TEMA 1

DESARROLLO WEB EN ENTORNO SERVIDOR

EJERCICIOS

REBECA SÁNCHEZ PÉREZ
IES LOS SAUCES

INDICE

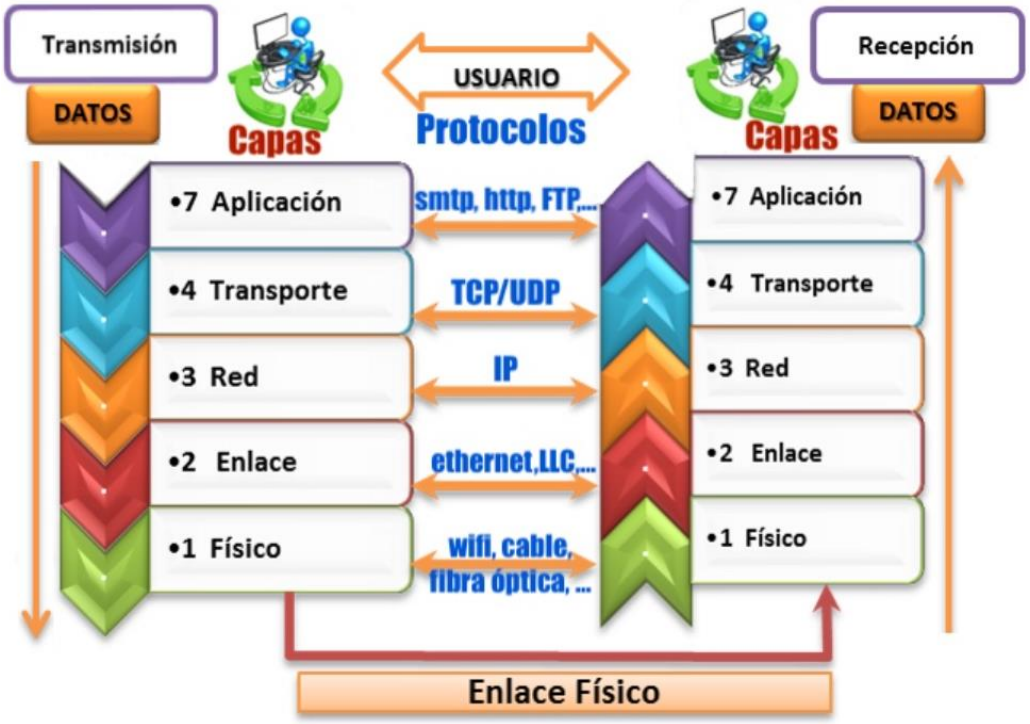
1. Protocolos de comunicaciones: IP, TCP, HTTP, HTTPS

IP: (Protocolo de Internet) es un protocolo de comunicación que tiene como objetivo transmitir paquetes de datos de una forma bidireccional (que tiene un origen y un destino) a través de una red. Pertenece a la capa 3 del modelo OSI (capa de red) en la que se enruta el paquete dándole una dirección IP.

TCP: (Protocolo de control de transmisión) es un protocolo de comunicación que establece las comunicaciones entre los equipos para que los usuarios podamos navegar por internet y servicios en red. Pertenece a la capa de transporte del modelo OSI.

HTTP: (Hypertext Transfer Protocol) es un protocolo el cual nos permite realizar peticiones de datos que sigue la estructura de cliente-servidor (protocolo en el que se basan las aplicaciones web). Pertenece a la capa aplicación del modelo OSI.

HTTPS: consiste en lo mismo que el protocolo HTTP solamente que en este la comunicación es cifrada (agrega SSL).



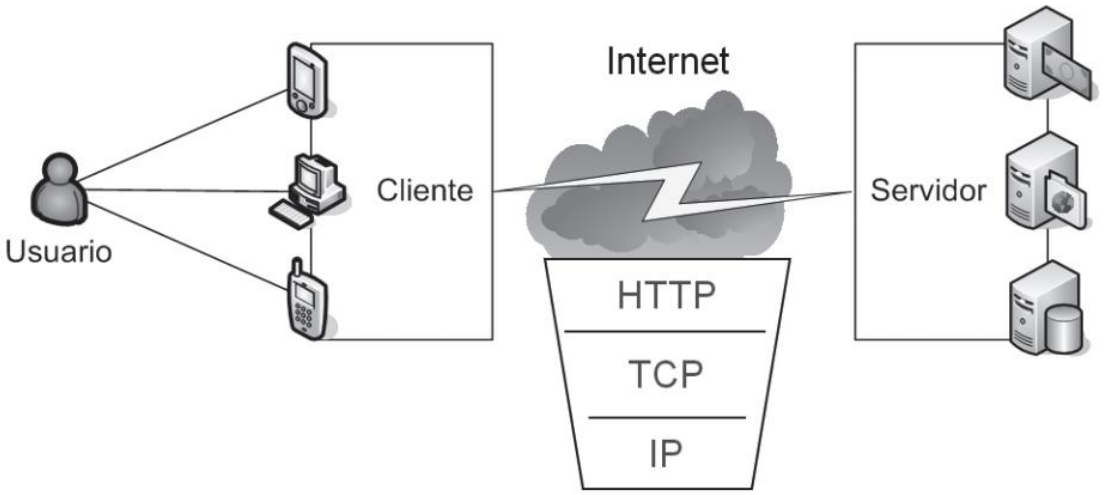
Enlaces:

https://www.cloudflare.com/es-es/learning/network-layer/internet-protocol/#:~:text=El%20Protocolo%20de%20Internet%20(IP)%20es%20un%20protocolo%2C%20o,trozos%20m%C3%A1s%20peque%C3%B1os%2C%20llamados%20paquetes.

https://developer.mozilla.org/es/docs/Web/HTTP/Overview

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

Es un modelo de diseño de software que se compone de 2 partes (el cliente y el servidor) en el que cliente realiza una petición HTTP/S al servidor, que la procesa y envía una respuesta. Este tipo de arquitectura es la más utilizada en el desarrollo de aplicaciones web ya que nos permite conectar varios clientes a los servicios que provee un servidor. La mayoría de los protocolos de la capa de aplicación del modelo OSI utilizan en este modelo de comunicaciones (por ejemplo, http, dns, ssh, ftp...)



