**UNIDAD 1 – INTRODUCCIÓN A LAS APLICACIONES WEB**

**PROGRAMACIÓN WEB**

**1.1 EVOLUCIÓN DE LAS APLICACIONES WEB**

INTERNET – Se basa en la tecnología Cliente/Servidor.

La Web **es un sistema de estándares aceptados universalmente** para almacenamiento, recuperación, formateado y visualización de información o crear una página Web.

Combina **texto**, **hipermedia**, **sonidos** y **gráficos**, utilizando **interfaces gráficas** de usuario para una visualización fácil.

La Web esta constituida por millones de documentos enlazados entre sí, denominados páginas Web.

Una **página web** normalmente esta construida por texto, imágenes, audio y video.

Un **sitio web** esta organizado alrededor de una página inicial que sirve como página de entrada y punto de enlace a otras páginas del sitio.

Cada página web tiene una dirección conocida como **URL**(Uniform Resource Locator).

La web ser basa en un lenguaje estándar de hipertexto denominado **HTML (Hypertext Markup Language)** que da **formatos** a **documentos** e **incorpora enlaces dinámicos** a **otros documentos** almacenados en la misma computadora o en computadoras remotas.

(**Los navegadores web están programados de acuerdo con este estándar**)

**Aplicación web:** Es un programa que ayuda al usuario de un ordenador a procesar una tarea específica. **Funcionalidad – Resolver un problema.**

Con la introducción de comercio electrónico y las páginas dinámicas, los sitios ya han dejado atrás los folletos y han pasado a ser auténticas aplicaciones de software

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAGINA WEB | Informativa  **ITCULIACAN** | Publicidad | Vista 🡨 Bonita | Dar contexto a algo |
| APLICACIÓN WEB | **SITEC** | Funcionalidad | Una sola cosa | Resolver problema |

Utiliza como medio físico de **comunicación** la red **internet** y el protocolo **HTTP** (Hypertext transference Protocol). **HTTP** es el protocolo de **transferencia de hipertexto** que los navegadores utilizan para realizar peticiones a los servidores web y para recibir respuesta de ellos. (**Es el protocolo que se utiliza para ver páginas web**)

**Funcionamiento de la Web**

1.- El primer paso consiste en **traducir la parte nombre del servidor de la URL** en una **dirección IP** usando la base de datos distribuida de Internet conocida como **DNS**. Esta dirección IP es necesaria para **contactar con el servidor web** y poder **enviarle paquetes de datos**.

2.- El siguiente paso es **enviar una petición HTTP al servidor web** solicitando el recurso. En el caso de una página web típica, **primero** se solicita el **texto HTML** y luego es inmediatamente analizado por el navegador, el cual, después, hace **peticiones adicionales** para los **gráficos** y **otros ficheros** que formen parte de la página.

3.- Al recibir los ficheros solicitados desde el servidor web, el **navegador** representa (**renderiza**) la página tal y como se describe en el código HTML, el CSS y otros lenguajes web. Al final se incorporan las imágenes y otros recursos para producir la página que ve el usuario en su pantalla.

**WEB 1.0 – 1991, a veces se menciona 1989 o 1990**

Su difusión masiva comenzó en **1993** como **medio de comunicación universal.**

Es la forma más básica que existe, cuenta con **navegadores** que son los bastantes **rápidos** ya que es de **lectura** solamente.

Características:

* El usuario **no puede interactuar** con el contenido de la página (**Unidireccional - Estáticas**).
* Las páginas web estaban escritas a mano en código **HTML**. **Añadir** o **cambiar** algo en la página requería **mucho tiempo y trabajo**.
* **No había normas de diseño en las páginas web**, con lo que era fácil encontrarse con páginas con colores estrafalarios, marquesinas que saltaban o bailaban, etc. En resumen, paginas que mareaban y creaban desconcierto en el usuario.
* El **contenido** y el **diseñó** estaban **juntos** en el código HTML. Si se deseaba cambiar el color o el tipo de letra de un párrafo o cabecera común a todas las páginas, era necesario cambiarlo en cada una de ellas.
* **No se usaban las bases de datos** y, por tanto, no existian los usuarios ni el registros de usuarios, **todos eran visitantes**.

En la web 1.0 no existía Google; existía **Yahoo!**, **Altavista** y **otros** buscadores; **encontrar** **algo** con dichos buscadores era **muy complicado** y había gente que se ganaba la vida solo haciendo búsquedas. Uso de **correo electrónico**, **navegadores**, **foros**, chats, etc.

Formularios HTML enviados vía correo electrónico. Un usuario llenaba un formulario y después de hacer clic se enviaba a través de un cliente de correo electrónico, con el problema que en el código se podía observar los detalles del envío del correo electrónico.

**Una página web que se limita a mostrar información y que ni siquiera se actualiza, forma parte de la generación 1.0. En cambio, cuando las páginas ofrecen un nivel considerable de interacción y se actualizan con los aportes de los usuarios, se llaman la web 2.0.**

Lo que **diferencia** a la web 1.0 y la 2.0 no es el nivel tecnológico en los servidores, --aunque naturalmente se ha dado un considerable avance en el hardware--, es, sobre todo, la **finalidad de la red**, los **objetivos** y la **forma** en la que los usuarios **perciben y utilizan la información en línea**.

**WEB 2.0 – 2004 - web social**

**Es dinámica, interactiva, de lectura y escritura, desarrolla la inteligencia colectiva y favorece el trabajo colaborativo, etc.**

Comprende aquellos sitios web que facilitan el compartir información, el diseño centrado en el usuario y la colaboración en la World Wide Web. Permite a los usuarios interactuar y colaborar entre si como creadores de contenido generado por usuarios en la comunidad virtual. **NUEVA GENERACIÓN DE PÁGINAS Y CONTENIDO.**

Los sitios Web 2.0 utilizan **tecnologías** como **bases de datos**, **hojas de estilo** o **lenguajes de programación** y permiten a los usuarios **interactuar** entre **ellos** y **crear sus propios contenidos**.

Ejemplos de la Web 2.0 son las **comunidades web**, los **servicios web**, las **aplicaciones Web**, los servicios de red social, los servicios de alojamiento de videos, las wikis, blogs, mashups y folcsonomías.

