

Tema 1 - Desarrollo web en entorno servidor.

2º DAW

I.E.S. Los Sauces

Erika Martínez Pérez

*Desarrollo web en entorno de
servidor*

1. Protocolos de comunicaciones: IP, TCP, HTTP, HTTP/2, HTTPS	3
2. Modelo de comunicaciones cliente - servidor y su relación con las aplicaciones web.	4
3. Estudio sobre los métodos de petición HTTP/HTTPS más utilizados.	4
4. Estudio sobre el concepto 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.	5
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.	9
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.	10
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.	12
20. Propuesta de configuración del entorno de explotación para la asignatura de Desarrollo web del lado servidor en este curso.	12
21. Realizar un estudio sobre los siguientes conceptos y su relación con el desarrollo de aplicaciones web:	12

1. Protocolos de comunicaciones: IP, TCP, HTTP, HTTP/2, HTTPS

Protocolo de internet o IP (Internet Protocol), protocolo de comunicaciones de datos digitales clasificado por el modelo internacional OSI en la capa de red.

Más información: https://es.wikipedia.org/wiki/Protocolo_de_internet

Protocolo de control de transmisión o TCP (Transmission Control Protocol), protocolo que crea las conexiones en una red compuesta por ordenadores para intercambiar datos, clasificado en la capa de transporte por el modelo internacional OSI.

Más información:

https://es.wikipedia.org/wiki/Protocolo_de_control_de_transmisi%C3%B3n

Protocolo de transferencia de hipertexto o HTTP (Hypertext Transfer Protocol), protocolo de comunicación para la transferencia de información a través de archivos (XML, HTML, ...) que utiliza el puerto 80.

Más información:

https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto

Protocolo seguro de transferencia de hipertexto o HTTPS (Hypertext Transfer Protocol Secure) protocolo de aplicación basado en el protocolo http que utiliza el cifrado de la seguridad de texto SSL/TLS que utiliza el puerto 443.

Más información:

https://es.wikipedia.org/wiki/Protocolo_seguro_de_transferencia_de_hipertexto



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

El modelo de comunicaciones cliente - servidor, es un modelo de diseño software en el que participan dos partes esenciales clientes (demandantes de recursos o servicios) y los servidores (proveedores de los recursos específicos), el servidor web pone a disposición las aplicaciones web a las que acceden los clientes desde un navegador.

Más información: <https://es.wikipedia.org/wiki/Cliente-servidor> y <https://definicion.de/cliente-servidor/#:~:text=El%20servidor%20web%20pone%20a,navegador%20instala%20en%20su%20computadora>



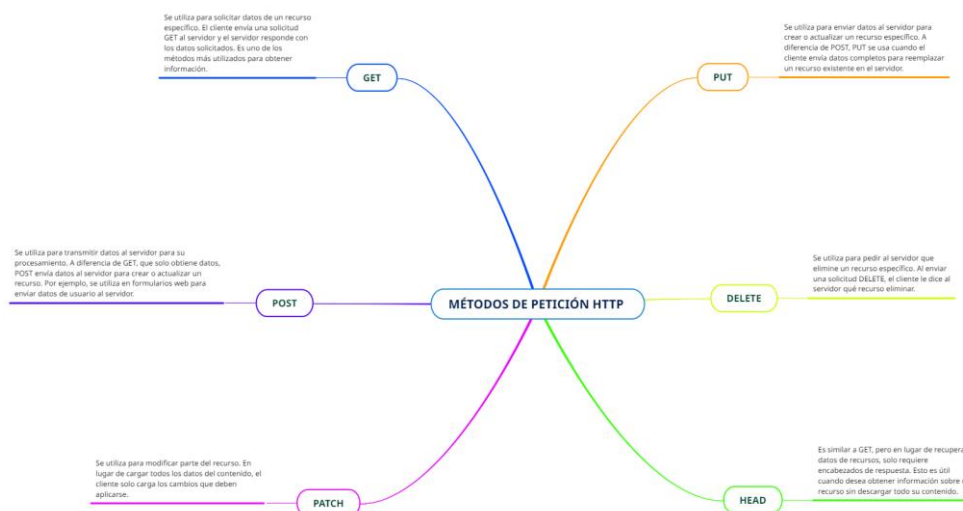
3. Estudio sobre los métodos de petición HTTP/HTTPS más utilizados.

