

# *Composer, un gestor de dependencias para proyectos en PHP*



A la hora de empezar el desarrollo de un proyecto en PHP, es necesario conocer todas las librerías que necesitaremos. La instalación de estas librerías puede ser una tarea complicada y tediosa si lo hacemos a mano, pero por suerte existen gestores de dependencias que se encargan de realizarla de forma automática, gestionando esas librerías de terceros de forma sencilla y eficaz. En este White Paper hablaremos sobre qué es **Composer**, cómo funciona y cómo instalarlo en tu sistema.

Mucha gente suele asociar a Composer la categoría de framework, lo cual es erróneo. Esto es debido a que por medio de él se puede poner en marcha la configuración de un entorno de desarrollo basado en **Symfony**. Composer se encargaría de realizar la instalación de todas las dependencias necesarias para que Symfony pudiera trabajar de forma correcta. Hemos puesto el ejemplo de Symfony pero también puede ser aplicado a cualquier otro framework.

## ¿Qué es un gestor de dependencias?

A la hora de hacer un desarrollo pequeño utilizando PHP, en muchos casos nos es suficiente con las funciones nativas que incluye el propio lenguaje, pero esto no ocurre cuando el proyecto va adquiriendo mayor envergadura. En estos casos todos los desarrolladores hacen uso de alguna librería de terceras personas para realizar algún tipo de función, por ejemplo a la hora de enviar un **email** autenticado se puede utilizar la librería PHPMailer o generar un archivo XLS con la librería PHPExcel. Cualquier cosa que necesites puede estar ya creada por otros desarrolladores, lo que ayuda a acotar los tiempos de desarrollo del proyecto.

Si sois de los que utilizáis este tipo de librerías en vuestros desarrollos, al empezar el proyecto deberéis acudir a la página web oficial de cada una de ellas para descargarlos y copiarlas en vuestro proyecto para empezar a utilizarlas. Esto mismo habría que hacer si el desarrollador lanza una nueva versión de ese código. Si son una o dos librerías, el tiempo empleado para este tipo de mantenimiento no es mucho, pero si se hace uso de muchas, o de un framework como Symfony que utiliza a su vez muchas otras funcionalidades externas a él, el proceso de instalación y mantenimiento sí que puede ser una tarea complicada.

Es aquí donde entran en acción los gestores de dependencias, cuya principal función es la de ayudarnos a realizar las tareas de instalación y mantenimiento de las versiones utilizadas en el proyecto de forma automática y sin necesidad de tener que visitar sitio por sitio oficial.

## Ventajas de utilizar Composer

Entre las principales ventajas que podemos encontrar a la hora de utilizar Composer están:

- Permite indicar exactamente la librería que necesitas y su versión
- Fomenta el uso de estándares
- Puedes utilizar paquetes PEAR
- Existe un repositorio público donde podrás encontrar librerías para cualquier tipo de acción. Un ejemplo de esto es la web [packagist.org](http://packagist.org)
- Permite también el uso de repositorios privados
- Permite la definición de procedimientos o llamadas a funciones que se ejecutarán después de realizar ciertas operaciones como actualizaciones, borrados, instalaciones...

## Pasos para instalar Composer en nuestro sistema



Dependiendo de si nuestro sistema es Windows o Linux, la forma de instalar Composer en él variará. Veamos cómo se instala para cada uno de los casos.

### a) Instalando Composer en Windows

A continuación veremos los pasos que hay que seguir para realizar la instalación de Composer en nuestro sistema.

Lo primero que haremos será entrar mediante nuestro navegador en esta dirección web <http://getcomposer.org/installer> y su contenido lo guardamos en cualquier parte de nuestro equipo con el nombre que queramos pero siempre con extensión “.php”, por ejemplo “instalador.php”.

Lo siguiente que deberemos hacer será entrar en la consola de Windows (Ejecutar -> CMD) y entrar en el directorio donde hemos guardado el archivo. Por ejemplo:

```
C:\> cd C:\Document and settings\acens\Escritorio
```

A continuación ejecutamos el siguiente comando para instalar Composer.

```
php instalador.php
```

Si todo ha funcionado bien, en ese mismo directorio tendremos que ver un nuevo archivo llamado “**composer.phar**”. Una vez que lo tenemos, ya podemos borrar el archivo PHP que nos habíamos descargado, en nuestro caso era “instalador.php”.

Por último sólo faltaría mover este nuevo archivo al directorio donde tenemos nuestro proyecto y donde queremos que se realice la instalación de las dependencias.

### b) Instalando Composer en Linux o Mac OS X

A continuación vamos a ver los pasos a seguir para realizar la instalación Composer en un entorno donde se trabaje con Linux o con Mac OS X.

Lo primero que tendremos que hacer es abrir la consola de comando y ejecutar la siguiente instrucción en cualquier directorio:

```
curl -s https://getcomposer.org/installer | php
```

Si todo ha ido bien, tendremos que ver un nuevo archivo llamado “**composer.phar**”.

A continuación movemos ese archivo a algún directorio ejecutable del sistema.

```
mv composer.phar /usr/local/bin/composer
```

Una vez hecho esto y si no hemos tenido problemas, al ejecutar la orden “**composer**” deberíamos ver un listado con los comandos que proporciona Composer.

## Cómo se usa Composer

Como hemos explicado, Composer es un gestor de dependencias que se encarga de realizar las tareas de instalación y mantenimiento de las librerías utilizadas en nuestros proyectos por nosotros. Para llevar a cabo esta tarea, es necesario indicarle a la herramienta qué es lo que necesitamos.

Para ello, lo primero que deberemos crear es un archivo cuyo nombre es “**composer.json**” y será en él donde indicaremos las dependencias que queremos instalar en nuestro proyecto. La forma de realizarlo es así:

```
{
  "require": {
    "monolog/monolog": "1.0.0"
  }
}
```

En este caso le estamos indicando que requerimos la librería “**monolog**” y en concreto la versión “**1.0.0**”, por lo que si salen nuevas versiones de este software, Composer no lo actualizará.

También le podríamos indicar que nos da igual la versión de nuestra librería, siempre y cuando sea superior a una versión indicada. Esto lo conseguiríamos mediante el siguiente código:

```
{
  "require": {
    "phpmailer/phpmailer": "5.2.*"
  }
}
```

Le estamos indicando que necesitamos alguna versión de “**phpmailer**” igual o superior a la versión “**5.2**”.

Una vez que hemos indicado todas las dependencias que utilizaremos en nuestro proyecto, es hora de indicar a Composer que empiece la instalación de éstas. Para ello desde consola deberemos ejecutar la siguiente instrucción:

```
php composer.phar
```

Como habéis podido leer a lo largo de todo este White Paper, la puesta en marcha de Composer es muy sencilla y los beneficios que obtendremos por ello son importante. Ahora bien, es muy importante que no lo confundamos con un framework, ya que Composer no ofrece un conjunto de archivos y directorios que facilitan la creación de aplicaciones web, sino que lo que hace es añadir librerías de terceras personas a nuestros proyectos, librerías que luego serán utilizadas bien dentro de un **framework** como Symfony o Zend.