

Laravel

Laravel es un popular framework de PHP. Permite el desarrollo de aplicaciones web totalmente personalizadas de elevada calidad.

Laravel es un framework PHP. Es uno de los frameworks más utilizados y de mayor comunidad en el mundo de Internet.

Como framework resulta bastante moderno y ofrece muchas utilidades potentes a los desarrolladores, que permiten agilizar el desarrollo de las aplicaciones web.

Laravel pone énfasis en la calidad del código, la facilidad de mantenimiento y escalabilidad, lo que permite realizar proyectos desde pequeños a grandes o muy grandes. Además permite y facilita el trabajo en equipo y promueve las mejores prácticas.

Las características más notables que aporta Laravel son las siguientes:

- **Blade:** Blade es un sistema de plantillas para crear vistas en Laravel. Este permite extender plantillas creadas y secciones en otras vistas en las cuales también tendremos accesibles las variables y con posibilidad de utilizar código PHP en ellas
- **Eloquent:** Eloquent es el ORM que incluye Laravel para manejar de una forma fácil y sencilla los procesos correspondientes al manejo de bases de datos en nuestro proyecto. Transforma las consultas SQL a un sistema MVC lo que no permite procesar consultas SQL directamente y así protegernos de la inyección SQL.
- **Routing:** Laravel proporciona un sistema de organización y gestión de rutas que nos permite controlar de manera exhaustiva las rutas de nuestro sistema.
- **Middlewares:** Son una especie de controladores que se ejecutan antes y después de una petición al servidor, lo que nos permite insertar múltiples controles, validaciones o procesos en estos puntos del flujo de la aplicación.
- **Comunidad y documentación:** Un gran punto a destacar de este framework es la gran comunidad y documentación que existe, una comunidad de profesionales activa que aporta conocimiento y funcionalidades, además de testear nuevas versiones y detectar fallos del framework, lo que le da seguridad al framework. Y una documentación muy completa y de calidad pensada para los propios desarrolladores.

La página oficial de Laravel es <https://laravel.com>

Actualmente Laravel está en su versión 11, pero vamos a trabajar con su versión 9, ya que XAMPP nos ofrece PHP 8.0, el cual es compatible con esta versión. Si instalamos y configuramos un servidor web y configuramos la última versión de PHP, entonces, se puede utilizar la última versión de Laravel (no es necesario).

MVC

El MVC (modelo, vista, controlador) es un patrón arquitectónico de software que separa una aplicación en tres capas descritas como su acrónimo lo indica. Laravel, así como la mayoría de frameworks en PHP implementan este patrón de diseño en donde cada capa maneja un aspecto de la aplicación. Pero antes de ver cómo Laravel está diseñado para implementar

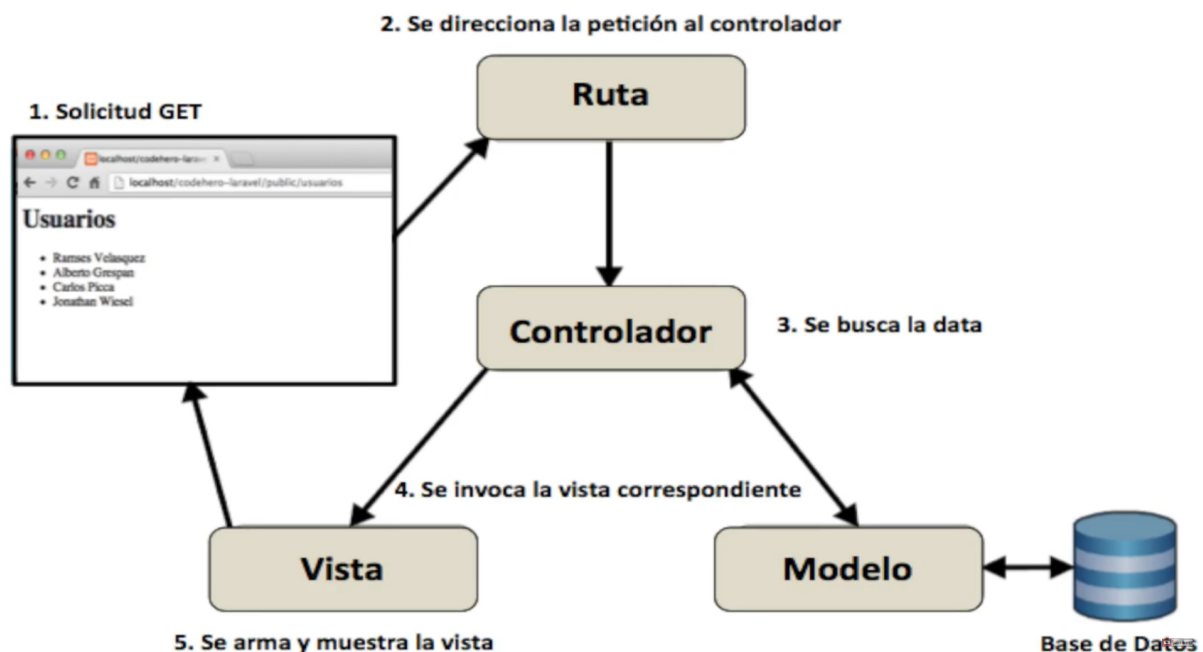
este patrón de software, vamos a tratar de dejar este concepto un poco más claro definiendo cada una de sus partes.

