequando à nossa realidade;
- Foram acrescentado outros itens que achamos importantes.
- Todos o resto foi mantido, assim como a formatação mais próxima existente no material base e sua licença.

FORMATO DO ARQUIVO PHP

Identação

- Tab com tamanho de 4 espaços.

Codificação do Arquivo

- UTF-8;

NOMENCLATURAS

Padrão CamelCase

- É a denominação em inglês para a prática de escrever palavras compostas ou frases, onde cada palavra é iniciada com Maiúsculas, e unida sem espaços.
- É um padrão largamente utilizado em diversas linguagens de programação, como Java, Ruby e Python, principalmente nas definições de Classes e Objetos. (Fonte: wikipedia.com)
- Divisões: lowerCamelCase (iPod, iPhone) e UpperCamelCase (OpenOffice, StarTrek)

UpperCamelCase

- Classes
  - O nome da classe deverá conter primeiramente o ***cls*** que identifica que é uma classe, seguido do nome Representativo da classe. O nome Representativo é definido sobre os objetivos e atividades que a classe exercerá, os mesmo devem seguir o padrão UpperCamelCase. Ex.: clsBancoDados; clsEnviarArquivo.

LowerCamelCase

- Variáveis
  - *$isRoot, $itensCarrinho*
- Propriedades
  - *$this->caminhoUpload, $this->nomeCompleto*
- Funções e Métodos
  - *this->CalculaFrete(), $this->GeraMiniatura(), ContaPalavras($frase)*

UPPERCASE

- Contantes
  - *ORM_SERVIDOR, BASE_UPLOAD, URL_API*

ESTILO DE CÓDIGO (Coding Style)

Demarcação de Código PHP

- Não usar short_tags(<? E <?=)
- Usar tags completas (<?php e <?php echo)

Strings Literais

- Se a string não contiver variáveis de substituição, deve-se usar aspas simples.
  - *$tmpStr = 'Exemplo de String';*

Strings Literais com Apóstrofos

- Pode-se usar aspas simples, mas é recomendado o uso de aspas duplas para evitar slashes [\]
  - *$tmpSql = "SELECT id, nome FROM cliente WHERE name='Walker'";*

Substituição de Variáveis e Concatenação de Strings

- Usar aspas simples.
- Usar espaço antes e depois do operador ".", melhorando assim a visibilidade.
  - *$tmpStr = 'Exemplo ' . $de . ' String com ' . $variavel;*
- Quando concatenar mais de uma string longa, alinhe o operador "." abaixo do operador "=".
  - *$tmpSql = 'SELECT id,nome '*
    . 'FROM cliente '
    . "WHERE name = 'Walker' ";

Classes

- Nomear em clsUpperCamelCase.
- As chaves "{" virá na linha frente do nome da classe e o "}" virão na linha abaixo do nome da Classe.
- Toda classe deve ter um bloco de documentação em conformidade com o Padrão do PHPDocumentor.
- Qualquer código dentro da classe precisa ser identado com um TAB.
- Só é permitida uma classe por arquivo PHP.

```
/**
* Envia E-mail conectando num servidor SMTP.
*
* @author  Cleyton Ferrari <cleyton@w7br.com>
* @copyright 2009 W7Br Soluções
* @license http://www.w7br.com
* @version 1.1
*/
class W7br_SMTP {
    /* Ensinar ao Mr. M como fazer Mágica aqui. */
}
```

Variáveis de Classes (Propriedades)

- Nomear em _lowerCamelCase.
- Devem ser declaradas no topo da classe, antes de qualquer declaração de métodos.
- Sempre declarar sua visibilidade: private, protected ou public.
- Preferencialmente não utilizar declaração de variáveis de classes como public, para incentivar o uso de (set/get)
- Se a propriedade for a chave primária ou chave estrangeira de uma entidade do banco de dados, devera ser adicionado um _id no final de sua nomenclatura, ex: _chavePrimaria_id; _chaveEstrangeira_id; _campoSimplesBD;

```
/**
* Envia E-mail conectando num servidor SMTP.
*
* @author  Cleyton Ferrari <cleyton@w7br.com>
* @copyright 2009 W7Br Soluções
* @license http://www.w7br.com
* @version 1.1
*/
class W7br_SMTP {
    /* Ensinar ao Mr. M como fazer Mágica aqui. */
    private $_servico_id;
    function setServico_id ($value) {
            $this->_servico_id = $value;
    }
    function getServico_id(){
            return $this->_servico_id;
    }
```

```
    private $_nome;
    function setNome ($value) {
            $this->_nome = $value;
    }
    function getNome(){
            return $this->_nome;
    }
}
```

