

Estructura de proyecto MVC

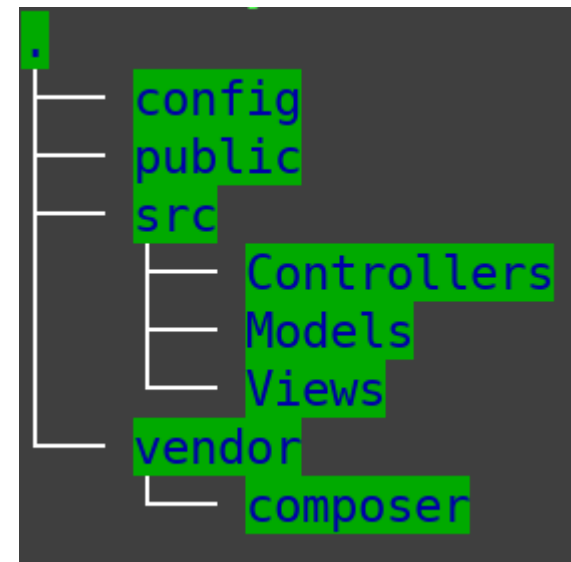
2º DAW - Desarrollo Web en Entorno Servidor

David Soler Talens

Estructura típica



- **Las carpetas pueden variar de un proyecto a otro**
 - Config -> ficheros de configuración, como de conexión a bbdd, etc
 - Public -> raíz del sitio = carpeta visible al usuario
 - Src -> lógica de la aplicación
 - Vendor -> librerías de terceros
 - Composer -> gestor de dependencias



Composer (I)



- **Es una herramienta para la gestión de dependencias en PHP.**
- **Permite declarar librerías de las que depende tu proyecto y las administra.**
- **Es un gestor de paquetes por proyecto, dentro de cada proyecto.**
- **Por defecto no instala nada globalmente.**
- **inspirado en *'npm'* de Node y *'bundler'* de Ruby.**

Composer (II)



- **Instalación manual**

- Descargar el instalador

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
```

- Verificar el instalador

```
php -r "if (hash_file('sha384', 'composer-setup.php') ===  
'a5c698ffe4b8e849a443b120cd5ba38043260d5c4023dbf93e1558871f1f0  
7f58274fc6f4c93bcfd858c6bd0775cd8d1') { echo 'Installer verified'; }  
else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo  
PHP_EOL;"
```

- Ejecute el instalador

```
php composer-setup.php
```

- Quitar el instalador

```
php -r "unlink('composer-setup.php');"
```

Composer (III)



- **Para usar Composer**

- Podemos copiar y pegar composer.phar a cada directorio raíz de cada proyecto nuevo que creemos
- Podemos habilitar su uso de forma global, estableciendo en el PATH la ruta
- Se puede cambiar el nombre *composer.phar* a *composer*, para hacer más fácil su uso



Composer (IV)



- **Esquema mínimo de composer.json**

Desarrollador Nombre de Proyecto

```
1 {  
2     "name": "daw2-dwes/mvc-simple",  
3     "autoload": {  
4         "psr-4": {  
5             "App\\" : "src/",  
6             "Config\\" : "config/"  
7         }  
8     },  
9     "require": {}  
10 }
```

Autoload de Composer

Dependencias instaladas
(en este caso, ninguna...)

- **Podemos crear el fichero con**

composer init -n --name "daw2-dwes/mvc-simple"

Autoload de Composer (I)



- **Se genera con el comando**

composer install

- **Debemos añadir a la página .php la línea**

require_once 'vendor/autoload.php';



Atención a la
dirección relativa!!!

- **Composer localiza las clases y/o interfaces relacionando carpetas con los Namespace**

```
"autoload": {  
    "psr-4": {"MiNamespace\\": "mi_carpeta/"},
```

Autoload de Composer (II)



- Cada vez que generamos **nuevos namespace** o **cambiamos el autoload** de composer, es necesario ejecutar:

composer dump-autoload

- **Criterios generales**

- Las clases se llaman igual que el fichero que las contiene
- Cada fichero solo contiene código de definición de clases
- Cada fichero contiene una sola definición de clase
- El sistema es sensible a mayúsculas/minúsculas