



# DESARROLLO WEB ENTORNO SERVIDOR

---

Arkaitz Rodriguez

[arkaitz.rodmar@educa.jcyl.es](mailto:arkaitz.rodmar@educa.jcyl.es)

23-09-2020

# INDICE

Protocolos de comunicaciones: IP, TCP, HTTP, HTTPS .....	4
Modelo de comunicaciones cliente – servidor y su relación con las aplicaciones web .....	5
Estudio sobre los métodos de petición HTTP /HTTPS más utilizados .....	6
Modelo de desarrollo de aplicaciones multicapa – comunicación entre capas – componentes –funcionalidad de cada capa.....	7
Modelo de división funcional front-end / back-end para aplicaciones web .....	8
Página web estática – página web dinámica – aplicación web – mashup.....	9
Componentes de una aplicación web .....	10
Programas ejecutados en el lado del cliente y programas ejecutados en el lado del servidor – lenguajes de programación utilizados en cada caso.....	11
Lenguajes de programación utilizados en el lado servidor de una aplicación web (características y grado de implantación actual).....	12
Características y posibilidades de desarrollo de una plataforma XAMPP .....	13
En que casos es necesaria la instalación de la máquina virtual Java (JVM) y el software JDK en el entorno de desarrollo y en el entorno de explotación .....	14
IDE más utilizados (características y grado de implantación actual).....	15
Servidores HTTP /HTTPS más utilizados (características y grado de implantación actual).....	16
Apache HTTP vs Apache Tomcat.....	17
Navegadores HTTP /HTTPS más utilizados (características y grado de implantación actual).....	18
Generadores de documentación HTML (PHPDoc): PHPDocumentor, ApiGen, .....	19
Repositorios de software – sistemas de control de versiones: GIT , CVS, Subversion...	20
Propuesta de configuración del entorno de desarrollo para la asignatura de Desarrollo web del lado servidor en este curso (incluyendo las versiones): xxx-USED y xxx-WXED .....	21
Propuesta de configuración del entorno de explotación para la asignatura de Desarrollo web del lado servidor en este curso (incluyendo las versiones): xxx-USEE.....	22
Realizar un estudio sobre los siguientes conceptos y su relación con el desarrollo de aplicaciones web .....	23

CMS – Sistema de gestión de contenidos .....	23
ERP – Sistema de planificación de los recursos empresariales .....	23
Elegir y realizar un estudio y una presentación para la exposición del trabajo sobre una de las siguientes arquitecturas de desarrollo de Aplicaciones Web: Laravel .....	24
¿Qué es Laravel? .....	24
¿Qué objetivo tiene? .....	25
¿Cómo funciona Laravel? Características .....	26
¿Por qué elegir Laravel? .....	27
¿Cuándo utilizar Laravel y no un CMS? .....	28
¿Ventajas e inconvenientes de Laravel? .....	29

### ***Protocolos de comunicaciones: IP, TCP, HTTP, HTTPS***

IP: Protocolo de comunicación de datos digitales clasificado funcionalmente en la capa de red según el modelo internacional OSI. Su función principal es el uso bidireccional en origen o destino de comunicación para transmitir datos.

TCP: Protocolo de control de transmisión, garantiza que los datos serán entregados en su destino sin errores y en el mismo orden de transmisión.

HTTP: Protocolo de transferencia de hipertexto, permite las transferencias de información en la www.

HTTPS: Protocolo seguro de transferencia de hipertexto, esta basado en el protocolo HTTP, y es destinado a la transferencia segura de datos, es la versión segura de HTTP.

SOCKET: Relación entre dos puntos en internet, ip:protocolo-ip:protocolo, siendo uno el cliente y otro el servidor, en el modelo cliente-servidor.

## ***Modelo de comunicaciones cliente – servidor y su relación con las aplicaciones web***

La arquitectura cliente-servidor es un modelo de diseño de software en el que las tareas se reparten entre los servidores y los clientes. Un cliente realiza peticiones al servidor, y este, le da respuesta.

La relación es que la aplicación web utiliza protocolo HTTP y HTTPS, y los dos protocolos son modelo cliente-servidor, en un equipo se carga la información, y en el otro equipo se ejecuta.