Algunas empresas como Yahoo!, Ebay, Amazon generaron mucho dinero. A partir de estas empresas, que añadían webs más dinámicas incorporando sistema de gestión de contenido, nació la web 2.0Características:

* Interactiva y colaborativa - **bidireccional**
* El usuario ha dejado de ser un actor pasivo para convertise en un **actor activo** y **participativo** en el **uso** y **desarrollo** de **aplicaciones web**.
* Uso de **bases de datos** que permiten el registro de usuarios, la creación de perfiles, la inserción de comentarios y discusiones, etc.
* Aparecen las **redes sociales** que permiten que las personas esten en **permanente contacto** con amigos y familiares, así como conocer nuevas personas.
* **Los usuarios pueden almacenar y compartir documentos en la red de una forma muy sencilla.**
* Aparecen **aplicaciones web colaborativas** al alcance de los usuarios, que reemplazan a las aplicaciones de escritorio y a las que se accede mediante un navegador web.
* Cualquier usuario puede crear su propio espacio, publicar sus fotos, sus opiniones, videos, etc. **La mayor parte del contenido es creado por usuarios.**

Así podemos entender la Web 2.0 como “todas aquellas utilidades y servicios de internet que se sustentan en una base de datos, la cual puede ser modificada por los usuarios del servicio, ya sea en sus contenidos (Añadiendo, cambiando o borrando información asociando datos a la información existente), bien en la forma de presentarlos o en contenido y forma simultáneamente” (Xavier Ribes, 2007)

En la web 2.0 nos encontramos con los siguientes tipos de servicios:

* **Redes sociales:** destacan facebook y MySpace.
* **Foros:** donde los usuarios mantienes converdaciones sobre algun tema de interes común.
* **Compartir y organizar fotografías:** destacan las herramientas Flirk y Picasa.
* **Compartir video y musica:** Youtube, Vimeo, DailyMotion.
* **Ofimatica web:** Zoho o Google Docs.
* **Blog**: es un sitio web donde se recogen cronologicamente articulos u opiniones de uno o varios autores. Destaca wordpress
* **Microbloggin**: es un sitio web donde los usuarios pueden enviar y publicar mensajes breves, de 120 caracteres como máximo. Destaca twitter.

Para que todo lo indicado hasta aquí sea técnicamente posible estas aplicaciones utilizan un gestor de contenidos, (CMS), que permite la **creación y administración de contenidos en páginas web**. Consiste en una interfaz que **controla una o varias bases de datos** donde se aloja el contenido de la web. El sistema permite **manejar de manera independiente el contenido por una parte y el diseño por otra**. Así, es posible manejar el contenido y darle, cuando se quiere, un diseño distinto, sin tener que darle un formato nuevo al contenido, además de permitir a varios editores la publicación en el sitio, fácil y controlada.

**WEB 3.0 – 2010 (operativa) - web semántica – 2006 apareció el termino**

Expresión que se utiliza para **describir la evolución del uso y la interacción de las personas a través de diferentes formas**. Entre los que se incluyen la transformación de la red en una base de datos, un movimiento social hacia crear contenidos accesibles por múltiples aplicaciones non-browser, el empuje de las **tecnologías de inteligencia artificial**, la **web semántica**, la **web geoespacial** o la **web 3D.**

Se basa en la idea de añadir **metadatos semánticos y ontológicos** a la World Wide Web.

Esas **informaciones adicionales** —que **describen** el **contenido**, el **significado y** la **relación** de los **datos**— se deben **proporcionar de manera formal**, para que así sea posible **evaluarlas(rastreadas) automáticamente por máquinas de procesamiento**.

El **objetivo** es **mejorar** Internet **ampliando la interoperabilidad entre los sistemas informáticos usando "agentes inteligentes"**.

**Agentes inteligentes** son **programas** en las computadoras que **buscan información sin operadores humanos.** Con la web 3.0 se busca que los usuarios puedan **conectarse desde cualquier lugar, cualquier dispositivo y a cualquier momento.**

**Información contenida en Internet esté mucho mejor organizada.**

Mientras la web 2.0 está gestionada por el propio usuario humano, la web 3.0 está **gestionada en la nube** y **ejecutada desde cualquier dispositivo** con un alto grado de complejidad y personalización; constituye un nuevo tipo de web en la que se añade contenido semántico a los documentos que la forman y ello conlleva que la ejecución de esta sea realizada **por máquinas que, basándose en nuestros perfiles en la red, descubren información relevante para nosotros.**

Está muy **asociada** al concepto de **personalización**. Ofrece un flujo de información y de **contenidos adaptados a nuestros gustos y preferencias**.

La web 3.0 apunta a que todos podamos disfrutar de la información y de las herramientas de Internet sin importar el aparato a través del cual nos conectemos, ya que busca una flexibilidad y una versatilidad que superen las barreras del formato y la estructura.

Entre sus innovaciones destacan:

* Bases de datos
* Inteligencia artificial
* Web semántica y SOA
* Evolución al 3D

**WEB 4.0 – 2020 ¿?**

Ofrecer un **comportamiento** **más inteligente y predictivo**, de modo que podamos, con **sólo realizar una afirmación o una llamada, poner en marcha un conjunto de acciones que tendrán como resultando aquello que pedimos, deseamos o decimos**.

Las aplicaciones ya no estarán en nuestras PC’s, estarán en la Internet y por ende en todos lados. Pasaremos de una red “tonta” a una red **“inteligente**” donde el **objetivo primordial será el de unir las inteligencias donde tanto las personas como las cosas se comuniquen entre sí para generar la toma de decisiones.**

Para el 2020 o quizás antes se espera que haya “agentes” en la Web que conozcan, aprendan y razonen como lo hacemos las personas.

La Web Ubicua es un concepto que está aún en desarrollo, pero me llama curiosamente la atención como se van complementando algunas tecnologías que nos permiten imaginar o soñar lo que podemos esperar en un futuro no muy lejano. Imagínese recibir información en su celular en la calle por la que camina y que su propio equipo le haga una reservación en el restaurante de la esquina con solo saber sus gustos.

