**Especificación para la documentación del código fuente en PHP de los proyectos de Mueblería Casa Morales**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **ELABORO:** | **REVISÓ:** | **APROBÓ:** |
| **RESPONSABLE** | Ernesto Rico Gutiérrez | Marco Alejandro Trenado Herrera | Gilberto Sánchez Castro |
| **CARGO** | Programador | Encargado de Pruebas | Líder del proyecto |
| **FECHA** | 12.06.2019 | 21.06.2019 | 21.06.2019 |

# Introducción.

Documentar el código de un programa es añadir suficiente información como para explicar lo que hace, de forma que no sólo los ordenadores sepan qué hacer, sino que además los humanos entiendan qué están haciendo y por qué.

Documentar un programa no es sólo un acto de buen hacer del programador, ya que es una necesidad que sólo se aprecia en su debida magnitud cuando hay errores que reparar o hay que extender el programa.

Todo programa que tenga éxito será modificado en el futuro, bien por el programador original, bien por otro programador que lo sustituya. Pensando en esta revisión de código es por lo que es importante que el programa se entienda: para poder repararlo y modificarlo.

# Especificación general de los documentos compatibles con phpDocumentor.

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

<!--Se obtinen todas las promociones por medio del foreach-->

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

<?php

/\*\*

\* Página de comentarios

\* En esta página se enviarán los comentarios de parte del cliente

\*

\* @author Ernesto Rico Gutiérrez

\* @package application/views/

\*

\* @version 1.0.0

\* Creado ‎viernes, ‎7‎ de ‎junio‎ de ‎2019, ‏‎01:39 pm

\* Última Modificación el 21/06/2019 a las 12:08 pm

\*/

?>

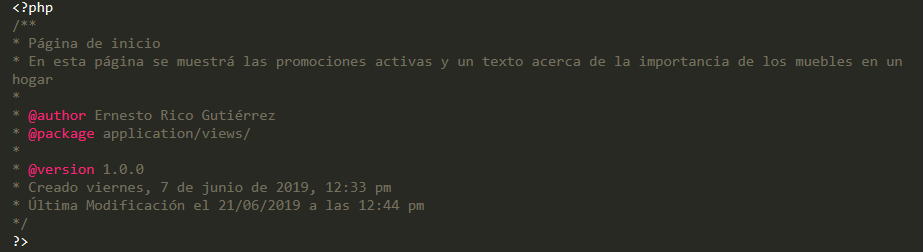
## El comentario que documenta una estructura se debe ubicar …

En la parte superior de la página y encima de los bloques php en el contenido de las views.

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

|  |  |  |
| --- | --- | --- |
| Etiquetas obligatorias | | |
| @author | | Autor del elemento con la posibilidad de incluir un elemento dentro de <>, esto acorde al gusto del programador. |
| @package | | Especifica el paquete que agrupa de manera documental las clases y funciones especificadas. |
| @version | | Versión del elemento |
| Etiquetas opcionales | | |
| @abstract | Especifica que una clase, un método o una variable son abstractos. | |
| @param tipo1|tipo2|... $nombre\_de\_variable descripción | Indica el tipo y el nombre del parámetro | |
| @static | La variable o la función es estática | |
| @link URL [texto] | Despliega un hiperenlace en la documentación. | |
| @example | Se utiliza para incluir una ejemplo | |

# Especificación práctica.



# Generación de la documentación.

## Instalación.

1. Descargue Composer desde  https://getcomposer.org/
2. Ejecutar a través de la instalación con la configuración predeterminada. Localizará php para usted automáticamente. Si tiene algún problema con la instalación, busque la respuesta en Google. Aunque no debería haber ningún error.
3. Si tiene CMD abierto, ciérrelo ahora.
4. Abre CMD y escribe compositor. Si algo empieza a suceder, Composer está instalado correctamente.
5. Vaya al directorio en el que desea instalar phpDocumentor .
6. En el directorio de su proyecto, cree un archivo JSON. Nómbrelo, composer.json y añádele el siguiente código

"require-dev": {

"phpdocumentor/phpdocumentor": "2.\*"

}

1. Por ejemplo, digamos que el directorio de su proyecto se encuentra en C: / wamp / www / your\_project , querrá abrir CMD y asegurarse de que haya navegado a ese directorio. Recuerda cd ../ para subir un directorio. Si está en C: /, escriba cd C: / wamp / www / your\_project para ingresar a ese directorio.
2. Ahora escriba instalar compositor
3. Esto llevará algún tiempo (generalmente menos de un par de minutos) pero instalará un nuevo directorio llamado proveedor
4. En CMD, escriba cd vendor / bin . Vas a estar en C: \ wamp \ www \ your\_project \ vendor \ bin
5. Ahora puedes escribir en phpdoc seguido de un comando
6. Si desea crear documentación para un solo archivo, usará -f . Si desea crear documentación para todo el directorio, usará -d .
7. Vas a querer seguirlo con una -t que indica a dónde irá la documentación. Cree una carpeta de documentos en C: \ wamp \ www \ your\_project \ docs .
8. El último segmento que tiene la opción de añadir es --template . Vaya a https://www.phpdoc.org/templates para obtener las plantillas disponibles. es decir, limpio
9. Por ejemplo, para crear documentación para un archivo, vaya a CMD y en el directorio bin, escriba phpdoc -f "C: \ wamp \ www \ your\_project \ classes \ some\_class.php" -t "C: \ wamp \ www \ your\_project \ docs "--template" responsive-twig "
10. Para crear documentación para todo el directorio, escriba lo siguiente:  phpdoc -d "C: \ wamp \ www \ su\_proyecto \ clases" -t "C: \ wamp \ www \ su\_proyecto \ docs" --template "responsive-twig"

## Ejecución para generar la documentación.

phpDocumentor permite generar la documentación de varias formas y en varios formatos.

**Opciones para generar documentación:**

1. Desde línea de comandos (php CLI - Command Line Interpreter)
2. Desde interfaz web (incluida en la distribución)
3. Desde código. Como phpDocumentor está desarrollado en PHP, podemos incluir su funcionalidad dentro de scritps propios.

**¿Qué hay que especificar?**

1. El directorio en el que se encuentra nuestro código. phpDocumentor se encarga de recorrer los subdirectorios de forma automática
2. Opcionalmente los paquetes (@pakage) que deseamos documentar, lista de ficheros incluidos y/o excluidos y otras opciones interesantes para personalizar la documentación.
3. El directorio en el que se generará la documentación
4. Si la documentación va a ser pública (sólo interfaz) o interna (en este caso aparecerán los bloques private y los comentarios @internal).
5. El formato de salida de la documentación

**Formatos de salida**

1. HTML a través de un buen número de plantillas predefinidas (podemos definir nuestra propia plantilla de salida)
2. PDF
3. XML (DocBook). Muy interesante puesto que a partir de este dialecto podemos transformar (XSLT) a cualquier otro utilizando nuestras propias reglas y hojas de estilo.

# Enlaces asociados.

<http://dinocajic.blogspot.com/2016/06/how-to-install-and-use-phpdocumentor.html>

<http://www.epsilon-eridani.com/cubic/ap/cubic.php/doc/phpDocumentor---documentacion-para-codigo-PHP-246.html>

<http://www.juntadeandalucia.es/servicios/madeja/sites/default/files/historico/1.4.0/contenido-libro-pautas-103.html>