



TEMA 1 – DESARROLLO WEB EN ENTORNO SERVIDOR

DAW2 - DWES

CRISTIAN MATEOS VEGA

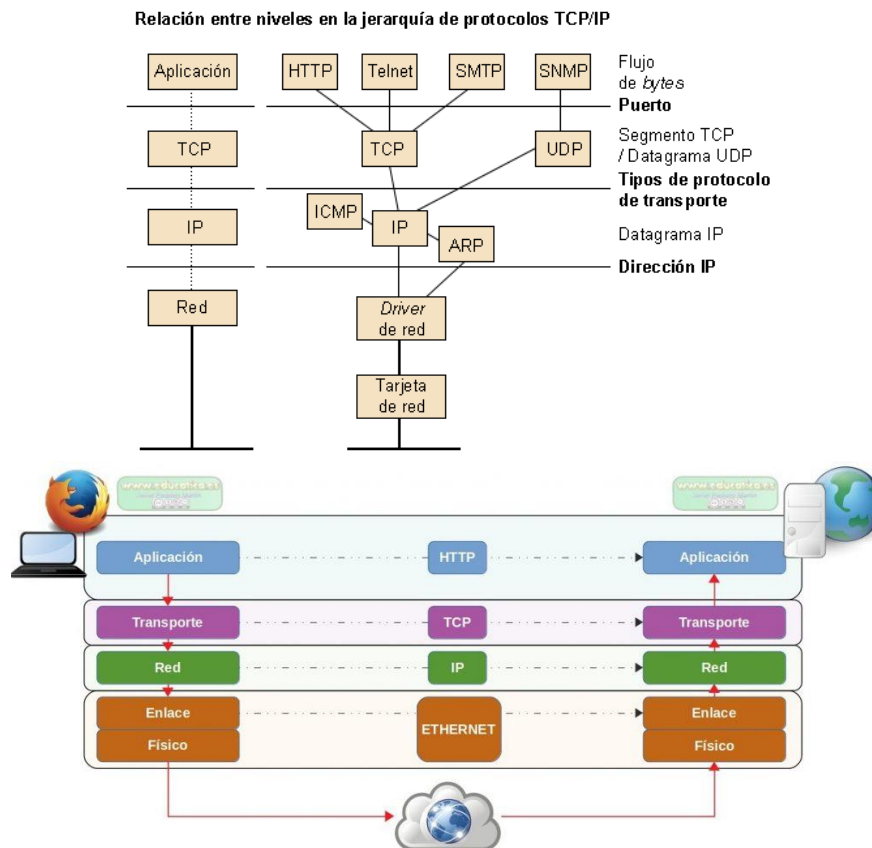
Contenido

1. Protocolos de comunicaciones: IP, TCP, HTTP, HTTPS.	3
2. Modelo de comunicaciones cliente – servidor y su relación con las aplicaciones web.	3
3. Estudio sobre los métodos de petición HTTP /HTTPS más utilizados.	4
4. Estudio sobre el concepto de URI (Identificador de Recursos Uniforme) /URL/URN, estructura, utilidad y relación con el protocolo HTTP/HTTPS.	5
5. Modelo de desarrollo de aplicaciones multicapa – comunicación entre capas – componentes – funcionalidad de cada capa.	6
6. Modelo de división funcional front-end / back-end para aplicaciones web.	6
7. Página web estática – página web dinámica – aplicación web – mashup.	7
8. Componentes de una aplicación web.	7
9. Programas ejecutados en el lado del cliente y programas ejecutados en el lado del servidor - lenguajes de programación utilizados en cada caso.	8
10. Lenguajes de programación utilizados en el lado servidor de una aplicación web (características y grado de implantación actual).	8
11. Características y posibilidades de desarrollo de una plataforma XAMPP.	8
12. 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.	9
13. IDE más utilizados (características y grado de implantación actual).	9
14. Servidores HTTP /HTTPS más utilizados (características y grado de implantación actual).	10
15. Apache HTTP vs Apache Tomcat	11
16. Navegadores HTTP /HTTPS más utilizados (características y grado de implantación actual).	11
17. Generadores de documentación HTML (PHPDoc): PHPDocumentor, ApiGen,	11
18. Repositorios de software – sistemas de control de versiones: GIT, CVS, Subversion,	11
19. 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.	11
20. Propuesta de configuración del entorno de explotación para la asignatura de Desarrollo web del lado servidor en este curso (incluyendo las versiones): xxx-USEE.	11
21. Realizar un estudio sobre los siguientes conceptos y su relación con el desarrollo de aplicaciones web:	11
CMS – Sistema de gestión de contenidos:	11
ERP – Sistema de planificación de los recursos empresariales.	11
22. Elegir y realizar un estudio y una presentación para la exposición del trabajo sobre una de las siguientes arquitecturas de desarrollo de Aplicaciones Web:	12
• MEAN (con MongoDB y con MySQL)	12
• Java EE vs Spring	12