Modelo: Hace referencia a la estructura de datos de la aplicación. Los datos pueden ser transferidos desde la base de datos, una clase, un servicio, u otros, directamente a la vista o ser transformados en el controlador para ser actualizados nuevamente al origen.

Vista: Es la representación de la información en una interfaz de usuario. Por lo general en interfaces no estáticas se representan los datos que vienen directamente del modelo o estos son transformados en un proceso intermedio en el controlador. En vistas estáticas por lo general no hace falta que las vistas sean renderizadas con datos enviados del controlador.

Controlador: Es el lugar en donde se implementa la lógica de la aplicación, los procedimientos, algoritmos y rutinas que hacen que funcione el software. Actúa como interfaz entre los componentes de modelo y vista aplicando las transformaciones y lógica necesarias.

¿Cómo funciona Laravel?



1. Se hace una petición por medio de una ruta.
2. La ruta llama a un controlador.
3. El controlador se encarga de extraer los datos de la base de datos con ayuda del modelo.
4. El controlador manda los datos extraídos a la vista.
5. La vista retorna los datos obtenidos.

Pre-requisitos para trabajar con Laravel

En este curso trabajaremos con Laravel 9, el cual requiere de PHP 8.0.2 o superior, la extensión PDO de PHP para la conexión con la base de datos, así como de otras extensiones

Requisitos que debe cumplir el servidor:

- PHP >= 8.0.2
- Extensión BCMath para PHP
- Extensión Ctype para PHP
- Extensión DOM para PHP
- Extensión Fileinfo para PHP
- Extensión JSON para PHP
- Extensión Mbstring para PHP
- Extensión OpenSSL para PHP
- Extensión PCRE para PHP
- Extensión PDO para PHP
- Extensión Tokenizer para PHP
- Extensión XML para PHP

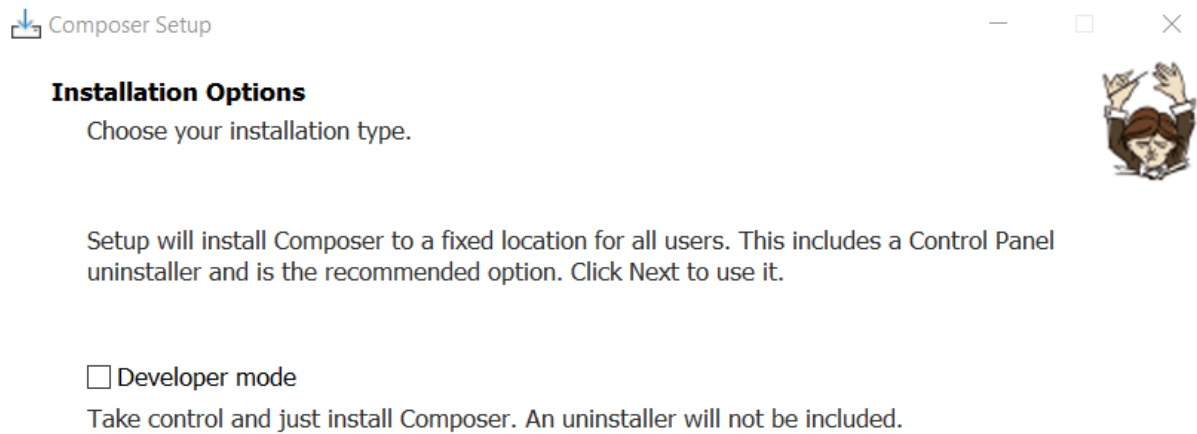
Al instalar XAMPP ya tenemos todas las extensiones necesarias para trabajar con Laravel con lo que sólo necesitaremos instalar “Composer”.

