

República Bolivariana de Venezuela
Ministerio del Poder Popular para la Educación Universitaria
Instituto Universitario Tecnológico “Dr. Federico Rivero Palacio”
Departamento de Informática

Manual de PHPDoc

Profesor:

Johan Alamo

C. I.: 16.148.116

ÍNDICE

PHPDoc	1
¿Qué es PHPDoc?.....	1
¿Por qué preferir PHPDoc?.....	1
¿Qué fines persigue PHPDoc?.....	1
¿Cómo puedo tener PHPDoc en mi máquina?.....	2
Usando PEAR.....	2
Usando COMPOSER.....	2
Usando PHAR.....	3
Requisitos Mínimos.....	3
Usando PHPDoc para Documentar.....	4
¿Qué elementos pueden ser documentados?.....	4
DockBlocks o Bloques de Documentación.....	4
¿Cómo luce un DockBlock?.....	4
Etiquetas o Tags.....	6
@api.....	6
@author.....	6
@copyright.....	6
@filesource.....	6
@ignore.....	7
@license.....	7
@method.....	7
@package.....	8
@param.....	8
@return.....	8
@see.....	8
@throws.....	9
@todo.....	9
@var.....	9
@version.....	9
Creando Auto Documentación Web.....	10

PHPDoc

¿Qué es PHPDoc?

PhpDoc o PhpDocumentor es, si se quiere etiquetar, un plugin para hacer auto documentación o documentación automática para aplicaciones nativas en PHP.

Está disponible desde el año 2000 y a diferencia de sus similares, no es un programa ejecutable como tal, sino que es un conjunto de códigos PHP. Por ende, la documentación en PHP, básicamente, está realizada con el mismo PHP de tu máquina.

¿Por qué preferir PHPDoc?

La debida documentación de un código fuente, sea el lenguaje que sea, es la clave para su duración en el tiempo, ya que es un requisito dispensable para que otras personas puedan darle mantenimiento, puedan entender las funciones y procedimientos, corregir errores y expandir el código. Con PHPDoc, se dispone de un patrón para documentar y se puede crear una página web (esto último es automático de PHPDoc) con todas las funciones, procedimientos, métodos y atributos de cada clase comentada bajo los estándares de PHPDoc.

¿Qué fines persigue PHPDoc?

PHPDoc forma parte del universo del software libre, y tiene como su principal objetivo el siguiente:

“Proveer a los desarrolladores de PHP con todas las herramientas y recursos necesarios para escribir efectiva y comprensiva documentación con el menor esfuerzo posible”.

Esta aplicación trata de alcanzar este objetivo de la siguiente forma:

- Mantener los estándares de PHPDoc para explicar y describir código fuente PHP usando sus **CodeBlocks** o documentación por bloques.
- Proveer herramientas a la aplicación PHPDoc para:
 - Generar documentación de la aplicación.
 - Crear reportes de errores, elementos depreciados y cosas por hacer o revisar.
 - Generar gráficos para de esta forma comprender cómo está engranado el proyecto.
- Proveer plantillas para cambiar el aspecto de los reportes para que se adapten a los gustos de cada usuario.

¿Cómo puedo tener PHPDoc en mi máquina?

PHPDoc como buena aplicación (si se quiere) de software libre, hace uso de las mejores herramientas de control, actualización e instalación de software en LINUX, por esta razón la instalación de PHPDoc se realiza usando alguna de las siguientes formas:

USANDO PEAR:

PEAR dispone de la última versión oficial de **PHPDoc** y es una forma fácil de instalarlo en tu máquina.

Para usar **PEAR** como forma de instalación, primero debemos asegurarnos de tener **PEAR** instalado. Esto podemos hacerlo con el comando:

apt-get install pear o aptitude install pear

El siguiente paso sería agregar el canal de PHPDoc, esto lo hacemos con el comando:

pear channel-discover pear.phpdoc.org

Luego de descubrir el canal (que es similar a añadir un repositorio en un distro de Linux) podemos proceder a instalar el PHPDoc con el siguiente comando:

pear install phpdoc/phpDocumentor

Con esto se procederá a instalar el PHPDoc en nuestra máquina. Una vez finalizado, podemos probar si la instalación fue exitosa ejecutando el comando ***phpdoc*** en el terminal.

