**Especificación para la documentación del código fuente en PHP de los proyectos de “*LedIt”.***

|  |  |  |  |
| --- | --- | --- | --- |
|  | ELABORÓ: | REVISÓ: | APROBÓ: |
| RESPONSABLE | Óscar David López | Miguel Angel Ríos  José Eduardo Soreque  Óscar López | Miguel Angel Ríos |
| CARGO | Documentador | Equipo de CQA | Equipo de Programadores |
| FECHA | 07/06/2018 | 16/06/2018 | 16/06/2018 |

# Introducción.

El siguiente documento es para presentar la forma en que se documentará la codificación del proyecto “Sitio Web con Comercio electrónico LedIt”.

El presente documento tiene por finalidad especificar la sintaxis básica de los comentarios requeridos para generar la documentación tentativa de código de un proyecto basado en el lenguaje de programación PHP.

Este estándar cumple las especificaciones de la herramienta phpDocumentor.

# Especificación general de los documentos compatibles con phpDocumentor.

## Los comentarios de documentación se enmarcan de la siguiente forma.

/\*\*

\*

\*

\*

\*/

Cualquier línea de comentario que no inicie con el carácter asterisco (“\*”) es ignorada.

## Un bloque de comentarios consta de la siguiente estructura.

/\*\*

\* Descripción corta (1 Línea)

\* Descripción extensa (2 líneas o más)

\* (cuatro espacios separados del asterisco) pueden agruparse con etiquetas <p>.

Las líneas de comentario extensas

**Otras etiquetas a utilizar**

<strong> negritas

<em> cursivas

<ul> numeración en forma de listado

<ol>

<code> mostrar código fuente

\*/

## El comentario que documenta una estructura se debe ubicar …

Se colocarán justo antes de la declaración de cada entidad.

Indicar los documentos susceptibles a esta documentación:

* Métodos

/\*\*  
 \* Bloque de Documentación aquí  
 \*/  
class Foo  
{  
    /\*\*  
     \* Bloque de Documentación aquí  
     \*/  
    public function bar()  
    {  
        // el contenido de la función  
        // debe separarse con cuatro espacios  
    }  
}

/\*\*  
 \* Bloque de Documentación aquí  
 \*/  
class Foo  
{  
    /\*\*  
     \* INCORRECTO  
     \*/  
    public function bar()  
    {  
        return($this->bar);  
    }  
  
    /\*\*  
     \* CORRECTO  
     \*/  
    public function bar()  
    {  
        return $this->bar;  
    }  
}

* Clases

/\*\*  
 \* Descripción corta de la clase  
 \*  
 \* Descripcion larga de la clase (si la hubiera)...  
 \*  
 \* @category   Zend  
 \* @package    Zend\_Magic  
 \* @subpackage Wand  
 \* @copyright  Copyright (c) 2005-2011 Zend Technologies USA Inc. (http://www.zend.com)  
 \* @license    http://framework.zend.com/license   BSD License  
 \* @version    Release: @package\_version@  
 \* @link       http://framework.zend.com/package/PackageName  
 \* @since      Class available since Release 1.5.0  
 \* @deprecated Class deprecated in Release 2.0.0  
 \*/

/\*\*  
 \* Bloque de Documentación aquí  
 \*/  
class SampleClass  
{  
    // el contenido de la clase  
    // debe separarse con cuatro espacios  
}

* Secciones
  + Sentencias include[\_once]
  + Sentencia require[\_once]
  + Sentencias define[\_once]
* Ciclos y bifurcadores->(bifurcar es dividir aquí entrarían if y while)

**SWITCH**

Las declaraciones de control escritas con la declaración "switch" deben tener un único espacio en blanco antes del paréntesis de apertura del condicional y después del paréntesis de cierre.

Todo contenido dentro de una declaración "switch" debe separarse usando cuatro espacios. El contenido dentro de cada declaración "case" debe separarse usando cuatro espacios adicionales.

* switch ($numPeople) {  
      case 1:  
          break;  
    
      case 2:  
          break;  
    
      default:  
          break;  
  }

La construcción **default** no debe omitirse nunca en una declaración **switch** .

**IF – ELSE IF**

Las sentencias de control basadas en las construcciones **if** y **elseif** deben tener un solo espacio en blanco antes del paréntesis de apertura del condicional y un solo espacio en blanco después del paréntesis de cierre.