Los métodos de petición del protocolo HTTP/HTTPS son:

- a) GET: método para solicitar la representación de un recurso específico.
- b) HEAD: método para solicitar una respuesta a la petición GET, pero sin el cuerpo de la respuesta.
- c) POST: método de envío de una entidad a un recurso específico.

Más información:

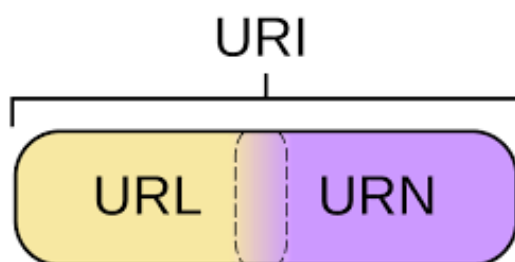
<https://developer.mozilla.org/es/docs/Web/HTTP/Methods> y <https://keepcoding.io/blog/metodos-de-peticion-http/>



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

El identificador uniforme de recursos (URI en sus siglas en inglés: Uniform Resource Identifier) identifica un recurso por su nombre, por su ubicación o por ambos.

La URI está compuesta por una parte URL (Uniform Resource Locator) y una parte URN (Uniform Resource Name).



Partes de una URL

`http://www.sitioweb.com/carpeta/pagina/#etiqueta`

1 2 3 4 5 6 7

1. Protocolo.
2. Subdominio.
3. Dominio.
4. TLD.
5. Carpeta o subcarpeta.
6. Página.
7. Etiqueta.

Todas las URLs utilizan los protocolos HTTP con el puerto 80 y HTTPS con el puerto 443, para indicar como se accederá al recurso que contienen la aplicación web.

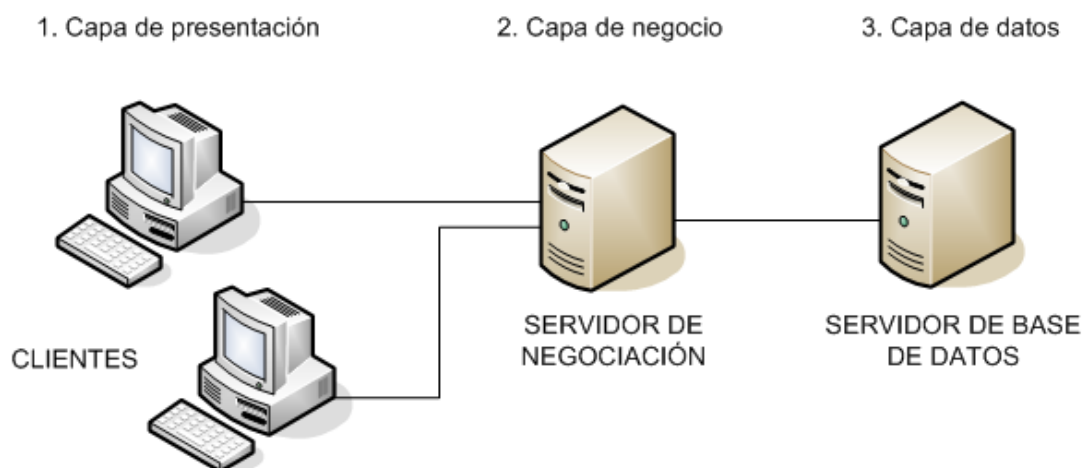
Más información:

https://es.wikipedia.org/wiki/Identificador_de_recursos_uniforme

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

Es una arquitectura cliente-servidor en la que las funciones de presentación, lógica de negocio y gestión de datos están separadas físicamente.

- Capa de presentación: contiene la interfaz gráfica y recoge la información creada por la interacción del usuario.
- Capa de negocio: gestiona las funciones de la aplicación web, y puede estar situada tanto de lado de cliente como de servidor.
- Capa de persistencia: contiene los datos almacenandolos y dando el acceso a los mismos.

