Contents

[**¿Qué es composer?** 1](#_Toc56747441)

[**Configurar composer** 1](#_Toc56747442)

[**Cómo iniciar un proyecto con composer** 3](#_Toc56747443)

[**Introducción a JSON** 18](#_Toc56747444)

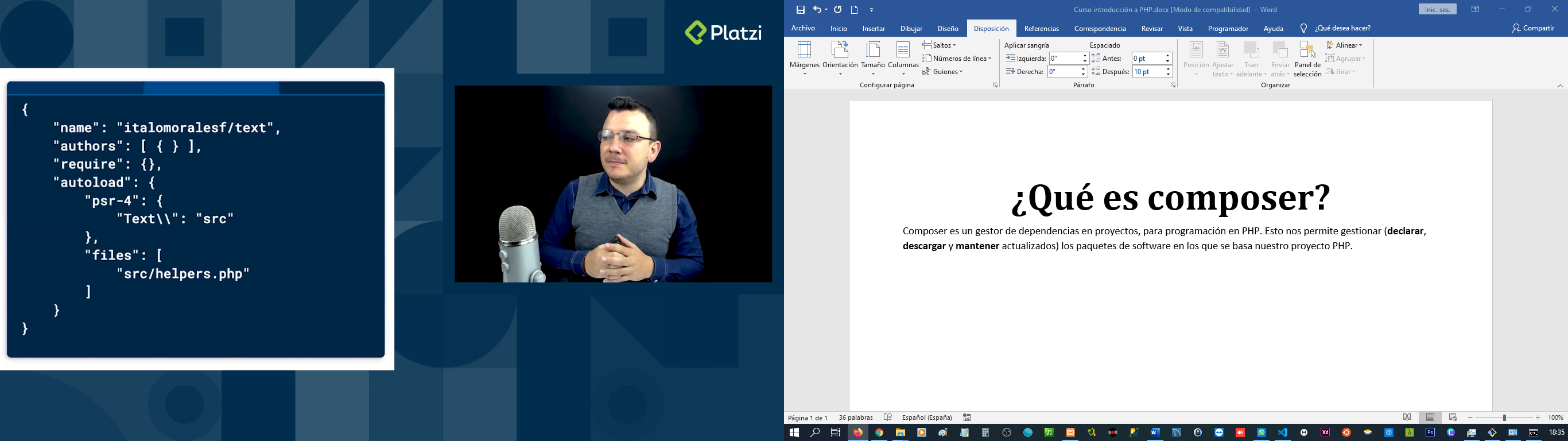
[**Autoload** 19](#_Toc56747445)

[**Comandos de Composer** 21](#_Toc56747446)

[**composer.lock** 22](#_Toc56747447)

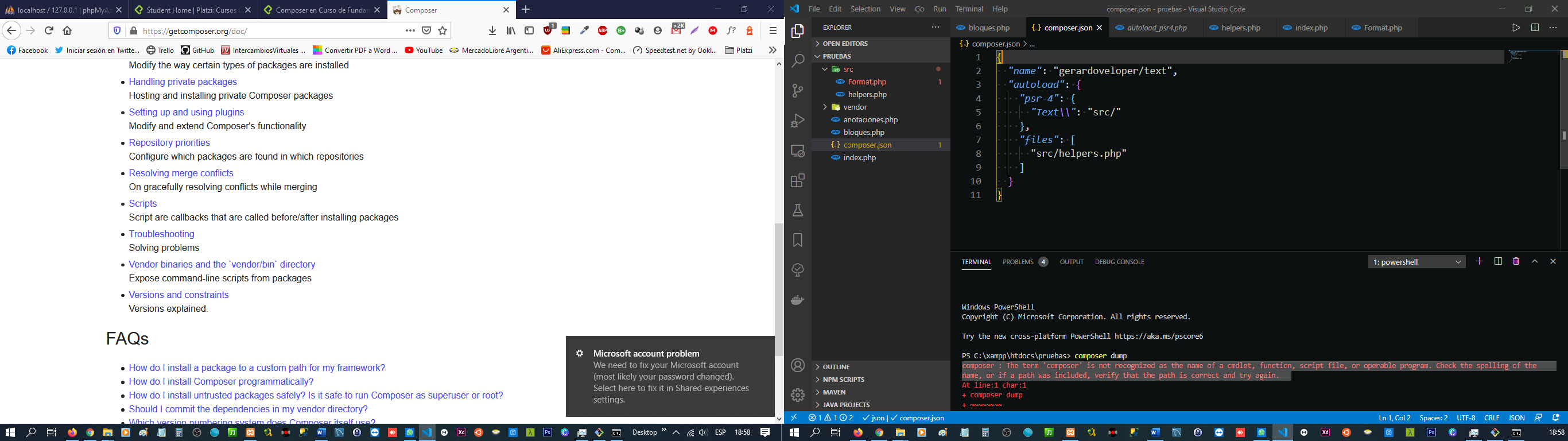
# **¿Qué es composer?**

Composer es un gestor de dependencias en proyectos, para programación en PHP. Esto nos permite gestionar (**declarar**, **descargar** y **mantener** actualizados) los paquetes de software en los que se basa nuestro proyecto PHP.