Dentro de las sentencias condicionales entre paréntesis, los operadores deben separarse con espacios, por legibilidad. Se aconseja el uso de paréntesis internos para mejorar la agrupación lógica en expresiones condicionales más largas.

La llave de apertura "{" se escribe en la misma línea que la sentencia condicional. La llave de cierre "}" se escribe siempre en su propia línea. Cualquier contenido dentro de las llaves debe separarse con cuatro espacios en blanco.

if ($a != 2) {  
    $a = 2;  
}

Si la instrucción condicional hace que la longitud de la línea exceda la longitud máxima de la línea y tiene varias cláusulas, puede dividir la condicional en varias líneas. En tal caso, rompa la línea antes de un operador lógico, y rellene la línea de modo que se alinee bajo el primer carácter de la cláusula condicional. El paréntesis de cierre en el condicional se colocará en una línea con el paréntesis de apertura, con un espacio que los separa, en un nivel de sangrado equivalente al enunciado de control de apertura.

if (($a == $b)  
    &amp;&amp; ($b == $c)  
    || (Foo::CONST == $d)  
) {  
    $a = $d;  
}

La intención de este último formato de declaración es evitar problemas al agregar o quitar cláusulas del condicional durante las revisiones posteriores.

Para las declaraciones "if" que incluyan "elseif" o "else", las convenciones de formato son similares a la construcción "if". Los ejemplos siguientes demuestran el formato correcto para declaraciones "if" con construcciones "else" y/o "elseif":

if ($a != 2) {  
    $a = 2;  
} else {  
    $a = 7;  
}  
  
if ($a != 2) {  
    $a = 2;  
} elseif ($a == 3) {  
    $a = 4;  
} else {  
    $a = 7;  
}  
  
if (($a == $b)  
    &amp;&amp; ($b == $c)  
    || (Foo::CONST == $d)  
) {  
    $a = $d;  
} elseif (($a != $b)  
          || ($b != $c)  
) {  
    $a = $c;  
} else {  
    $a = $b;  
}

**PHP** permite escribir sentencias sin llaves -{}- en algunas circunstancias. Este estándar de código no hace ninguna diferenciación- toda sentencia "if", "elseif" o "else" debe usar llaves.

El uso de la construcción "elseif" está permitido, pero no se aconseja, en favor de la combinación "else if".

* Archivos

/\*\*  
 \* Descripción corta del fichero  
 \*  
 \* Descripción larga del fichero (si la hubiera)...  
 \*  
 \* LICENSE: Some license information  
 \*  
 \* @category   Zend  
 \* @package    Zend\_Magic  
 \* @subpackage Wand  
 \* @copyright  Copyright (c) 2005-2011 Zend Technologies USA Inc. (http://www.zend.com)  
 \* @license    http://framework.zend.com/license   BSD License  
 \* @version    $Id:$  
 \* @link       http://framework.zend.com/package/PackageName  
 \* @since      File available since Release 1.5.0  
\*/

## Es posible definir listas simples con líneas de comentarios que empiezan por

“-”, “°”, “+”, “#”

/\*\*

\* Bloque de código que requiere lista

\* @todo lo que vaya

\* - ítem 1

\* + ítem 1.1

\* # ítem 1.1.1

\*/

Clases

/\*\*  
 \* Descripción corta de la clase  
 \*  
 \* Descripcion larga de la clase (si la hubiera)...  
 \*  
 \* @category   Zend  
 \* @package    Zend\_Magic  
 \* @subpackage Wand  
 \* @copyright  Copyright (c) 2005-2011 Zend Technologies USA Inc. (http://www.zend.com)  
 \* @license    http://framework.zend.com/license   BSD License  
 \* @version    Release: @package\_version@  
 \* @link       http://framework.zend.com/package/PackageName  
 \* @since      Class available since Release 1.5.0  
 \* @deprecated Class deprecated in Release 2.0.0  
 \*/

## Las etiquetas son [obligatorias | opcionales] y proporcionan información adicional acerca de la entidad documentada.

* Son obligatorias para clases
* Para archivos

# Especificación práctica.

[Ejemplos de la codificación]

# Generación de la documentación.

Zend Framework es un framework de código abierto para desarrollar aplicaciones web y servicios web con PHP5. Zend Framework es una implementación que usa código 100% orientado a objetos. La estructura de los componentes de Zend Framework es algo único; cada componente está construido con una baja dependencia de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. A menudo se refiere a este tipo de diseño como "use-at-will" (uso a voluntad).