## ***Estudio sobre los métodos de petición HTTP /HTTPS más utilizados***

Los métodos mas utilizados son:

- Get: Se utiliza cuando se necesita un archivo o un recurso de un servidor web, devuelve la cabecera y el recurso.
- Put: Crea o carga un recurso nuevo en el servidor, si ya existe, lo reemplaza.
- Delete: Borra un recurso solicitado al servidor web.
- Post: Envía información a un recurso web del servidor.
- Head: Es similar al método GET, la diferencia es que HEAD solicita los metadatos del recurso y no todo el elemento.
- Trace: Diagnostico de errores o detección de servidores intermedios en la conexión mediante la monitorización de mensajes
- Options: Describe opciones de comunicación del recurso de destino
- Patch: Se utiliza para aplicar modificaciones parciales a un recurso
- Connect: Establece el túnel hacia el servidor identificado por el recurso

## ***Modelo de desarrollo de aplicaciones multicapa – comunicación entre capas – componentes –funcionalidad de cada capa***

Consiste en que la aplicación no se desarrolle en una capa, es decir, que no esté guardada en una carpeta, sino cada archivo en la carpeta que corresponde, la comunicación entre capas en MVC, el controlador es el encargado de realizar la comunicación entre el modelo y la vista.

Las funcionalidades vienen cuando por ejemplo hay un error en la vista, y tu tienes que buscar ese error, entonces si tienes desarrollado el modelo multicapa en la aplicación web, sería mucho mas fácil encontrar el error, ya que sabes donde estaría o entre que archivos podría estar, de la otra forma sería mucho mas difícil.

## ***Modelo de división funcional front-end / back-end para aplicaciones web***

Front-end: Parte de la aplicación web, que se desarrolla para usuarios normales o invitados.

Back-end: Parte de la aplicación web, que desarrollamos para usuario con unos permisos especiales.



## ***Página web estática – página web dinámica – aplicación web – mashup***

Página web estática: Es una página en la cual yo entre desde un equipo u otro siempre va a ser la misma página, y la visualización del contenido va a ser la misma.

Página web dinámica: Es una página que se puede abrir en dos equipos diferentes, y no tiene porque visualizarse lo mismo.

Aplicación web: Programa que se codifica en un lenguaje interpretado y que se puede utilizar accediendo al servidor web mediante un cliente.

Mashup: Página web híbrida que consta de otras aplicaciones web o web services, con el fin de reutilizar su contenido o funcionalidad.

## ***Componentes de una aplicación web***

Todas las aplicaciones web tienen tres componentes primarios:

- Cliente: Un navegador web que será el encargado de enviar peticiones al servidor web.(Mozilla Firefox)
- Servidor web de aplicaciones: Este servidor es el encargado de ejecutar ciertas aplicaciones pedidas por el cliente.(Apache)
- Servidor de bases de datos: Es el encargado de la persistencia de los datos de una aplicación web.(MYSQL)

***Programas ejecutados en el lado del cliente y programas ejecutados en el lado del servidor – lenguajes de programación utilizados en cada caso***

Los programas ejecutados en el lado del cliente son aquellos que mandan peticiones a los servidores web mediante una aplicación web, los programas ejecutados en el lado del servidor son aquellos que el servidor ejecuta por si mismo, a diferencia que el cliente, que es mandado por el servidor.

Lenguajes cliente:

- JAVASCRIPT
- HTML

Lenguajes servidor:

- PHP
- ASP

***Lenguajes de programación utilizados en el lado servidor de una aplicación web (características y grado de implantación actual)***

Enlace

## ***Características y posibilidades de desarrollo de una plataforma XAMPP***

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.

***En que casos es necesaria la instalación de la máquina virtual Java (JVM) y el software JDK en el entorno de desarrollo y en el entorno de explotación***

El JVM siempre será necesaria, cuando este escribiendo código java y quiero compilarlo, y el JDK siempre es necesario que uses java

## *IDE más utilizados (características y grado de implantación actual)*

- Eclipse: Es un IDE muy popular de código abierto y ampliamente extendido, desarrollado por IBM
- NetBeans: Es un IDE basado en código abierto, es el IDE oficial de Java
- IntelliJ Idea: Es un IDE que cuenta con 2 versiones, la community edition que era libre de descarga y la ultimate edition que necesita suscripción anual, Node JS, PHP, Python...

***Servidores HTTP /HTTPS más utilizados (características y grado de implantación actual)***

APACHE - ORACLE

TOMCAT - ORACLE

IIS - MICROSOFT



## ***Apache HTTP vs Apache Tomcat***

Tomcat puede ejecutar varias especificaciones diferentes de Java, como Java Servlet, JavaServer Pages (JSP), Java EL y WebSocket. Tomcat se creó específicamente para aplicaciones Java, mientras que Apache es un servidor HTTP de propósito general

## ***Navegadores HTTP /HTTPS más utilizados (características y grado de implantación actual)***

- Google Chrome
- Firefox
- Opera
- Safari
- Internet Explorer

***Generadores de documentación HTML (PHPDoc):  
PHPDocumentor, ApiGen, ...***

Esta pregunta es para mas adelante.