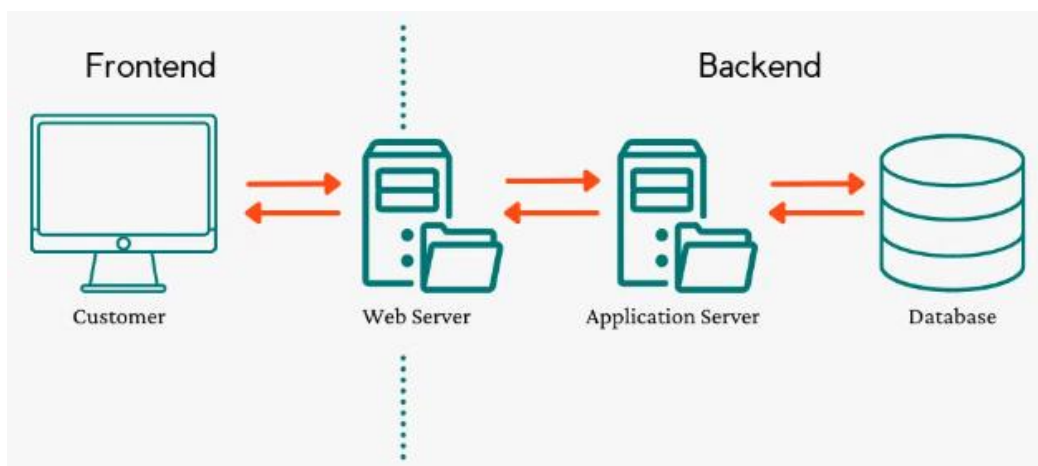


6. Modelo de división funcional front-end / back-end para aplicaciones web.

El front-end es el desarrollo web que se dedica a la parte frontal y visual de un servicio web (página web, aplicación web, mashup, etc..), desde la estructura de esta hasta otros aspectos estéticos.

Y el back-end es el desarrollo web que se dedica a la programación y ejecución de esta desde el navegador, procesando la información para mostrarla a continuación en la parte del front.

Más información: <https://blog.hubspot.es/website/frontend-y-backend>



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

Página web estática es una página web que se entrega al navegador del cliente mostrándola como está almacenada, es decir, presenta la misma información a todos los clientes y no reciben actualizaciones.

Más información:

https://es.wikipedia.org/wiki/P%C3%A1gina_web_est%C3%A1tica

Página web dinámica es una página web que es generada bajo demanda, su contenido cambiará en función de datos como puede ser la zona horaria y geográfica, el nombre de usuario, preferencias del usuario, etc...

Más información:

https://es.wikipedia.org/wiki/P%C3%A1gina_web_din%C3%A1mica

Aplicación web es un software web que utilizan los usuarios accediendo a un servidor web a través de internet mediante un navegador, es decir, un programa que se codifica en un lenguaje de programación interpretable por los navegadores.

Más información: https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web

Mashup o aplicación web híbrida, es una aplicación que combina servicios de varias páginas web en una sola página con la finalidad de reutilizar su contenido.

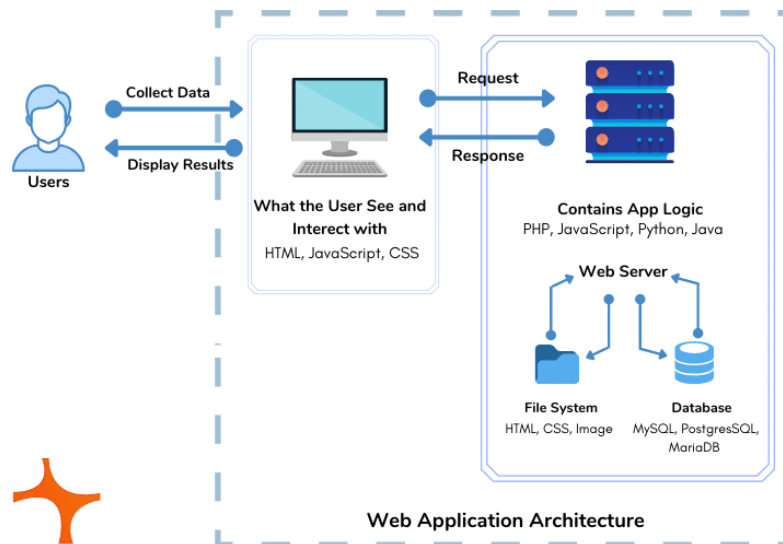
Más información:

