

LICENCIATURA EN SISTEMAS DE LA INFORMACION

**TP 1**

# **INTRODUCCION A LA WEB**



**UADER FCyT**

## **PROGRAMACION AVANZADA**

**ALUMNOS**

**ERRANDONEA GONZALO  
ROMERO GONZALO**

**PROFESORES**

**LEDESMA ERNESTO**

**19/07/2022**

## Contenidos

Arquitectura y Protocolos de la World Wide Web .....	¡Error! Marcador no definido.
Bibliografía.....	1
Introducción .....	1
Preguntas.....	1

## Bibliografía

- <http://www.w3.org/>
- [TS06] A. Tanenbaum/ M. Van Steen: *Distributed Systems Principles and paradigms* - Prentice Hall – ed. 2 – 2006
- [OR02] R. Orfali/D.Harkey/J Edwards: *Cliente/Servidor y Objetos Guía de supervivencia* – 3ra ed. – Oxford University Press.

## Introducción

En esta práctica veremos los elementos del sistema World Wide Web.

## Preguntas

### *Historia de la WWW*

1. **¿Como surgió el sistema de documentos distribuidos WWW?**  
Tim Berners-Lee, en aquella época le inspiró su creación fue ver cómo físicos de todo el mundo se reunían para debatir cuestiones científicas en los edificios del CERN.
2. **¿En que año se creo la www? ¿Quien la creo? ¿Cuál fue el propósito inicial?**  
La World Wide Web (la Web) nació en 1989 en el CERN, el Centro Europeo de Física Nuclear, en Ginebra (Suiza), de la mano del ingeniero y físico británico Tim Berners-Lee como un sistema de intercambio de datos entre los 10.000 científicos que trabajaban en la institución
3. **¿Sobre que red funcionaba? ¿Cuales fueron los componentes iniciales?**  
El primer servidor web fue corrido en un NeXT Computer  
Componentes.  
El protocolo HTTP dirige las transferencias de datos entre el servidor y el cliente.  
Para acceder a un componente de la Web, el cliente proporciona un único identificador universal, llamado URL por sus siglas en inglés de Localizador Uniforme de Recursos (Uniform Resource Locator)  
HTML por sus siglas en inglés de Lenguaje de Marcas de Hipertexto (Hypertext Markup Language) es el formato más común para publicar documentos web.

### *WWW en la actualidad*

4. **Nombre y describa las distintas etapas de evolución en la web. Realice una cronología de tecnologías que surgieron en la web hasta la actualidad.**  
Evolución de la web

Web 1.0: Fue la primera que apareció en el año 1990 y en ella solo se puede consumir contenido. Se trata de información a la que se podía acceder, pero sin posibilidad de interactuar; era unidireccional. La web 1.0 es la forma más básica que existe de navegadores de solo texto. es muy primitiva para lo que hoy ofrece la web.

Web 2.0: Cuando hablamos de la Web 2.0 o la Web Social, nos referimos a un modelo de páginas Web que facilitan la transmisión de información, la interoperabilidad y la colaboración entre sus usuarios, mediante un diseño centrado en sus necesidades, más que en las de la empresa. En otras palabras, se trata de una tendencia en la Internet que aboga por una red más interactiva, menos unilateral, en la que los usuarios no ocupen un rol meramente pasivo.

Web 3.0: está relacionada a lo que se conoce como web semántica. Los usuarios y los equipos, en este marco, pueden interactuar con la red mediante un lenguaje natural, interpretado por el software. De esta manera, acceder a la información resulta más sencillo. Dicho de otro modo, todos los datos alojados en la web 3.0 deberían ser “entendidos” por las máquinas, que podrían procesarlos con rapidez.

La web 3.0, en definitiva, está relacionada con la inteligencia artificial. Los sitios web incluso tendrían la capacidad de conectarse entre sí de acuerdo a los intereses del usuario.

1989 -1991: Tim Berners inventa la WWW basada en HTTP HTML Y URL

1992 – 1993: Surgen los primeros navegadores graficos ViolaWWW, Mosaic, SpyceGlass

1994 – Se lanza Netscape, también nace Yahoo!, y también nace PHP, y w3c (fija estándares para la web)

1995 – Nace internet explorer, el cual venia gratis con Windows 95

1995 – Se crea JavaScript desarrollado por Brendan Eich de Netscape, nace MySql y JAVA

1996 - Surge CSS1, Macromedia flash,

1998 - Surge CSS2, Tambien surge Google y html4

1999 – Surge CSS3, saleforce y ajax

2001 - Aparece Wikipedia, c#, y Json

2002 – Surge Firefox, Linked In

2005 – Nace google maps, sass, y youtube

2006 – Nace JQuery }

2007 – Aparece github y también se empezó a pensarse en una web para Mobile y el concepto del meta viewport

2008 – Aparece HTML5

2009 – Nace NodeJS, mongoDB y bitcoin

2010 – Aparece AngularJS y se define el termino responsive web design

2012 – Aparece Polymer, con el concepto de web components

2013 – Se crea ReactJS con el mismo concepto de web components, y tambien aparece Slack y tambien paso lo de Edward Snowden

2014 – HTML5 se estandariza, y también aparece VueJS

2015 – Se crea React Native

2016 – Aparece Figma, TypeScript, NextJS, Github Pages y Aparece Angular 2

2017 – Aparece web assembly, y Flutter

2019 – Se creo el 5G

2020 – Se popularizo muchísimo el trabajo remoto

5. **¿Qué es un Servidor WWW?**

Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. El código recibido por el cliente es renderizado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se usa el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del modelo OSI. El término también se emplea para referirse al ordenador donde se ejecutan esas tareas.

