Laravel, un framework de PHP

Teléfono: 900 103 253
www.hostalia.com

En el mundo del diseño y desarrollo web, el lenguaje PHP se ha convertido en el más utilizado en todo el mundo superando con creces a otros lenguajes de programación en servidores como ASP.NET, Java J2EE o Ruby on Rails. De hecho algunos de los CMS más utilizados hoy en día, como puede ser WordPress, PrestaShop o Moodle, están desarrollados utilizando este lenguaje de programación. Ahora bien, llevar a cabo un proyecto desde cero con este lenguaje u otro cualquiera puede ser una tarea compleja y tediosa.

Para solucionar esto y hacer la vida un poco más fácil a los programadores, podemos hacer uso de frameworks que simplifican el proceso de desarrollo de un proyecto. Hoy en el White Paper de Hostalia nos centraremos en Laravel, de código abierto y de los más fáciles de asimilar para PHP.

¿Qué es un framework?

Antes de meternos de lleno con Laravel, creemos que es importante conocer bien qué es un framework y todo lo que puede ofrecer a los desarrolladores.

Un framework puede ser definido como un entorno de trabajo para el desarrollo de aplicaciones, ya sean web o de escritorio, que ofrece componentes que facilitan el trabajo a los programadores, tales como bibliotecas de funciones, uso de plantillas, administración de recursos en tiempo de ejecución y otras muchas cosas. Esto permite llevar a cabo el proyecto sin tener que escribir mucho código, consiguiendo que el trabajo sea más eficiente y recursivo (es decir, favoreciendo la reutilización de código).

La arquitectura más utilizada en la mayoría de los frameworks es conocida como MVC (Modelo-Vista-Controlador) que permite la división de cualquier proyecto en tres grandes partes:

- **Modelo**: Hace referencia a los datos de la aplicación y su reglamentación
- **Vista**: Es la forma que utilizamos para presentar los datos
- **Controlador**: Es la parte del programa encargada de procesar las peticiones de los usuarios y controlar el flujo de la ejecución del sistema

Entre las principales características que ofrece el uso de framework podemos destacar:

- Autenticación mediante login y password, que permite restringir el acceso y el tipo de permisos de los diferentes usuarios
- Configuración de acceso a los datos mediante el uso de archivos TXT o XML
- Abstracción de URLs y sesiones, encargándose el framework de su manejo y liberando de esta tarea al programador
- Internacionalización que facilita la inclusión de varios idiomas en el desarrollo
- Controladores fácilmente adaptables a las necesidades del proyecto que gestionan las peticiones y eventos

¿Qué es Laravel?

Laravel es el nombre de un framework creado para trabajar con PHP creado en el año 2011 por Taylor Otwell y que con el paso del tiempo, ha ido ganando terreno a otros framework para trabajar con PHP como Symfony o Zend Framework.

Se trata de framework de desarrollo con una curva de aprendizaje muy rápida y que maneja una sintaxis expresiva, elegante, con el objetivo de eliminar la molestia del desarrollo web facilitando las tareas comunes, como la autenticación, enrutamiento, sesiones y caché.

Proporciona potentes herramientas necesarias para construir aplicaciones robustas y que puede ser utilizado tanto para proyectos a nivel empresarial como para proyectos más sencillos, lo que significa que es perfecto para todos los tipos de proyectos.

Ventajas de utilizar Laravel en el desarrollo web

Aquellos desarrolladores que se decantan por el uso de Laravel a la hora de llevar a cabo sus proyectos obtienen las siguientes ventajas:

- Reducción de costos y tiempos en el desarrollo y posterior mantenimiento de la aplicación creada
- Curva de aprendizaje relativamente baja si se compara con otros frameworks de PHP
- Flexible y adaptable no sólo al uso del sistema MVC tradicional, sino que para reducir las líneas de código propone lo que denomina "**Routes with clousures**"
- Modular y con un amplio sistema de paquetes y drivers con el que se puede extender las funcionalidades de forma sencilla, robusta y segura

- Sencillez a la hora de utilizar los datos mediante Eloquent, que se trata de un ORM cuya interacción con las bases de datos es totalmente orientada a objetos, siendo compatible con la gran mayoría de bases de datos del mercado
- Facilita el manejo de las rutas de nuestra aplicación, así como la generación de URLs amigables que ayudan a mejorar el posicionamiento web
- Uso del sistema de plantillas Blade, que se caracterizan por ser más simples y que además incluyen un sistema de caché que las hace más rápidas
- Una gran comunidad y mucha documentación, sobre todo en su sitio oficial
- Cuenta con una herramienta de líneas de comando llamada **Artisan** que permite programar tareas programadas como por ejemplo para ejecutar migraciones, pruebas a determinadas horas...