Enlaces:

https://nucba.medium.com/qu%C3%A9-es-la-arquitectura-cliente-servidor-eb9f402506cc

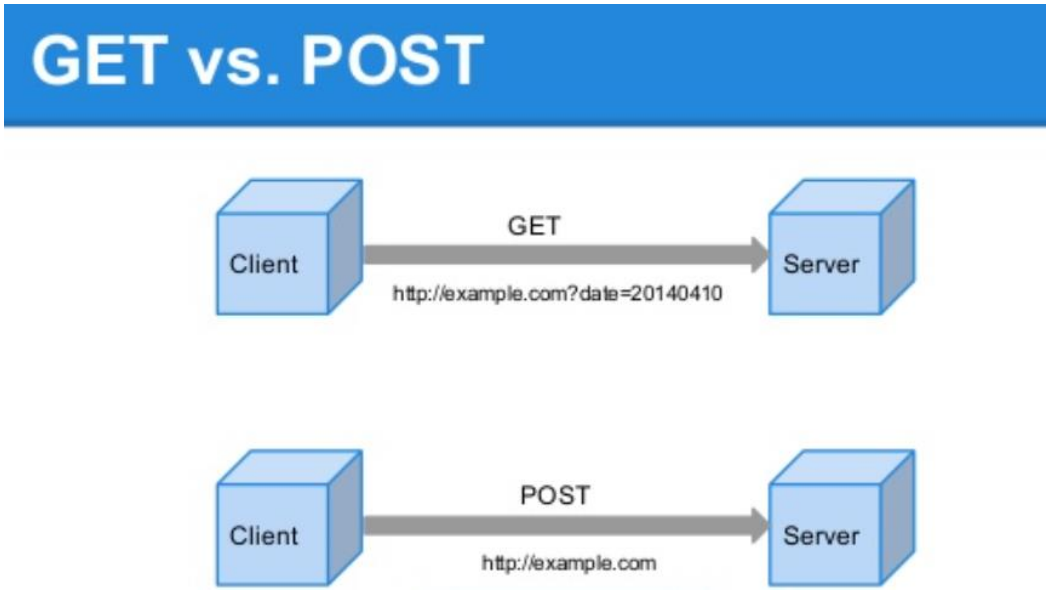
3. Estudio sobre los métodos de petición HTTP /HTTPS más utilizados

Los métodos de petición HTTP/S se encargan de devolver la información al cliente y mandar peticiones al servidor (solicitudes, por ejemplo, formularios html). Los más utilizados son los métodos GET y POST, ambos sirven para lo mismo, responden a solicitudes del cliente, pero cada uno imprime los parámetros de forma diferente.

- GET: Cuando envía información al servidor, los parámetros introducidos aparecen en la barra del navegador.
- POST: Cuando envía información al servidor, los parámetros no se pueden ver desde el lado del cliente y solo son accesibles desde el lado del servidor.

Otros métodos de peticiones HTTP/S son:

- HEAD: recibe la información de la cabecera de la página web
- PUT: sube, carga o actualiza un recurso específico
- DELETE: borra un recurso



Enlaces:

https://developer.mozilla.org/es/docs/Web/HTTP/Methods

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 utilizada en un navegador para acceder a un recurso mediante la dirección o nombre (o ambos) del mismo. Dentro de este término se incluyen las URL y URN (Ej: https://website.com/drive/info#part1).

URL: (Localizador uniforme de recursos) es una cadena de caracteres utilizada en un navegador para acceder a un recurso mediante la dirección del mismo y que especifica el protocolo que utiliza (Ej: https://elpais.com/ file:///localhost/8.8.8.8). Todas las URL son URI.

URN: (Nombre uniforme de recurso) es una cadena de caracteres utilizada en un navegador para acceder a un recurso mediante el nombre del mismo pero que no ofrece el protocolo utilizado (HTTP) ni la dirección del recurso (Ej: urn:nbn:de:101:3-2019075675872913). Las URN son un tipo de URI.

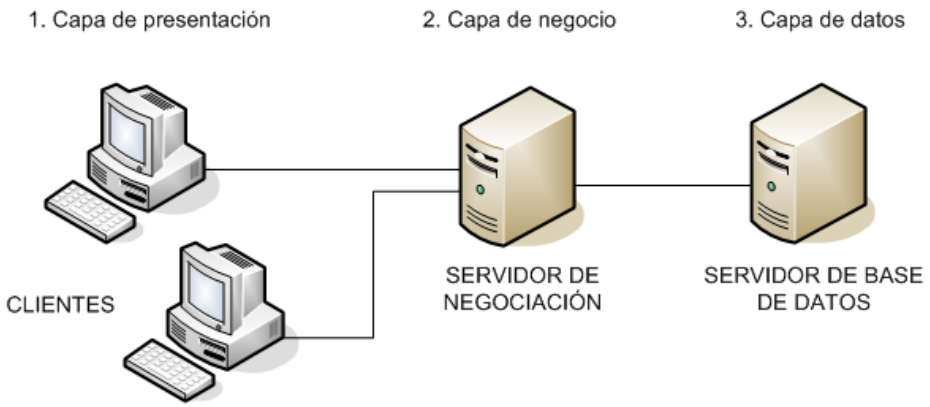
Para hacer una petición desde un cliente a un servidor es necesario usar el protocolo HTTP seguido de una URL.