6. **¿Qué es un cliente web? explique detalladamente.**

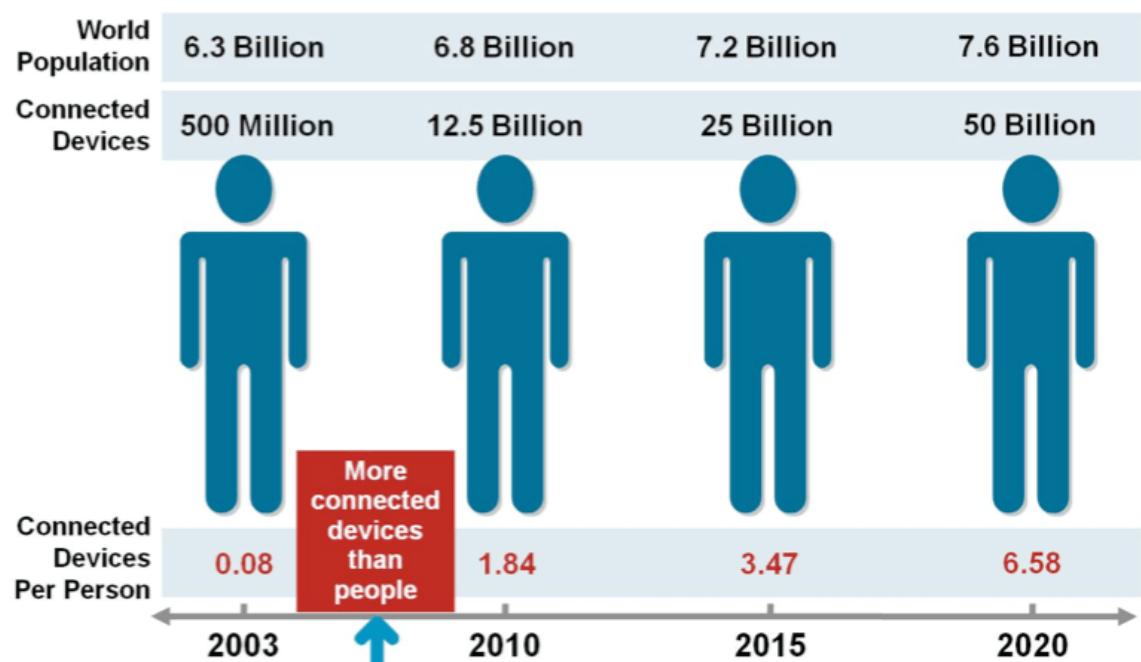
La arquitectura cliente-servidor es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. De acuerdo al diccionario de informática y tecnología, un cliente web es cualquier aplicación que sirve para utilizar la web, como por ejemplo, un navegador de internet.

7. **¿Como está compuesto el sistema WWW actual?**

La World Wide Web se basa en hipertextos, es decir, archivos de texto (páginas) en los que se pueden insertar hipervínculos o enlaces que conducen a los usuarios de una página web a otra, o a otro punto de esa misma página. Su función es ordenar y distribuir la información que existe en internet.

8. **Realice un diagrama de crecimiento de host conectados a Internet desde su inicio hasta la actualidad.**

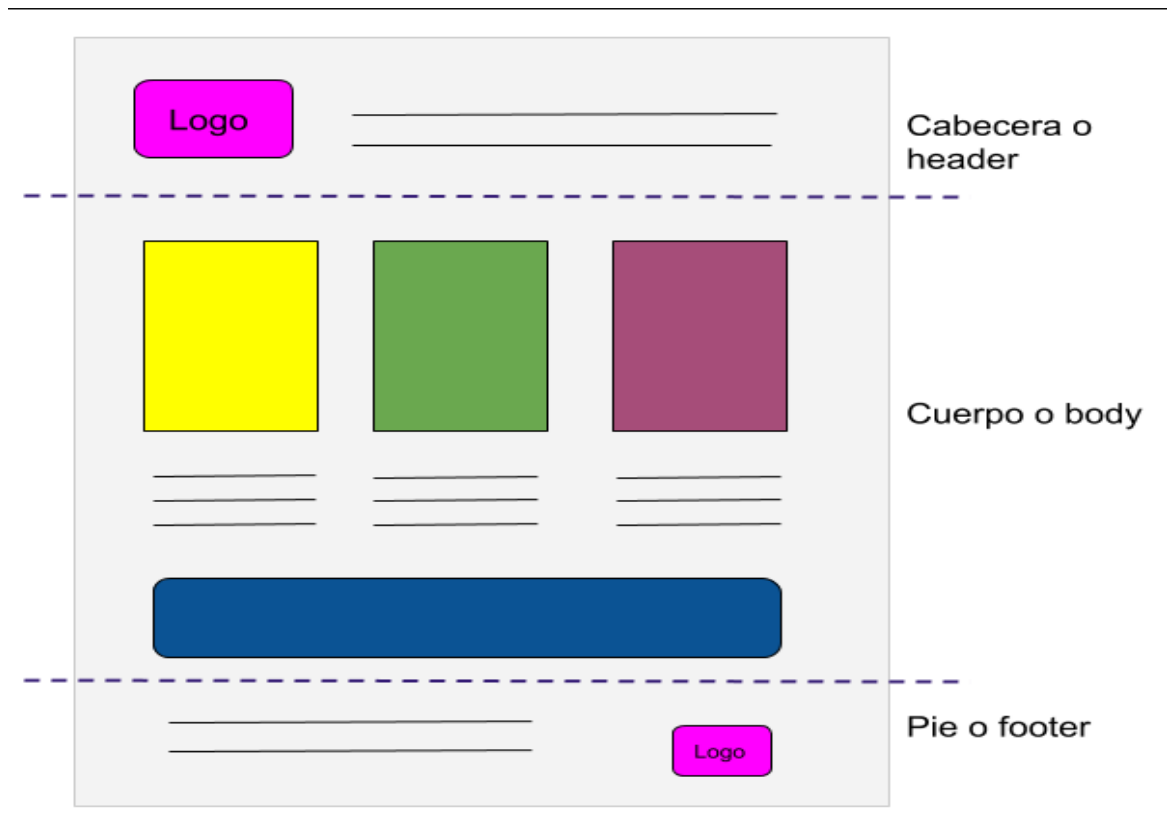
Figure 1. The Internet of Things Was "Born" Between 2008 and 2009