• Microsoft .NET	12
• Angular 7	12
• Symfony	12
• Laravel	12
• CakePHP	12
• CodeIgniter.....	12

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

- **IP:** Se encarga de la dirección y envío de datos entre dispositivos. Usa direcciones IP para identificar origen y destino y no garantiza la entrega. (Capa de internet)
- **TCP:** Establece la conexión fiable entre dos dispositivos. Controla orden, integridad y transmisión de los datos y los divide en segmentos asegurando que lleguen completos y en orden. (Capa de transporte)
- **HTTP:** Es el protocolo que usan los navegadores y servidores web para pedir y enviar páginas web en texto plano. (Capa de aplicación y usa el puerto 80)
- **HTTPS:** Es lo mismo que http, pero con seguridad, cifrando la información pedida y enviada. (Capa de aplicación y usa el puerto 443)



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

Cliente: Es un proceso sencillo que solicita recursos o servicios

Servidor: Es un proceso que ofrece esos recursos o servicios que se han pedido

Funcionamiento:

1. El cliente (navegador) envía una solicitud al servidor (por ejemplo, pedir una página web) mediante HTTP/HTTPS.
2. El servidor procesa esa solicitud, accede a la información necesaria y envía la respuesta (la página web) al cliente.
3. El cliente recibe y muestra esa información para que el usuario pueda interactuar con ella.

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

- **GET:** Solicita la representación de un recurso específico. Es el método más usado para obtener datos

Características:

- No tiene cuerpo en la petición
- Si se hace la misma petición varias veces devuelve lo mismo
- Los parámetros suelen enviarse en la URL
- Se usa para obtener páginas web con recursos estáticos (imágenes, CSS...)

- **POST:** Envía datos al servidor para crear o modificar recursos.

Características:

- Puede contener un cuerpo con datos (JSON, formularios, archivos...)
- Si se hace la misma petición varias veces puede devolver cosas diferentes
- Se usa para enviar formularios, subir archivos, crear registros en una base de datos...

- **PUT:** Reemplaza un recurso existente o crea uno nuevo si no existe.

Características:

- El cuerpo contiene el recurso completo
- Actualiza completamente un recurso

- **DELETE:** Elimina un recurso específico.

Características:

- Se usa para eliminar recursos como usuarios o archivos

- **HEAD:** Obtiene los encabezados de una respuesta, sin el cuerpo del recurso.

Características:

- Sin cuerpo
- Se trata igual que el GET, pero omite el cuerpo
- Es muy eficiente para comprobar el estado de un recurso sin descargarlo completamente

Tecsify presenta...

Métodos de petición HTTP

El protocolo HTTP regula la forma en la que el cliente realiza peticiones y la forma en la que responde el servidor, para esto emplea diferentes métodos de petición.

A continuación, te contaremos más detalles acerca de estos métodos.

GET: El método GET solicita una entidad específica. Las peticiones que usan el método GET solo deben recuperar datos.	HEAD: Este método pide una respuesta idéntica a la de una petición GET, pero sin el cuerpo de la respuesta, únicamente con el encabezado de la solicitud.	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.
PUT: Es similar al POST, solo que el método PUT se utiliza para la actualización de información existente, es semejante a un UPDATE a nivel de base de datos.	DELETE: Permite eliminar un recurso específico, generalmente se utiliza para eliminar información existente, es semejante a un DELETE a nivel de base de datos.	PATCH: Este método se emplea generalmente para realizar modificaciones parciales de un recurso en particular.

Los métodos **PUT & DELETE** son **idempotentes**; es decir, puede ser ejecutados **varias veces** y tener el mismo efecto, caso contrario a un **POST** que cada vez que se ejecuta realiza la agregación de un **nuevo objeto**.