Enlaces:

https://geekflare.com/es/difference-between-url-uri-and-urn/

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

Lo modelos aplicaciones multicapa o de 3 capas consisten en agrupar el código de la aplicación en capas dependiendo de la función que desempeñan y que permiten la comunicación entre ellas.



- Capa de datos: en la que las líneas de código se agrupan en ficheros, dicho de otra forma, donde se encuentran los datos que utiliza la aplicación para el cliente.
- Capa de negocio o aplicación: es el código que se encarga de realizar cálculos, es decir, el programa que estructura la aplicación.
- Capa de presentación: las líneas de código que le dan apariencia a la web y muestra información al cliente desde su navegador (interfaz gráfica).

El número de capas de una aplicación web depende del tamaño de los componentes, de la naturaleza del servicio ofrecido y del reparto de las funciones entre el cliente y el servidor. Generalmente se componen por 3 capas, pero pueden aparecer más si trabamos con un modelo de aplicación MVC o si la aplicación está destinada a la extracción de datos para un web service o para generar una API para el cliente.

Enlaces:

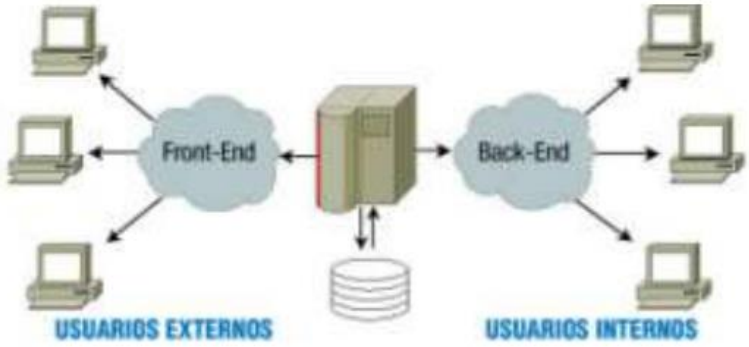
https://es.wikipedia.org/wiki/Arquitectura_multicapa

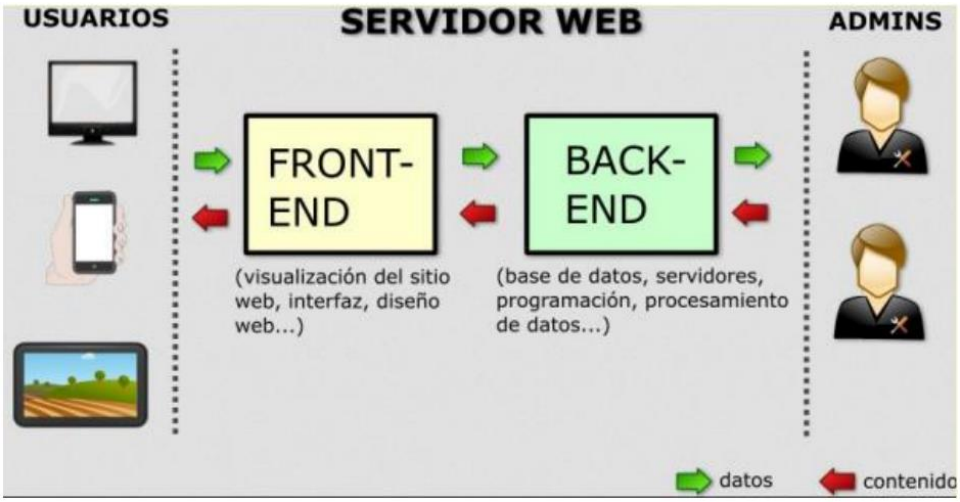
https://elvex.ugr.es/decsai/csharp/design/layers.xml

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

Este modelo se basa en dividir la aplicación en 2 partes dependiendo de los usuarios que la usan (usuarios externos y usuarios externos) y de su funcionalidad.

- El Front-end es el conjunto de páginas que están destinadas a ser utilizadas por los clientes. Se refiere a la parte de una aplicación web con la que los usuarios interactúan directamente, incluyendo botones, formularios, etc. Es responsable de la presentación de datos, la experiencia del usuario y la interacción en tiempo real.
- El Back-end es otro conjunto de páginas dinámicas que utilizan las personas que producen el contenido y las que administran la aplicación web (administradores, publicadores, censores, reguladores...). Se refiere a la parte de una aplicación web que funciona detrás de escena y no es visible para el usuario final incluyendo el servidor web que recibe las solicitudes del navegador del usuario, almacena y gestiona los datos en una BD y contiene una lógica en la que procesa datos y autentica usuarios y a su vez proporciona una API





En conclusión, El frontend se ocupa de la presentación y la experiencia del usuario, mientras que el backend se encarga de la lógica de la aplicación.