¿Cómo llegamos a la Web 4?0.? Gracias a la propia **evolución de la tecnología**. Empresas como Google, Microsoft o Facebook, entre otras, están desarrollando nuevos sistemas que gracias al **Deep Learning** y **Machine Learning** serán capaces de procesar información de forma similar a como lo haría el cerebro humano. Tenemos los avances que los **asistentes de voz** están logrando. Siri, Google Now o Cortana entienden cada vez de forma más precisa y correcta lo que les decimos o solicitamos. Es más, ya hay smartphones que siempre están “escuchando” para activarse en el preciso momento que oigan “Oye, Siri” u “Hola, Google Now”, etc. para contestar de inmediato.

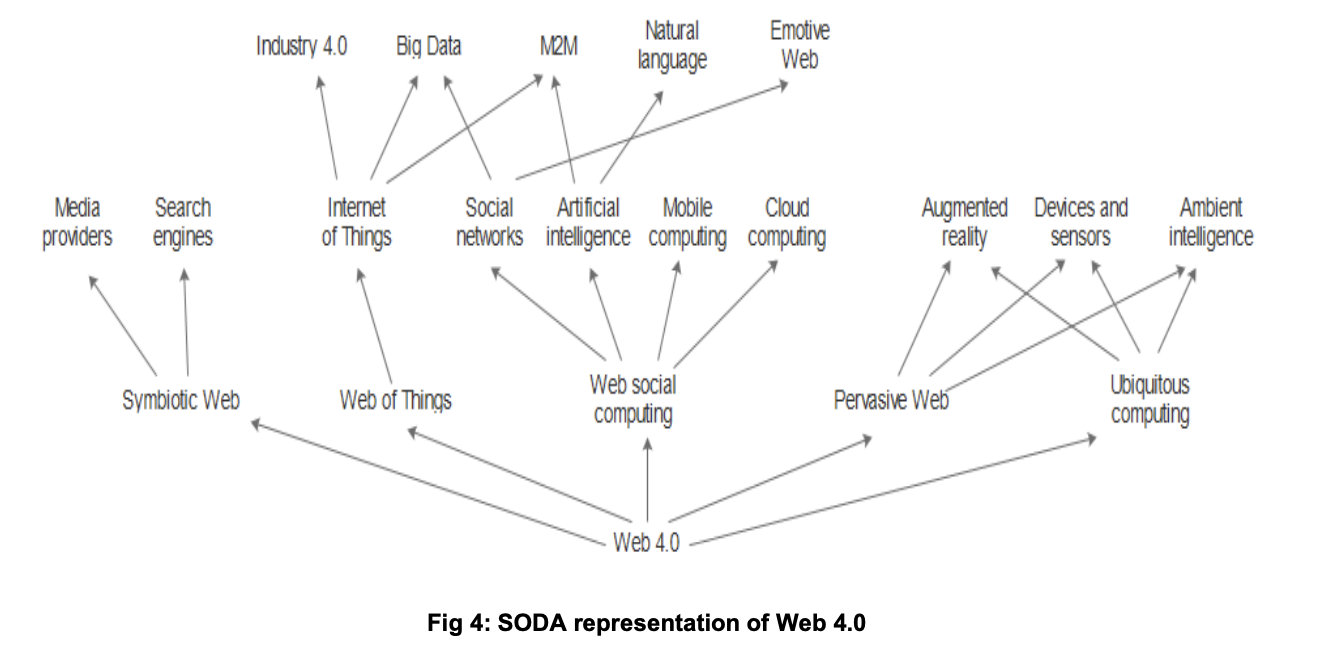
Otro indicador de hacia dónde vamos son los **bots**. Son **programas de interacción** que traerá la web 4.0. Por ahora dependen de la introducción de texto, pero, con la evolución en tecnologías de “hablar al texto” (Speech to text), seremos capaces de conversar con ellos de igual modo que lo haríamos con un amigo en una cafetería. En la web 3.0 los buscadores son elementos esenciales en el día a día. Cuando se entra en Google, Bing, Go, etc. e introduces palabras clave, aparecen muchos resultados que hay que revisar hasta encontrar lo que se desea.

Pues bien, la web 4.0 mejora esa experiencia mediante el uso de nuevas tecnologías que **permiten un nivel de interacción más completo y personalizado**. Es decir, se puede decir verbalmente al dispositivo digital --que puede ser un smartphone, computadora, etc.--: “Compra un boleto de avión con tales características”, “Pide un taxi para la hora y tal lugar” y automáticamente ejecutará la acción sin más intervención propia. Así, pasamos de una web que nos **proporciona información** a la web que nos **proporciona soluciones**. Como se puede observar la web 4.0 ofrece soluciones a partir de toda la información que le damos y que existe en la web. Para lograrlo, se fundamenta en cuatro pilares:

* **La comprensión del lenguaje natural hablado, escrito y tecnologías (de voz a texto y viceversa).**
* **Nuevos sistemas de comunicación máquina a máquina (M2M)**
* **Uso de la información del contexto. Por ejemplo, ubicación que aporta el GPS, ritmo cardíaco que registra el smartwatch, etc.**
* **Nuevo modelo de interacción con el usuario.**

La Web 4.0 permite **adelantarse a situaciones cotidianas**, como ver que llegas tarde al trabajo y enviar un mensaje avisando de lo que sucede, tomar una ruta alternativa a la habitual porque está colapsada, etc. Si tu smartwatch detecta que el ritmo cardíaco es elevado y tienes registrado que sufres problemas de corazón, avisará a la asistencia sanitaria y aparecerá una ambulancia.

La web 4.0 permite la computación cognitiva. A través de potentes ordenadores se almacenan en la nube y procesan los datos, peticiones, etc. permitiendo, no sólo que cualquier dispositivo con conexión a internet nos ayude a aprovecharnos de ellos, sino también **sirviendo cada uno de nosotros** como **suministradores de datos a las máquinas.**



**GLOSARIO**

**Hipertexto:** Sistema de organización y presentación de datos que se basa en la vinculación de fragmentos textuales o gráficos a otros fragmentos, lo cual permite al usuario acceder a la información no necesariamente de forma secuencial sino desde cualquiera de los distintos ítems relacionados.

