



# Desarrollo web Entorno Servidor

## Tema 1

Cristina Manjon Lacalle

Fecha: 25/09/2020

I.E.S. Los Sauces

## Sumario

1. Protocolos de comunicaciones: IP, TCP, HTTP, HTTPS.....	4
2. Modelo de comunicaciones cliente – servidor .....	4
3. Estudio sobre los métodos de petición HTTP /HTTPS.....	4
4. Modelo de desarrollo de aplicaciones multicapa.....	5
5. Modelo de división funcional front-end / back-end .....	5
6. Página web estática – página web dinámica – aplicación web – mashup.....	5
7. Componentes de una aplicación web.....	6
8. Programas ejecutados en el lado del cliente y en el lado del servidor.....	6
9. Lenguajes de programación utilizados en el lado servidor.....	6
10. Características y posibilidades de desarrollo de una plataforma XAMPP.....	7
11. Instalación de la máquina virtual Java (JVM) y el software JDK .....	7
12. IDE más utilizados (características y grado de implantación actual).....	8
13. Servidores HTTP /HTTPS más utilizados.....	9
14. Apache HTTP vs Apache Tomcat.....	10
15. Navegadores HTTP /HTTPS más utilizados.....	10
16. Generadores de documentación HTML (PHPDoc).....	11
17. Repositorios de software – sistemas de control de versiones.....	11
18. Propuesta de configuración del entorno de desarrollo.....	13
19. Propuesta de configuración del entorno de explotación .....	14
20. Elegir y realizar un estudio y una presentación .....	14

Realiza un estudio sobre los siguientes conceptos:

### 1. Protocolos de comunicaciones: IP, TCP, HTTP, HTTPS.

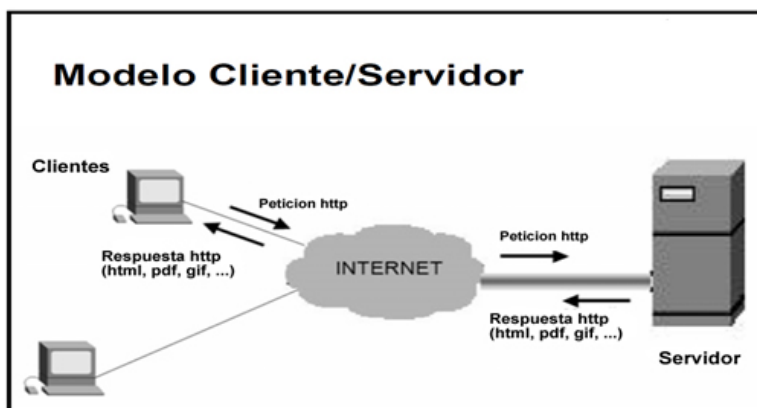
**IP:** protocolo de internet.

**HTTP:** protocolo de transferencia de hipertexto, que es popular porque se utiliza para acceder a las páginas web.

**HTTPS:** Protocolo seguro de transferencia de hipertexto

**TCP:** Protocolo de control de transmisión

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



**Cliente:** Navegador

**Servidor:** Servidor web

**Protocolos:** HTTP – HTTPS

TCP

IP

**Lenguaje de marcas:** Páginas web

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

El método **GET** solicita una representación de un recurso específico. Las peticiones que usan el método GET sólo deben recuperar datos.

El método **HEAD** pide una respuesta idéntica a la de una petición GET, pero sin el cuerpo de la respuesta.

El método **POST** se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor.

El modo **PUT** reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición.

El método **DELETE** borra un recurso en específico.

El método **CONNECT** establece un túnel hacia el servidor identificado por el recurso.

El método **OPTIONS** es utilizado para describir las opciones de comunicación para el recurso de destino.

El método **TRACE** realiza una prueba de bucle de retorno de mensaje a lo largo de la ruta al recurso de destino.

El método **PATCH** es utilizado para aplicar modificaciones parciales a un recurso.

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

**Capa de presentación:** esta capa es la que ve el usuario.

**Capa de negocio:** conoce y gestiona las funcionalidades que esperamos del sistema o aplicación web

**Capa de acceso a datos:** donde residen los datos y la encargada de acceder a los mismos.

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

**Una parte externa o front-end,** que es el conjunto de páginas que ven la gran mayoría de usuarios que las usan (usuarios externos).

