

Documentación y Control de Versiones

Francisco Javier Sueza Rodríguez

8 de mayo de 2024

Centro:	IES Aguadulce
Ciclo Formativo:	Desarrollo Aplicaciones Web (Distancia)
Asignatura:	Despliegue de Aplicaciones Web
Tema:	Tema 5 - Documentación y Control de Versiones

Índice

1	Actividad 1: phpDocumentor	4
1.1	Actividad 1.1	4
1.1.1	Solución	4
1.2	Actividad 1.2	5
1.2.1	Solución	5

Índice de figuras

1.1	Instalación de phpDocumentor	4
1.2	Generación de la documentación de phpDocumentor	6
1.3	Visualización de la documentación generada	6

1. Actividad 1: phpDocumentor

Este ejercicio relacionado con generación de documentación consistirá en los siguientes apartados :

1.1. Actividad 1.1

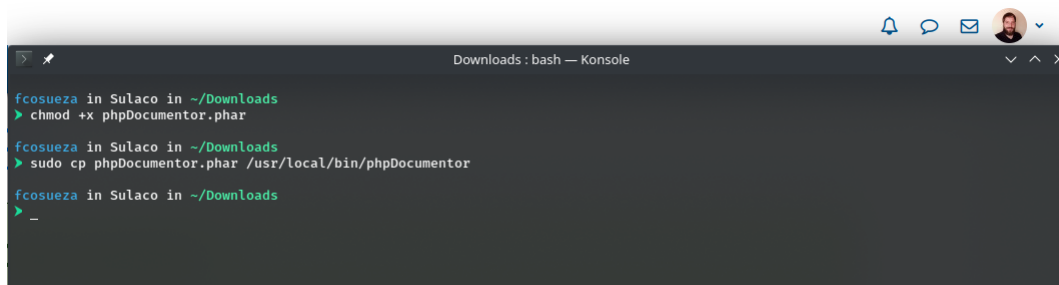
Instala la herramienta phpdocumentor en tu servidor Linux y comenta los aspectos más importantes de tal herramienta, así como las etiquetas principales que se usan.

1.1.1. Solución

En este ejercicio vamos a realizar la instalación de **phpDocumentor** en la distribución **Kubuntu 22.04** de Linux. Hay varios métodos para realizar la instalación, se puede hacer mediante **Docker**, usando **Phive** o **manualmente**. La instalación mediante composer no es recomendada por los desarrolladores de phpDocumentor, ya que hay una alta probabilidad de que se generen conflictos con otras librerías.

Nosotros hemos optado por la **opción de instalación manual**. Para ello, solo hemos tenido que descargar el **paquete de phpDocumentor** del repositorio de **repositorio de Github** de la aplicación, desde donde hemos descargado la **versión 3.4.3**.

Una vez descargada la versión simplemente le hemos **agregado permisos de ejecución** y la hemos movido al directorio `/usr/local/bin`, para que se puede ejecutar desde cualquier parte del sistema, como podemos ver en la siguiente captura.



```
Downloads: bash — Konsole
fcosueza in Sulaco in ~/Downloads
> chmod +x phpDocumentor.phar
fcosueza in Sulaco in ~/Downloads
> sudo cp phpDocumentor.phar /usr/local/bin/phpDocumentor
fcosueza in Sulaco in ~/Downloads
> _
```

Figura 1.1: Instalación de phpDocumentor

Actualmente, **phpDocumentor** es la aplicación más empleada para la **generación de documentación** en aplicaciones desarrolladas con PHP. Es el equivalente para PHP de **JavaDoc** para Java. Su instalación, como hemos visto, se hace de forma muy sencilla y su uso es igualmente sencillo, pudiendo ejecutarse desde la **línea de comandos**, la **interfaz web** o desde el **propio código PHP**.