Aunque se pueden utilizar de forma individual, los componentes de la biblioteca estándar de Zend Framework conforman un potente y extensible framework de aplicaciones web al combinarse. Zend Framework ofrece un gran rendimiento y una robusta implementación MVC, una abstracción de base de datos fácil de usar, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos. Otros componentes, como Zend\_Auth y Zend\_Acl, proveen autentificación de usuarios y autorización diferentes a las tiendas de certificados comunes . También existen componentes que implementan bibliotecas de cliente para acceder de forma sencilla a los web services más populares. Cualesquiera que sean las necesidades de su solicitud, usted tiene todas las posibilidades de encontrar un componente de Zend Framework que se pueda utilizar para reducir drásticamente el tiempo de desarrollo, con una base completamente sólida.

El principal patrocinador del proyecto Zend Framework es Zend Technologies, pero muchas empresas han contribuido con componentes o características importantes para el marco. Empresas como Google, Microsoft y StrikeIron se han asociado con Zend para proporcionar interfaces de servicios web y otras tecnologías que desean poner a disposición de los desarrolladores de Zend Framework.

Zend Framework no podría haber proporcionado y apoyado todas estas características sin la ayuda de la vibrante comunidad de Zend Framework. Los miembros de la comunidad, incluidos los contribuyentes, están disponibles en las, listas de correo, canales de IRC, y en otros foros. Cualquier duda que tenga acerca de Zend Framework, la comunidad está siempre disponible para responder.

## Instalación.

Véase el apéndice requisitos para una lista detallada de requisitos para Zend Framework.

La instalación del Zend Framework es muy simple. Una vez que haya descargado y descomprimido el framework, deberá añadir la carpeta /library de la distribución al principio de su "include path". También puede mover la carpeta "library" a cualquier otra posición (compartida o no) de su sistema de archivos.

Descargar la última versión estable. Esta versión esta disponible en formatos .zip . .tar.gz , es una buena opción para aquellos que comienzan o son nuevos en Zend Framework.

Descargue la última instantánea nocturna. Para aquellos que se atreven a la vanguardia, las instantáneas nocturnas representan el último avance del desarrollo de Zend Framework. Las instantáneas se incluyen con documentación en inglés solamente o en todos los idiomas disponibles. Si prevé trabajar con los últimos desarrollos de Zend Framework, considere usar un cliente de Subversion (SVN).

Usando un cliente de Subversión (SVN). Zend Framework es un software de código abierto, y el repositorio de Subversión utilizado para su desarrollo está a disposición del público. Considere usar SVN para obtener Zend Framework si ya usa SVN para el desarrollo de su aplicación, desea contribuir de nuevo al marco, o necesita actualizar su versión de framework con más frecuencia de lo que ocurre.

La exportación es útil si desea obtener una revisión de marco particular sin los directorios .svn creados en una copia de trabajo.

Eche un vistazo a una copia de trabajo si desea contribuir con Zend Framework, una copia de trabajo puede actualizarse en cualquier momento con la actualización svn y los cambios pueden ser enviados a nuestro repositorio SVN con el comando svn commit.

Una definición externa es muy conveniente para los desarrolladores que ya usan SVN para administrar las copias de trabajo de su aplicación.

La URL para el trunk del repositorio SVN de Zend Framework es:

http://framework.zend.com/svn/framework/standard/trunk

Una vez que tenga disponible una copia de Zend Framework, su aplicación necesita poder acceder a las clases del framework. Aunque hay diferentes maneras de lograr esto , su include\_path de PHP necesita contener una ruta a la librería de Zend Framework.

Zend proporciona un QuickStart para ponerlo en funcionamiento lo más rápido posible. Esta es una excelente manera de comenzar a aprender sobre el marco con énfasis en ejemplos del mundo real sobre los que puede basarse.

Ya que los componentes de Zend Framework están débilmente conectados, tiene la opción de usar cualquier combinación de ellos en sus aplicaciones. Los siguientes capítulos presentan una referencia exhaustiva de Zend Framework, componente a componente.

## Ejecución para generar la documentación.

# 5. Enlaces asociados.

<http://manual.zfdes.com/es/doc-standard.file-formatting.html>

<http://manual.zfdes.com/es/introduction.installation.html>

<https://phpdoc.org/>

<http://manual.zfdes.com/es/index.html>