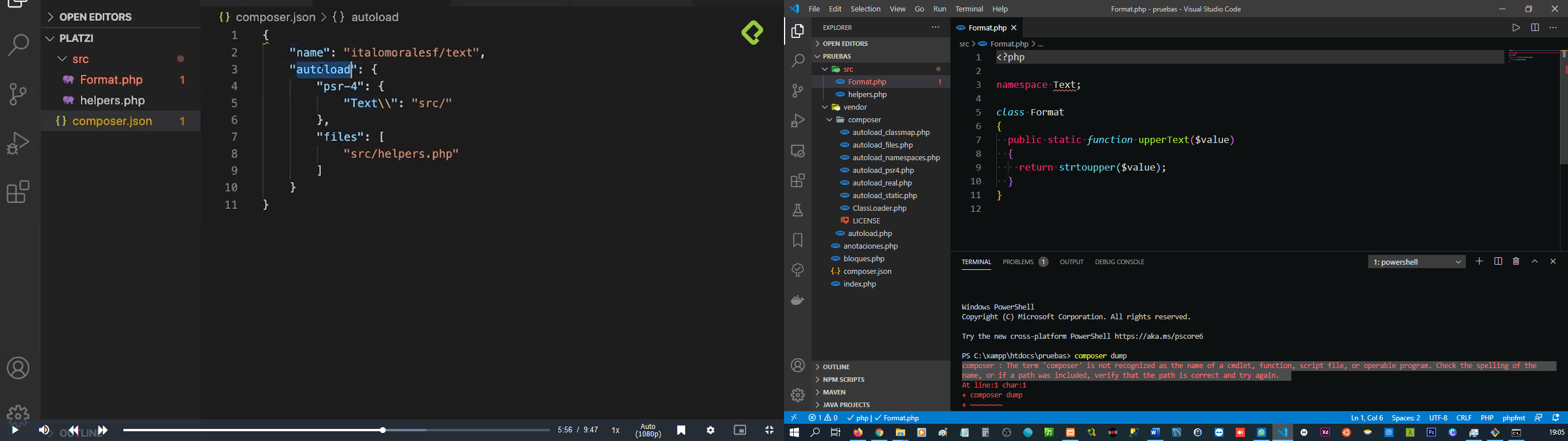


## **Configurar composer**

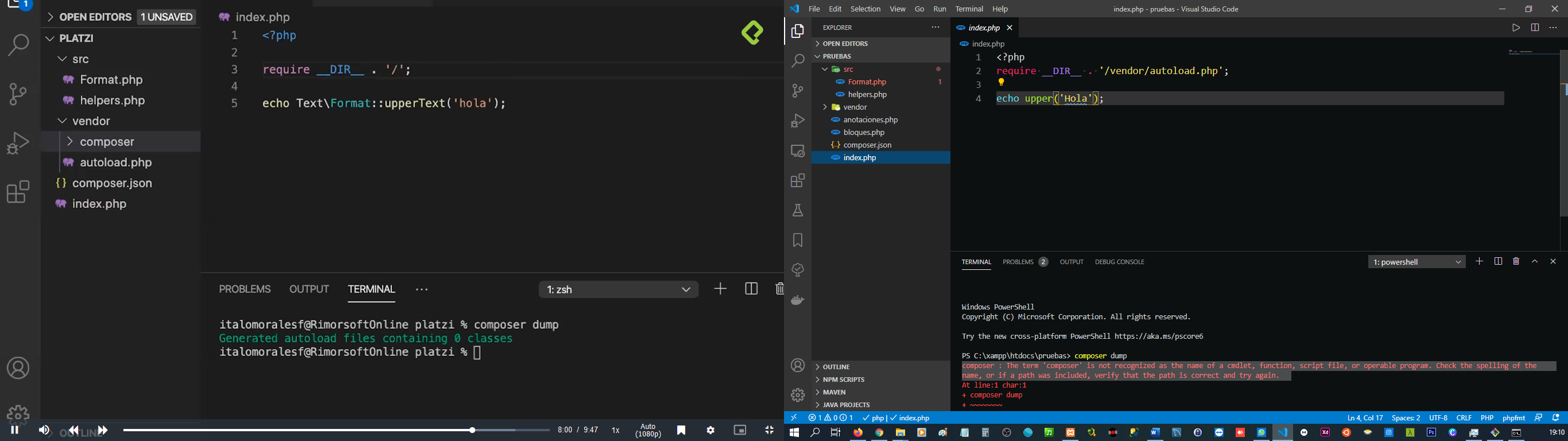
1. Crear un archivo **.json** con el nombre de **composer**.
2. Colocamos lo siguiente dentro del archivo **composer.json**



1. Creamos una carpeta de nombre **src**, dentro de ella un archivo con el nombre **helpers.php** y otro con el nombre **Format.php**
2. Dentro de **Format.php** colocamos las funciones que necesitamos en nuestro proyecto.

