TEMA 1

DESARROLLO WEB EN ENTORNO SERVIDOR

EJERCICIOS

REBECA SÁNCHEZ PÉREZ
IES LOS SAUCES

INDICE

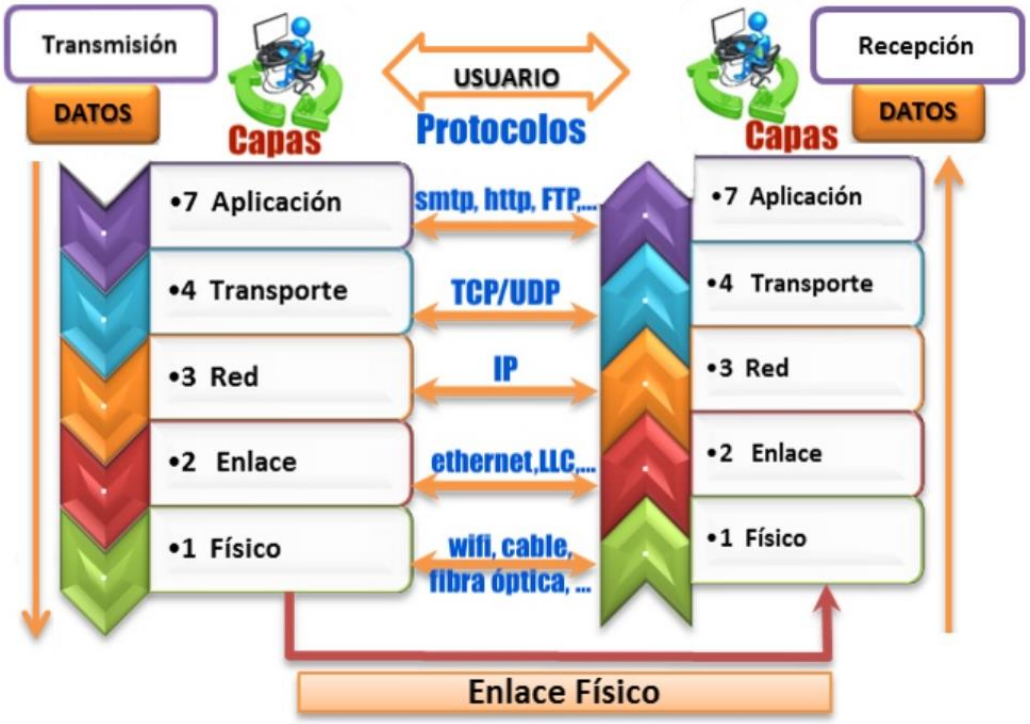
1. Protocolos de comunicaciones: IP, TCP, HTTP, HTTPS

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).



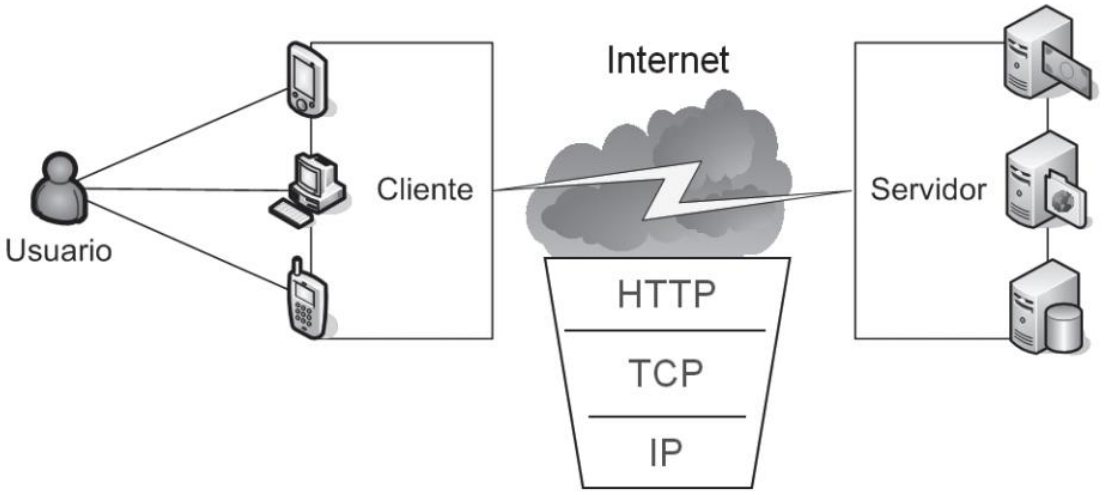
Enlaces:

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.



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 varían dependiendo de la acción que se debe realizar para devolver la información al cliente o mandar peticiones (solicitud). Los más utilizados son los métodos GET y POST.