USANDO COMPOSER:

Para usar **COMPOSER** como forma de instalación, primero debemos asegurarnos de tener **COMPOSER** instalado. Esto podemos hacerlo con el comando:

apt-get install composer o aptitude install composer

El siguiente paso sería obtener los directorios de PHPDoc, esto lo podemos lograr con el comando:

composer require "phpdocumentor/phpdocumentor:2.*"

Con esto creamos un directorio *host* para PHPDocumentor. Podemos comprobar la instalación ejecutando el comando ***phpdoc*** en el terminal.

USANDO PHAR:

Para usar **PHAR** como método de instalación, debemos descargar el archivo PHAR desde <http://www.phpdoc.org/phpDocumentor.phar>.

Una vez descargado, procedemos a ejecutar:

php phpDocumentor.phar -d . -t docs/api

Donde phpDocumentor será el archivo PHAR que antes instalamos. De no estar dentro del directorio, debemos añadir la ruta exacta del archivo.

! Algunas instalaciones de PHP tienen problemas al ejecutar el archivo PHAR.

REQUISITOS MINIMOS:

Para cualquier método de instalación, los requisitos mínimos son:

- PHP 5.3.3 o superior.
- Extensión intl para PHP.
- Graphviz (si se desean crear diagramas de clases)

Usando PHPDoc para Documentar.

Una vez instalado, podemos disfrutar de las ventajas que nos otorga PHPDoc. Por ende, pasaremos a explicar cómo generar documentación con PHPDoc.

¿Qué elementos pueden ser documentados?

En PHPDoc se reconocen los siguientes elementos estructurales y por ende, pueden ser documentados:

- Funciones
- Constantes
- Clases
- Interfaces
- Constantes de Clases
- Propiedades
- Métodos
- Archivos

PHPDoc reconoce todos los elementos de la programación orientada a objetos en PHP, e incluso, se pueden documentar archivos.

DocBlocks o Bloques de Documentación

Un *DocBlock* o *Bloque de Documentación* es segmento del código cuya utilidad es contener la documentación. Su objetivo es informar sobre la clase, función o método al que va dirigido.

¿Cómo luce un *DocBlock*?

Un *DocBlock* común comienza con `"/*` y termina con `*/`. Esto se vería así:

```
/*  
 * Este es un ejemplo de DocBlock  
*/
```

En PHPDoc cambia el formato que conocemos de los DocBlocks, luciendo de la siguiente forma:

```
/**
 * Este es un ejemplo de DocBlock según el estándar de          * PHPDoc
 */
```

Para documentar con PHPDoc, se debe tener presente el siguiente formato:

```
<?php
/**
 * Una línea con una descripción corta de lo que se va a documentar.
 *
 * Una descripción más detallada de lo que hace aquello que se desea documentar
 * puede llevarse más de dos líneas, a diferencia de la descripción corta.
 *
 * @param string $miArgumento Etiquetas de PHPDoc
 *
 * @throw Exception Etiquetas de PHPDoc.
 */
function Ejemplo($miArgumento){
    throw new Exception("Exception");
}
```

- **Descripción Corta** (Primera línea bajo el doble asterisco): Es la encargada de explicar de forma poco detallada el objetivo de lo que se va a documentar.
- **Descripción Detallada:** Tiene como objetivo informar al programador de forma bastante detallada el objetivo e información pertinente sobre lo que se desea documentar.
- **Etiquetas:** PHPDoc tiene a su disposición una cantidad de etiquetas para facilitar la documentación.

ETIQUETAS o TAGS

PHPDoc tiene a su disposición una cantidad de **Etiquetas** que pueden usarse en los DocBlocks para especificar todos los detalles técnicos de lo que se desea documentar.

Las **Etiquetas** son:

@api

La etiqueta **@api** es usada para indicar que una estructura puede ser usada por terceros. Los elementos con esta etiqueta van separados del resto en la documentación web.