Creación de un proyecto con Laravel

Empezar un proyecto con Laravel es un proceso muy sencillo gracias al uso de una herramienta muy útil llamada **Composer**, que no es más que un manejador de dependencias de PHP con el que poder instalar paquetes que otros usuarios comparten con la comunidad.

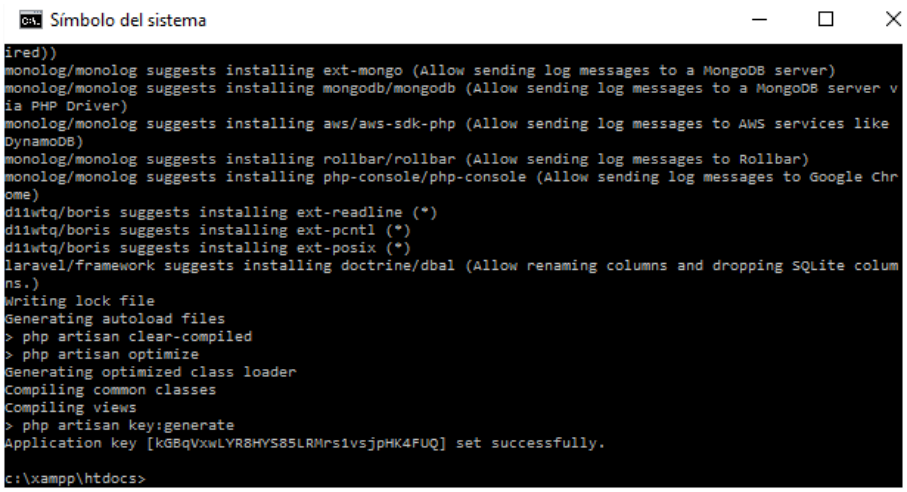
Para instalar Composer, basta con entrar a la página de Composer e ir al apartado "**Download**" para descargar su última versión. Dependiendo del sistema operativo con el que trabajemos deberemos seguir unos pasos u otros.

Una vez que lo tengamos instalado, para crear el proyecto tendremos que abrir la consola de líneas de comandos y situarnos en la carpeta donde queremos llevar a cabo nuestro desarrollo. Por ejemplo, si utilizamos una instalación XAMPP en un servidor local, tendremos que irnos a la carpeta "htdocs". Una vez allí, deberemos ejecutar la siguiente instrucción:

composer create-project laravel/laravel {directory} --prefer-dist

Donde "**{directory}**" será cambiado por el nombre del proyecto que queremos crear y donde no podremos utilizar nombres con espacios ni acentos.

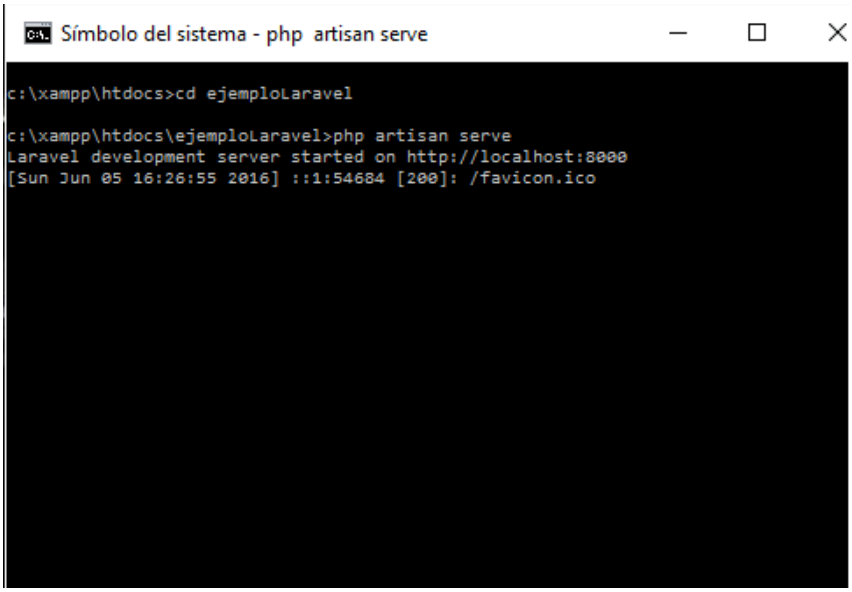
Una vez finalizado el proceso, nos debería aparecer algo parecido a lo que vemos en la siguiente imagen que os dejamos.