GET: Método utilizado para solicitar recursos (Ejemplo: entrar en la página de “El País”)

POST: Método utilizado para enviar datos (Ejemplo: al darle al botón enviar de un formulario)

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 (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.

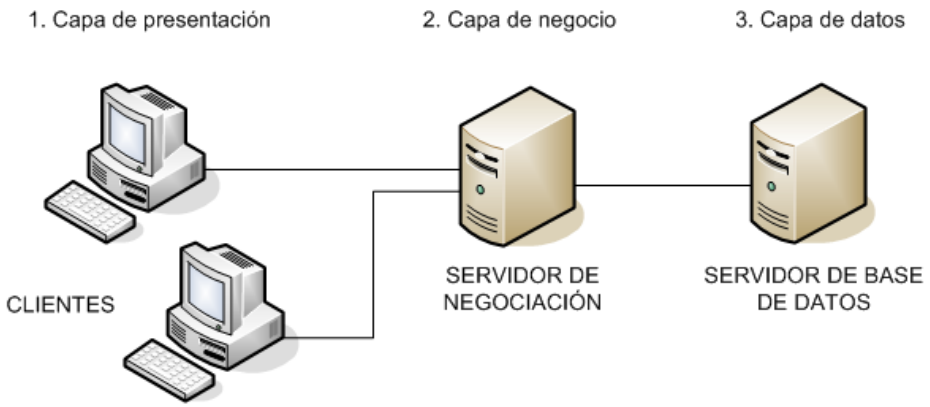
Para hacer una petición desde un cliente a un servidor es necesario usar el protocolo HTTP seguido de una URL.

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 cogido 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).

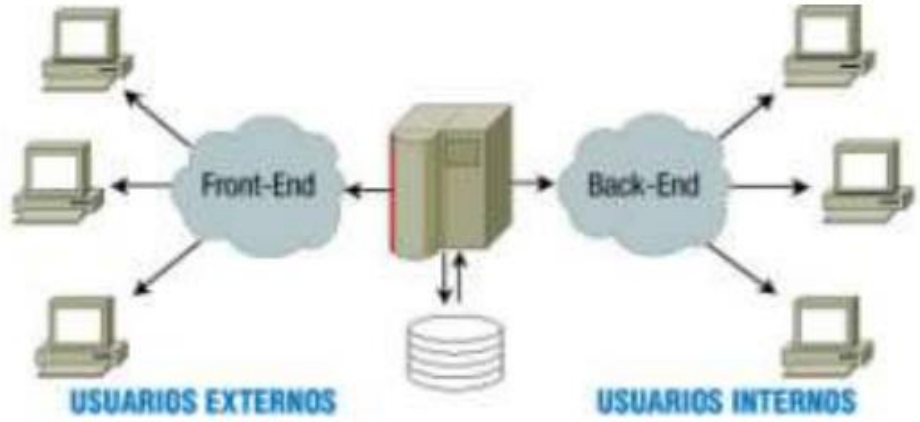
El número de capas de una aplicación web depende del tamaño de los componentes, de la naturaleza del servicio ofrecido y del reparto de las funciones entre el cliente y el servidor. Generalmente se componen por 3 capas, pero pueden aparecer más si trabamos con un modelo de aplicación MVC o si la aplicación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

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).

El Front-end es el conjunto de páginas que están destinadas a ser utilizadas por los clientes (Interfaz, diseño web...)

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...)



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: son programas ejecutados en el lado del cliente que no están conectados a ninguna base de datos. Generalmente trabajan con JavaScript, HTML y CSS.

Página web dinámica: son programas ejecutados en el lado del servidor que están asociados a una base de datos. Trabajan con HTML, CSS, PHP, MySQL...

Aplicación web: es lo mismo que una web dinámica solo que aumenta su grado de complejidad ya que añade control de acceso, sistema gestor de base de datos...

Single page aplicación

Mashup: es una mezcla de las webs dinámicas y estáticas ya que el código es ejecutado tanto del lado del servidor como del cliente y que añade más tecnologías (Java, PHP...) para dar soluciones ya sea para un cliente o para un web service.