Enlaces:

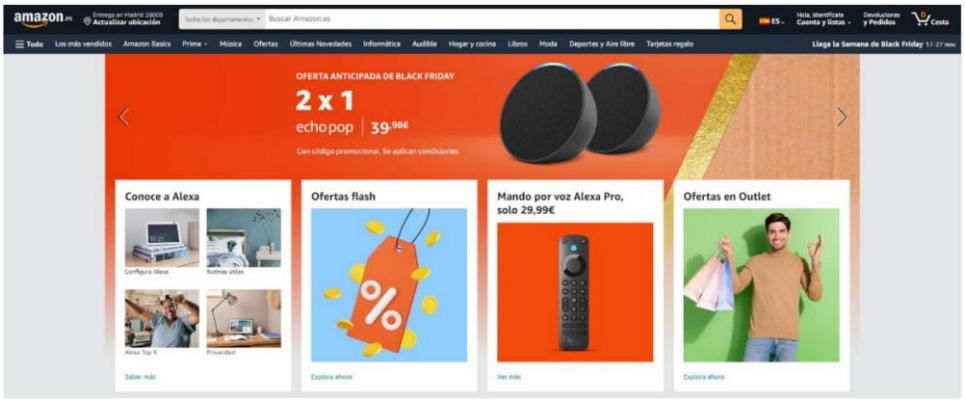
https://aws.amazon.com/es/compare/the-difference-between-frontend-and-backend/#:~:text=El%20front%20end%20es%20aquello,permiten%20que%20la%20aplicaci%C3%B3n%20funcione

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

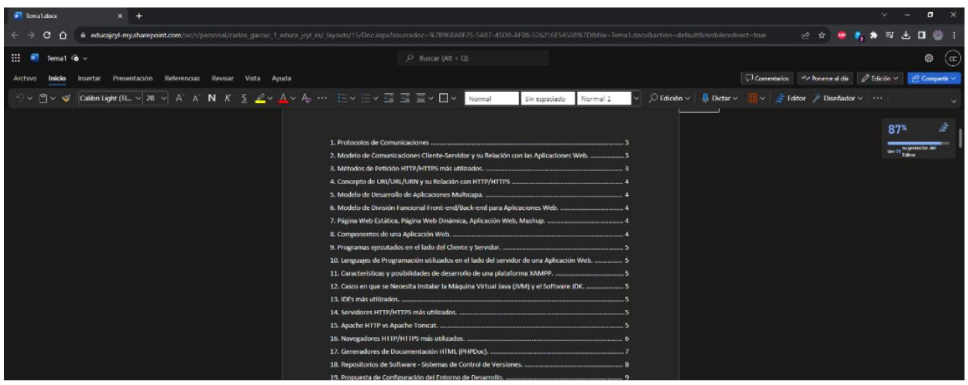
La diferencia principal entre estos conceptos es la complejidad, a medida que se va bajando en la escala aumenta el grado de complejidad en la web:

Página web estática: Contenido fijo que no cambia según la interacción del usuario, un ejemplo es el sitio "The World's Worst Website Ever," que se creó como un ejemplo de lo que no se debe hacer en diseño web.

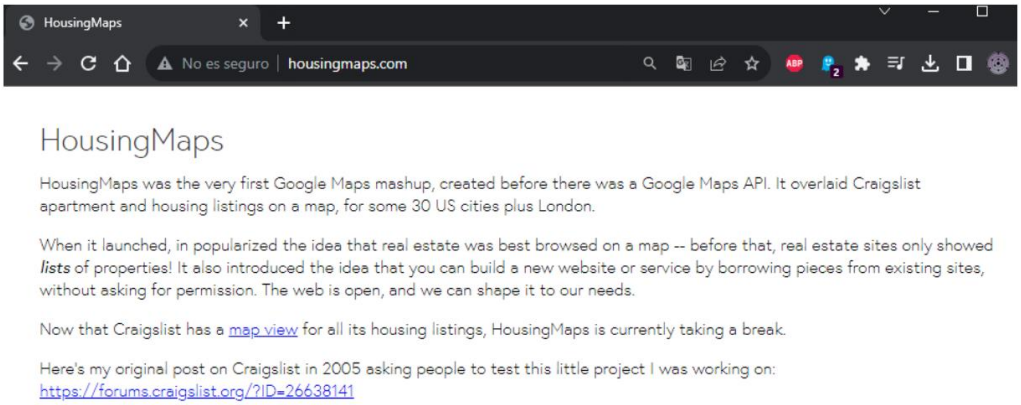
Página web dinámica: Contenido generado en tiempo real, adaptado según la interacción del usuario o datos externos, tiene control de acceso y gestión de datos. Un ejemplo sería "Amazon.es"



Aplicación web: Es un programa interactivo ejecutado en un navegador, un ejemplo de una aplicación web es "Google Docs".

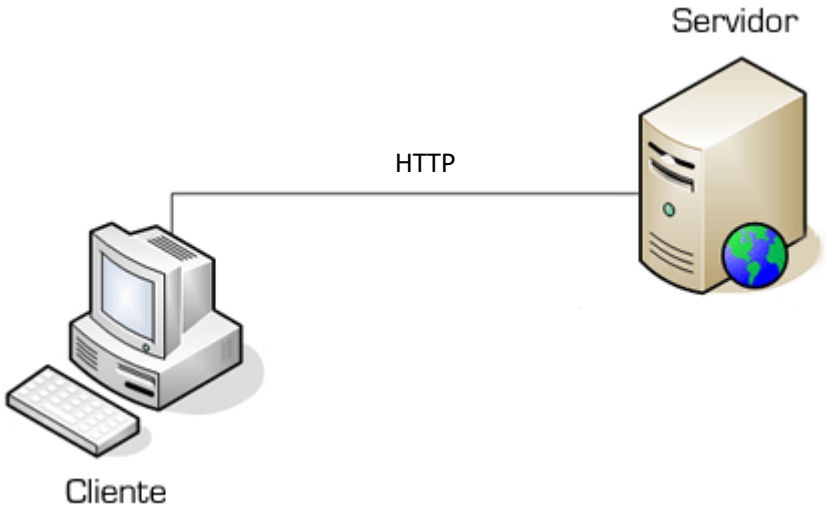


Mashup: Integración de diferentes servicios web para crear una nueva aplicación compuesta, un ejemplo sería "HousingMaps," que combina datos de Craigslist y Google Maps para ayudar a los usuarios a encontrar alquileres y propiedades en un mapa interactivo.