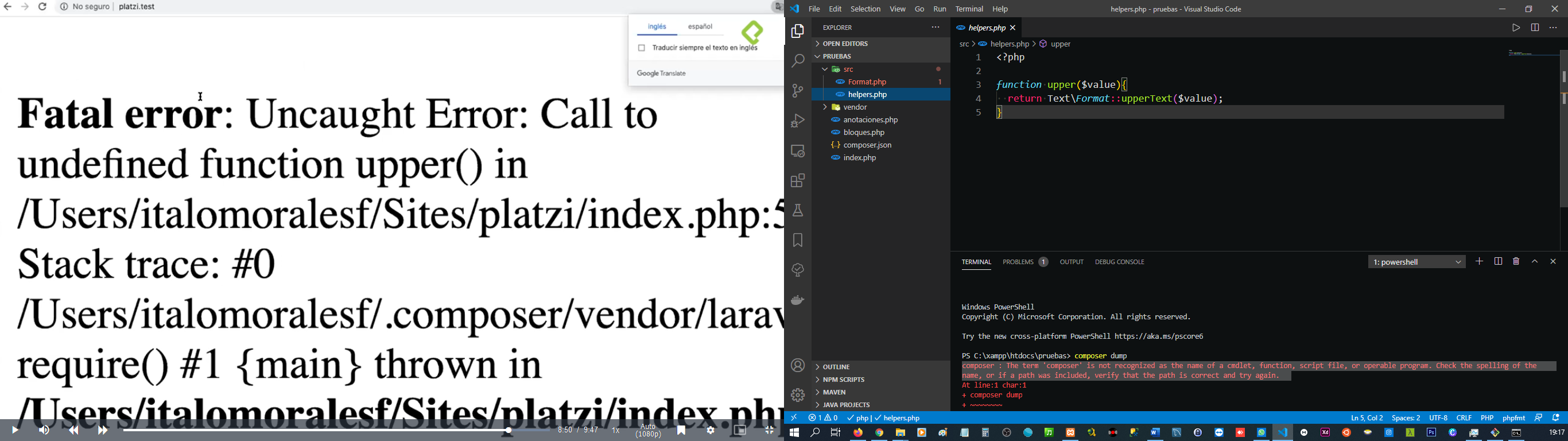


1. Abrimos un terminal y le decimos a nuestro proyecto que estamos utilizando el sistema de autocarga de composer, para ello tipeamos en la terminal **composer dump**.
2. Se crea una carpeta llamada **vendor** donde se encuentra toda nuestra configuración necesaria.
3. Creamos un archivo **index.php** en la raíz del proyecto. Dentro colocamos **require \_\_DIR\_\_ . ‘/vendor/autoload.php’** y todas las funciones que necesitemos.



Con ese **require** le decimos a nuestro proyecto que tenemos un sistema de clases de carga instalado.

1. En el archivo **helpers.php** colocamos.

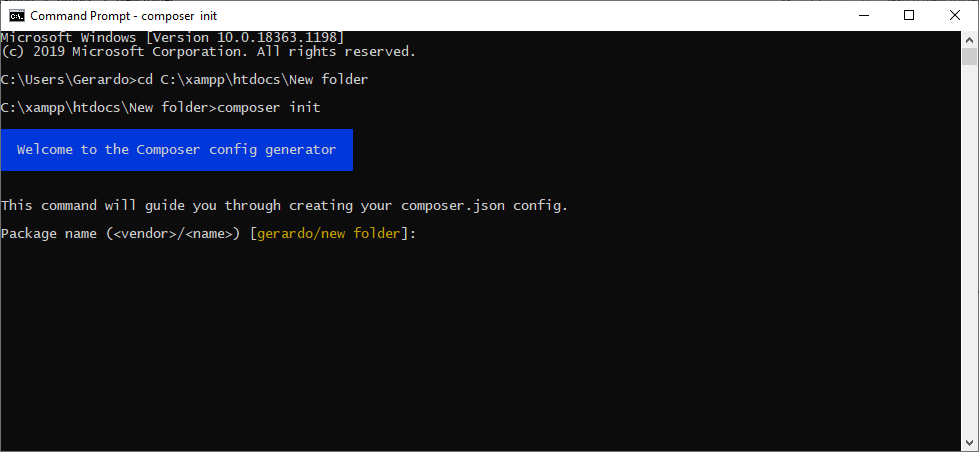


Aquí colocamos todas las funciones que necesite nuestro sistema.

[Sitio oficial](https://getcomposer.org/) de composer.

# **Cómo iniciar un proyecto con composer**

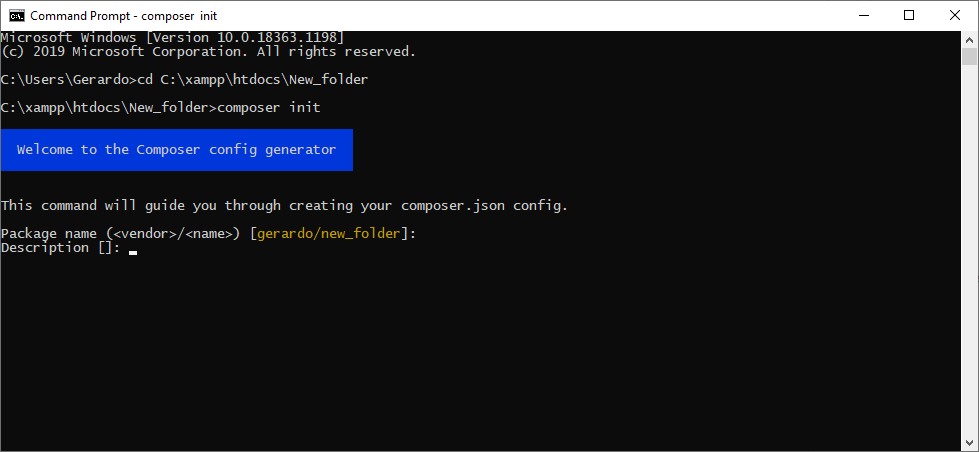
1. Abrimos la terminal y escribimos **composer init**.
2. Composer nos dará la bienvenida, nos irá preguntando distintas cosas, como el **nombre del proveedor** y el **nombre del paquete**.



