

Introducción

Prerequisitos

Para seguir esta guía únicamente necesitarás **conocimientos básicos/medios de PHP** y motivación para aprender. Si todavía no dispones de estos conocimientos, puedes utilizar el material gratuito disponible en <http://jonvadillo.com/learn> para comenzar tu aprendizaje.

!!! tip "Consejo" Cualquier editor de texto te servirá también para programar. No obstante, te recomiendo [VS Code](#) o [PhpStorm](#), los cuales considero sin duda alguna dos de los editores para PHP más potentes en la actualidad.

¿Qué es Laravel?

Tal y como dice la guía oficial, Laravel es un **framework de desarrollo de aplicaciones web** con una sintaxis elegante que nos permitirá desarrollar aplicaciones web de forma rápida y segura.

El objetivo de [Laravel](#) es permitir a los desarrolladores crear aplicaciones web robustas y profesionales, de forma ágil y con una estructura adecuada. Laravel facilita la implementación de cualquier funcionalidad que toda aplicación profesional pueda necesitar (interacción con bases de datos, seguridad, servicios web, etc.).

En este libro aprenderás a **crear aplicaciones web con Laravel desde cero**, desde lo más básico hasta funcionalidades más complejas que incluyan aspectos como la seguridad o control de acceso.

Características principales

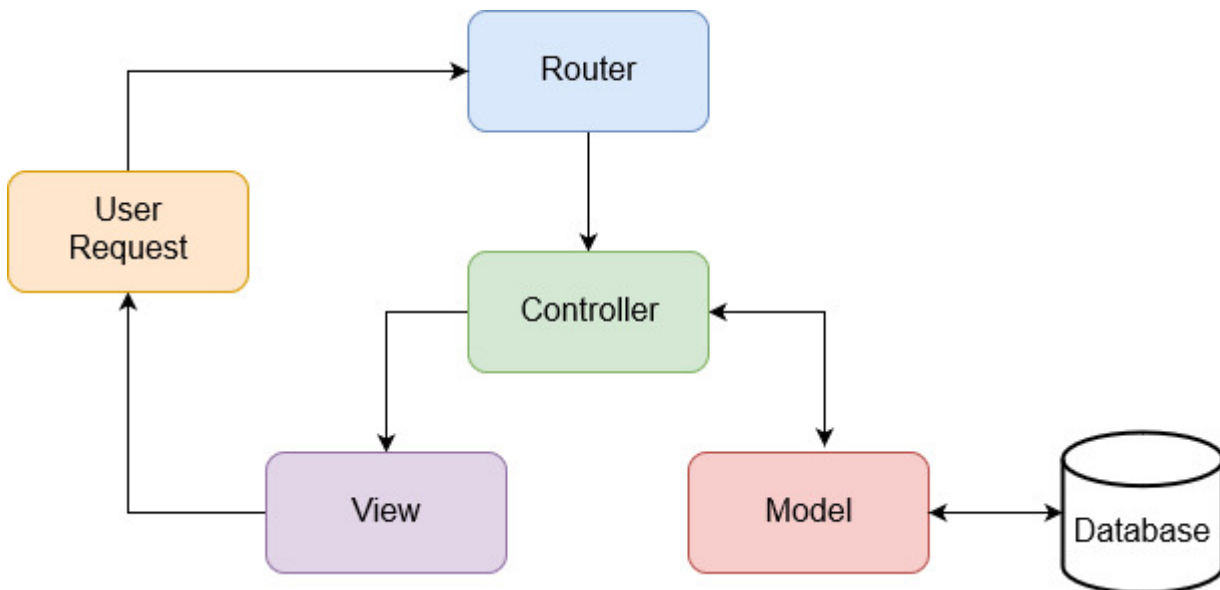
Algunas de las características de Laravel son:

- Sistema intuitivo de rutas.
- Motor de plantillas Blade para la generación de interfaces de usuario de forma flexible.
- Soporte para cualquier base de datos relacional (**SQLite, MySQL, Postgres y SQL Server**) mediante Eloquent ORM.
- Uso de la arquitectura MVC.
- Dispone de multitud de componentes capaces de resolver las problemáticas más comunes del desarrollo de aplicaciones web.
- Una grande comunidad de desarrolladores y expertos.

Ecosistema Laravel

Es importante conocer bien los actores principales del ecosistema Laravel:

- **Router:** recibe todas las peticiones y las envía al controlador adecuado (también puede ejecutar algún middleware específico antes de llamar al controlador).
- **Controladores (Controllers):** contienen toda la lógica para reaccionar a las peticiones entrantes.
- **Vistas (Views):** contienen el código HTML y separan la presentación de la lógica de la aplicación (controlador).
- **Modelos (Models):** se utilizan para interactuar con la base de datos y aplicar la lógica de negocio.



Tal y como muestra la imagen anterior, el flujo de una petición en una aplicación de Laravel sería el siguiente:

1. El punto de entrada de todas las peticiones es el archivo `public/index.php` el cual se encargará de lanzar una instancia de nuestra aplicación Laravel.
2. La petición se envía al router, el cual la reenvía al controlador correspondiente.
3. El controlador atiende la petición y realiza las acciones correspondientes (por ejemplo, puede interactuar con la base de datos para cargar o almacenar información).
4. Por último, el controlador genera la vista correspondiente y se la envía al cliente.

Estructura de carpetas

Laravel crea una estructura de ficheros como punto de partida para todo tipo de aplicaciones:

- `/app` : contiene el código fuente de la aplicación y la mayoría de las clases que crearemos. Importante comentar la carpeta `/app/Http` que almacenará los controladores de la aplicación que vayamos creando.
- `/bootstrap` : contiene el archivo `app.php` , el cual arranca el framework. También contiene la carpeta `/cache` en la que Laravel almacena los archivos generados y así optimizar su rendimiento.
- `/config` : contiene la configuración de la aplicación, de la conexión a base de datos, sistema de ficheros, etc. A pesar de ello, es conveniente realizar los cambios de configuración en el archivo `.ENV` y no en los archivos de configuración de esta carpeta.
- `/database` : contiene todos los ficheros relacionados con el almacenamiento de datos.
- `public` : contiene el fichero `index.php` , el cual será el punto de entrada a nuestra aplicación, es decir, recibirá todas las peticiones que reciba nuestra aplicación. `/public` es el único directorio accesible públicamente, por lo que contiene todos los archivos estáticos: `.js`, `.css` y las imágenes.
- `/resources` : contiene las vistas de la aplicación que se utilizarán para generar los ficheros HTML y los archivos CSS y JS no compilados (una vez compilados Laravel los guardará en la carpeta `/public`). También contiene los archivos de traducción.
- `/routes` : contiene los archivos de rutas a nuestra aplicación o API. La diferencia entre estos dos, es que el segundo no contiene información sobre las sesiones, cookies, etc.
- `/storage` : contiene las vistas compiladas y otros archivos generados por laravel. La carpeta `storage\app\public` se utilizará para almacenar los archivos subidos por nuestros usuarios (por ejemplo, fotos de perfil).