[https://es.wikipedia.org/wiki/Mashup_\(aplicaci%C3%B3n_web_h%C3%ADbrida\)](https://es.wikipedia.org/wiki/Mashup_(aplicaci%C3%B3n_web_h%C3%ADbrida)) y <https://www.arimetrics.com/glosario-digital/mashup-aplicacion-web-hibrida>

8. Componentes de una aplicación web.

Los componentes básicos y mínimos de una aplicación web son del lado del cliente uno o más clientes con un navegador, del lado del servidor necesitamos un servidor web, un módulo de interpretación del lenguaje de programación, el código escrito como HTML, PHP, JS, CSS, y para la persistencia de la información una base de datos.

Más información: <https://www.ibm.com/docs/es/db2-for-zos/11?topic=environment-components-web-based-applications>



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

Lado del cliente:

- HTML → <https://www.w3schools.com/html/>
- CSS → <https://www.w3schools.com/css/default.asp>
- JavaScript → <https://www.w3schools.com/js/default.asp>

Lado del servidor:

- PHP → <https://www.w3schools.com/php/default.asp>
- Python → <https://www.w3schools.com/python/default.asp>
- Java → <https://www.w3schools.com/java/default.asp>
- C# → <https://www.w3schools.com/cs/index.php>

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

PHP

- Es un código abierto y gratuito.
- Es un lenguaje orientado a objetos

Python

- Un lenguaje interpretado.
- Un lenguaje orientado a los objetos.

Java

- Lenguaje compilado
- Un lenguaje orientado a los objetos.

Node.js

- Asíncrono y controlado por eventos
- Procesos en un solo hilo, pero altamente escalable

C#

- Multiplataforma.
- Sintaxis similar a C, C++, Java y otros.

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

XAMPP es un paquete de software libre, sus siglas hacen referencia a su contenido X indicando que funciona en cualquier sistema operativo, A de Apache el servidor web, M de MariaDB/MySQL/MongoDB, PHP de lenguaje de programación y Perl.

Algunas características de XAMPP:

- a) Utilización de licencia GNU (General Public License).
- b) Software de carácter libre.
- c) Multiplataforma, compatibilidad con sistemas operativos.

Más información: <https://es.wikipedia.org/wiki/XAMPP>, <https://blogdelinformatico-reizer.blogspot.com/2015/11/caracteristicas-de-xampp.html> y <http://www.elpadawan.com/apache/xampp-ventajas-y-desventajas>

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.

Para entender en que caso necesitaríamos realizar la instalación de una máquina virtual Java (JVM) y el software JDK deberemos saber estas dos:

- Entorno de explotación o entorno de producción, entorno real en el que se ejecuta el software y se utiliza por los usuarios finales y clientes software que utilizan JVM.
- Entorno de desarrollo, espacio de trabajo que permite a los desarrolladores crear una aplicación o realizar cambios en ella sin afectar a la versión real del producto software que utilizaran JDK.

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

Visual Studio Code → Entorno de desarrollo integrado (IDE) de código abierto desarrollado por Microsoft.

Apache NetBeans → Entorno de desarrollo integrado (IDE) de código abierto desarrollado por Apache.

Eclipse → conjunto de herramientas de programación de código abierto multiplataforma para desarrollar el proyecto llamado "Aplicaciones de Cliente Enriquecido".