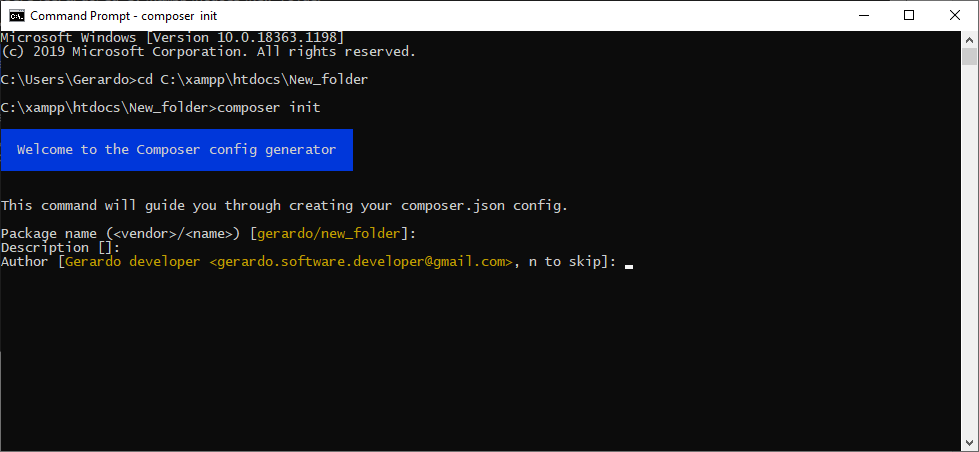
Si queremos dejar todo como esta presionamos Enter.

**Nota**: Composer toma las configuraciones básicas que tengamos en git.

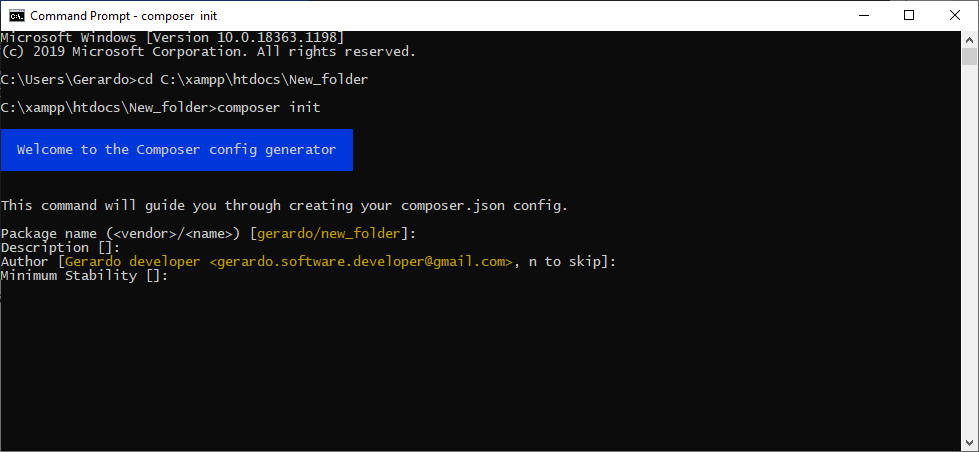
1. Nos pedirá una descripción. Si queremos dejarlo en blanco presionamos **enter**.



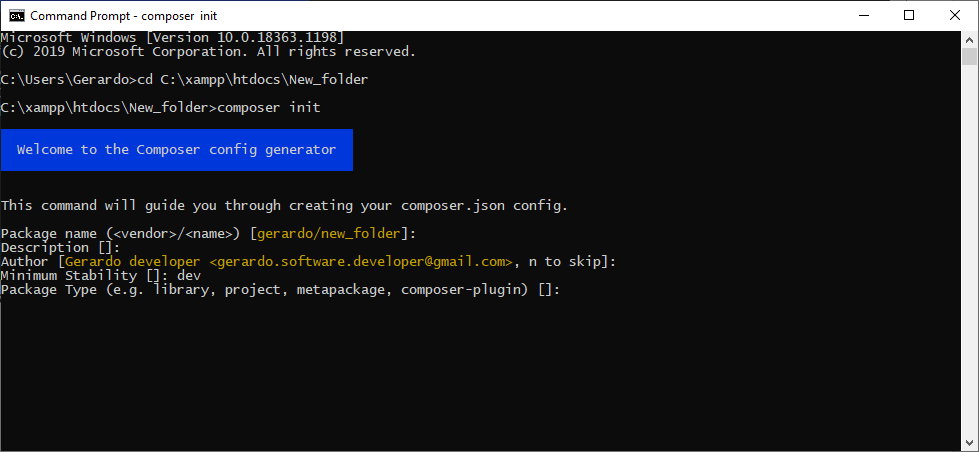
1. Nos mostrara la información del correo. Enter para continuar.