Algunos métodos solo pueden aplicarse en ciertos contextos, por ejemplo, el método **CONNECT**, que crea una conexión directa y protegida por medio de un proxy (tunneling).

[Tecsify](#) [Tecsify](#) [Tecsify](#) [Tecsify](#) [Tecsify](#) [Tecsify](#) [Tecsify](#) [Tecsify](#) [Tecsify](#) [Tecsify](#)

[www.Tecsify.com/blog](#)

[Explicación detallada](#)

4. Estudio sobre el concepto de URI (Identificador de Recursos Uniforme) /URL/URN, estructura, utilidad y relación con el protocolo HTTP/HTTPS.

- **URI (Identificador de Recursos Uniforme):** Es una cadena de caracteres que identifica de manera única un recurso de internet o en cualquier red

Ejemplo:

<https://www.ejemplo.com>

[Explicación detallada de URI](#)

- **URL (Localizador de Recursos Uniforme):** Es un tipo específico de URI que no solo identifica un recurso, sino que también proporciona los medios para localizarlo, es decir, la dirección o ruta para acceder al recurso)

Ejemplo:

<https://www.ejemplo.com/index.html>

[Explicación detallada de URL](#)



- **URN (Nombre de Recursos Uniforme):** Es otro tipo específico de URI que identifica un recurso de manera única sin referirse a su ubicación, es decir, es un nombre permanente para el recurso. Estos tienen unos estándares de identificación específicos, como por ejemplo ISBN para los libros

Ejemplo:

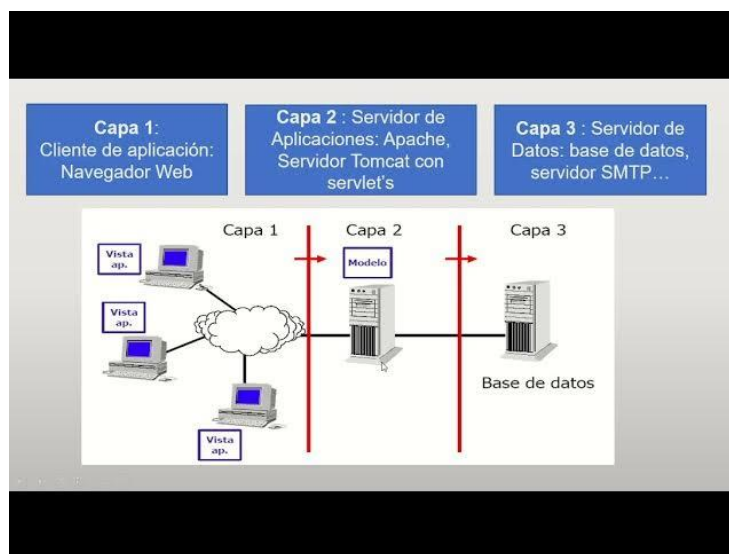
`urn:isbn:979-10-91414-08-1`

[Explicación detallada de URN](#)

5. Modelo de desarrollo de aplicaciones multicapa – comunicación entre capas – componentes – funcionalidad de cada capa.

El modelo de desarrollo de aplicaciones multicapa es una forma de organizar una aplicación dividiéndola en varias capas o niveles, donde cada capa tiene una función específica. Esto facilita el mantenimiento, la escalabilidad y la reutilización del código.

- **Capa de Presentación:** Se encarga de mostrar y captar la información
 - Interfaz con el usuario.
 - Muestra datos y recibe inputs.
- **Capa de Lógica de Negocio:** Toma decisiones y ejecuta reglas
 - Procesa la información.
 - Contiene las reglas y operaciones del negocio.
- **Capa de Datos:** Guarda y recupera la información
 - Gestiona el almacenamiento y recuperación de datos.
 - Se conecta con bases de datos o servicios externos.



6. Modelo de división funcional front-end / back-end para aplicaciones web.

1. Front-end (lado cliente)

- Es la parte que ve y usa el usuario directamente en su navegador.
- Incluye todo lo relacionado con la interfaz visual: diseño, botones, menús, animaciones, formularios, etc.
- Se desarrolla principalmente con HTML, CSS y JavaScript.
- Su función es presentar los datos y facilitar la interacción del usuario con la aplicación.
- Puede comunicarse con el back-end para pedir o enviar datos.