Un **ejemplo** notorio de **hipertexto** son los artículos de internet, que en el cuerpo de texto presentan diversos links o enlaces de **hipertexto** en las palabras o en los temas que estén relacionados con el tema principal o en desarrollo, lo que permite al lector tener una lectura más activa

**Hipermedia:** Sirve para designar al conjunto de métodos o procedimientos para escribir, diseñar o componer contenidos que integren soportes tales como: texto, imagen, video, audio, mapas y otros soportes de información emergentes, de tal modo que el resultado obtenido, además, tenga la posibilidad de interactuar con los usuarios.

**Ontologías**: Son recopilaciones de enunciados que engloban las relaciones entre los conceptos y las reglas de la lógica que son necesarias para entenderlos y razonar con ellos.

Es una definición formal de tipos, propiedades, y relaciones entre entidades que realmente o fundamentalmente existen para un dominio de discurso en particular.

“La Web semántica requiere necesariamente de ontologías para representar conocimiento, ya son el mecanismo que permite exponer el modelo conceptual que existe detrás de cada página o recurso Web”

**Agentes**: Se definen como un software capaz de trabajar continuamente y de manera autónoma.

**Metadatos:** Para varios campos de la informática, como la recuperación de información o la web semántica, los metadatos en etiquetas son un enfoque importante para construir un puente sobre el intervalo semántico, ya que cualquier recurso tiene, cuando está almacenado juntamente con otros, la necesidad de ser descrito para facilitar las búsquedas que pudieran tratar de encontrarlo a partir de sus características distintivas. Esto es cierto para cualquier tipo de recurso, tanto para un vídeo como para un libro en una biblioteca o para un hueso en el armario de un paleontólogo.

**Deep learning:** el sistema aprende de forma autónoma a tomar las decisiones. De forma práctica esto se traduce en una función en la que a partir de una entrada se obtiene una salida, con lo que el problema radica en construir un modelo de esta función matemática de forma automática.

**Machine Learning:** Es una derivación de inteligencia artificial que crea sistemas que aprenden de manera automatizada, es decir, identificar patrones complejos en millones de datos, predecir comportamientos todo mediante un algoritmo y que además de todo son capaces de mejorarse de manera independiente con el tiempo.

[**https://web.archive.org/**](https://web.archive.org/)

**1.2 ARQUITECTURA DE LAS APLICACIONES WEB**

**URL (Uniform Resource Locator – Localizador uniforme de recursos)**

La URL es una dirección unica que poseen todos los elementos de internet, existe una URL por cada página, imagen, video y documento en internet, lo que nos permitirá ubicarlos facilmente gracias a la información que contiene.

Siempre que visitemos una página web, será a través de su URL. Cada que pulsemos un enlace que contenga la página o sitio, también nos dirigirá a otra URL.

El navegador utiliza la información que contiene la URL para hallar el contenido que estamos buscando.

Combina el **protocolo** o esquema que será necesario **utilizar para recuperar los datos**, el **nombre de la computadora que provee el recurso**, el **directorio** o **subdirectorio** donde se encuentra y el **documento** que seá accedido.

****

**FTP**

**TELNET**

**SSH**

**HTTPS**

**Básicamente, el http es un protocolo de comunicación entre el navegador y el servidor web.**

Un protocolo es un método establecido para intercambiar o transferir datos alrededor de una red informática.

Dos puntos: En todas las URL, luego del identificador de protocolo encontramos dos puntos (:), los cuales básicamente **actúan como un mecanismo de separación entre el protocolo y el resto de la dirección web**.

Barras inclinadas: En todos los casos una URL incluye dos barras inclinadas (//), las cuales se ubican separando el identificador de protocolo y el nombre de la fuente, a continuación de los dos puntos que se escriben después del http. **Su función principal es la de ser una iniciación para el contacto**, es decir que **notifican la dirección de internet para lograr el contacto con otro servidor.**

Index.html: En muchos casos podemos encontrar el llamado index.html, que es el nombre de un archivo solicitado, el cual por lo general suele ser una página por defecto dentro de un determinado sitio web. Este index.html puede llamarse también default.html

**Extensión:** En la mayoría de los casos las direcciones URL incluyen la extensión **.com**, aunque también podemos encontrar otros muy utilizados como **.net .info .gob .org** y demás, siendo el más habitual el .com, y este se refiere al dominio utilizado.

Nombre de dominio: Cuando hablamos del nombre de dominio completo nos referimos a una **dirección web completa**, es decir **www.nombredelafuente.com**, que en muchos casos puede llegar a prescindir del www. Y aparecer de la siguiente manera: tecnología-informatica.com, ese es el nombre de dominio, que se refiere puntualmente a la última parte de la dirección web.

**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.

**Parametros**: 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.

CADA UNA DE LAS DIRECCIONES URL EQUIVALE A UNA DIRECCIÓN IP, la cual básicamente es una serie de números que le indican a nuestra PC donde hallar en internet el documento que estamos buscando.

Las direcciones URL fueron creadas respondiendo a la necesidad del usuario de poder recordar estas ubicaciones en forma sencilla.

El DNS (Domain Name Server o Sistema de nombres de dominios), es el sistema encargado de traducir la inmensa cantidad de peticiones en direcciones IP, es decir el proceso inverso, pero que pueden manejar con extrema exactitud las computadoras que componen Internet.

http utiliza el puerto 80.