Source: Cisco IBSG, April 2011

9. **Realiza un diagrama conceptual de composición de la web. ¿Cuáles son los Componentes de la www?**

Los componentes de la www son los URL, el protocolo HTTP y el lenguaje HTML



**10. ¿Cuál es la Arquitectura del sistema distribuido de documentos?**

La arquitectura del sistema distribuida se conoce como arquitectura basada en eventos.

**11. ¿Cómo funciona en sistema www?**

- El direccionamiento URL (Localizador Uniforme de Recursos) sirve para encontrar archivos en internet utilizando nombres. Cada recurso en la red tiene un URL único que conduce al navegador hasta esa información y no a otra.

¿Cómo funciona? Primero, el navegador contacta con un servidor DNS (Sistema de Nombres de Dominio) para que traduzca ese nombre a una dirección IP, numérica. Una vez recibida, la usa para dirigirse al servidor y recurso concreto en Internet. Con este sistema, aunque cambien las direcciones IP (que lo hacen, y a menudo), el navegador solo tiene que recordar el nombre. Si la base de datos DNS está actualizada (suelen tardar un rato), encontrará el recurso.

## ***Tecnologías, Servidores y Clientes.***

**12. ¿Que es HTML? ¿De donde proviene?**

HTML (Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language) es el componente más básico de la Web. Define el significado y la estructura del contenido web.

- El origen de HTML se remonta a 1980, cuando el físico Tim Berners-Lee, investigador del CERN (Organización Europea para la Investigación Nuclear) propuso un nuevo sistema de "hipertexto" para compartir documentos.
- Tras finalizar el desarrollo de su sistema, Tim Berners-Lee lo presentó a una convocatoria organizada para desarrollar un sistema de hipertexto para internet. Después de unir sus fuerzas con el ingeniero de sistemas Robert Cailliau, presentaron la propuesta ganadora llamada WorldWideWeb (W3). El primer documento formal con la descripción de HTML se publicó en 1991 bajo el nombre "HTML Tags" (Etiquetas HTML) y todavía hoy puede ser consultado en línea a modo de reliquia informática.

**13. ¿A que se le llama pagina web estática? ¿y pagina dinámica? ¿y página activa?**

Las páginas web dinámicas son generadas en el servidor por la ejecución del lenguaje Script sobre el cual se encuentran programadas y el resultado es una salida HTML despachada al navegador. Sin embargo, una web estática es un archivo de texto HTML que despacha el servidor hacia el navegador, por lo que requiere menor procesamiento del lado servidor, ya que solo necesita localizar el archivo HTML, leerlo y despacharlo al navegador.

En las páginas web estáticas no se utilizan bases de datos ni se requiere programación. Este tipo de webs son mas económicas ya que el tiempo de programación es mucho menor que en las paginas dinámicas.

Una página web activa es un tipo de página HTML (identificada por un nombre de archivo. Asp) que incluye scripts (pequeños programas) que se procesan en un servidor web antes de que la página web se sirva al navegador web del usuario.

**14. ¿Qué es una URL? ¿qué función cumple en la Web? ¿Qué es un servidor de archivos basado en URL?**

Junto con el Hipertexto y HTTP, las URL son uno de los conceptos claves de la Web. Es el mecanismo usado por los navegadores para obtener cualquier recurso publicado en la web.

URL significa Uniform Resource Locator (Localizador de Recursos Uniforme). Una URL no es más que una dirección que es dada a un recurso único en la Web. En teoría, cada URL valida apunta a un único recurso. Dichos recursos pueden ser páginas HTML, documentos CSS, imagenes, etc. En la practica, hay algunas excepciones, siendo la más común una URL apuntando a un recurso que ya no existe o que ha sido movido. Como el recurso representado por la URL y la URL en si son manejadas por el servidor Web, depende del dueño del servidor web manejar ese recurso y su URL asociada adecuadamente.