2. Back-end (lado servidor)

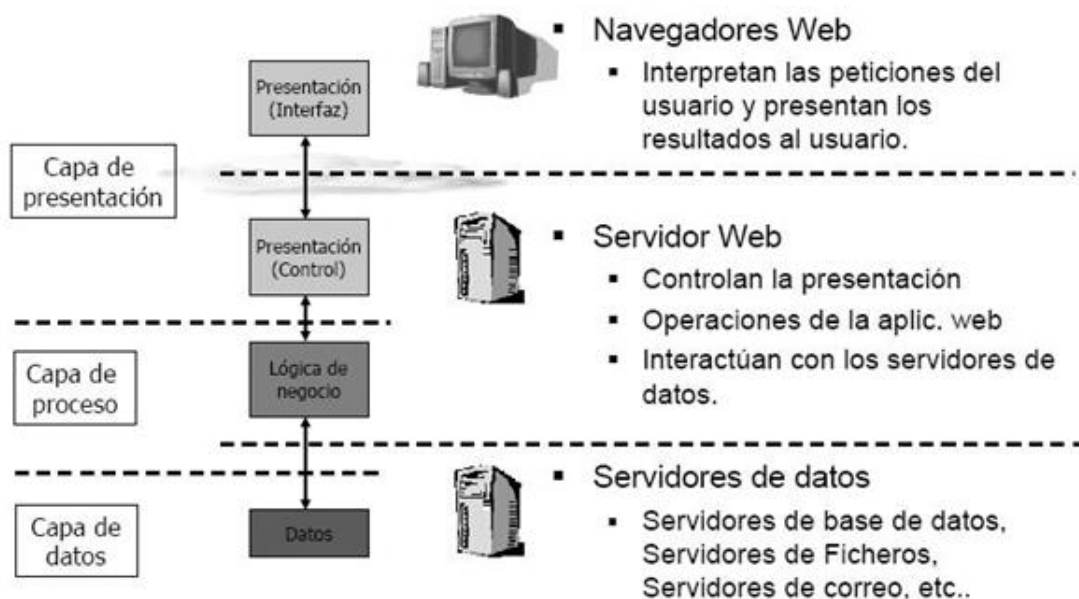
- Es la parte que corre en el servidor y se encarga de la lógica, gestión de datos y seguridad.
- Procesa las solicitudes que llegan desde el front-end, accede a bases de datos, realiza cálculos o validaciones, y prepara las respuestas.
- Se desarrolla con lenguajes como Python, Java, PHP, Node.js, Ruby...
- Proporciona servicios web que el front-end consume para mostrar información actualizada.

7. Página web estática – página web dinámica – aplicación web – mashup.

- **Página web estática:** página con contenido fijo que no cambia a menos que se modifique manualmente.
- **Página web dinámica:** página web cuyo contenido puede cambiar en función del usuario o de datos externos, generalmente mediante programación del lado del servidor.
- **Aplicación Web:** programa completo que se ejecuta con un navegador.
- **Mashup:** aplicación web que combina datos o funcionalidades de dos o más fuentes externas para crear un nuevo servicio.

8. Componentes de una aplicación web.

- **Navegador:** Programa que permite a los usuarios acceder y visualizar páginas web (ej. Chrome, Firefox).
- **Servidor Web:** Computadora o software que almacena, procesa y entrega páginas web a los navegadores.
- **Módulo encargado de ejecutar el código:** Componente del servidor que interpreta y ejecuta código dinámico (ej. PHP, Node.js).
- **Base de datos:** Sistema que almacena y organiza información para ser consultada y manipulada por la web.
- **Control de acceso:** Mecanismo que gestiona permisos y autentica usuarios para restringir acceso a recursos.
- **Ficheros escritos en lenguajes de programación:** Archivos con código fuente que definen la funcionalidad y lógica de la aplicación web.



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

Programas del lado del cliente:

Estos programas se ejecutan directamente en el navegador o dispositivo del usuario. Su función principal es la interacción con el usuario y la manipulación de la interfaz.

Características:

- Se ejecutan en el navegador o dispositivo del usuario.
- Permiten interacción directa con la interfaz.
- No necesitan comunicación con el servidor para funcionar
- Seguridad limitada, ya que el código es visible y accesible para el usuario.

Lenguajes comunes: JavaScript, HTML y CSS, TypeScript...

Programas del lado del servidor:

Estos programas se ejecutan en el servidor, procesan datos, gestionan bases de datos, autenticación, lógica de negocio, etc.