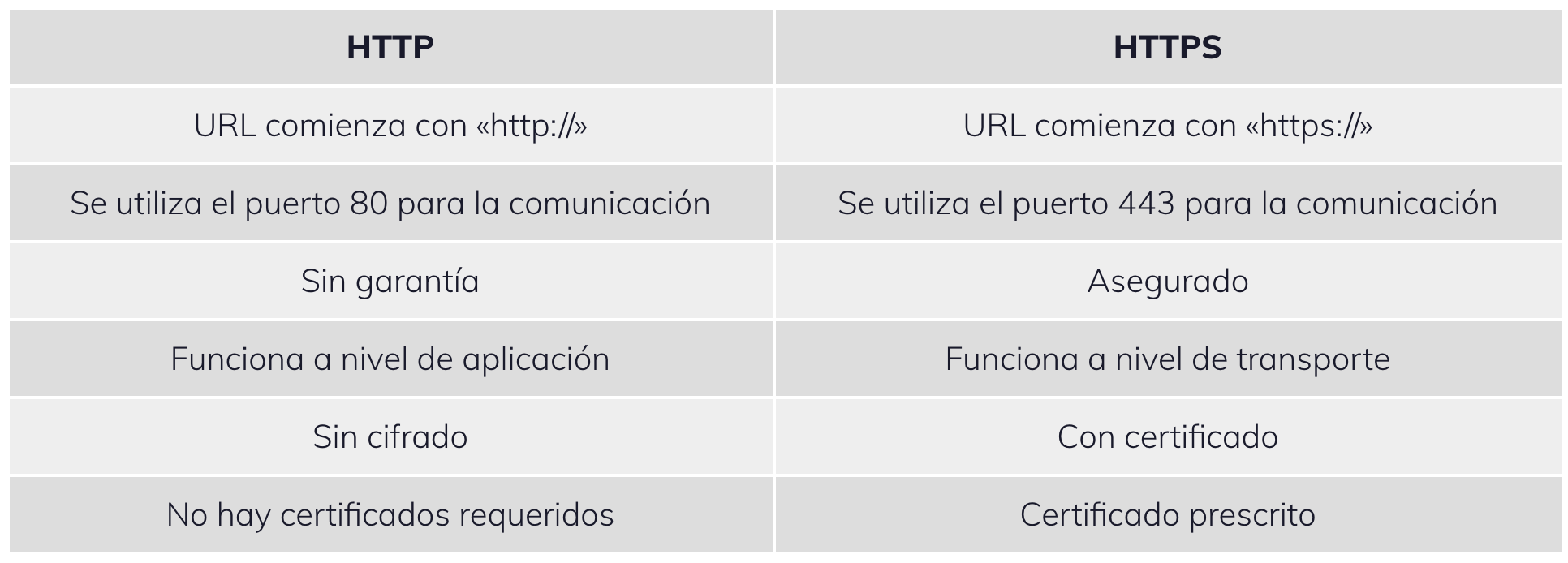
* http: (Hypertext Transfer Protocol) o “Protocolo de Transferencia de Hipertexto”, el tipo de esquema más utilizado en la red.
* https: (Hypertext Transfer Protocol sobre Secure Sockets Layer). Este tipo de protocolo es utilizado mayoritariamente para proporcionar un nivel más alto de seguridad en las comunicaciones en Internet.
* ftp: (File Transfer Protocol): Protocolo de Transferencia de Archivos. Usado para subir y bajar archivos desde la red.
* mailto: Utilizado para proporcionar un marco desde el cual los usuarios tengan la posibilidad de enviar un correo electrónico en forma sencilla desde la web.
* file: Este protocolo es usado para localizado para localizar y acceder a recursos en una PC local o en la red.
* news: Utilizado para el acceso a grupo de noticias.
* telnet: Este protocolo es utilizado para acceder a otras PCs utilizando la red.

HTTPS es la forma más segura de navegar por Internet, ya que cualquier dato que introduzcamos en el navegador viajará cifrado y por lo tanto no podrá ser analizado para su uso, tanto con fines comerciales o delincuenciales.

En HTTPS, se basa en una combinación de dos protocolos diferentes, HTTPS y SSL/TLS.

Cómo podemos ver, la utilización del protocolo **HTTPS es de vital importancia para cualquier actividad que** involucre el **uso de datos personales** como los utilizados en **entidades bancarias, tiendas en línea, correos electrónicos** y en cualquier otro sitio en el cual debamos introducir **passwords**, **números de tarjeta de crédito** u **otra** **información personal.**

Para que el sistema pueda funcionar, se debe implementar el esquema de “**Certificado**”, teniendo que estar este **debidamente firmado por una autoridad**. En el caso de los navegadores web, este método es transparente para el usuario, ya que los certificados necesarios como para poder explorar Internet sin problemas son obtenidos con el navegador.

****

**Arquitecturas cliente/servidor**

Las aplicaciones web son un tipo especial de aplicaciones cliente/servidor.

Cliente/Servidor es una arquitectura de red en la que cada ordenador o proceso en la red es cliente o servidor. **Normalmente los servidores son ordenadores potentes dedicados a gestionar unidades de disco, impresoras, trafico de red, datos, o incluso aplicaciones, mientras que los clientes son máquinas menos potentes y usan los recursos que ofrecen los servidores.**

Esta arquitectura implica la existencia de una relación entre procesos que solicitan servicios(clientes) y procesos que responden a estos servicios(servidores). Estos dos tipos de procesos pueden ejecutarse en el mismo procesador o en distintos.

La arquitectura cliente/servidor permite la creación de aplicaciones distribuidas. La principal ventaja de esta arquitectura es que facilita la separación de las funciones según su servicio, permitiendo situar cada función en la plataforma más adecuada para su ejecución. Además, también presenta las siguientes ventajas:

* Las redes de ordenadores permiten que múltiples procesadores puedan ejecutar partes distribuidas de una misma aplicación, logrando concurrencia de procesos.
* Existe la posibilidad de migrar aplicaciones de un procesador a otro con modicaciones mínimas en los programas.
* Se obtiene una escalabilidad de la aplicación. Permite la ampliación horizontal o vertical de las aplicaciones. La escalabilidad horizontal se refiere a la capacidad de añadir o suprimir estaciones de trabajo que hagan uso de la aplicación (clientes), sin que afecte sustancialmente al rendimiento general (Figura 3.1). La escalabilidad vertical se refiere a la capacidad de migrar hacia servidores de mayor capacidad o velocidad, o de un tipo distinto de arquitectura sin que afecte a los clientes (Figura 3.2).
* Posibilita el acceso a los datos independientemente de donde se encuentre el usuario.

Una aplicación web es proporcionada por un servicio web y utiliza por usuarios que se conectan desde cualquier punto vía clientes web (browsers o navegadores). La arquitectura de un sitio web tiene tres componentes principales:

El servidor Web distribuye páginas de información formateada a los clientes que las solicitan. Los requerimientos son hechos a través de una conexión de red, y para ello se usa el protocolo HTTP. Una vez que se solicita esta petición mediante el protocolo HTTP y la recibe el servidor Web, éste localiza la página Web en su sistema de archivos y la envía de vuelta al navegador que la solicitó.

Existen diversas variantes de la arquitectura básica según cómo se implementen las diferentes funcionalidades de la parte servidor.