Una URL está compuesta de diferentes partes, algunas obligatorias y otras opcionales. Veamos las partes más importantes usando la siguiente URL:

- Http es el protocolo. La primera parte de la URL indica qué protocolo debe usar el navegador. Un protocolo es un método establecido para intercambiar o transferir datos alrededor de una red informática. Por lo general, para sitios web es el protocolo HTTP o su versión segura, HTTPS. La Web requiere uno de estos dos, pero los navegadores también saben cómo manejar otros protocolos como mailto: (para abrir un cliente de correo) o ftp: para manejar la transferencia de archivos, así que no se sorprenda si ve tales protocolos.
- www.example.com es el nombre de dominio. Indica qué servidor web se solicita. Alternativamente, es posible usar directamente un dirección IP, pero debido a que es menos conveniente, no se usa con frecuencia en la Web.
- :80 es el puerto. Indica la "puerta" técnica utilizada para acceder a los recursos en el servidor web. Por lo general, se omite si el servidor web utiliza los puertos estándar del protocolo HTTP (80 para HTTP y 443 para HTTPS) para otorgar acceso a sus recursos. De lo contrario es obligatorio.
- /path/to/myfile.html es la ruta al recurso en el servidor web. En los primeros días de la Web, una ruta como esta representaba la ubicación de un archivo físico en el servidor web. Hoy en día, es principalmente una abstracción manejada por servidores web sin ninguna realidad física.

**15. ¿Dónde y como se ejecuta el lenguaje HTML?**

HTML se ejecuta en el navegador web.

**16. ¿Qué es un lenguaje script en la tecnología WWW?**

Un lenguaje de script es un pequeño lenguaje de programación cuyo código se inserta dentro del documento HTML. Este código se ejecuta en el navegador del usuario al cargar la página, o cuando sucede algo especial como puede ser el pulsar sobre un enlace.

Estos lenguajes permiten variar dinámicamente el contenido del documento, modificar el comportamiento normal del navegador, validar formularios, realizar pequeños trucos visuales, etc... Sin embargo, conviene recordar que se ejecutan en el navegador del usuario y no en la máquina donde estén alojadas, por lo que no podrán realizar cosas como manejar bases de datos. Esto hace que los contadores (por ejemplo) se deban realizar de otra manera, utilizando programas CGI.

**17. ¿Qué es un plugin? ¿qué es una cookie?**

Plugin: Un plugin es un software que puede ser instalado en un programa y que permite ampliar las capacidades del mismo. Aplicados a los sitios web o sitios ecommerce, se trata de aplicaciones que añaden funcionalidades a una estructura básica web

Cookie: es un término que hace referencia a una pequeña información enviada por un sitio web y almacenada en el navegador del usuario, de manera que el sitio web puede consultar la actividad previa del navegador.

**18. ¿Qué es JavaScript? ¿Dónde y quien ejecuta JavaScript? ¿Cómo nació JavaScript?**

JavaScript es un lenguaje de programación ligero, interpretado, con funciones de primera clase. Si bien es más conocido como un lenguaje de scripting para páginas web, y es usado en muchos entornos fuera del navegador, tal como Node.js, Apache CouchDB y Adobe Acrobat JavaScript es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa (por ejemplo programación funcional).

Como se dijo anteriormente, JS no solo se ejecuta en navegadores, sino se pueden hacer aplicaciones con el, además de poder utilizarse en backend, sin embargo, en la parte web, el script que se ejecuta en la parte del cliente, lo hace el mismo navegador del usuario, respondiendo, por ejemplo, a ciertos eventos.

JavaScript, en su momento LiveScript, fue creado por Brendan Eich, desarrollado en tan solo 1 semana, y nace a raíz del problema de que las páginas eran estáticas.

Netscape para evitar problemas entre JS y JScript (JScript era el lenguaje que había desarrollado Microsoft, al haber hecho ingeniería inversa al navegador de Netscape)

propone estandarizar JS, para así no haya problemas para los usuarios a la hora de navegar en diferentes navegadores.

**19. ¿Qué es un Applet de Java? ¿Cómo puede un Browser ejecutar código Java?**

El Applet es un programa que puede incrustarse en un documento HTML (página Web). Cuando un Navegador carga una página Web que contiene un Applet, éste se descarga en el navegador Web y comienza a ejecutarse, lo cual nos permite crear programas que cualquier usuario puede ejecutar. Uno de los requisitos para el applet funcione es tener instalado Java y que esté activado a través del explorador web.

**20. ¿Cómo funciona la tecnología Flash de Macromedia?**

Adobe Flash es una herramienta de autoría para la creación de contenido multimedia e interactivo. Permite conectar componentes gráficos y programación técnica. Se pueden crear nuevos objetos gráficos o integrar los ya existentes. Al mismo tiempo, pueden ser fácilmente dirigidos y animados por código ActionScript. Además, los elementos de audio se pueden añadir rápidamente. Por lo que se aplica desde juegos hasta banners.

Flash es una aplicación que nos permite trabajar películas (movie), y como en toda película tenemos varias escenas, así mismo flash nos permite crearlas. De este modo, podemos crear submenús que nos enlacen a escenas que se encuentran dentro de la misma película dándonos así una sensación de navegación e interactividad.

Otra ventaja que posee este software, es que nos permite trabajar por capas. Estas capas funcionan como acetatos que se ubican uno encima de otro. Es cada acetato se coloca un elemento, el cual tapará total o parcialmente a los objetos que se encuentren debajo de él, sólo si se encuentran en la misma posición. De otra manera, dejaría ver los objetos que están ubicados a la parte transparente que corresponderá a la parte no dibujada de la capa. (similar al manejo de Photoshop).

**21. ¿Para que se utiliza la tecnología CGI? ¿Qué lenguajes de programación se pueden utilizar para los scripts CGI?**

Interfaz de entrada común (en inglés Common Gateway Interface, abreviado CGI) es una importante tecnología de la WWW que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web. CGI especifica un estándar para transferir datos entre el cliente y el programa. Es un mecanismo de comunicación entre el servidor web y una aplicación externa cuyo

resultado final de la ejecución son objetos MIME. Las aplicaciones que se ejecutan en el servidor reciben el nombre de CGIs.