**Una parte interna o back-end,** que es otro conjunto de páginas dinámicas que utilizan las personas que producen el contenido y las que administran la aplicación web (usuarios internos) para crear contenido, organizarlo, decidir la apariencia externa, etc.

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

**Una Web estática** es aquella enfocada a mostrar una información **permanente**, donde el navegante se limita a obtener dicha información, sin poder interactuar con la página web visitada.

**Una web dinámica** contiene aplicaciones dentro de la propia web, otorgando mayor **interactividad** con el usuario.

**Mashup – Aplicación Web Híbrida:** es una aplicación que combina servicios de varias páginas web en una sola para ofrecer un **nuevo servicio**

## 7. Componentes de una aplicación web.

- ◆ Un servidor web para recibir las peticiones de los clientes web (normalmente navegadores) y enviarles la página que solicitan
- ◆ El módulo encargado de ejecutar el código o programa y generar la página web resultante.
- ◆ Una aplicación de base de datos, que normalmente también será un servidor.
- ◆ El lenguaje de programación que utilizarás para desarrollar las aplicaciones

## 8. Programas ejecutados en el lado del cliente y programas ejecutados en el lado del servidor - lenguajes de programación utilizados en cada caso.

### LADO SERVIDOR:

Navegadores, Entornos de programación IDE, lenguajes de programación como Java, PHP, Perl, ASP.NET, un servidor HTTP que puede ser Apache , Nginx, Tomcat.

### LADO CLIENTE

Lenguaje de JavaScript, editores html, css, editores de documentos, software para tratamiento de imágenes, software para creación y administración de base de datos, software para transferencia de archivos SFTP

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

**PHP** : es uno de los lenguajes más extendidos actualmente. Sus características de gratuidad, código abierto, la posibilidad de ser portado y ejecutado en diferentes plataformas

**ASP.NET** : utilizado por programadores de todo el mundo para la creación y el desarrollo de sitios web dinámicos, los sitios web creados con este lenguaje de programación del lado del servidor, se pueden ejecutar en todos los navegadores del mercado.

Una de las principales ventajas es que permite el control de usuario personalizado y, además, ofrece una división entre la capa de diseño y el código. En cuanto a las desventajas, la más destacada es que consume una gran cantidad de recursos.

**Perl:** PERL es un lenguaje de programación basado en una serie de herramientas UNIX. Su principal punto fuerte es que resulta muy útil en el procesamiento tanto de textos como de archivos.

Es uno de los lenguajes del lado del servidor preferidos por parte de los programadores ya que el desarrollo de aplicaciones con PERL es muy rápido. Además, ofrece una gran libertad para crear y desarrollar el programa “a medida”.

**Python:** Un lenguaje de programación multiplataforma y multiparadigma, que también es de propósito general y el año pasado ha superado al que conocerás en el puesto número. Su simpleza, legibilidad y similitud con el idioma inglés lo convierten en un gran lenguaje ideal para principiantes

**JSP :** No debemos confundirlo con Java. Son lenguajes distintos. Javascript es un lenguaje de programación que puede ser utilizado para crear programas que luego son acoplados a una página web o dentro de programas más grandes. Sirve para crear efectos y realizar acciones interactivas.

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

**XAMPP** es un servidor independiente de plataforma de código libre. Te permite instalar de forma sencilla Apache en tu propio ordenador, sin importar tu sistema operativo (Linux, Windows, MAC o Solaris). Y lo mejor de todo es que su uso es gratuito.

XAMPP incluye además servidores de bases de datos como MySQL y SQLite con sus respectivos gestores phpMyAdmin y phpSQLiteAdmin. Incorpora también el intérprete de PHP, el intérprete de Perl, servidores de FTP como ProFTPD ó FileZilla FTP Serve, etc. entre muchas cosas más.

Sin embargo, es bueno acotar que la seguridad de datos no es su punto fuerte, por lo cual no es suficientemente seguro para ambientes grandes o de producción.

## **11. En qué 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.**

Máquina virtual JVM cuando escribo y compilo, entonces sería para el **entorno de desarrollo**

Software JDK cuando voy a interpretar java, para el **entorno de explotación**



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

**NETBEANS** : entorno de desarrollo integrado basado en código abierto, gracias a su enfoque podemos desarrollar nuestras aplicaciones como módulos (como un componente de un software). Lo cual resulta sumamente cómodo pues podemos reutilizar nuestro código para diferentes proyectos las veces que queramos sin mayor complicación.