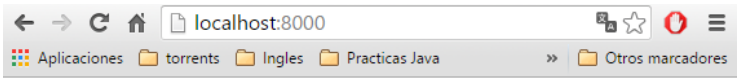
Para saber si la instalación ha sido correcta, podemos arrancar un servicio de pruebas para verlo en funcionamiento. Para ello, entraremos dentro de la carpeta de nuestro proyecto, en nuestro caso la hemos llamado "**ejemploLaravel**" y desde la línea de comandos ejecutaremos lo siguiente.

php artisan serve

Si todo ha ido bien, veremos un mensaje como el que se muestra en la siguiente pantalla.



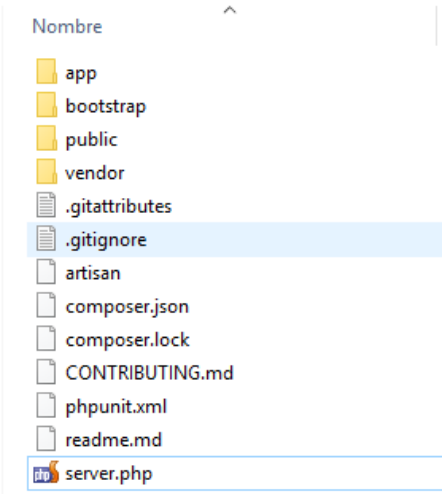
Por último, si vamos a un navegador y ponemos como dirección "**localhost:8000**" deberíamos ver una pantalla como la que os mostramos a continuación.



You have arrived.

Estructura de un proyecto Laravel

Una vez que hemos visto cómo iniciar un proyecto con Laravel, es muy importante que conozcamos la estructura de directorios que se crea y para qué está pensado cada uno de ellos. Si entramos en nuestro proyecto de ejemplo creado podremos ver la siguiente estructura:



De todos los archivos y carpetas que podemos ver en la imagen anterior, los más importantes son "**app**", "**public**" y "**vendor**".

Directorio app

Se trata de la carpeta más importante y donde se suele trabajar la mayor parte del tiempo. En su interior nos encontraremos nuestros modelos de base de datos, los controladores, las vistas o los archivos de configuración de la aplicación, todo ello ordenado en una serie de carpetas y archivos que explicamos a continuación:

- **app/router.php**: Será el archivo donde se definirán las URLs de nuestra aplicación
- **app/filter.php**: En este archivo se indicarán los distintos filtros que se aplicarán a las rutas que definamos. Por ejemplo, si se quiere acceder a una zona privada y no se está logueado, que le redirija al formulario de login
- **app/config**: En el interior de esta carpeta, nos encontraremos los archivos de configuración de nuestra aplicación
- **app/controller**: Dentro de este directorio, se almacenarán los diferentes controladores que se utilizarán para el funcionamiento de nuestra página web
- **app/database**: Se guardarán las migraciones que son versiones de la base de datos, escritas con Schema Builder propio de Laravel
- **app/models**: Esta otra carpeta contendrá los modelos que interaccionan con nuestra base de datos y que pueden ser modificados en cualquier momento
- **app/views**: Hace referencia a la vista de la aplicación, es decir, todo el código HTML que se utilizaría en la página web que fuéramos a construir

Directorio public

En el interior de este directorio, nos encontramos el archivo "**index.php**" que es el encargado de lanzar la aplicación. Además de esto, también será donde almacenaremos todos nuestros archivos CSS, JS e imágenes que utilizaremos para desarrollar la aplicación.

Directorio Vendor

En la carpeta vendor, lo que nos encontramos es todo el código correspondiente al framework Laravel, así como código de terceras personas o empresas que lo comparten con el resto de la comunidad para que lo utilicen en cualquier proyecto.

A lo largo de este White Paper hemos conocido un poco sobre el framework Laravel y las ventajas que nos encontraremos a la hora de trabajar con él. Hemos aprendido a iniciar un proyecto y el significado de la estructura de archivos y directorios que forman parte del proyecto. Lo siguiente que quedaría sería ponerse manos a la obra y empezar a picar código para hacer realidad el proyecto que tenemos en mente.