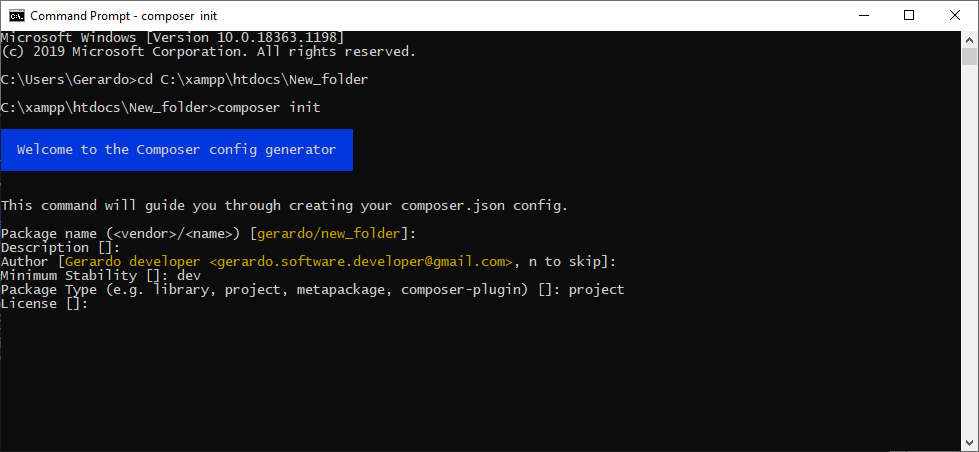
1. Nos preguntará las etapas de nuestro proyecto, por ej: dev, alpha, beta.



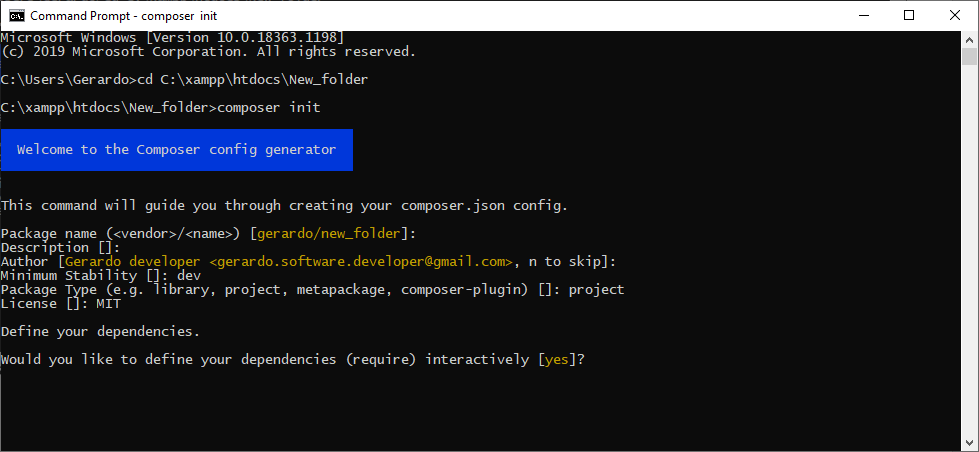
1. Nos pregunta que tipo de proyecto estamos haciendo.



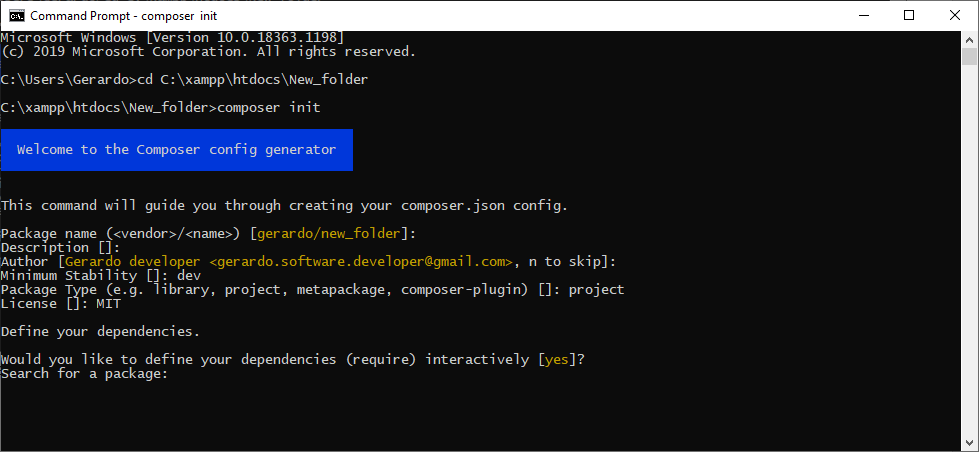
1. Nos pregunta el tipo de licencia.