NetBeans es el IDE oficial de Java pero también podemos desarrollar en otros lenguajes como PHP, C, C++ e incluso HTML 5

**ECPLISE** : Es un IDE muy popular de código abierto y ampliamente extendido. Fue desarrollador originalmente por IBM aunque actualmente cuenta con una gran comunidad que le da soporte con una gigantesca biblioteca de plugins. Es multiplataforma (Windows, Linux y Mac) y dispone de nuevas versiones cada año. Con Eclipse es posible desarrollar todo tipo de aplicaciones para web, escritorio o móviles usando Java, C, C++, JSP, perl, Python, Ruby y PHP.

**IntelliJ Idea**: podemos programar sobre diferentes lenguajes y proporciona soporte para trabajar con JAVA , Node JS, PHP, Python, Ruby, Sass, TypeScript, AngularJS, CoffeeScript, CSS, HTML, JavaScript, LESS, etc.

**BlueJ**: Como los IDE anteriores de esta lista, es multiplataforma. Podemos usarlo sobre Windows, macOS y GNU/Linux y cuenta con una gran documentación que lo hace especialmente indicado para estudiantes.

Su principal característica es que nos permite generar una vista UML de todos los objetos que componen nuestra aplicación.

**Codenvy**: es un IDE comercial de Java que está construido sobre el Eclipse Che. Lo que hace interesante a Codenvy es que está pensado para ofrecer un espacio de trabajo en la nube, colaborativo y bajo demanda. Además de Java, Codenvy proporciona soporte para una gran cantidad de lenguajes de programación interpretados y compilados. Esto incluye C ++, C #, JavaScript y PHP.

**Xcode**: Algunos de los aspectos más destacados importantes de Xcode son un depurador incorporado, un generador de GUI y un autocompletado de perfiles. Además de Java, Xcode proporciona soporte para AppleScript, C, C ++, Objective-C, Objective-C ++, Python, Ruby y Swift.

**JGRASP :** Los anteriores IDE tienen un “defecto” común. Todos devoran ingentes cantidades de memoria RAM. Si buscas algo ligero para comenzar tu desarrollo quizá jGRASP pueda ayudarte. Con jGRASP podemos desarrollar además aplicaciones basadas en C, CPP, Objective-C, Python, ADA y VHD.

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

**Apache http:** Es un proyecto open-source de larga trayectoria que nació en 1996 y durante más de dos décadas lideró el mercado de servidores y aunque hoy día a perdido algo de popularidad al diversificarse el mercado sigue siendo el servidor web de código abierto más conocido.

**Nginx:** es un servidor de software libre y de código abierto, que está licenciado bajo una licencia BSD, es un servidor multiplataforma, por lo que corre en sistemas GNU/Linux, BSD, Solaris, Mac OS X) así como también en sistemas Microsoft Windows.

**LiteSpeed :** primer servidor licenciado de esta lista, nacido en 2003 y comercializado por LiteSpeedTech encontramos en LiteSpeed es un servidor comercial muy veloz y eficiente que compite directamente con Nginx en cuestión de rendimiento.

Gracias a su completa integración con cPanel es un servidor candidato para sustituir Apache en servicios de web hosting que utilizan cPanel como panel de control

**Microsoft IIS:** En el mundo web ISS nunca fue muy popular, su principal nicho de mercado siempre estuvo en el mundo empresarial y hoy día se ha popularizado para ofrecer servicios en la nube, principalmente en Azure la plataforma de Cloud Hosting de Microsoft.

**Lighttpd :** Lanzado en 2003 fue una de las primeras alternativas para la lentitud de Apache, diseñado para ser rápido, consumir muy pocos recursos de CPU y RAM, además de permitir miles de conexiones simultáneas, es software libre y se distribuye bajo la licencia BSD.

**Caddy :** Es uno de los tipos de servidores web más nuevos y que mayores novedades incluye, Caddy nacido en 2015 aunque ya está siendo conocido y haciéndose de un nombre, como curiosidad está escrito en GO un lenguaje joven de Google surgido hace pocos años.

**Tomcat:** Lanzado en 1999 Apache Tomcat es un servidor desarrollado por Apache Software Foundation. Tomcat no es un servidor HTTP/HTTPS convencional sino un



servidor para contenedores web para soportar servlets Java y JavaServer Pages (JSP), tampoco hay que confundirlos con un servidor de aplicaciones Java tipo JBoss.