Instalación de Composer

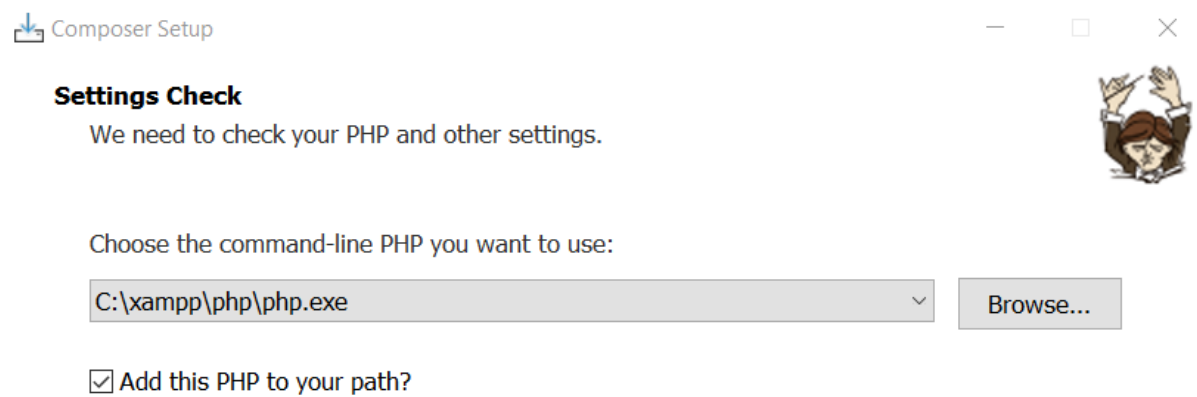
Antes de instalar el framework y componentes de PHP debemos instalar Composer. **Composer** es el manejador de dependencias para PHP. Una dependencia puede ser tanto un framework (como Laravel o Symfony) así como paquetes o componentes.

Composer nos permite instalar y actualizar el framework y todos los componentes de PHP que requiera nuestro proyecto con tan solo ejecutar un comando, es decir, sin tener que buscar y descargar archivos, descomprimirlos, copiarlos, pegarlos, etc. Esto es muy conveniente sobre todo para mantener nuestro proyecto actualizado con correcciones de bugs y parches de seguridad. Composer también nos permite detectar problemas entre nuestras dependencias, por ejemplo si una versión de un componente no es compatible con nuestra versión de PHP o del framework Laravel, etc.

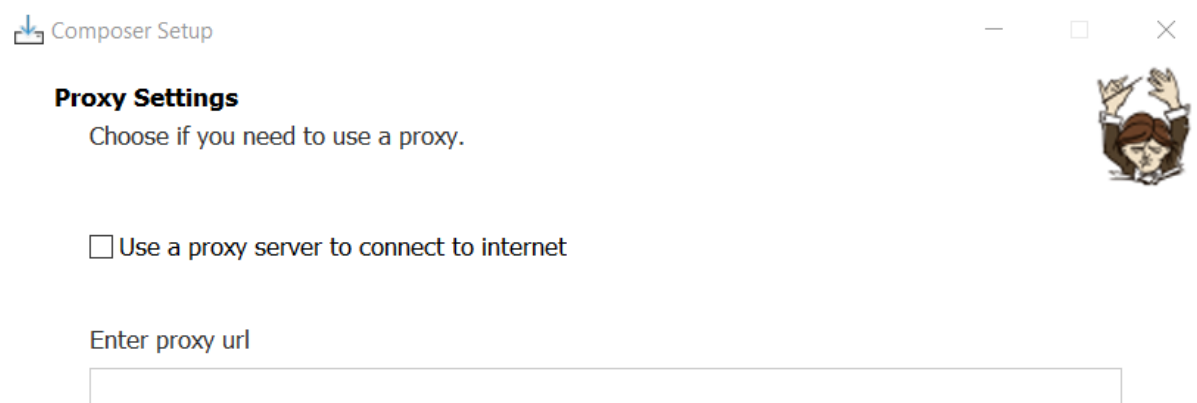
Instalar Composer en Windows es muy sencillo gracias a su instalador que puedes descargar desde el siguiente enlace: <https://getcomposer.org/Composer-Setup.exe>. Este instalador instala la última versión de Composer y configura el PATH por ti para que puedas ejecutar composer en tu consola desde cualquier directorio.



En la instalación “Developer Mode” significa que seleccionará automáticamente la ruta de PHP si tienes más de una versión de PHP. Si se prefiere una versión específica, tiene que estar desmarcada la opción.



Marcamos la opción de añadir PHP al path



No marcamos la opción de utilizar un servidor proxy para conectarse a Internet

Composer Setup

Ready to Install

Setup is now ready to download Composer and install it on your computer.



Please review these settings. Click Install to continue with the installation.

PHP version 8.1.6
C:\xampp\php\php.exe

Proxy: none

Add to System path:
C:\xampp\php

Back

Install

Cancel

Pulsamos el botón **Install** y se procederá a la instalación.