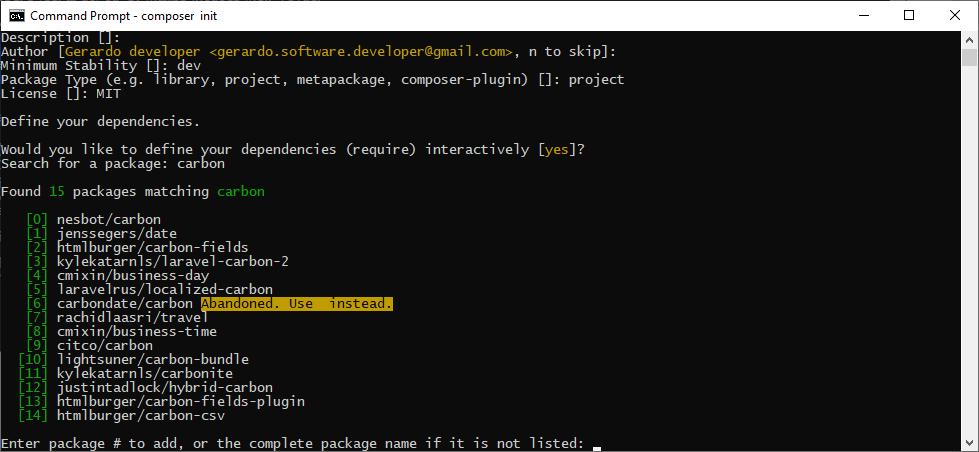
1. Nos pregunta si necesitamos paquetes en producción. Enter para dejar vacío.



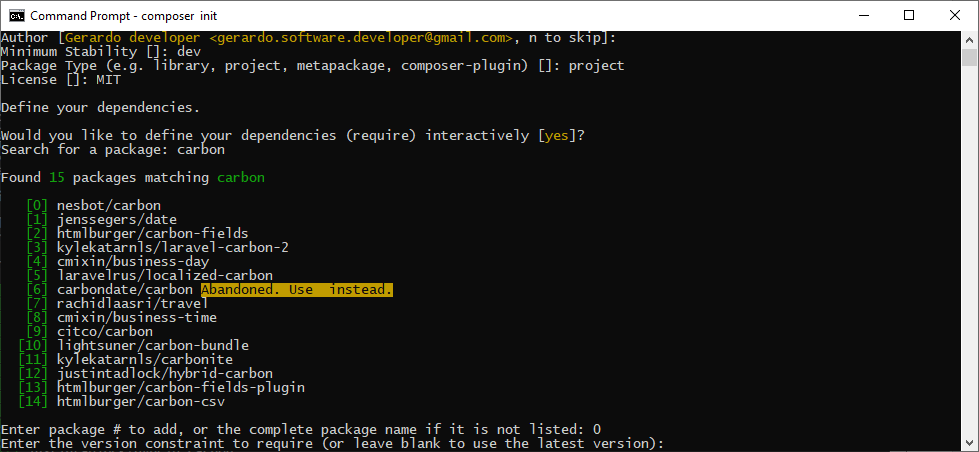
1. Si le decimos que si nos preguntará que paquetes queremos buscar y descargar.



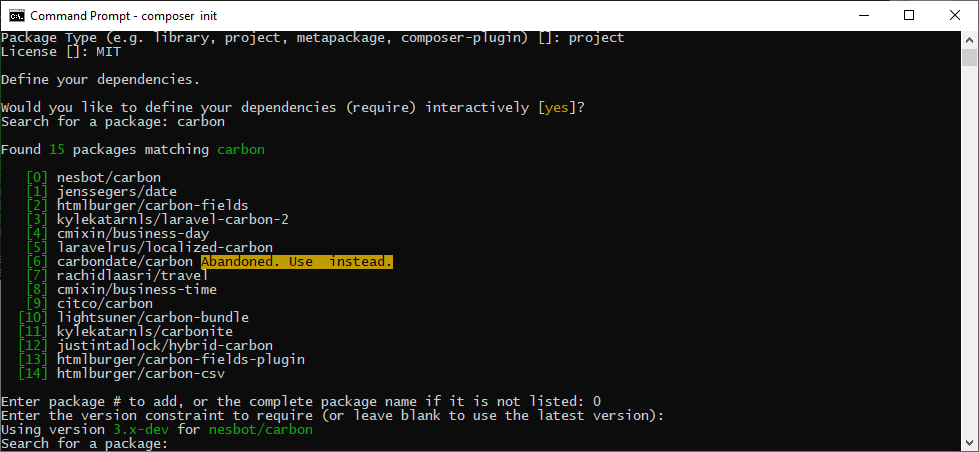
1. Colocamos el numero de elementos donde se encuentra el paquete deseado.