@author

La etiqueta **@author** es usada para indicar el creador de una estructura. Comúnmente se suele colocar el nombre del autor y un correo de contacto.

```
/**
 * @author Mi nombre
 * @author Mi nombre <mi.nombre@ejemplo.com>
 */
```

@copyright

La etiqueta **@copyright** es usada para indicar los derechos de autor a los que se somete una estructura.

```
/**
 * @copyright 2014-2015 Grupo Ejemplo PHP
 */
```

@filesource

La etiqueta **@filesource** es usada para indicarle al PHPDoc que debe incluir la ruta del archivo en la documentación.

```
/**
 * @filesource
 */
```


@ignore

La etiqueta ***@ignore*** es usada para indicarle al PHPDoc que no debe procesar la estructura en la documentación web.

```
/**  
 * @ignore  
 */
```

@license

La etiqueta ***@license*** es usada para indicar cuál licencia es aplicada para la estructura en cuestión.

PHPDoc agregará el link directo a la licencia en caso de ser:

- GPL
- LGPL
- BSD
- FreeBSD
- MIT

```
/**  
 * @license GPL  
 */
```

@method

La etiqueta ***@method*** es usada cuando una clase contiene métodos mágicos (`__call()`) e indica qué métodos mágicos son llamados.

```
/**  
 * @method string getString()  
 * @method void setInteger(integer $integer)  
 * @method setString(integer $integer)  
 */  
  
Class Hijo extends Padre{  
    <...>  
}
```

@package

La etiqueta **@package** es usada para categorizar las estructuras en subdivisiones lógicas. Para entenderlo más fácil, agrupa todas las estructuras que tengan en común el paquete.

```
/**
 * @package Controladores\EjemploControlador
 */
```

@param

La etiqueta **@param** es usada para definir los parámetros que recibe una función o un método. Se indica el tipo de dato del parámetro, el nombre del mismo y una descripción corta.

```
/**
 * @param int $codigo Código utilizado como ejemplo.
 */
```

@return

La etiqueta **@return** es usada para definir lo que retorna la función o el método. Se debe indicar el tipo de dato del retorno y la descripción.

```
/**
 * @return array|null Retorna un arreglo o null.
 */
```

@see

La etiqueta **@see** es usada para indicar una referencia desde un elemento estructural asociado o una dirección web.

```
/**
 * @see http://ejemplo.com/mi/ejemplo Documentación de Ejemplo.
 * @see MiClase::ejemplo() Método usado como ejemplo.
 */
```

@throws

La etiqueta **@throws** es usada para definir aquellas excepciones lanzadas dentro de la estructura a documentar. Se debe indicar el tipo de excepción y la descripción de la misma.

```
/**  
 * @throws EjemploException Excepción usada como ejemplo.  
 */
```

@todo

La etiqueta **@todo** es un marcador, se usa para indicar a un programador que se debe revisar, trabajar o corregir alguna sección de la estructura. Se debe indicar una descripción de lo que debe hacerse.

```
/**  
 * @todo Documentar mejor la función y evaluar los valores de retorno.  
 */
```

@var

La etiqueta **@var** es usada para definir los atributos de una clase o variables usadas dentro de las estructuras no parametrizadas. Se debe indicar el tipo de dato, el nombre de la variable y una descripción.

```
/**  
 * @var int $cont Un contador usado como ejemplo.  
 */
```

@version

La etiqueta **@version** indica la versión actual de una estructura. Se debe indicar la versión y una descripción.

```
/**  
 * @version 1.0.0.1 Un ejemplo de la etiqueta @version  
 */
```

Estas son las etiquetas no depreciadas en el PHPDoc 2. Para más información, no dude consultar <http://www.phpdoc.org/docs/latest/index.html>

CREANDO AUTO DOCUMENTACIÓN WEB

Una vez documentada nuestras estructuras, nos dirigimos al terminal y ejecutamos:

phpdoc DIRECTORIO_DE_APLICACION

Esto creará una carpeta en el directorio actual con una serie de elementos html y un index.html. Éste se debe abrir en un explorador web y con esto se tendrá la auto documentación web de PHPDoc.