Un programa CGI puede ser escrito en cualquier lenguaje de programación que produzca un archivo ejecutable. Entre los lenguajes más habituales se encuentran: C, C++, Perl, Java, Visual Basic, Cobol, etc...

## 22. ¿Qué es un “form” HTML? ¿para que se utiliza?

El elemento HTML form (<form>) representa una sección de un documento que contiene controles interactivos que permiten a un usuario enviar información a un servidor web.

## 23. ¿Qué son y cómo funcionan las tecnologías ASP, PHP y JSP?

ASP: ASP (Active Server Pages) es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página web, utilizando el lenguaje Visual Basic Script o Jscript.

El lenguaje ASP a grandes rasgos funciona así: un computador cliente hace una petición de una página ASP. El computador servidor interpreta esta petición y le envía la página web. El resultado final es una página HTML que se le envía al cliente

PHP: PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

En lugar de usar muchos comandos para mostrar HTML (como en C o en Perl), las páginas de PHP contienen HTML con código incrustado que hace "algo". El código de PHP está encerrado entre las etiquetas especiales de comienzo y final <?php y ?> que permiten entrar y salir del "modo PHP".

Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga.

JSP: JSP es un acrónimo de Java Server Pages. Es, pues, una tecnología orientada a crear páginas web con programación en Java.

Con JSP podemos crear aplicaciones web que se ejecuten en variados servidores web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. Por tanto, las JSP podremos escribirlas con nuestro editor HTML/XML habitual.

## 24. ¿Qué es DHTML? ¿y XHTML?

DHTML: HTML dinámico o DHTML (Dinamic HyperText Markup Language) es un término colectivo que sirve para designar una combinación de nuevas etiquetas del lenguaje HTML y nuevas opciones de estilo y programación que permiten crear páginas web más dinámicas y animadas, a la vez que ofrecen una mayor interactividad por parte del usuario que las versiones previas de HTML.

XHTML: XHTML (eXtensible HyperText Markup Language) es, básicamente, HTML expresado como XML válido. Es más estricto a nivel técnico, pero esto permite que posteriormente sea más fácil al hacer cambios o buscar errores entre otros.

## 25. ¿Qué es XML? explique detalladamente.

XML es un lenguaje de marcado similar a HTML. Significa Extensible Markup Language (Lenguaje de Marcado Extensible) y es una especificación de W3C como lenguaje de marcado de propósito general. Esto significa que, a diferencia de otros lenguajes de marcado, XML no está predefinido, por lo que debes definir tus propias etiquetas. El propósito principal del lenguaje es compartir datos a través de diferentes sistemas, como Internet.

Para que un documento XML sea correcto, debe ser un documento bien formado, cumpliendo todas las reglas de sintaxis de XML, y válidas, acorde a las reglas de un lenguaje específico. Un ejemplo de un documento que no está bien formado es uno que tiene una etiqueta de apertura y no tiene una de cierre, ni se cierra en sí misma.

## 26. ¿Que son los Servicios Web? De un ejemplo de aplicación utilizando Servicios WEB.

Un servicio web (en inglés, web service o web services) es una tecnología que utiliza un conjunto de



protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web.

Un ejemplo de app con un servidor es con el modelo cliente servidor, básicamente el cliente, (un computador que esta en nuestro sitio) hace una petición al servidor, y este, servidor, responde a esa petición, dando algún recurso, como un dato, imagen, etc., a su vez, el servidor, no responde a un solo cliente, sino a muchos clientes, a su vez, el servidor para comunicarse, usa el protocolo HTTP.

## ***URL's, Dominios y Direcciones.***

### **27. ¿Qué es una URL? ¿qué función cumple en la web? ¿Cuál es la estructura de una URL?**

Junto con el Hipertexto y HTTP, las URL son uno de los conceptos claves de la Web. Es el mecanismo usado por los navegadores para obtener cualquier recurso publicado en la web.

URL significa Uniform Resource Locator (Localizador de Recursos Uniforme). Una URL no es más que una dirección que es dada a un recurso único en la Web. En teoría, cada URL valida apunta a un único recurso. Dichos recursos pueden ser páginas HTML, documentos CSS, imagenes, etc. En la práctica, hay algunas excepciones, siendo la más común una URL apuntando a un recurso que ya no existe o que ha sido movido. Como el recurso representado por la URL y la URL en si son manejadas por el servidor Web, depende del dueño del servidor web manejar ese recurso y su URL asociada adecuadamente.

Una URL está compuesta de diferentes partes, algunas obligatorias y otras opcionales. Veamos las partes más importantes usando la siguiente URL:

<http://www.example.com:80/path/to/myfile.html?key1=value1&key2=value2#SomewhereInTheDocument>

http es el protocolo. La primera parte de la URL indica qué protocolo debe usar el navegador. Un protocolo es un método establecido para intercambiar o transferir datos alrededor de una red informática. Por lo general, para sitios web es el protocolo HTTP o su versión segura, HTTPS. La Web requiere uno de estos dos, pero los navegadores también saben cómo manejar otros protocolos como mailto: (para abrir un cliente de correo) o ftp: para manejar la transferencia de archivos, así que no se sorprenda si ve tales protocolos.

www.example.com es el nombre de dominio. Indica qué servidor web se solicita. Alternativamente, es posible usar directamente un dirección IP, pero debido a que es menos conveniente, no se usa con frecuencia en la Web.