Características:

- Ejecutados en un servidor remoto.
- Control total sobre la lógica, seguridad y acceso a datos.
- Pueden interactuar con bases de datos, servicios externos, etc.
- No son visibles directamente para el usuario final.

Lenguajes comunes: JavaScript (Node.js), Python, PHP, Ruby, Java...

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

1. PHP (73,4%)
2. Ruby (6,4%)
3. Java (5,3%)
4. JavaScript (5%)
5. ASP.NET (4,8%)

11. Características y posibilidades de desarrollo de una plataforma XAMPP.

Características:

- Facilidad de uso y configuración
- Portabilidad y compatibilidad multiplataforma
- Soporte para múltiples lenguajes de programación y bases de datos

Posibilidades de Desarrollo:

- Desarrollo de sitios web dinámicos
- Pruebas locales de aplicaciones web
- Integración con CMS populares como WordPress
- Desarrollo de scripts automatizados
- Configuración personalizada
- Desarrollo colaborativo en local
- Migración y respaldo de bases de datos y proyectos entre entornos

12. 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.

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

1. Visual Studio Code (70%)

El IDE de Visual Studio es una interfaz de desarrollo integrada impulsada por Microsoft desarrollada para ayudar a los desarrolladores de software con los desarrollos web. El IDE utiliza funciones de inteligencia artificial para aprender de la edición del programador en sus códigos, lo que facilita completar líneas de código automáticamente.

Una de las características principales que a muchos desarrolladores les ha gustado de Visual Studio es que ayuda al desarrollo colaborativo entre equipos en el desarrollo en vivo. Esta característica es muy importante, especialmente durante el proceso de depuración. El IDE también permite a los usuarios compartir servidores, comentarios y terminales.

Además, Visual Studio tiene la capacidad de admitir el desarrollo de aplicaciones móviles, web y juegos. También es compatible con el lenguaje Python, Node.js, ASP.NET y Azure. Con Visual Studio, los desarrolladores pueden crear fácilmente un entorno de desarrollo en la nube.

2. IntelliJ IDEA (25%)

Ha existido durante años y ha servido como uno de los mejores IDE para la programación Java. La interfaz de usuario de IntelliJ Idea está diseñada de una manera elegante que hace que la codificación sea atractiva para muchos desarrolladores de Java.

Con este IDE, el código se puede indexar, proporcionando sugerencias relevantes para ayudar a completar las líneas de código. También lleva esta codificación sugerente más allá al automatizar varias tareas que pueden ser repetitivas.

Además de ser compatible con la programación Java web, empresarial y móvil, también es una buena opción para la programación JavaScript, SQL y JPQL.

3. Eclipse (10-15%)

Eclipse es uno de los IDE más populares. Es una herramienta multiplataforma con una potente interfaz de usuario que admite arrastrar y soltar. El IDE también incluye algunas características importantes, como herramientas de análisis estático, capacidades de depuración y creación de perfiles. Eclipse es compatible con el desarrollo empresarial y permite a los desarrolladores trabajar fácilmente en el desarrollo de software escalable y de código abierto.