La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información **en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos**. Se denomina **cliente** al proceso que **inicia el diálogo o solicita los recursos** y **servidor** al proceso que **responde a las solicitudes**. En este modelo las aplicaciones se dividen de forma que el **servidor** contiene la **parte que debe ser compartida por varios usuarios,** y en el **cliente permanece solo lo particular de cada usuario**.

**Descripción de un sistema cliente/servidor:**

Un sistema cliente/servidor suele presentar las siguientes características:

1. Una combinación de la parte cliente (también llamada front-end) que interactúa con el usuario (hace de interfaz entre el usuario y el resto de la aplicación) y la parte servidor (o back-end) que interactúa con los recursos compartidos (bases de datos, impresoras, módems).

2. La parte cliente y servidor tienen diferentes necesidades de recursos a la hora de ejecutarse: velocidad de procesador, memoria, velocidad y capacidad de los discos duros, dispositivos de entrada/salida, etc.

3. El entorno suele ser heterogéneo y multivendedor 3 . El hardware y sistema operativo del cliente y el servidor suelen diferir. El cliente y el servidor se suelen comunicar a través de una Application Program Interface (API) y Remote Procedure Call (RPC) conocidas (por ejemplo, Open DataBase Connectivity (ODBC) para acceder a bases de datos).

4. Normalmente la parte cliente se implementa haciendo uso de una interfaz gráca de usuario, que permite la introducción de datos a través de teclado, ratón, lápiz óptico, etc.

|  |  |
| --- | --- |
| **Cliente** | **Servidor** |
| * Inicia solicitudes o peticiones. * Espera y recibe las respuestas del servidor. * Puede conectarse a varios servidores a la vez. * Interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario. | * Al iniciarse esperan a que lleguen las solicitudes de los clientes * Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente. |

**Separación de funciones**

La arquitectura cliente/servidor nos permite la separación de funciones en tres niveles.

**Lógica de presentación:** Se encarga de la entrada y salida de la aplicación con el usuario. Sus principales tareas son: obtener información del usuario, enviar la información del usuario a la lógica de negocio para su procesamiento, recibir los resultados del procesamiento de la lógica de negocio y presentar estos resultados al usuario

**Lógica de negocio** (o aplicación): Se encarga de gestionar los datos a nivel de procesamiento. Actúa de **puente entre el usuario y los datos.** Sus principales tareas son: recibir la entrada del nivel de presentación, interactuar con la lógica de datos para ejecutar las reglas de negocio (business rules) que tiene que cumplir la aplicación (facturación, cálculo de nóminas, control de inventario, etc.) y enviar el resultado del procesamiento al nivel de presentación

**Lógica de datos.** Se encarga de gestionar los datos a nivel de almacenamiento. Sus principales tareas son: almacenar los datos, recuperar los datos, mantener los datos y asegurar la integridad de los datos.

Los componentes de una aplicación Web son:

1. Lógica de negocio.

* Parte más importante de la aplicación.
* Define los procesos que involucran a la aplicación.
* Conjunto de operaciones requeridas para proveer el servicio.

2. Administración de los datos.

* Manipulación de BD y archivos.

3. Interfaz

* Los usuarios acceden a través de navegadores, móviles, PDAs, etc.
* Funcionalidad accesible a través del navegador.
* Limitada y dirigida por la aplicación.

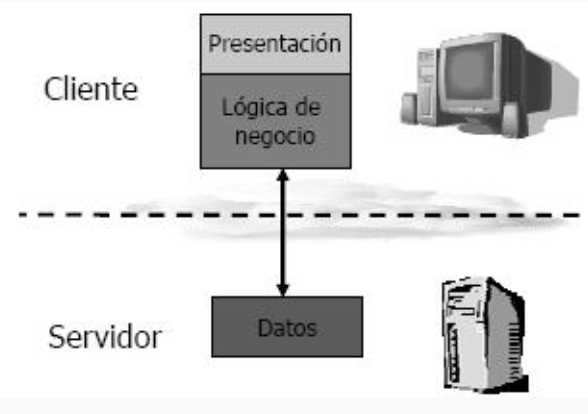
**MODELO 2 CAPAS**

**Las capas son:**

**Cliente (fat client):** La lógica de negocio está inmersa dentro de la aplicación que realiza el interfaz de usuario, en el lado del cliente.

**Servidor:** Administra los datos.

Las limitaciones de este modelo son.

* Es difícilmente escalable
* Número de conexiones reducida
* Alta carga de la red.
* La flexibilidad es restringida
* La funcionalidad es limitada.

**MODELO DE 3 CAPAS**

Las capas de este modelo son:

1. Capa de presentación (parte en el cliente y parte en el servidor)

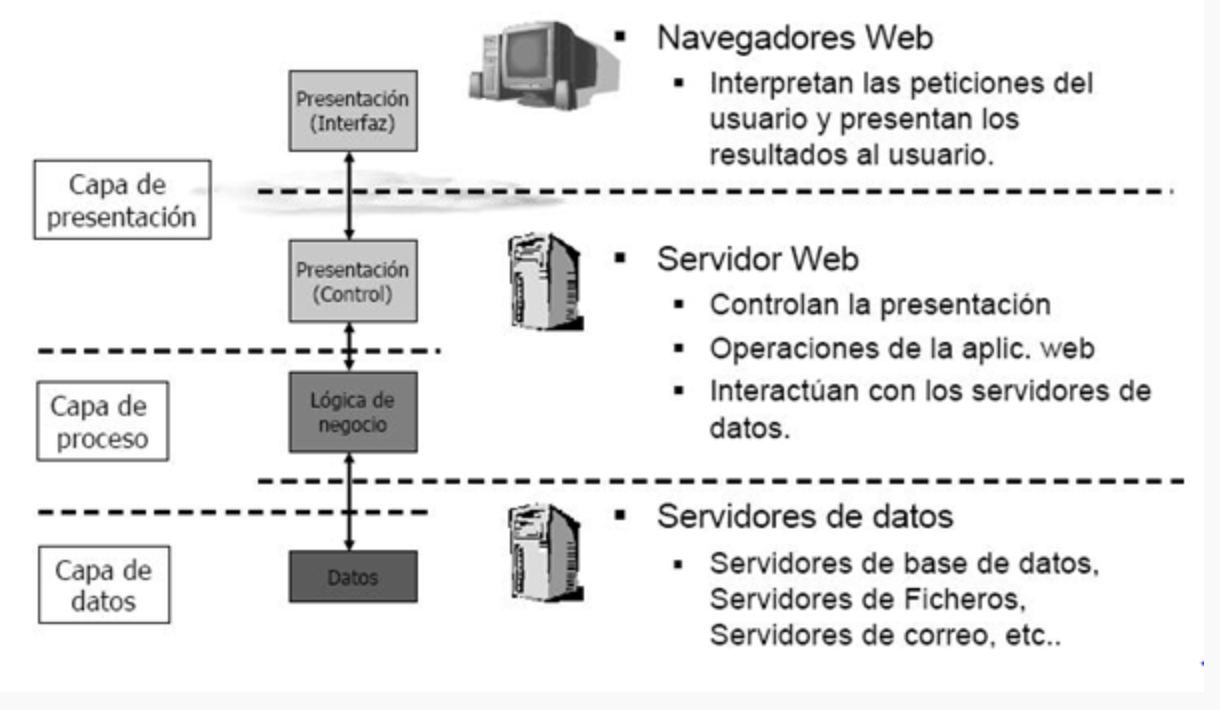
* Recoge la información del usuario y la envía al servidor (cliente)
* Manda información a la capa de proceso(NEGOCIO) para su procesado
* Recibe los resultados de la capa de proceso
* Generan la presentación
* Visualizan la presentación al usuario (cliente)

