

# Presentación del curso



## 1. ¿Por qué este curso?

Desde hace muchos años, y aún hoy en día, PHP es el lenguaje de desarrollo en servidor más utilizado a nivel mundial. Prueba de ello es que alrededor del 80% de las webs que podemos encontrar en Internet se han desarrollado con este lenguaje. Además, la gran mayoría de proveedores de hosting ofrecen servicio gratuito o muy económico con PHP, lo que, en parte, facilita que el lenguaje se mantenga muy en forma hoy en día.

Sin embargo, el desarrollo de aplicaciones web empleando lenguaje PHP puede llegar a ser algo caótico si no sabemos cómo estructurar correctamente nuestra aplicación, y eso es especialmente relevante cuando el proyecto que desarrollamos es complejo. Es ahí donde los frameworks PHP entran en juego, y ofrecen al desarrollador una forma más guiada de estructurar los proyectos, además de ofrecerle una serie de herramientas que facilitan, entre otras cosas, la definición de las distintas vistas o páginas de la aplicación, o la conexión y acceso a diferentes fuentes de datos.

Por tanto, se hace necesario, por un lado, conocer las características del lenguaje PHP, y por otro, comprender el funcionamiento de un framework basado en este lenguaje, si queremos formar parte de esa gran mayoría de desarrollo en *backend* que supone el ecosistema PHP.

## 2. Recursos necesarios

A nivel de conocimientos, es conveniente tener nociones de programación tradicional, estructurada y modular, y un conocimiento a nivel intermedio del lenguaje PHP, en cuanto a definición de funciones, estructuras de control (condiciones, bucles), y también algunas nociones básicas de programación orientada a objetos en PHP, en cuanto a la definición de clases, ya que tanto Laravel como otros frameworks PHP que podamos encontrar son frameworks orientados a objetos, que trabajan con diferentes clases y espacios de nombres.

En cuanto a recursos software para seguir el curso, echad un vistazo a lo que necesitaremos instalar...

### 2.1. Máquina virtual

Con el fin de intentar homogeneizar la experiencia de todos en el curso, se os proporcionará una máquina virtual Lubuntu para VirtualBox. Sobre esa base instalaremos el software necesario para el desarrollo de dichos proyectos: Visual Studio Code, Apache, MySQL, PHP, y también el framework Laravel junto con alguna

otra herramienta adicional que nos servirá para instalar librerías adicionales, como son los gestores de paquetes *composer* y *npm*. También se os proporcionará una segunda versión de esa máquina virtual con todo este software instalado, para que sólo os tengáis que preocupar de probar los ejemplos de las sesiones, y hacer los ejercicios correspondientes. Podéis consultar el software necesario, cómo instalarlo y el acceso a las máquinas virtuales en el documento de [software necesario](#).

En cuanto a la máquina virtual proporcionada, se ha creado un usuario llamado **alumno**, con password **alumno**, con el que podremos realizar instalaciones adicionales, si queremos, a través del comando `sudo`. En la versión completa de la máquina virtual veremos el escritorio preparado para el curso, con accesos directos a los elementos que utilizaremos (navegadores, Visual Studio Code, XAMPP, etc).

### 3. ¿Cómo está estructurado el curso?

---

Los contenidos de este curso están estructurados en 8 sesiones de unas 5 horas de duración cada una, aproximadamente, para cubrir las 40 horas de curso:

1. **Introducción a los frameworks PHP y a Laravel.** Haremos una rápida introducción a lo que suponen los frameworks PHP y sus características principales, y veremos cuáles son los frameworks más populares actualmente, y la razón por la que elegimos Laravel como candidato. Después, daremos unos primeros pasos con Laravel a la hora de crear proyectos y ponerlos en marcha, viendo los elementos que componen un proyecto y sus principales características.
2. **Rutas y vistas.** En esta sesión veremos cómo organiza Laravel el acceso a las diferentes rutas que pueden componer una aplicación, y cómo empezar a construir nuestras vistas a través de HTML y el motor de plantillas Blade, integrado en Laravel.
3. **Controladores e inyección de dependencias.** Trataremos aquí el uso de controladores para separar la lógica de nuestra aplicación en módulos independientes, y veremos cuáles son las operaciones típicas que suele hacer un controlador. También trataremos el tema de la inyección de dependencias, o cómo acceder desde cualquier parte de la aplicación a recursos compartidos, como variables globales o acceso a fuentes de datos.
4. **El modelo y los datos (I). Migraciones y modelos simples.** En esta sesión tendremos una primera toma de contacto con cómo organiza Laravel los datos en una aplicación, y cómo es posible comunicarnos con una base de datos a través del ORM Eloquent, incorporado en Laravel, para gestionar tablas como si fueran colecciones de objetos. También veremos para qué sirven las migraciones, a la hora de inicializar nuestras bases de datos.
5. **El modelo y los datos (II). Seeders, factories y relaciones entre modelos.** En esta segunda sesión sobre el modelo de datos en Laravel veremos cómo establecer relaciones entre diferentes modelos de la aplicación, y cómo utilizar *seeders* y *factories* para poblar ciertas tablas con datos de antemano.
6. **Formularios y validación de datos.** Aquí veremos cómo definir formularios mediante Blade, y establecer mecanismos de validación a través de Laravel.

7. **Autenticación de usuarios.** En esta sesión trataremos los mecanismos de autenticación y validación de usuarios que ofrece Laravel en sus aplicaciones.
8. **Desarrollo de servicios REST.** Para finalizar el curso, veremos cómo podemos definir controladores que proporcionen servicios REST a los clientes que lo necesiten, en lugar de renderizar vistas más complejas y dependientes del navegador cliente.

### 3.1. Evaluación del curso

Al final de los apuntes de cada sesión habrá una serie de ejercicios obligatorios a realizar, más alguno optativo en algunos temas. Para superar el curso, deberéis entregar los ejercicios obligatorios de cada sesión en un archivo comprimido (ZIP, RAR, TGZ...) dentro de los plazos establecidos.

### 3.2. Sobre los ejemplos y archivos de prueba

En cada sesión del curso se os mostrarán algunos ejemplos de código para probar, antes de pasar a los ejercicios correspondientes que deberéis entregar. Estos ejemplos los podéis (deberíais) hacer paso a paso a medida que se van explicando, pero como guía por si acaso os perdéis en algún punto, se facilitarán completos en un archivo ZIP junto a cada sesión, para poderlos descargar y probar directamente, o bien contrastar lo que habéis hecho con lo que debería haber, y no perdáis un tiempo excesivo probando el ejemplo en sí, antes de pasar al ejercicio.

Eso es todo por ahora. Comenzamos!