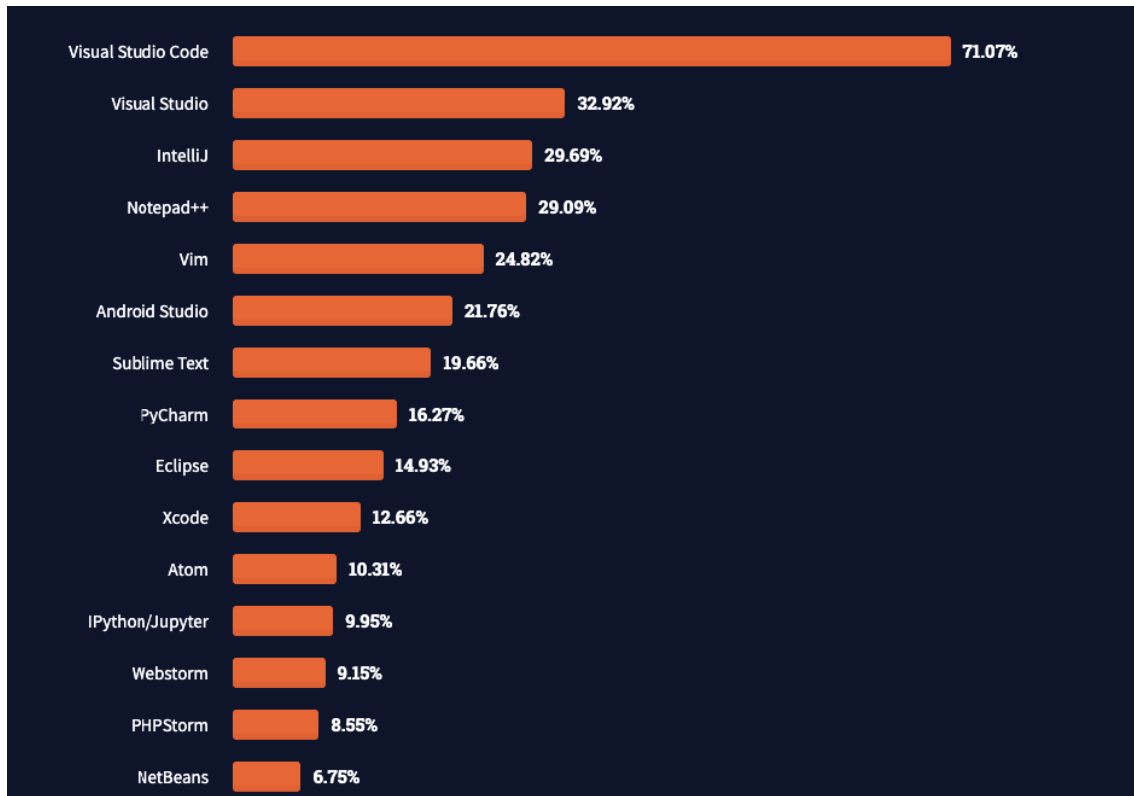
Aunque Eclipse se asocia mejor con Java, también admite varios lenguajes de programación. Además, los usuarios pueden agregar sus complementos preferidos al IDE para respaldar proyectos de desarrollo de software.

4. Android Studio (20%)

Android Studio es uno de los mejores IDE para el desarrollo de aplicaciones de Android. Este IDE es compatible con los lenguajes de programación Kotlin y Java.

Algunas funciones importantes que los usuarios pueden obtener de Android Studio son alertas automáticas, integraciones de cámara y otras funciones de tecnología móvil.

Los desarrolladores también pueden crear variantes y diferentes APK con la ayuda de este IDE flexible, que también ofrece compatibilidad con plantillas ampliadas para los servicios de Google.



[Estos y más IDEs](#)

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

1. Nginx (33,3%)

- Alta eficiencia para manejar tráfico masivo
- Rendimiento ligero y asíncrono
- Funciona también como proxy

2. Apache (25,5%)

- Gran compatibilidad con varias tecnologías y lenguajes
- Facilita la gestión de contenidos dinámicos

3. Cloudflare Server (24,5%)

- Mejora la seguridad, acelera la carga de las páginas y además protege de ataques

4. LiteSpeed (14,8%)

- Muy rápido y eficiente para sitios con mucho tráfico
- Funciona muy bien sobre todo con PHP y WordPress

5. Node.js (5%)

- Usa JavaScript para crear aplicaciones web interactivas y en tiempo real
- Soporta muchas conexiones simultaneas

15. Apache HTTP vs Apache Tomcat

Apache HTTP: Es un servidor web estándar para páginas estáticas, usa php

Apache Tomcat: Es un servidor para páginas dinámicas basadas en servlets y JSP (JavaServer Pages).

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

1. Chrome (69,23%)
2. Safari (14,98%)
3. Edge (5,03%)
4. Firefox (2,26%)
5. Samsung Internet (1,97%)
6. Opera (1,85%)

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

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

19. 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.

[Configuración del servidor](#)

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

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

CMS – Sistema de gestión de contenidos:

Es un software que permite a los usuarios crear, gestionar, editar y publicar contenido en un sitio web sin necesidad de programar esa web.

Ejemplos: WordPress, Joomla, Shopify, Wix...

ERP – Sistema de planificación de los recursos empresariales

Es un software que unifica y automatiza todos los procesos de una empresa en una sola plataforma utilizando una base de datos centralizada.

Ejemplos: SAP, Microsoft Dynamics...

22. 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)
- Java EE vs Spring
- Microsoft .NET
- Angular 7
- Symfony
- Laravel
- CakePHP
- CodeIgniter

Me gustaría estudiar WordPress