## ***Repositorios de software – sistemas de control de versiones: GIT , CVS, Subversion...***

Los repositorios son una manera de almacenar los archivos digitales en un equipo de trabajo, para la facilitación del trabajo en grupo

***Propuesta de configuración del entorno de desarrollo para la asignatura de Desarrollo web del lado servidor en este curso (incluyendo las versiones): xxx-USED y xxx-WXED***

Sistema operativo: Ubuntu Server/Windows (RAM, Disco, Particiones, Cuentas, Password, Red)

Servidor administración remota :SSH

Servidor de transferencia de ficheros :SFTP (SSH)

Repositorio: GIT Lab, GIT Hub

Servidor Web: Apache HTTP (mod\_php, mod\_ssl,...), XAMPP, Apache Tomcat (JRE, JDK)(Actualizados) (para JSP, Servlets,EJB)

SGBD: Mariadb/ MySQL, Phpmyadmin, Workbench...

Navegador: Mozilla, Chrome, ... (Actualizados)

IDE: NetBeans / NotePad++ /Eclipse / Microsoft Visual Estudio

Ofimática, multimedia, generador html,...: Libre office, Audacity, Gimp, Openshot, Dreamweaver, Microsoft FrontPage, ...

Framewoks PHP: Symfony / CakePHP

Cliente ssh: NetBeans / Notepad++ / FileZilla client / Putty

***Propuesta de configuración del entorno de explotación para la asignatura de Desarrollo web del lado servidor en este curso (incluyendo las versiones): xxx-USEE***

Sistema operativo: Ubuntu Server/Windows (RAM, Disco, Particiones, Cuentas, Password, Red)

Servidor administración remota :SSH

Servidor de transferencia de ficheros :SFTP (SSH)

Repositorio: GIT Lab, GIT Hub

Servidor Web: Apache HTTP (mod\_php, mod\_ssl,...), XAMPP, Apache Tomcat (JRE, JDK)(Actualizados) (para JSP, Servlets,EJB)

SGBD: Mariadb/ MySQL, Phpmyadmin, Workbench...

Cliente ssh: NetBeans / Notepad++ / FileZilla client / Putty

Navegador: Mozilla, Chrome, ... (Actualizados)

***Realizar un estudio sobre los siguientes conceptos y su relación con el desarrollo de aplicaciones web***

***CMS – Sistema de gestión de contenidos***

***ERP – Sistema de planificación de los recursos empresariales***

CMS: Es un sistema en el cual puedes hacer una aplicación web sin escribir ninguna línea de código

- WORDPRESS

- Joomla

ERP: Son programas que se hacen cargo de las operaciones internas de una empresa, como las nominas o los clientes dados de alta

- NAVISION

- SAP

***Elegir y realizar un estudio y una presentación para la exposición del trabajo sobre una de las siguientes arquitecturas de desarrollo de Aplicaciones Web: Laravel***

### ***¿Qué es Laravel?***

Es un framework de código abierto para PHP, es simple, potente y tiene una interfaz elegante. Se creó el 9 de Junio de 2011 por Taylor Otwell.



### *¿Qué objetivo tiene?*

Laravel quiere ser un framework que permita el uso de una sintaxis elegante y expresiva, para poder crear código de forma sencilla, permitiendo multitud de funcionalidades

Intenta aprovechar lo mejor de otros frameworks y aprovecha las características de las ultimas versiones de PHP

## *¿Cómo funciona Laravel? Características*

### Blade:

- Es un sistema de plantillas para crear vistas en Laravel, 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:

- 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 unos 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

### *¿Por qué elegir Laravel?*

La potencia radica en su integración, escalabilidad y facilidad de mantenimiento respecto a otros desarrollos en lenguajes nativos

### *¿Cuándo utilizar Laravel y no un CMS?*

Lo utilizaremos cuando:

- Los CMS no cubran nuestras necesidades
- Necesitemos un desarrollo ágil, seguro y con una fácil escalabilidad
- Puede necesitar una aplicación web que este integrada con otras aplicaciones(Google, servicios de mailing, ERP's)

## *¿Ventajas e inconvenientes de Laravel?*

### Ventajas:

- Documentación espléndida
- Artisan, un CLI que comprende herramientas avanzadas para hacer tareas y migraciones
- Utiliza el motor de plantillas Blade para acelerar las tareas de compilación, y se pueden incluir nuevas características de una manera sencilla

### Inconvenientes:

- Es muy lento
- Los desarrolladores amateurs suelen liarse extendiendo código y clases
- No cuenta con una gran comunidad detrás, comparado con otras plataformas
- Mucho métodos incluidos en el enrutamiento inverso son complejos