2. Capa de proceso (servidor web)

* Recibe la entrada de datos de la capa de presentación
* Interactúa con la capa de datos para realizar operaciones
* Manda los resultados procesados a la capa de presentación

3. Capa de datos (servidor de datos)

* Almacena los datos
* Recupera datos
* Mantiene los datos
* Asegura la integridad de los datos



**El objetivo de aumentar el número de niveles en una aplicación distribuida es lograr una mayor independencia entre un nivel y otro, lo que facilita la portabilidad en entornos heterogéneos y la escalabilidad en caso de incorporación de nuevos clientes.**

**1.4 PLANIFICACIÓN DE APLICACIONES WEB**

****

PLAN DEL PROYECTO

* Resumen del proyecto
* Estrategia
* Audiencia
* Requerimientos o funcionalidades
* Alcance
* Métricas
* Mantenimiento y soporte posterior
* Equipos y roles
* Presupuesto
* Tiempo
* Requerimientos tecnologicos

PRUEBAS DE CALIDAD

* Prueba de los contenidos y revisión de los mismos: comprobación de enlaces, revisión de textos e imágenes, etc.
* Pruebas funcionales: pruebas sobre la aplicación y sobre elementos interactivos
* Pruebas de integración
* Pruebas de navegación
* Planificación final del proceso de calidad
* Pruebas en los distintos navegadores

Planificar una aplicación web implica pensar y definir muchas cuestiones de importancia:

* **Planificar el futuro**: Planear el futuro de algo que aún no existe, que pasa si en un futuro tuviéramos problemas de escalabilidad, y sí tuviéramos que migrar las bases de datos de MySql a Oracle, de una buena planificación depende la dificultad en un futuro de realizar cambios en la plataforma que sostiene nuestra aplicación.
* **Documentación:** Documenta todo lo que puedas de la aplicación aquellas partes de esta que hayan requerido un mayor tiempo de desarrollo.
* **No empieces por los pequeños detalle**s: Plantear el desarrollo de pequeños servicios cómo pueden ser los RSS o una galería de imágenes, pero es fundamental comenzar por lo que vendrá a ser la base de nuestra aplicación, de este modo podremos **ir completando nuestra aplicación en un orden lógico**.
* **Diseño de la interfaz:** Es importante conseguir un buen diseño para nuestra interfaz, es importante conseguir un interfaz que se atractivo y útil para la gran mayoría de los usuarios, lo más importante es llegar a conectar con el usuario y la mejor manera es ofrecerle la mayor cantidad de información posible para que se llegue a sentir parte del proyecto pero todo esto muchas veces no lo llegamos a conseguir por una mala elección en la combinación de los colores que se integran en nuestra interfaz.
* **Evitar construir librerías o frameworks innecesarios**: Es mejor que cada desarrollador utilice sus propias librerías, pero la realidad es que en muchas ocasiones esto supone una gran pérdida de tiempo, es mejor buscar una librería o framework que se parezca lo más posible a lo deseado que desarrollarlo por ti mismo
* **Un plan de proyecto**: Cada desarrollador que trabaje en el proyecto sabe qué es lo que debe de hacer y cuándo tiene que hacerlo, teniendo todo esto redactado en un documento te ayudará a identificar problemas.
* **Elegir el lenguaje de programación adecuado**: depende de para que quieras utilizarlo, por eso asegúrate de escoger el lenguaje adecuado.
* **Evita las distracciones:** Optimiza tu tiempo, cuando estés trabajando que nada te distraiga, ni llamadas telefónicas, ni clientes de mensajerías instantáneas, proponte un buen número de horas al día de trabajo aislado dedicado al proyecto.
* Un buen control de calidad
* **Final del desarrollo:** Con frecuencia se deberá actualizar y que se necesita para que este funcione.

**HTML**

**Lenguaje de marcado que permite crear estructuras.**

**Maquetar: dar formato**

* HTML es un lenguaje de marcado de hipertexto, la mayoria de los sitios web estan escritos en HTML, es utilizado para crear páginas y hacerlas funcionales.
* Hipertexto significa que el documento contiene links que permiten al lector conectar a otros lugares del documento o de un documento a otro. La ultima versión es HTML5.
* Un lenguaje de marcado es la forma en la que las computadoras se comunicas unas a otras para controlar como el texto es procesado y presentado, para esto HTML utiliza etiquetas y atributos.