**NodeJS:** no es muy diferente de Tomcat. Node es un entorno en tiempo de ejecución multiplataforma basado en JavaScript y entre sus módulos podemos encontrar Node HTTP module o Express que permiten montar un servicio HTTP/HTTPS

**Cherokee:** es un servidor que apareció en 2001, es software libre y multiplataforma, su objetivo primordial es ser liviano, al menos más que la competencia, por lo que está escrito completamente en C. Lo interesante de este servidor es que además puede integrarse embebido en otro software y además soporta complementos para aumentar sus capacidades

**GWS:** servidor que casi nadie ha oído nombrar, básicamente porque no se puede descargar ni usar por particulares ni empresas y aun así es uno de los servidores web más usados de la web.

Es uno de los tipos de servidores web más usados, aunque casi nunca sabes que lo estás usando.

Información : <https://hostingdiario.com/tipos-de-servidores-web/>

#### 14. Apache HTTP vs Apache Tomcat

-El nombre

**-El propósito** es la diferencia fundamental entre **Tomcat** y el servidor web **Apache**.

El propósito del servidor **HTTP Apache** es simplemente servir archivos estáticos como texto, HTML, imágenes, archivos de audio y video a clientes basados en la web. Por el contrario, el servidor **Apache Tomcat** entrega contenido que cambia según quién es el cliente, si el cliente ha iniciado sesión y qué ha hecho en interacciones anteriores con el servidor.

Aquí dejo una página muy interesante para saber más :

[Apache Tomcat y http](#)

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

**Google Chrome:** Creado por google, es uno de los más conocidos y usados, básicamente porque es el que asegura una velocidad mayor.



Algunos de estos son

**GIT** Con Git vamos a poder controlar todos los cambios que se hacen en nuestra aplicación y en nuestro código y vamos a tener control absoluto de todo lo que pasa en el código, pudiendo volver atrás en el tiempo, pudiendo abrir diferentes ramas de desarrollo, etc.

#### **Debes decantarte por Git cuando**

- No quieres depender de una conexión de red permanente, pues quieres trabajar en tu proyecto desde cualquier lugar.
- Quieres seguridad en caso de fallo o pérdida de los repositorios principales.
- No necesitas contar con permisos especiales de lectura y escritura para los diferentes directorios (aunque, de ser así, será posible y complejo implementarlo).
- La transmisión rápida de los cambios es una de tus prioridades.

**Subversion** se basa en un sistema de control de versiones centralizado. Esto significa que existe un almacén central de datos (el repositorio) accesible a todos los usuarios. Dado que los cambios realizados no pueden ser fusionados entre sí, el sistema evita que dos usuarios puedan editar un mismo archivo al mismo tiempo.

#### **Subversion será la opción indicada, si :**

- Necesitas permisos de acceso basados en rutas de acceso para las diferentes áreas de tu proyecto.
- Deseas agrupar todo tu trabajo en un solo lugar.
- Trabajas con numerosos archivos binarios de gran tamaño.
- También quieres guardar la estructura de los directorios vacíos (estos son rechazados por Git, debido a que no contienen ningún tipo de contenido).

[Sus diferencias se pueden ver aquí](#)

	SVN	Git
Control de versiones	Centralizada	Distribuida
Repositorio	Un repositorio central donde se generan copias de trabajo	Copias locales del repositorio en las que se trabaja directamente
Autorización de acceso	Dependiendo de la ruta de acceso	Para la totalidad del directorio
Seguimiento de cambios	Basado en archivos	Basado en contenido
Historial de cambios	Solo en el repositorio completo, las copias de trabajo incluyen únicamente la versión más reciente	Tanto el repositorio como las copias de trabajo individuales incluyen el historial completo
Conectividad de red	Con cada acceso	Solo necesario para la sincronización

**18. 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-W7ED. tecnologías o programas para instalar en esas maquinas Ide, so, versiones etc, ubuntu server20.4, windows 10 64 bits educación**

- **SO:** En nuestro caso Ubuntu Server y Windows10
- **Navegadores :** como Chrome, mozilla, edge..
- **Servicio SSH :** para conexión remota
- **Servicio SFTP :** y una aplicación como Filezilla para usarlo
- **Servicio APACHE2:** servidor web de HTTP
- **Implementación de PHP :** el lenguaje de programación con módulos que necesitaremos

