Instalación de Composer

[Descargar Composer](https://getcomposer.org/download/)

* **Primeros pasos**

En la raíz del directorio donde vaya a estar nuestro proyecto y desde la terminal tecleamos "composer init" a continuación se nos hacen unas preguntas y vamos contestando:

* + Package name (<vendor>/<name>) : aquí nos pregunta organización y nombre podemos poner "usuario/usuario" por ejemplo
  + Description [] : Ponemos una descripción del proyecto
  + Author : ponemos autor y correo en el formato siguiente: autor <correo>, por ejemplo: "usuario <usuario@correo.es>"
  + Minimum Stability [] : Lo podemos dejar en blanco.
  + Package Type [] : Aquí elegiremos project.
  + License [] : Ponemos la licencia por ejemplo **GPL**
  + A las siguientes preguntas sobre definir nuestras dependencias, ahora decimos que no.

Al final nos muestra un archivo JSON , se nos pregunta que si queremos generarlo, contestamos que sí y nos genera el archivo "composer.json". Una vez generado el archivo tecleando "composer install" nos instalará las dependencias.

**Autocarga** de clases/librerías: crearnos, en la raíz del proyecto que vayamos a usar, un directorio de nombre "**src**" y modificar el archivo "composer.json". Para el "**autoload**" de clases y librerías se hace uso de los namespaces.

{

    "name": "roberto/app",

    "description": "App SOAP",

    "type": "project",

    "license": "GPL",

    "config": {

        "optimize-autoloader": true,

        "allow-plugins": {

            "kylekatarnls/update-helper": true

        }

    },

    "autoload": {

        "psr-4": {

            "Roberto\\App\\": "src/"

        }

    },

    "authors": [

        {

            "name": "robertorodriguezNet",

            "email": "robertorodriguez@robertorodriguez.net"

        }

    ],

    "require": {}

}

Modificamos la línea en la que se declara nuestro namespace con las librerías que se cargarán con autoload.

    "autoload": {

        "psr-4": {

            "Clases\\": "src/"

        }

    },

Cada vez que se modifique el fichero **composer.json** hay que ejecutar **composer install**.

**Incluir un paquete**

composer require wsdl2phpgenerator/wsdl2phpgenerator

**Especificar una versión**

En general, debes usar el comando composer require seguido del **nombre del paquete**, de **dos puntos** y de la **versión del paquete** que quieres instalar:

composer require nombre/paquete:version

Por ejemplo, en caso de que quieras instalar la versión 7.1 del paquete [Guzzle](https://github.com/guzzle/guzzle), tendrías que ejecutar el siguiente comando:

composer require guzzlehttp/guzzle:~7.1

Al agregar el carácter ~ hemos especificado un rango de versiones, que será mayor o igual a la versión 7.1 y menor que la versión 7.2. De entre el rango de versiones anterior se instalará la más reciente. Es decir, que obtendremos las actualizaciones de seguridad.

También podríamos haber especificado el rango de versiones usando el carácter ^, que expandirá el rango a las versiones mayores:

composer require guzzlehttp/guzzle:^7.2

El rango de versiones anterior indica que la versión mínima a instalar será mayor o igual a la versión 7.2 y menor que la versión 8.

También puedes instalar una versión específica:

composer require guzzlehttp/guzzle:7.2.3

O también:

composer require guzzlehttp/guzzle:7.2

Si no sabes la versión del paquete que está instalada actualmente, puedes comprobar el archivo composer.json o también usar el comando composer show, que mostrará una lista con los paquetes locales instalados en el proyecto. De querer ver los paquete globales, tendrías que usar el comando composer global show, aunque se recomienda tener cuidado al instalar una versión en concreto de un paquete global, ya que puede afectar a varios proyectos.

Cabe decir que no se recomienda instalar versiones menores específicas de un paquete. Es por ello que hemos usado ~7.0 en lugar de instalar por ejemplo la versión 7.1.3, ya que esto último además nos dará un aviso. En general, la versiones menores solamente se usan para corregir bugs, no afectando a las funcionalidades del paquete. Así, además, el paquete se actualizará a la versión menor más reciente cuando ejecutes el comando composer update:

composer update guzzlehttp/guzzle

En ocasiones necesitarás usar alguna versión mayor de un paquete cuando otro paquete lo requiera. Esto puede que se te pida cuando intentes instalar o actualizar un paquete. Las versiones menores nunca suelen ser requisito de ningún paquete. Si quieres comprobar si existe algún problema con algún paquete instalado o su respectiva versión, tendrás que ejecutar el comando composer diagnose.