:80 es el puerto. Indica la "puerta" técnica utilizada para acceder a los recursos en el servidor web. Por lo general, se omite si el servidor web utiliza los puertos estándar del protocolo HTTP (80 para HTTP y 443 para HTTPS) para otorgar acceso a sus recursos. De lo contrario es obligatorio.

/path/to/myfile.html es la ruta al recurso en el servidor web. En los primeros días de la Web, una ruta como esta representaba la ubicación de un archivo físico en el servidor web. Hoy en día, es principalmente una abstracción manejada por servidores web sin ninguna realidad física.

?key1=value1&key2=value2 son parámetros adicionales proporcionados al servidor web. Esos parámetros son una lista de pares clave/valor separados con el símbolo &. El servidor web puede usar esos parámetros para hacer cosas adicionales antes de devolver el recurso. Cada servidor web tiene sus propias reglas con respecto a los parámetros, y la única forma confiable de saber si un servidor web específico está manejando parámetros es preguntando al propietario del servidor web.

#SomewhereInTheDocument es un ancla para otra parte del recurso en sí. Un ancla representa una especie de "marcador" dentro del recurso, dando al navegador las instrucciones para mostrar el contenido ubicado en ese lugar "marcado". En un documento HTML, por ejemplo, el navegador se desplazará hasta el punto donde se define el ancla; en un video o documento de audio, el navegador intentará ir a la hora que representa el ancla. Vale la pena señalar que la parte después del #, también conocido como el identificador de fragmento, nunca se envía al servidor con la solicitud.

### **28. ¿Qué es un servidor de archivos basado en URL?**

Un file server (o servidor de archivos) es un servidor central de una red de ordenadores, que pone a disposición del cliente conectado un conjunto de archivos o parte del mismo. De este modo, el servidor de archivos

proporciona a los usuarios un lugar de almacenamiento centralizado para los archivos en sus propios soportes de datos, disponible para todos los clientes autorizados. El administrador del servidor establece unas directrices estrictas para determinar qué usuarios tienen derechos de acceso y a qué datos: por ejemplo, configurando el sistema u otorgando los permisos del sistema de archivos, se puede determinar qué carpetas puede ver y abrir un usuario o grupo de usuarios concreto y si los archivos solo son visibles o también pueden guardarse, modificarse o eliminarse.

Teniendo esto en cuenta, se podría decir que, un servidor de archivos basados en URL no es mas que lo descrito anteriormente, con lo que, mediante URL, se podrían acceder a dichos archivos

## 29. ¿Cuál es el formato de una URL en IPv6?

El tamaño y el formato de la dirección IPv6 amplían la capacidad del espacio para direcciones.

El tamaño de las direcciones de IPv6 es de 128 bits. La representación preferente de direcciones de IPv6 es x:x:x:x:x:x:x:x, donde cada x es el valor hexadecimal de las 8 partes de 16 bits de la dirección. Las direcciones IPv6 abarcan desde 0000:0000:0000:0000:0000:0000:0000 a ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff.

## 30. ¿Qué es un Dominio www? ¿Qué relación tiene con una URL?

Es el nombre único e irrepetible que se le da a un sitio en Internet para que las marcas (propietarias de dichas páginas) sean identificadas de forma cómoda y sencilla por los usuarios y sus clientes. La relación que tiene el dominio con la URL, es que, este hace referencia a un recurso de un servidor en internet, por ejemplo, [www.facebook.com](http://www.facebook.com), es la URL para acceder a Facebook, a su vez, cuando se ingresa, se obtienen el html, el css, y el js, que nos provee el servidor mediante la request que se hace al introducir dicho URL en nuestro navegador.

Aclarar que, una URL es una dirección web completa que se utiliza para encontrar una página web específica. Mientras que el dominio es el nombre del sitio web, una URL conduce a alguna página individual dentro del sitio web. Cada URL contiene un nombre de dominio, así como otros componentes necesarios para localizar la página o el contenido específicos.

## 31. ¿Cuál es la función del DNS en la WEB?

DNS son las iniciales de Domain Name System (sistema de nombres de dominio) y es una tecnología basada en una base de datos que sirve para resolver nombres en las redes, es decir, para conocer la dirección IP de la máquina donde está alojado el dominio al que queremos acceder.

Cuando un ordenador está conectado a una red (ya sea Internet o una red casera) tiene asignada una dirección IP. Si estamos en una red con pocos ordenadores, es fácil tener memorizadas las direcciones IP de cada uno de los ordenadores y así acceder a ellos pero ¿qué ocurre si hay miles de millones de dispositivos y cada uno tiene una IP diferente? Pues que se haría imposible, por eso existen los dominios y las DNS para traducirlos.

Por lo tanto, el DNS es un sistema que sirve para traducir los nombres en la red, y está compuesto por tres partes con funciones bien diferenciadas.

**Cliente DNS:** está instalado en el cliente (es decir, nosotros) y realiza peticiones de resolución de nombres a los servidores DNS.

**Servidor DNS:** son los que contestan las peticiones y resuelven los nombres mediante un sistema estructurado en árbol. Las direcciones DNS que ponemos en la configuración de la conexión, son las direcciones de los Servidores DNS.

**Zonas de autoridad:** son servidores o grupos de ellos que tienen asignados resolver un conjunto de dominios determinado (como los .es o los .org).