(Referencias: Estudio Teórico de Carlos García Cachón)

8. Componentes de una aplicación web



- Servidor web
  - Modulo interprete que ejecuta un lenguaje de programación
  - Directorios y ficheros escritos en un lenguaje de programación del lado del servidor:
    - HTML, JS, PHP, SQL, JSON, XML...
  - Sistema gestor de Base de Datos y datos
- Cliente web (navegador): integra su propio intérprete de javascript, html...

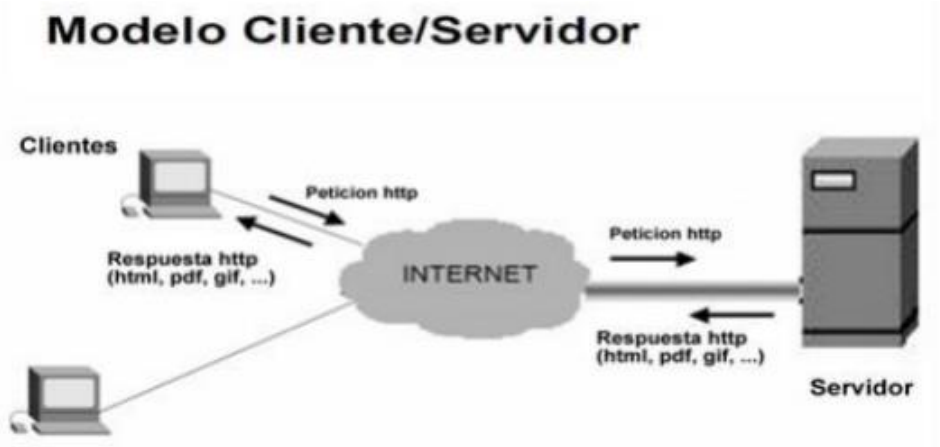
Enlaces:

https://sistemasredes-virus.es.tl/Cliente_Servidor-.--.--.-.htm

https://es.wikipedia.org/wiki/Cliente-servidor

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

- Lado del Cliente: Navegador web (por ejemplo, Google Chrome): Se ejecutan en el lado del cliente y su función principal es interpretar y representar contenido web, incluyendo páginas HTML, CSS y JavaScript. Los navegadores permiten a los usuarios acceder a sitios web, interactuar con aplicaciones web y mostrar el contenido de manera amigable para el usuario. Además, pueden ejecutar scripts de JavaScript en el navegador del usuario para mejorar la interactividad y la experiencia del usuario en la web.
- Lado del Servidor: Servidor web (por ejemplo, Apache): Un servidor web es un programa que se ejecuta en el lado del servidor y responde a las solicitudes de los navegadores web de los clientes. Su función principal es recibir las solicitudes HTTP/HTTPS de los clientes, procesarlas y entregar las respuestas adecuadas, que generalmente incluyen páginas web, datos o recursos. Los servidores web también pueden ejecutar aplicaciones y lógica de servidor, como PHP o Python, para generar contenido dinámico antes de enviarlo al cliente. Estos servidores son esenciales para alojar sitios web y aplicaciones web y servir su contenido a los usuarios.



(Referencias: Estudio Teórico de Carlos García Cachón)

Enlaces:

https://w3techs.com/technologies/overview/programming_language
https://w3techs.com/technologies/overview/client_side_language

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

- PHP: Esta especialmente diseñado para el desarrollo web y en la actualidad se suelen usar frameworks como Laravel o Symfony
- ASP.NET: Es un framework propiedad de Microsoft en el que se usan lenguajes muy populares como C# y .net
- Ruby: Es un lenguaje orientado a objetos fácil de leer, fue muy popular en la década pasada y actualmente ha perdido popularidad.
- Java: Es un lenguaje de programación orientado a objetos que ofrece alta escalabilidad y rendimiento. Es muy utilizado en aplicaciones empresariales y grandes sistemas se suele usar con el framework Spring.
- JavaScript: Es versátil y se utiliza tanto en el lado del cliente como en el lado del servidor. Node.js es un entorno de ejecución de JavaScript en el lado del servidor que permite desarrollar aplicaciones web rápidas no tan usado para crear grandes aplicaciones.