Esta aplicación está desarrollada en PHP y usa el lenguaje para la generación de la documentación, empleando para ello los **DocBlocks**, que son trozos de código incluidos en el código PHP que sirven para describir la funcionalidad, parámetros, valores devueltos, etc., de una función, constante, propiedad, etc.

Además, phpDocumentator nos permite generar la documentación en una **variedad de formatos**, entre los se incluyen **HTML**, **PDF** y **XML**. Esta última opción es muy interesante, ya que mediante el uso de XSL podemos realizar transformaciones sobre el documento, extraer información, etc.

Dentro de los DocBlocks podemos usar un **conjunto de etiquetas** para describir diferentes elementos de la función que estamos documentando, siendo las más empleadas las siguientes:

- **@package**: se emplea para añadir documentación general en un fichero a nivel de clase.

- **@access:** se usa para especificar si se va a generar o no la documentación de un elemento concreto. En caso de que no queramos que esta genere, por ejemplo, porque es un método privado, podemos indicarle el valor *private*, con lo que la documentación de ese elemento no se generará.
- **@author:** se emplea para indicar el autor de código.
- **@copyright:** indicar información sobre los derechos de uso del elemento.
- **@deprecated:** se usa para indicar que un elemento ha sido deprecado y que no debería usarse.
- **@internal:** se usa para incluir documentación que solo estará disponible para los desarrolladores.
- **@link:** incluye un enlace a un recurso.
- **@see:** se usa para crear un enlace interno a un elemento, por norma general relacionado con el elemento que se esta documentando.
- **@version:** indica la versión del elemento.
- **@global:** se usa para especificar variables globales dentro de una función.
- **@param:** describe los parámetros que recibe una función.
- **@return:** describe el valor devuelve por una función.

Como podemos ver, para aquel que este familiarizado con JavaDoc todo esto ya le sonará, ya que las similitudes son muchas e incluso la sintaxis que se emplea es prácticamente la misma.

1.2. Actividad 1.2

En esta actividad debes crear en tu servidor un script PHP con el nombre practica-XXXXXX.php, donde XXXXXX será tu apellido. A continuación escribe dentro de este script bloques de código y DocBlocks para que luego se pueda generar la documentación correspondiente. El script debe contener al menos dos funciones documentadas, indicando mediante las etiquetas vistas en la unidad los siguientes elementos:

- Parámetros de entrada de la función.
- Parámetros de devuelve la función.
- Autor y versión del script.
- Una anotación que solo sea visible en la documentación para desarrolladores.

1.2.1. Solución

En este ejercicio se ha creado un script, *practica-SUEZA.php*, que genera un mensaje sencillo y procesa una fecha indicando la fecha actual. El script se adjunta con el trabajo por lo que no vamos a entrar en detalles sobre éste.

Una vez creado el script, se ha usado **phpDocumentor** para generar la documentación empleando para ello el comando *phpDocumentor -run -d . -t docs/*, donde se le indica que el directorio donde tiene que el actual y que la documentación generada la almacene en el directorio *docs*.

En la siguiente captura, vemos la ejecución de phpDocumentor y la estructura de directorios que ha generado en el proceso.

```
DAW-Tema5 : bash — Konsole

fcosueza in Sulaco in ~/NetBeansProjects/DAW-Tema5 via v8.1.2
> ls
practica-SUEZA.php

fcosueza in Sulaco in ~/NetBeansProjects/DAW-Tema5 via v8.1.2
> phpDocumentor run -d . -t docs/
phpDocumentor v3.4.3

Parsing files
1/1 [=====] 100%
Applying transformations (can take a while)

All done in 0 seconds!

fcosueza in Sulaco in ~/NetBeansProjects/DAW-Tema5 via v8.1.2
> ls -las docs/
total 52K
4,0K drwxr-xr-x 10 fcosueza fcosueza 4,0K may 8 12:50 .
4,0K drwxrwxr-x 4 fcosueza fcosueza 4,0K may 8 12:50 ..
4,0K drwxr-xr-x 2 fcosueza fcosueza 4,0K may 8 12:50 css
4,0K drwxr-xr-x 2 fcosueza fcosueza 4,0K may 8 12:50 files
4,0K drwxr-xr-x 2 fcosueza fcosueza 4,0K may 8 12:50 graphs
12K -rw-rw-r-- 1 fcosueza fcosueza 9,2K may 8 12:50 index.html
4,0K drwxr-xr-x 2 fcosueza fcosueza 4,0K may 8 12:50 indices
4,0K drwxr-xr-x 2 fcosueza fcosueza 4,0K may 8 12:50 js
4,0K drwxr-xr-x 2 fcosueza fcosueza 4,0K may 8 12:50 namespaces
4,0K drwxr-xr-x 2 fcosueza fcosueza 4,0K may 8 12:50 packages
4,0K drwxr-xr-x 2 fcosueza fcosueza 4,0K may 8 12:50 reports
```