## *Protocolos utilizados en la www*

## 32. ¿Cuál es la relación entre HTML y HTTP?

La forma en la que se relaciona html y http es de la siguiente manera: HTML define la estructura de las páginas web, la dirección URL, una forma derivada del URI, define cómo se localiza a un recurso (por ejemplo, una página web) en Internet. El protocolo HTTP, en cambio, regula cómo el servidor envía este recurso al cliente

## 33. ¿Cuáles son las versiones de HTTP y cuales sus diferencias?

Las versiones de HTTP son las siguientes: HTTP/1, HTTP/1.1, HTTP/2, HTTP/3

La versión 1.0 y 1.1 establecieron los elementos básicos que conocemos, como por ejemplo los códigos 2xx, 3xx, 4xx o 5xx.

La versión 2.0 requiere del uso de un sistema de cifrado si se quiere aprovechar al máximo. Sigue funcionando por el sistema TCP y añade un sistema principal de streams. Esto permite mejorar la paralelización de peticiones.

La versión 3.0 mejora el sistema funcionando por UDP mediante QUIC.

La mayor diferencia entre el HTTP2 y el HTTP3 es el protocolo sobre el que funciona. El sistema TCP tiene sistemas que permiten saber que algo que se manda a un usuario desde un servidor llega a su destino, mientras que UDP no, aunque el sistema UDP, al no tener estos sistemas de verificación funciona mucho más rápido. Con el sistema QUIC se añaden ciertas medidas de fiabilidad en la interconexión de datos.

#### 34. **¿Qué es MIME?**

El tipo Extensiones multipropósito de Correo de Internet (MIME) es una forma estandarizada de indicar la naturaleza y el formato de un documento, archivo o conjunto de datos. Está definido y estandarizado en IETF RFC 6838. La Autoridad de Números Asignados de Internet (IANA) es el organismo oficial responsable de realizar un seguimiento de todos los tipos MIME oficiales, y puede encontrar la lista más actualizada y completa en la página de tipos de medios (Media Types).

Los navegadores a menudo usan el tipo MIME (y no la extensión de archivo) para determinar cómo procesará un documento; por lo tanto, es importante que los servidores estén configurados correctamente para adjuntar el tipo MIME correcto al encabezado del objeto de respuesta.

#### 35. **Explica cómo es una solicitud HTTP.**

HTTP se basa en un modelo solicitud / respuesta, de modo que hay dos tipos de mensajes HTTP: la solicitud y la respuesta. El navegador abre una conexión a un servidor y realiza una solicitud. El servidor procesa la solicitud del cliente y devuelve una respuesta.

La solicitud HTTP

La línea de salida de una petición HTTP se conoce como línea de la petición. Siempre es la primera línea del mensaje de solicitud y contiene tres campos:

- Un método HTTP
- Un identificador universal de recursos (URI)
- Una versión del protocolo HTTP

Aunque hay varios métodos de HTTP para recuperar datos de un servidor, los dos más utilizados son GET y POST.

El método GET solicita un recurso del servidor indicado en el campo URI. Si la URI apunta a una base de datos de producción de recursos como un servlet, los datos serán devueltos dentro del mensaje de respuesta.

El método POST se utiliza para pasar explícitamente datos al servidor en el propio mensaje de solicitud. Estos datos estarán a disposición del recurso URI especificado en la solicitud.

La petición HTTP puede contener cero o más campos de cabecera. Estos campos de cabecera permiten pasar al servidor información del propio cliente y de la solicitud misma. El formato de un campo de cabecera, tanto para la solicitud y la respuesta, es el nombre del campo de cabecera seguido de dos puntos (:) y el valor. Si se especifican varios valores para un único campo de cabecera, deben estar separados por comas. Algunos de los más comúnmente utilizados son:

Accept: Indica que los tipos de medios son aceptables para la respuesta. En caso de que el campo Accept no aparezca en la cabecera el servidor puede asumir que el cliente acepta todos los tipos. Un ejemplo de un valor de la cabecera Accept es "image/gif, image/jpeg"

Accept-Charset: Indica lo que los conjuntos de caracteres son aceptables para la respuesta. Si éste campo no está presente en la solicitud, el servidor puede asumir que cualquier conjunto de caracteres es aceptable. La ISO-8859-1 juego de caracteres, puede suponerse que se acepta por todos los agentes de usuario.

Accept-Encoding: Similar a la cabecera Accept campo, pero que restringe aún más los valores aceptados por el cliente. Un ejemplo de un Accept-Encoding es "compress, gzip".

Accept-Language: Indica los idiomas que el cliente prefiere para la respuesta. Un ejemplo de una cabecera Accept-Language es "en-us, de-li, es-us".

Content-Encoding: Indica el mecanismo de codificación que se ha aplicado al cuerpo del mensaje y, por lo tanto, qué mecanismo de decodificación debe utilizarse para obtener la información. Un ejemplo de

un encabezado Content-Encoding es "gzip".

Content-Type: Indica el tipo del cuerpo del mensaje enviado al destinatario. Un valor posible para Content-Type es "text/html; charset = ISO-8859-1"

Host: Indica el nombre del host y el número de puerto de los recursos que se solicitan, obtenidos de la URL original. Un ejemplo de un encabezado de host puede ser "www.somehost.com"

Referer: Permite al cliente especificar la dirección (URI) de los recursos desde la que obtiene la respuesta. Esta cabecera se utiliza con propósitos de mantenimiento y seguimiento

User-Agent: Contiene información sobre el cliente que originó la petición. Esta cabecera se utiliza principalmente con fines estadísticos y de seguimiento. Un ejemplo de un User-Agent es "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5,0)".

