

MANUAL DE PADRONIZAÇÃO DE CODIFICAÇÃO PHP CODING STANDARDS

versão 0.1

SkySoft - Silas Ribas Martins

silasrm@gmail.com

Walker de Alencar Oliverira

contato@walkeralencar.com

Este conteúdo está sob licença **Creative Commons: BY-NC-SA**. Reporte falhas e dê sugestões, toda ajuda será devidamente referenciada.

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;
 - Formato Unix (final de linha em (LF) - [\n])

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 nome do Autor (deverá ser curto, não composto e não deverá conter caracteres especiais) seguido do nome Representativo da classe, sendo agrupados usando o "_" (underscore). O nome Representativo é definido sobre os objetivos e atividades que a classe exercerá, os mesmo devem seguir o padrão UpperCamelCase.
 - `Skysoft_EmailSmtp`, `Skysoft_Upload`
- **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 [\]
 - `$tmpSql = "SELECT id, nome FROM cliente WHERE name='Walker'";`
- **Substituição de Variáveis e Concatenação de Strings**
 - Usar aspas duplas.
 - Usar espaço antes e depois do operador ".", melhorando assim a visibilidade.
 - `$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 UpperCamelCase.
 - As chaves "{" e "}" 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 indentado com quatro espaços.
 - Só é permitida uma classe por arquivo PHP.
 - ```
/**
 * Envia E-mail conectando num servidor SMTP.
 *
 * @author Silas Ribas <silas@skysoft.com.br>
 * @copyright 2009 SkySoft Brasil
```

```

* @license http://www.skysoft.com.br/licenca/SSLL1.0.txt SkySoft Libr
ary License 1.0
* @version 1.1
* @category MAIL
* @package SkySoft_SMTTP
*/

class SkySoft_SMTTP
{
 /* 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)

- **Funções e/ou Métodos**

- Nomear em lowerCamelCase.
- As chaves "{" e "}" 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 indentado com quatro espaços.
- Sempre declarar a visibilidade: private, protect ou public.
- Na chamada, se a quantidade de parâmetros for maior que 4, ou se os parâmetros forem longos, deverá quebrar em 1 parâmetro por linha e a vírgula deverá está alinhada com "(".

- ```

/**
 * Conta quantas palavras tem na frase e retorna a quantidade.
 *
 * @author Silas Ribas <silas@skysoft.com.br>
 * @copyright 2009 SkySoft Brasil
 * @license http://www.skysoft.com.br/licenca/SSLL1.0.txt SkySoft Libr
ary License 1.0

```

```

* @param      string $frase texto que terÃi as palavras contadas
* @return     int quantidade de palavras que o texto possui
*/

```

```

function contaPalavras( $frase )
{
    /* Mágica aqui. */
}

```

- `criaFigura($comprimento`
 `, $altura`
 `, $corFundo`
 `, $corFigura`
 `, $borda);`

• Estruturas de Controle

◦ if / else / elseif

- Usar espaço simples depois do "(" e antes do ")" da condição no IF e no ELSEIF
- A chave "{" virá na linha abaixo da condição e a chave "}" virá na linha abaixo da última linha de conteúdo.
- Qualquer código entre as chaves "{" e "}" precisa ser indentado com quatro espaços.
- Se existe duas ou mais condições, as condições deverão estar uma por linha, iniciando a linha, da segunda condição em diante, com o comparador lógico seguido da condição, entre parênteses.

```

• 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 diabo você é?";
}

```

```
}
```

- **Operador Ternário (?:)**

- Deverá sempre ser usado fora de qualquer bloco de conteúdo
- Os resultados devem está uma linha abaixo da condição.
- Só deverá ser usado para uma condição, para mais condições use if/elseif/else
 - `$a = $root == true`
`? "Você é Rei." : "Você não é Rei";`

- **while/for/foreach**

- Usar espaço simples depois do "(" e antes do ")" da expressão.
- A chave "{" virá na linha abaixo da expressão, a chave "}" virá na linha abaixo da última linha de conteúdo.
- Qualquer código entre as chaves "{" e "}" precisa ser indentado com quatro espaços.

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

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

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

- **switch/case**

- Usar espaço simples depois do "(" e antes do ")"
- Sempre deverá ter o default
- A chave "{" virá na linha abaixo da Expressão, a chave "}" virá na linha abaixo da última linha de conteúdo

- Qualquer código entre as chaves "{" e "}" precisa ser indentado com quatro espaços
- Qualquer código dentro de: case e default, precisa ser indentado com quatro espaços, inclusive a palavra reservada: break.

```
• switch($intNivel)
 {
 case 1:
 // Qualquer conteúdo
 // precisa ser indentado com [Tab|4 espaços].
 break;
 case 2:
 // Qualquer conteúdo
 // precisa ser indentado com [Tab|4 espaços].
 break;
 default:
 // Qualquer conteúdo
 // precisa ser indentado 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
- Não deverá conter parênteses na chamada

- `require_once "SkySoft/SMTP.php";`

- **Erros**

- Deverão ser expostos somente em desenvolvimento
- Usar Exceções exaustivamente

## LICENÇA

- Este conteúdo está sob a licença Creative Commons: BY-NC-SA.
  - Sendo permitido:
    - copiar, distribuir, exibir e executar a obra.
    - criar obras derivadas.
  - Sob as seguintes condições:
    - Atribuição. Você deve dar crédito ao autor original, da forma especificada pelo autor ou licenciante.
    - Uso Não-Comercial. Você não pode utilizar esta obra com finalidades comerciais.
    - Compartilhamento pela mesma Licença. Se você alterar, transformar, ou criar outra obra com base nesta, você somente poderá distribuir a obra resultante sob uma licença idêntica a esta.
    - Para cada novo uso ou distribuição, você deve deixar claro para outros os termos da licença desta obra.
    - Qualquer uma destas condições pode ser renunciada, desde que Você obtenha permissão do autor.
    - Nada nesta licença prejudica ou restringe os direitos morais do autor.
  - Anexo:
    - Ao adotar este padrão em alguma instituição informar ao autor através do e-mail: [walkeralencar@gmail.com](mailto:walkeralencar@gmail.com) ou [contato@walkeralencar.com](mailto:contato@walkeralencar.com), para ser referenciado no site, isso engrandece o trabalho já realizado e incentiva continuidade de outros mais.
    - Favor informar também à Silas Ribas via [silasrm@gmail.com](mailto:silasrm@gmail.com)

## 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>