8. Componentes de una aplicación web

Servidor web

Modulo interprete

Lenguaje de programación

Sistema gestor de Base de Datos (y datos)

Cliente web (navegador)

REVISION

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

CLIENTE	SERVIDOR
HTML	PHP
CSS	ASP.NET
JavaScript	Ruby
	Java
	JavaScript
	HTML
	CSS
	Python
	JS

FALTA DESARROLLAR

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:

- Desarrollo de aplicaciones web dinámicas
- Programación orientada a objetos.
- Módulos externos para la aplicación web
- Código abierto y gratuito
- Se usa en frameworks como Lavarel o Symfony

ASP.NET:

- Desarrollo web basado en formularios web
- Basado en el modelo code-behind para separar el contenido de la presentación
- Creado por Microsoft y de pago
- Trabaja con el modelo MVC

Ruby:

- Trabaja con el modelo MVC
- Orientado a objetos
- Se usa en frameworks como Ruby on rails

Java:

- Orientado a objetos
- Se usa en el framework de Spring
-

JavaScript:

- Node.js

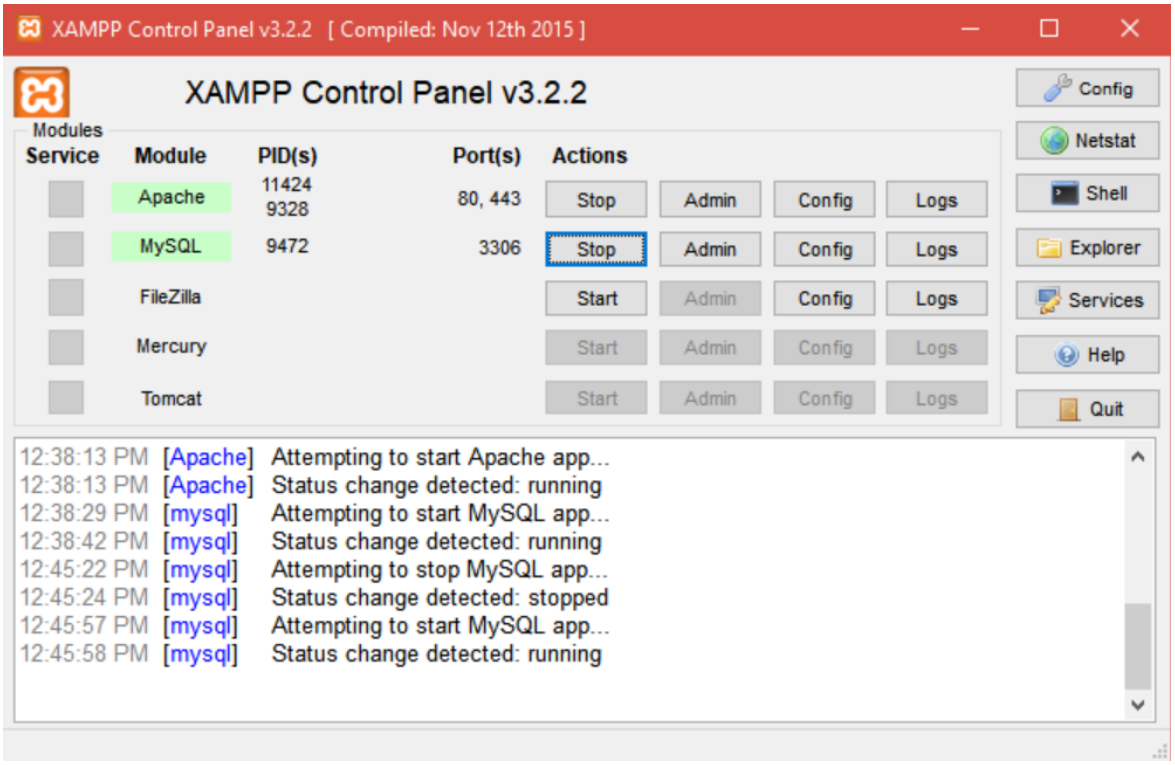
Enlaces:

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



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 en el caso de querer usar una aplicación necesitaremos el JVM ya que este es el intérprete de Java.

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

Eclipse: es un entorno de desarrollo integrado de código abierto multiplataforma. Emplea un sistema de módulos que nos permite incluir únicamente las funcionalidades que necesitemos. Sus características son:

- Un editor de texto con analizador sintáctico
- Compilación en tiempo real
- Pruebas unitarias con Junit
- Control de versiones con CVS
- Integración con Ant
- Asistentes para la creación de proyectos
- Sistema de refactorización

Atom:

NetBeans: es un entorno de desarrollo integrado libre, orientado principalmente para el lenguaje de programación Java (aunque ofrece la posibilidad de instalar módulos que nos permiten trabajar con otros lenguajes como por ejemplo PhP). Sus características son:

- Sistema de proyectos basado en Ant
- Control de versiones
- Sistema de refactorización

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

Nginx

Cloudflare Server

Microsoft IIS

LiteSpeed

FALTA DESARROLLAR

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, http está pensado para que el desarrollo del lado del servidor este escrito en php Pearl y el otro java.

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

Buscar página de navegadores web

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.

Tener la misma estructura de directorios en ambas maquinas

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.