Funções e/ou Métodos

- Nomear em UpperCamelCase.
- As chaves "{" virá na linha frente do nome da Função/Método e o "}" virão na linha abaixo do nome da Função/Método.
- Toda Função/Método deve ter um bloco de documentação em conformidade com o Padrão do PHPDocumentor.
- Qualquer código dentro da Função/Método precisa ser identado com um TAB.
- Sempre declarar a visibilidade: private, protect ou public.

```
/**
* Conta quantas palavras tem na frase e retorna a quantidade.
* @param string $frase texto que terá as palavras contadas
* @return int quantidade de palavras que o texto possui
*/
function contaPalavras( $frase ){
        /* Mágica aqui. */
}
```

Estruturas de Controle

- if / else / elseif
  - A chave "{" virá na linha da declaração da condição e a chave "}" virá na linha abaixo da última linha de conteúdo.
  - Qualquer código entre as chaves "{" e "}" precisa ser identado com um TAB.
  - Sempre deverá utilizar as “{” e “}” mesmo que a condição tenha somente uma linha.
  - O else e elseif deverão ficar logo após a chave de fechamento “}”

```
if( ( $root === true ) && ( $totalDinheiro >= 100 ) ){
        echo "Você é Rei. Ainda está rico, paga um churrasco?";
```

```
} elseif( ( $root === true ) && ( $totalDinheiro < 100 ) ){
        echo "Você é Rei. Mas está probrinho heim.";
} else {
        echo "É, que diaxo você é?";
}
```

- **Operador Ternário (?:)**
  - Só deverá ser usado para uma condição, para mais condições use if/elseif/else
  - Toda a expressão devera ficar somente e uma linha.

```
$a = $root == true ? "Você é Rei." : "Você não é Rei";
```

- **while/for/foreach**
  - A chave "{" virá na mesma linha da expressão, a chave "}" virá na linha abaixo da última linha de conteúdo.
  - Qualquer código entre as chaves "{" e "}" precisa ser identado com um TAB.

```
while(!feof($hFile)){
        // Qualquer conteúdo
        // precisa ser identado com [Tab|4 espaços].
}
```

```
for($intCount = 0; $intCount <= 10; $intCount++){
        // Qualquer conteúdo
        // precisa ser identado com [Tab|4 espços].
}
```

```
foreach($arrList as $mixKey => $mixValue){
        // Qualquer conteúdo
        // precisa ser identado com [Tab|4 espaços].
}
```

- **switch/case**
  - Sempre deverá ter o default
  - A chave "{" virá na linha da expressão, a chave "}" virá na linha abaixo da última linha de conteúdo;
  - Qualquer código entre as chaves "{" e "}" precisa ser identado com um TAB;
  - Qualquer código dentro de: case e default, precisa ser identado com um TAB, inclusive a palavra reservada: break.

```
switch($intNivel){
        case 1:
                // Qualquer conteúdo
                // precisa ser identado com [Tab|4 espaços].
                break;
        case 2:
                // Qualquer conteúdo
                // precisa ser identado com [Tab|4 espaços].
                break;
        default:
                // Qualquer conteúdo
                // precisa ser identado com [Tab|4 espaços].
}
```

Array

- Quando os índices não forem numéricos, deverá seguir o padrão lowerCamelCase
  - `$equipe = array( "silasRibas" => "Silas Ribas Martins", "rodrigoAlves" => "Rodrigo Alves" );`
- **Include e Require**
  - Usar include_once ou require_once somente
  - Deverá conter parênteses na chamada
    - `require_once ("SkySoft/SMTP.php");`

Erros

- Deverão ser expostos somente em desenvolvimento, após o desenvolvimento na versão final deverão ser usado do **error_reporting(0)**;
- Usar Exceções exaustivamente

REFERÊNCIAS

- **Appendix B. Zend Framework Coding Standard for PHP**
  - http://framework.zend.com/manual/en/coding-standard.html
- **PEAR Manual**
  - http://pear.php.net/manual/en/standards.classdef.php
- **CamelCase**
  - http://pt.wikipedia.org/wiki/CamelCase
- **PHP Coding Standards (Walker de Alencar)**
  - http://blog.walkeralencar.com/archives/11