© W3Techs.com	usage	change since 1 August 2023
1. PHP	76.9%	-0.5%
2. ASP.NET	6.8%	-0.1%
3. Ruby	5.5%	+0.1%
4. Java	4.7%	
5. JavaScript	3.0%	+0.3%

percentages of sites

(Referencias: Estudio Teórico de Carlos García Cachón)

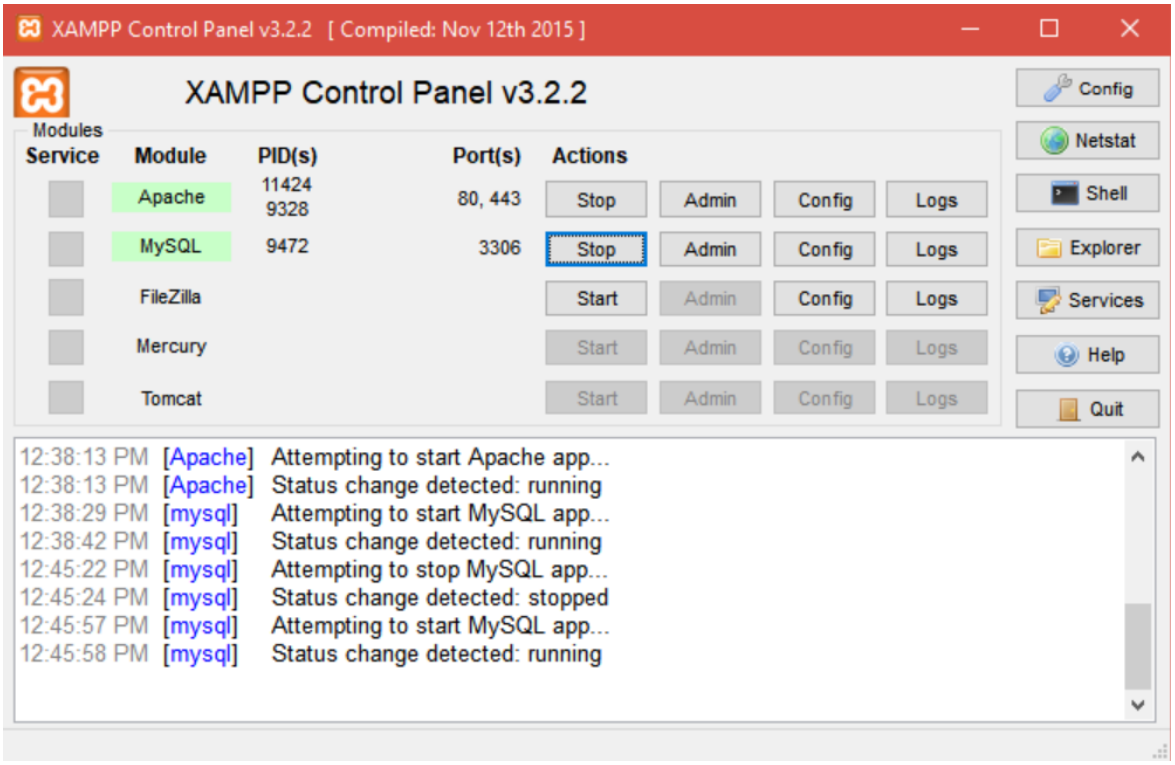
Enlaces:

https://w3techs.com/technologies/overview/programming_language

11. Características y posibilidades de desarrollo de una plataforma XAMPP

Es una plataforma versátil y fácil de usar que proporciona un entorno de desarrollo web completo para la creación y prueba de aplicaciones web basado en el lenguaje de programación Php o Pearl. Es útil para desarrolladores que desean trabajar en sus proyectos de manera local antes de llevarlos a un servidor en producción. Sus características son:

- Multiplataforma compatible con Windows, Linux y masOS
- Fácil de instalar y gran comunidad que ofrecen soporte y ayuda
- Compatibilidad con apache, conexión con bases de datos MySQL

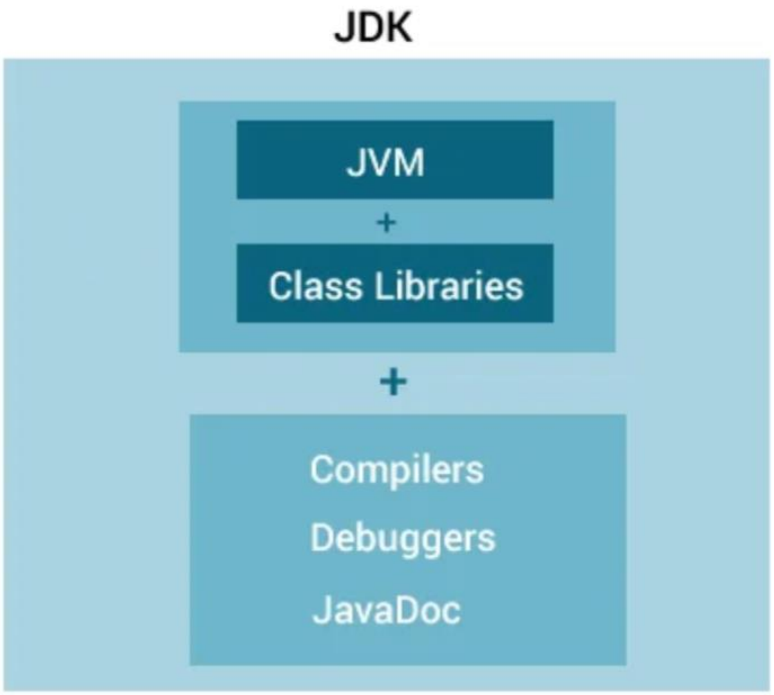


Enlaces:

https://www.apachefriends.org/es/index.html

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.

La instalación de JVM o JDK depende de si vamos a desarrollar una aplicación o solamente queremos usarla. Si queremos desarrollar una app en java necesitaremos el JDK (*Java Development Kit*) en el entorno de desarrollo (donde se llevan a cabo las pruebas) y de explotación (cuando ya puedes acceder a la app desde internet) y el JVM (*Java Virtual Machine*) para poder probar la aplicación o programa. En el caso de querer usar una aplicación solo necesitaremos el JVM ya que este es el intérprete de Java.