Figura 1.2: Generación de la documentación de phpDocumentor

Como podemos ver, se ha generado una estructura de directorios donde se almacenan diferentes elementos, en este caso que la salida ha sido en **HTML**, hay archivos de estilo CSS, archivos Javascript, etc...

En la siguiente captura, podemos ver la visualización del archivo principal de la documentación en una navegador.

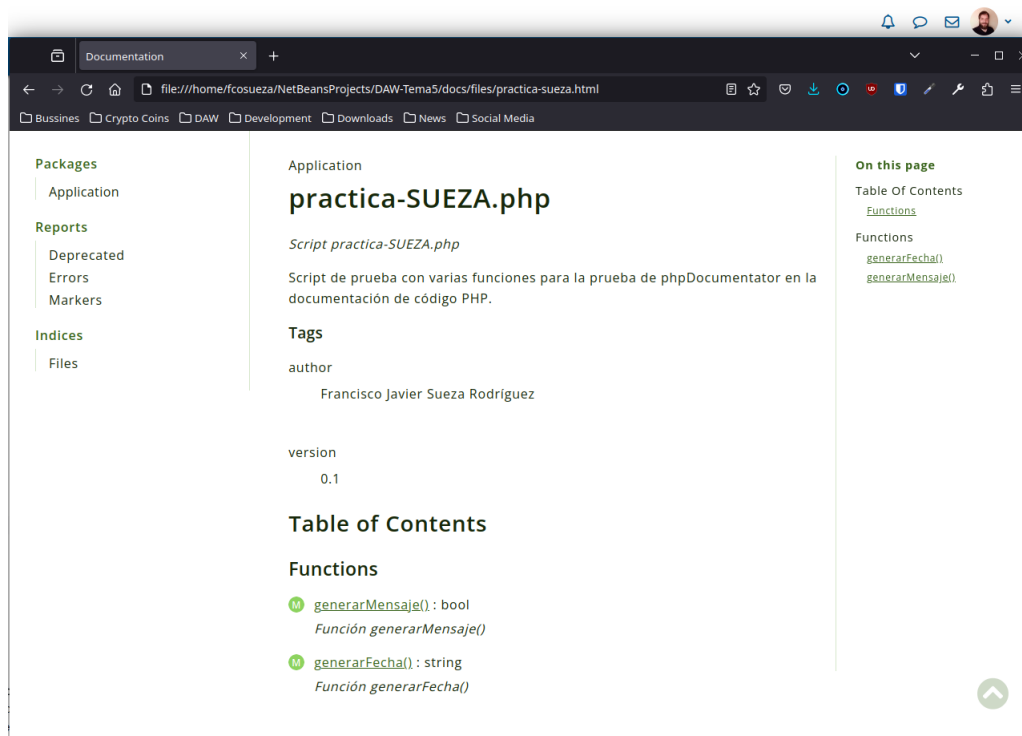


Figura 1.3: Visualización de la documentación generada

2. Ejercicio 2: Git y GitHub