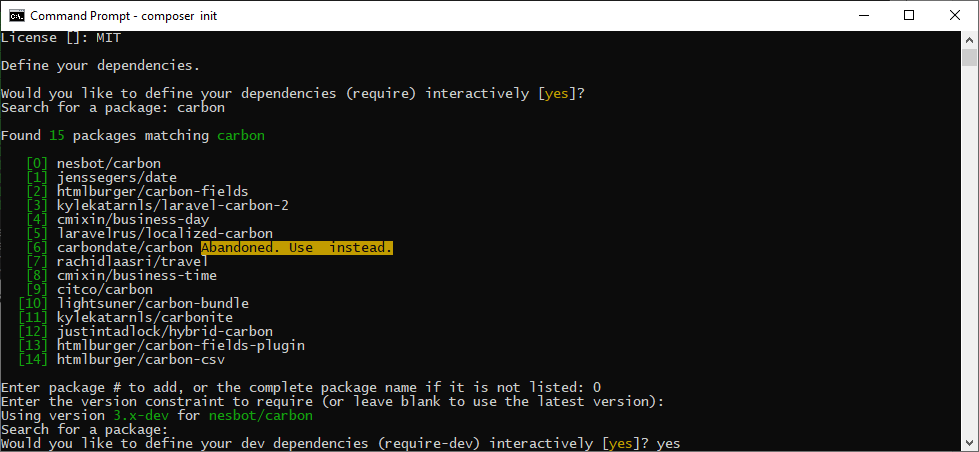
1. Nos pregunta si queremos la ultima versión del paquete, si es así debemos dejar en blanco, Enter para continuar.



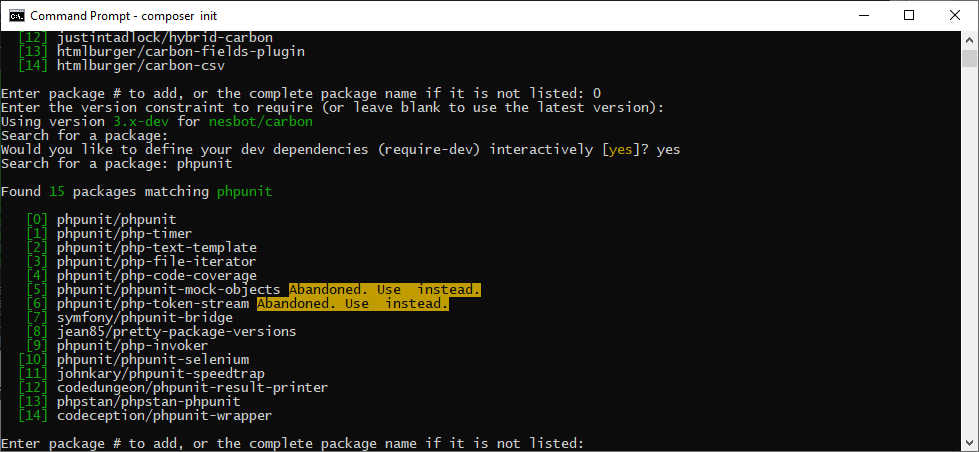
1. Al presionar **Enter** se conectará a internet y buscará el paquete indicado previamente. Nos preguntará nuevamente si queremos buscar otros paquetes. Enter para continuar si no se desea buscar más.



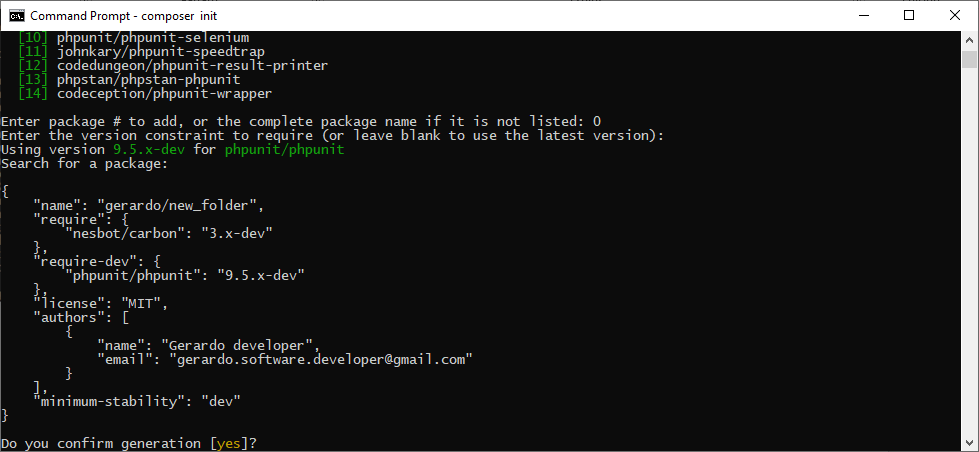
1. Nos pregunta que paquetes o componentes requerimos al momento de desarrollar. **Yes** para buscar.



El siglo es el mismo que lo anterior.