Enlaces:

https://www.javatpoint.com/difference-between-jdk-jre-and-jvm

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

Un IDE, o Entorno de Desarrollo Integrado, es una herramienta de software que proporciona un conjunto completo de funciones para el desarrollo de aplicaciones informáticas. Están diseñados para facilitar y agilizar el proceso de desarrollo junto con herramientas y características integradas en una interfaz de usuario. Las más utilizadas son:

Visual Studio Code: Desarrollado por Microsoft. Características:

- Amplia biblioteca de extensiones
- Ofrece sugerencias de código
- Integra depuración
- Control de versiones
- Multiplataforma
- Código abierto

Eclipse: Desarrollado originalmente por IBM y ahora mantenido por la Eclipse Foundation. Características:

- Soporte para múltiples lenguajes de programación(Java, C++, Python, …).
- Muy personalizable, por medio de extensiónes.
- Incluye resaltado de sintaxis
- Depuración de código
- Control de versiones
- Plataforma RCP (Proporciona un conjunto de componentes y bibliotecas que permiten desarrollar aplicaciones de escritorio personalizables de manera eficiente.)

NetBeans: Desarrollado originalmente por Sun Microsystem y ahora mantenido por Oracle Corporation. Características:

- Fácil uso y versatilidad particularmente en el desarrollo de aplicaciones Java
- Capacidad de arrastrar y soltar, que facilita la creación de interfaces de usuario gráficas y la construcción rápida de aplicaciones.
- Altamente extensible a través de complementos y módulos
- Admite numerosos lenguajes de programación como Java, PHP, HTML5, JavaScript.
- Proporciona herramientas de desarrollo web y móvil

(Referencias: Estudio Teórico de Carlos García Cachón)

Enlaces:

https://es.wikipedia.org/wiki/Eclipse_(software)#:~:text=archivos%20en%20tanto-,Caracter%C3%ADsticas,%2C%20etc.%2C%20y%20refactorizaci%C3%B3n.

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

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

Apache HTTP Server:

- Apache es uno de los servidores web más populares y de código abierto del mundo. Es conocido por su estabilidad y flexibilidad.
- Ofrece una amplia gama de módulos y configuraciones personalizables que permiten a los administradores de servidores adaptarlo a sus necesidades.
- Es compatible con una variedad de sistemas operativos, incluyendo Linux y Windows.

Nginx:

- Nginx es otro servidor web de código abierto ampliamente utilizado, conocido por su alto rendimiento y capacidad de servir grandes cantidades de tráfico.
- Está diseñado para ser eficiente y escalable, y es una elección común para servir sitios web y aplicaciones web de alto rendimiento.
- También se usa comúnmente como servidor proxy inverso y equilibrador de carga.

(IIS):

- IIS es el servidor web de Microsoft para sistemas operativos Windows.
- Es conocido por su integración con otros productos de Microsoft y su capacidad para ejecutar aplicaciones web ASP.NET y servicios de Microsoft.
- Es una elección común para empresas que utilizan tecnologías de Microsoft.

(Referencias: Estudio Teórico de Carlos García Cachón)

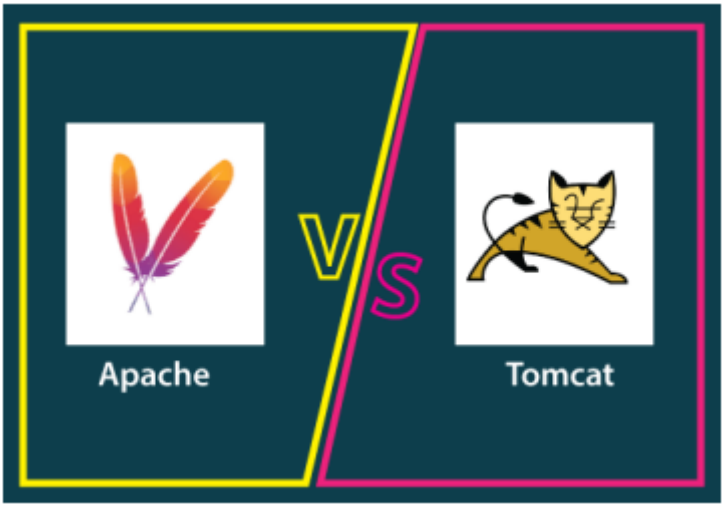
Enlaces:

https://www.stackscale.com/es/blog/top-servidores-web/#:~:text=Seg%C3%BAn%20las%20estad%C3%ADsticas%20de%20W3Techs%2C%20los%20servidores%20web%20en%20el,son%20Nginx%2C%20Apache%20y%20OpenResty.

https://www.sysadminok.es/blog/hosting/servidores-web-mas-utilizados/#:~:text=Apache%2C%20Nginx%2C%20Microsoft%20IIS%20y,web%20m%C3%A1s%20populares%20del%20mundo.

15. Apache HTTP vs Apache Tomcat

Los 2 son servidores web que atienden a las solicitudes http que realiza un cliente, la diferencia principal es que Apache HTTP está pensado para que el desarrollo del lado del servidor este escrito en php, Pearl y Python (el código que se interpreta para general la página web que solicita el cliente desde el lado del servidos) y Apache Tomcat trabaja con javaSCRIPT.



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

Google Chrome: es el navegador web y móvil desarrollado por Google. La mayor parte de su código fuente proviene de Chromium, el proyecto de software de código abierto y gratuito de Google. Algunas de sus características son:

- Gestión de contraseñas.
- Generación de contraseñas seguras.
- Autocompletar formularios.
- Modo oscuro.
- Múltiples perfiles.