- Extension de php **XDEBUG**: para depurar código y errores
- Repositorio de **GitHub** : para alojar proyecto y el control de versiones de estos.
- Servidor **Base de Datos**: Mysql o Mariadb, y phpmyadmin
- Entorno de desarrollo integrado **IDE**: como puede ser NetBeans, Eclipse

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

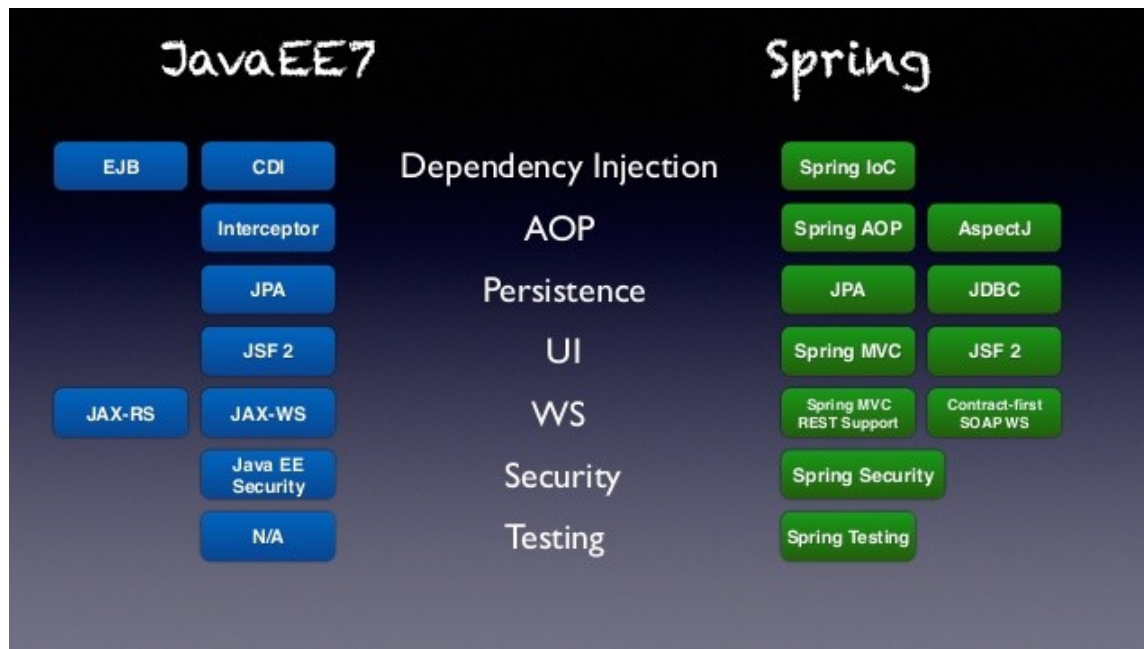
- SO: En nuestro caso Ubuntu Server y Windows10
- **Navegadores** : como Chrome, mozilla, edge..
- Servicio **SSH** : para conexión remota
- Servicio **SFTP** : y una aplicación como Filezilla para usarlo
- Servicio **APACHE2**: servidor web de HTTP
- Repositorio de **GitHub** : para alojar proyecto y el control de versiones de estos.
- Servidor **Base de Datos**: Mysql o Mariadb, y phpmyadmin

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

- MEAN (con MongoDB y con MySQL)

MEAN (formado por MongoDB, Express, AngularJS y NodeJS) al stack o compilación de estas tecnologías para desarrollar aplicaciones web usando como único lenguaje de programación javascript en el servidor, en el cliente y en la base de datos.

- Java EE vs Spring:



- Microsoft .NET

Es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permite un rápido desarrollo de aplicaciones. Basada en ella, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado.

- Angular 7

Es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página.

- Symfony

Es un framework diseñado para desarrollar aplicaciones web basado en el patrón Modelo Vista Controlador. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

- Laravel

Es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5 y PHP 7. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti"



- CakePHP

Es un framework para el desarrollo aplicaciones web escrito en PHP, creado sobre los conceptos de Ruby on Rails.

- CodeIgniter

Es un framework para el desarrollo de aplicaciones en php, que utiliza el MVC. Esto permite a los programadores o desarrolladores Web mejorar su forma de trabajar, además de dar una mayor velocidad a la hora de crear páginas Webs.