**36. ¿Cuáles son los métodos de HTTP 1? Enumere y describa el funcionamiento de cada uno.**

**1) GET: Solicita un documento al servidor.**

- Se pueden enviar datos en la URL

2) HEAD: Similar a GET, pero sólo pide las cabeceras HTTP.

- Comprobar enlaces
- Para consultar información sobre el fichero (fecha de modificación, tamaño, tipo de servidor, tipo de documento solicitado) antes de solicitarlo.

3) POST: Manda datos al servidor para su procesado.

- Similar a GET, pero además envía datos en el cuerpo del mensaje.
- La URL corresponde a una página dinámica que trata los datos enviados.

4) PUT: Almacena el documento enviado en el cuerpo del mensaje.

5) DELETE: Elimina el documento referenciado en la URL.

6) TRACE: Rastrea los intermediarios por los que pasa la petición.

7) OPTIONS: Averigua los métodos que soporta el servidor.

- En una caché sólo se guardan las respuestas de las peticiones realizadas con GET y HEAD (POST no).

8) Allow: métodos permitidos para el recurso

9) Content-Encoding: codificación de la entidad (p.e. compresión)

10) Content-Encoding: gzip

11) Content-Length: longitud de la entidad (importante en solicitud)

12) Content-Type: tipo MIME de la entidad

13) Content-Type: text/html; charset=iso-latin-1

14) Expires: fecha tope de validez en la caché

15) Last-Modified: fecha de la última modificación de la entidad

## **Bibliografía**

[https://www.alegsa.com.ar/Dic/cliente\\_web.php](https://www.alegsa.com.ar/Dic/cliente_web.php)

[https://developer.mozilla.org/es/docs/Learn/Common\\_questions/What\\_is\\_a\\_URL](https://developer.mozilla.org/es/docs/Learn/Common_questions/What_is_a_URL)

[https://www.cimat.mx/~amor/Academic/Books/Tutorial\\_HTML/script.html#:~:text=Un%20lenguaje%20de%20script%20es,el%20pulsar%20sobre%20un%20enlace](https://www.cimat.mx/~amor/Academic/Books/Tutorial_HTML/script.html#:~:text=Un%20lenguaje%20de%20script%20es,el%20pulsar%20sobre%20un%20enlace)

<https://www.beedigital.es/desarrollo-web/que-es-plugin-usos-tipos/>

<https://developer.mozilla.org/es/docs/Web/JavaScript>

<https://www.viafirma.com/faq/es/que-es-un-applet/>

[https://es.wikipedia.org/wiki/Interfaz\\_de\\_entrada\\_com%C3%BAn#:~:text=Interfaz%20de%20entrada%20com%C3%BAn%20\(en,el%20cliente%20y%20el%20programa](https://es.wikipedia.org/wiki/Interfaz_de_entrada_com%C3%BAn#:~:text=Interfaz%20de%20entrada%20com%C3%BAn%20(en,el%20cliente%20y%20el%20programa)

[https://es.wikipedia.org/wiki/Interfaz\\_de\\_entrada\\_com%C3%BAn#:~:text=Un%20programa%20CGI%20puede%20ser,Visual%20Basic%2C%20Cobol](https://es.wikipedia.org/wiki/Interfaz_de_entrada_com%C3%BAn#:~:text=Un%20programa%20CGI%20puede%20ser,Visual%20Basic%2C%20Cobol)

<https://developer.mozilla.org/es/docs/Web/HTML/Element/form>

<https://desarrolloweb.com/articulos/831.php>

<http://www.hipertexto.info/documentos/dhtml.htm>

<https://es.wikipedia.org/wiki/XHTML>

[https://developer.mozilla.org/es/docs/Web/XML/XML\\_introduction](https://developer.mozilla.org/es/docs/Web/XML/XML_introduction)

[https://developer.mozilla.org/es/docs/Learn/Common\\_questions/What\\_is\\_a\\_URL](https://developer.mozilla.org/es/docs/Learn/Common_questions/What_is_a_URL)

<https://www.ionos.es/digitalguide/servidores/know-how/file-server/>

<https://rockcontent.com/es/blog/que-es-un-dominio/#:~:text=Un%20dominio%20en%20Internet%20es,internauta%20pueda%20visitarlo%20e%20identificarlo.&text=Si%20sabes%20elegir%20el%20adecuado,digital%20marchar%C3%A1%20por%20excelente%20camino>

[https://domains.google/intl/es\\_mx/learn/web-terms-101/#:~:text=son%20lo%20mismo,-.URL,individual%20dentro%20del%20sitio%20web](https://domains.google/intl/es_mx/learn/web-terms-101/#:~:text=son%20lo%20mismo,-.URL,individual%20dentro%20del%20sitio%20web)

<https://www.xatakamovil.com/conectividad/como-funciona-internet-dns#:~:text=DNS%20son%20las%20iniciales%20de,dominio%20al%20que%20queremos%20acceder>

[https://developer.mozilla.org/es/docs/Web/HTTP/Basics\\_of\\_HTTP/MIME\\_types](https://developer.mozilla.org/es/docs/Web/HTTP/Basics_of_HTTP/MIME_types)

<https://sites.google.com/site/conceptoprogramacion/request-response>

<http://informatica.uv.es/iiguia/IST/Tema2.pdf>