Safari: esta específicamente diseñado y optimizado para funcionar en dispositivos Apple y algunas de sus características frente a otros navegadores son:

- Notas rápidas. (Cuentan con ella de manera nativa, aunque el resto tiene plug-in para usar también la misma tecnología.)
- Streaming de vídeo en 4K.

Todo ello de manera nativa y sin descargar o configurar nada.

Microsoft Edge: Esta desarrollado por Microsoft para reemplazar a Internet Explorer y algunas de sus características a destacar son:

- Modo de navegación infantil. (Aunque las otras tienen Control Parental, pero hay que configurarlo.)
- Software Propietario

Mozilla Firefox: es un navegador de código abierto. Comparte prácticamente todas las características anteriores que en Google Chrome.

Opera: es el más antiguo de los cinco y el código está basado en Chromium, el proyecto de software abierto de Google. La característica más destacable es:

- VPN integrada. (De manera nativa.)

(Referencias: Estudio Teórico de Carlos García Cachón)

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

phpDocumentor: Es uno de los generadores más antiguos y utilizados, proporciona una sintaxis simple y fácil de entender, es personalizable y extensible. Características destacadas:

- Genera documentación en formatos como HTML, PDF, y otros.
- Permite enlazar fácilmente documentación por medio de clases, métodos y propiedades.
- Proporciona una amplia variedad de opciones de configuración para adaptarse a las necesidades específicas de un proyecto.

ApiGen: Se basa en phpDocumentor pero simplifica la sintaxis de comentarios, es rápido y fácil de usar, permitiendo a los desarrolladores generar documentación rápidamente. Características destacadas:

- Proporciona una interfaz de línea de comandos simple para generar la documentación.
- Ofrece soporte para varias versiones de PHP y admite anotaciones de estilo PHPDoc.
- Permite personalización a través de temas y opciones de configuración.

Doxygen: Aunque no es específicamente un generador de documentación para PHP, es ampliamente utilizado y es compatible con varios lenguajes de programación. Características destacadas:

- Admite varios formatos de salida, incluidos HTML, LaTeX, PDF, y más.
- Permite generar gráficos de dependencia, diagramas de clases, y otros elementos visuales para ayudar a comprender la estructura del código.
- Proporciona un gran conjunto de características avanzadas para personalizar la documentación y adaptarla a diferentes necesidades

(Referencias: Estudio Teórico de Carlos García Cachón)

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

Git: Es el sistema de control de versiones más popular y utilizado actualmente debido a su eficiencia, velocidad y flexibilidad, es de código abierto, lo que fomenta una amplia aceptación en la industria de desarrollo de software y es utilizado por una gran cantidad de proyectos y organizaciones. Características destacadas:

- Distribuido: Cada usuario tiene una copia local completa del repositorio y puede trabajar de forma independiente, lo que mejora la velocidad y la flexibilidad.
- Ramificación (branching) y fusión (merging): Permite trabajar en diferentes ramas de desarrollo y fusionar los cambios de manera efectiva.
- Integración con plataformas populares: Git se integra fácilmente con herramientas y servicios populares, como GitHub, GitLab y Bitbucket, facilitando la colaboración y la gestión del código fuente.

Apache Subversion (SVN): Ha sido muy utilizado durante muchos años y ofrece una transición más sencilla para aquellos que están familiarizados con sistemas de control de versiones centralizados. Características destacadas:

- Centralizado: A diferencia de Git, SVN sigue un modelo centralizado, donde hay un único repositorio central que almacena toda la historia y las versiones del proyecto.
- Fácil administración de archivos binarios: SVN es eficaz para el manejo de archivos binarios, lo que lo hace adecuado para proyectos que contienen muchos recursos multimedia.
- Operaciones atómicas: Las operaciones de SVN son atómicas, lo que significa que se realizan por completo o no se realizan en absoluto, proporcionando una mayor integridad y coherencia en las versiones.

Mercurial: Es simple y eficaz, lo que lo convierte en una opción popular para muchos desarrolladores y proyectos, fácil de aprender y usar. Características destacadas:

- Distribuido: Al igual que Git, Mercurial es un sistema de control de versiones distribuido, lo que permite un flujo de trabajo flexible y descentralizado.
- Rendimiento y eficiencia: Es conocido por su rapidez en operaciones cotidianas y su eficiencia en términos de uso de recursos.
- Simplicidad: La interfaz de usuario y la estructura de comandos de Mercurial son intuitivas y fáciles de entender, lo que facilita su adopción y uso.

(Referencias: Estudio Teórico de Carlos García Cachón)

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.

- Sistema operativo: Ubuntu Server 22.0.4(RAM, Disco, Particiones, Cuentas, Password, Red...)
- Servidor administración remota: SSH
- Servidor de transferencia de ficheros: SFTP (SSH)
- Repositorio: GIT Hub
- Servidor Web: Apache HTTP (mod_php, mod_ssl, …),
- SGBD: MySQL
- Navegador: W3M

(Referencias: Estudio Teórico de Carlos García Cachón)

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.

- Sistema operativo: Windows 10 Pro (RAM, Disco, Particiones, Cuentas, Password, Red...)
- Servidor administración remota: SSH
- Navegador: Mozilla y Google Chrome (Actualizados)
- IDE: Visual Studio Code
- Ofimática, multimedia: Word, Gimp.
- Cliente SSH: Filezilla

(Referencias: Estudio Teórico de Carlos García Cachón)