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

Apache

- Gratuito
- Código abierto
- Multiplataforma

Nginx

- Mejor gestión del tráfico alto
- Código abierto
- Consume poca memoria para los usuarios

CloudFire

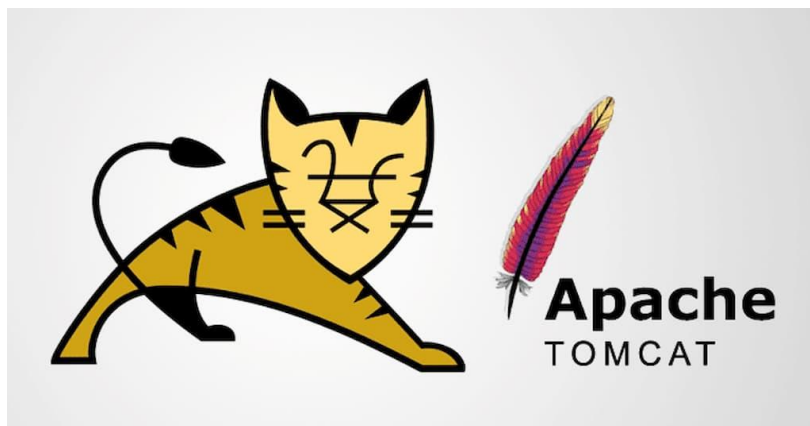
- Protección y aceleración de sitios web
- Versión gratuita y de pago
- Características de seguridad mejoradas

Más información: <https://w3techs.com/>

15. Apache HTTP vs Apache Tomcat.

Apache HTTP → servidor web que proporciona servicio para contenido estático, aunque también puede admitir contenido dinámico.

Apache Tomcat → servidor de aplicaciones para poder hacer ejecución de las aplicaciones web (mashup) Java.



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

Algunos de los navegadores más utilizados son:

- Google Chrome
- Firefox
- Microsoft Edge

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

Un generador de documentación es una herramienta de programación que genera documentación en lenguaje HTML destinada a los programadores o a usuarios futuros (documentación de API).

Mejorar explicación corta

Algunos ejemplos de estos generadores de documentación en PHP son:

- ApiGen → **enlaces a sus páginas**
- PhpDoc → **imágenes de sus tags o etiquetas**

Preguntarle a amor cual vamos a usar y buscar tutoriales

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

Los sistemas de control de versiones son un tipo de software que ayuda a hacer un seguimiento de los cambios realizados en el código a lo largo del tiempo. Algunos ejemplos de estos mismos sistemas de control de versiones son:

- Git
 - a) Ramificación (Branching) permite la realización de ramas dentro de un proyecto para un mejor desarrollo y la unión (Merging) de las mismas a una rama central una vez haya una versión estable.
 - b) Multiplataforma integrándose con facilidad con otros programas para su utilización como Visual Studio Code para tener de forma efectiva una gestión del código.
 - c) Distribución los usuarios tienen una copia local del repositorio, pudiendo trabajar de manera individual en el proyecto y mejorando así su desarrollo.

Incluir CVS

19. Propuesta de configuración del entorno de desarrollo para la asignatura de Desarrollo web del lado servidor en este curso.
20. Propuesta de configuración del entorno de explotación para la asignatura de Desarrollo web del lado servidor en este curso.
21. Realizar un estudio sobre los siguientes conceptos y su relación con el desarrollo de aplicaciones web:

CMS - Sistema de gestión de contenidos

Un sistema de gestión de contenidos (CMS en sus siglas en inglés: Content Management System) es una aplicación de software que se ejecuta en un navegador.

- a) Permite crear un entorno de trabajo para la creación y administración de contenidos.
- b) Contiene una interfaz que controla una o varias bases de datos.
- c) Permite manejar de manera independiente el contenido y el diseño.

Más información: https://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenidos



ERP - Sistema de planificación de los recursos empresariales

Los sistemas de planificación de recursos empresariales (ERP en sus siglas en inglés: Enterprise Resource Planning) es un tipo de software que las organizaciones utilizan para gestionar las actividades empresariales diarias.

- a) Se hace cargo de distintas operaciones internas de una empresa.
- b) Automatiza los procesos empresariales.
- c) Integra distintas bases de datos de una compañía en un solo programa.

Más información:

https://es.wikipedia.org/wiki/Sistema_de_planificaci%C3%B3n_de_recursos_empresariales