Composer Setup

Information

Please read the following information before continuing.



Important

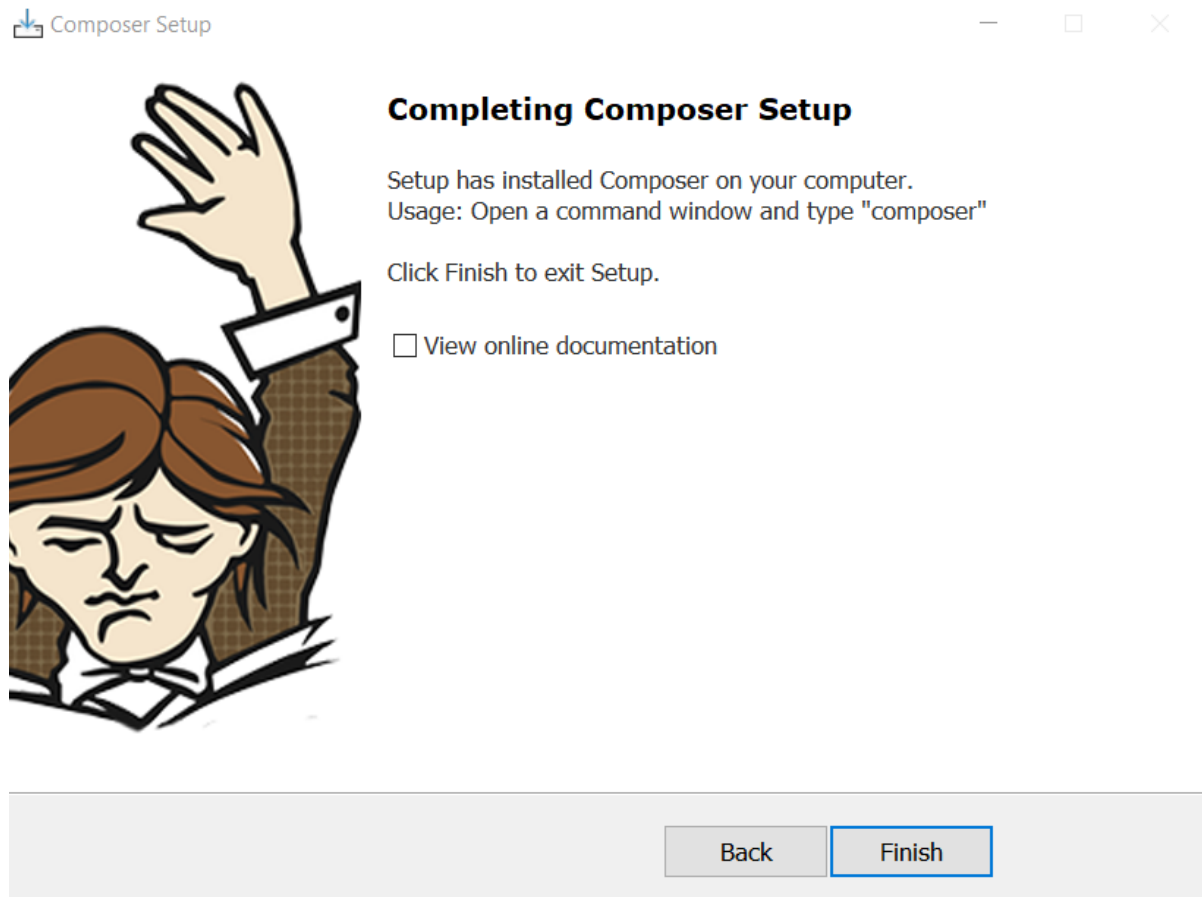
You must open a new command window to use Composer for the first time, because your environment has changed and running programs may not be aware of this.

If this does not work, you will have to do one of the following:

- Close all File Explorer windows, then open a new command window. OR
- Logoff and Logon again, then open a new command window.

As a last resort, you may need to restart your computer.

Tras finalizar la instalación se mostrará la siguiente pantalla



Después de completar la instalación, abre el símbolo del sistema. Escribe «cmd» y haz clic en OK. A continuación Ingresa el siguiente comando:

```
composer
```

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.22000.2538]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Fran>composer

Composer
Composer version 2.7.2 2024-03-11 17:12:18

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display help for the given command. When no command is given display help for the list
command
  -q, --quiet               Do not output any message
  -V, --version              Display this application version
                           Force (or disable --no-ansi) ANSI output
  -n, --no-interaction      Do not ask any interactive question
                           Display timing and memory usage information
                           Whether to disable plugins.
                           Skips the execution of all scripts defined in composer.json file.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
                           Prevent use of the cache
  -v|vv|vvv, --verbose      Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and
3 for debug
```