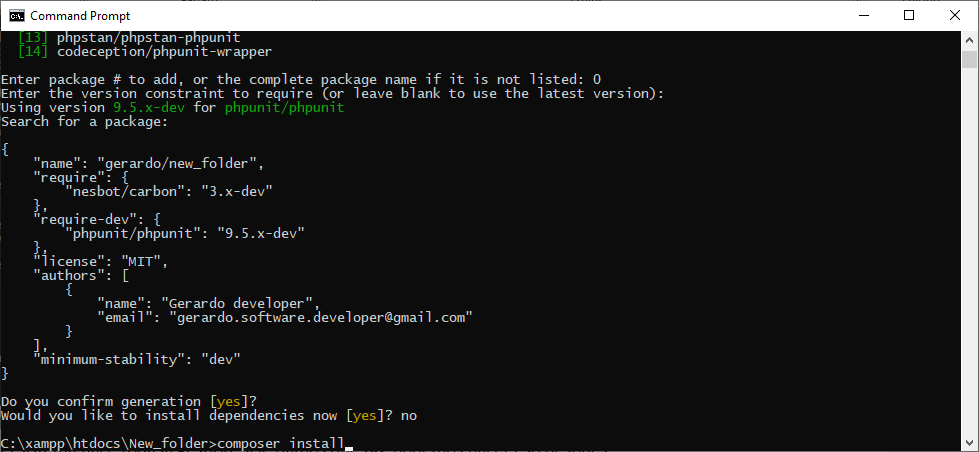


1. Pregunta si estamos de acuerdo con la información que se muestra en pantalla.

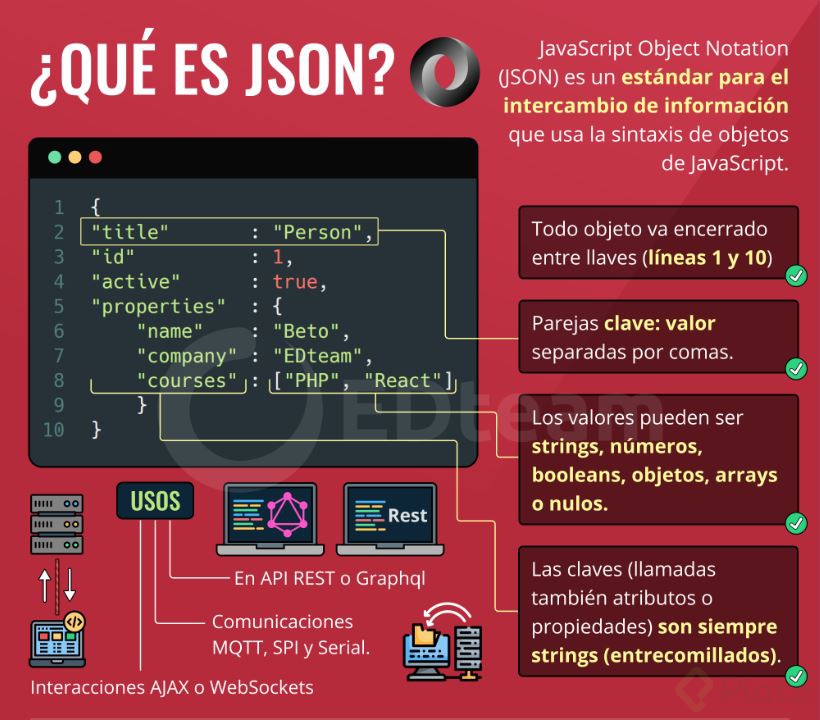


Si le damos Enter nos pregunta si queremos instalar las dependencias ahora, si decimos que **sí**, el sistema se conecta a internet y el sistema hace toda la gestión, si le damos que **no**, podemos hacer eso después.

1. Para hacer la instalación tipeamos en la terminal **composer install**, se conecta a internet, busca todos los paquetes, se crea la carpeta **vendor** y ahí dentro se van a instalar todas las dependencias de los paquetes.



# **Introducción a JSON**



# **Autoload**

**Files** → Permite cargar una serie de archivos con helpers. Aquí configuramos archivos ayudantes (helpers).

**Classmap** → Nos deja cargar carpetas de una manera directa carpetas que van a tener dentro de sí diferentes clases.

**Psr-0** → Ya que es más antiguo, tenemos que poner todas las rutas de las carpetas.

**Psr-4** → Al hacer énfasis a la carpeta principal, este entiende todas las rutas dentro de sí. (recomendado)



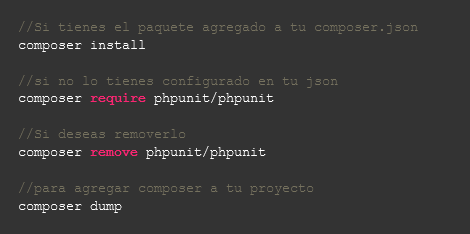
Siempre que utilicemos un archivo principal como el index PHP necesitamos invocar al archivo de autoload.

* El comando composer dump permite desarrollar todo el sistema de composer.
* Para evitar errores antes de ejecutar composer debemos crear las carpetas que contienen nuestras clases (si es el caso).
* autoload.php se encarga de configurar todo internamente (generado por composer).
* Siempre que se usa un archivo principal (ej. **index.php**) se necesita invocar el archivo de autoload.php.

# **Comandos de Composer**

El primer paso de todo proyecto es crear el archivo **composer.json**, esto es lo correcto. Si escribimos la palabra composer en la bash nos mostrará una lista de comando que podemos utilizar.

* **Install** → Si tienes el paquete agregado a tu composer.json
* **Require** → Si no tienes configurado en JSON, utilizar este comando para añadir las dependencias que requerimos.
* **Remove** → Si ya no requieres de algún componente, con este comando los removemos del proyecto (las desinstala todo).
* **Self-update** → Nos ayuda a actualizar a composer.
* **Update** → Este comando se conenccta a internet para saber si existe alguna actualización.
* **Dump** → Para agregar composer a nuestro proyecto.



Para ver todos los comandos más utilizados podemos escribir en la terminal **composer**.

# **composer.lock**

* **composer.json** -> Es la descripción exacta de los paquetes.
* **composer.lock** -> Es la descripción exacta y el detalle de los componentes y paquetes.