**¡SE DEBE DE CONTAR CON UN EDITOR DE TEXTO!**

**Etiquetas y atributos**

* La **etiquetas** y **atributos** son la **base** de **HTML**. Trabajan juntos pero realizan diferentes funciones.
* Las **etiquetas** son usadas para **marcar** el **inicio** de un **elemento** HTML, generalmente se incluyen entre <> (corchetes angulares)
* La mayoría de las etiquetas deben **abrirse** **<h1>** y **cerrarse** **</h1>** para que funcionen.
* Los atributos contienen información adicional, toman de una etiqueta de apartura y se coloca información adicional dentro. Un ejemplo es:

<img src="mydog.jpg" alt="A photo of my dog.">

En este caso, la fuente de la imagen (src) y el texto alternativo (alt) son atributos de la etiqueta <img>.

Reglas a recordar

* La gran mayoría de las etiquetas deben abrirse y cerrarse con la información del elemento, como un titulo o texto, entre las etiquetas.
* Cuando se utilizan varias etiquetas, las etiquetas deben cerrarse en el orden en que se abrieron. Ejemplo:

<strong><em>This is really important!</em></strong>

**Construcción básica de una página HTML**

Estas etiquetas deben colocarse una debajo de la otra en la parte superior de cada página HTML que cree.

**<! DOCTYPE html>:** esta etiqueta especifica el idioma que escribirá en la página. En este caso, el lenguaje es HTML 5.

**<html>:** Esta etiqueta indica que de aquí en adelante vamos a escribir en código HTML.

**<head>:** Aquí es donde van todos los metadatos de la página, material destinado principalmente a motores de búsqueda y otros programas informáticos.

**<body>:** Aquí es donde va el contenido de la página.

****

Más etiquetas

Dentro de una etiqueta **<head>**, hay una etiqueta que siempre se incluye: **<title>,** pero hay otras que son igualmente importantes.

<title>: Aquí es donde insertamos el nombre de la página como aparecerá en la parte superior de la ventana o pestaña del navegador.

<meta>: Aquí es donde se almacena la información sobre el cocumento: **codificación** de caracteres, **nombre**(contexto de la página), **descripción**.

**Agregar contenido**

La sección de <**body>** es donde **agregamos** el **contenido** que está diseñado para ser visto por los usuarios.

Incluye **texto**, **imágenes**, **tablas**, **formularios**, etc.

**<p>Párrafo</p> <- debe mostrarse así por semántica.**

**<pan></span> <- con css se le agrega estilo**

Encabezados

* <h1> - Solo se utiliza uno por página porque tíene valor semántico.
* <h2>
* <h3>
* <h4>
* <h5>
* <h6>
* <hr/> - Línea separadora

**<h1>** y **<h2>** se utilizan en los **titulos** más **importantes**, mientras que las **demás** **etiquetas** se utilizan para **subtítulos** y **texto menos importantes**.

Los bots de los motores de búsqueda utilizan este orden para descifrar que información es más importante en una página.

Otros elementos

(Ya no se suelen usar) - Obsoleto

**<b>** - negrita – información importante

**<strong>** - fuerte - resaltar texto clave – énfasis para el navegador

**<i>** - italica

**<em>** - texto enfatizado – pies de imagen

**<mark>** - texto marcado

**<small>** - texto pequeño

**<strike>** - texto tachado

**<u>** - subrayado

**<sub>** - subindice

**<sup>** - superindice

**<blockquote**> <- agregar cita

**<cite>** <- autor de la cita

Links

El internet se compone de muchos enlaces, casi todo en lo que hace clic mientras se navega por la web es un enlace que lo lleva a otra página dentro del sitio web que esta visitando o a uno externo.

Los **enlaces** se incluyen en un atributo abierto por la etiqueta **<a>**. Este elemento es el primero que conocemos que utiliza un atributo y, por lo tanto, se ve diferente a las etiquetas mencionadas anteriormente.

La etiqueta de **anchor (ancla)** se escribe:

<a href=" http://culiacan.tecnm.mx" target="\_blank"> El texto del enlace</a>

href: Apunta a la página que se abrirá una vez que se haga clic en el enlace.

El atributo **target específica donde abrir** el documento del link.

**\_blank -** Abre el documento vinculado en una nueva ventana o pestaña.

**\_self -** Abre el documento vinculado en el mismo marco en el que se hizo clic (esto es predeterminado).

**\_parent** -Abre el documento vinculado en el marco principal.

**\_top -** Abre el documento vinculado en todo el cuerpo de la ventana.

Vinculo a documentos de mi misma carpeta: <ahref="\*folder(s)\*/page2.html">Page2</a>

Imágenes

La etiqueta **<img>** tiene todo lo que necesita para mostrar imágenes en su sitio. Al igual que el elemento ancla <a>, <img> también **contiene** **un** **atributo**.

El atributo presenta información para su computadora con respecto a la **fuente**, **altura**, **ancho** y **texto** **alternativo** de la imagen.

Los **tipos de archivo** que se utilizan generalmente para archivos de imágenes en línea son: .**jpg**, .**png** y (cada vez menos) .**gif**.

El **texto** **alternativo** es **importante** para **garantizar** que **su sitio esté clasificado correctamente en los sitios de búsqueda** y también para los visitantes con discapacidad visual de su sitio.

<img src="yourimage.jpg" alt="Describe the image" height="X" width="X">

Ruta absoluta: link externo, tiene URL predefinida

Ruta relativa: dentro de mi sitio web, tengo otra carpeta, se accede mediante enlace relativo.

Comentarios

<!-- Esto es un comentario -->

Lista

**<ol>** - Lista ordenada de contenido.

Dentro de la etiqueta <ol> se necesita listar cada itam dento de una etiqueta **<li></li>**

**<li> -** Elementos de lista

<ol>

<li>primer item</li>

<li>segundo item</li>

<li>tercer item</li>

</ol>

Lista desordenada

**<ul>** - Lista desordenada – lista de **viñetas**

Tablas

Otra manera de mantener tus sitios web **ordenados** es mediante la utilización de tablas.

**<table>** - etiqueta de abertura.

**<tr>-** renglores - filas

**<td>** - celdas – columnas

**<th>** - encabezado

table *border*>

<tr>

<td>Row 1 - Column 1</td>

<td>Row 1 - Colunm 2 </td>

<td>Row 1 - Column 3 </td>

</tr>

<tr>

<td>Row 2 - Column 1</td>

<td>Row 2 - Column 2</td>

<td>Row 2 - Column 3</td>

</tr>

**Border** – pone borde en la equiqueta table

<thead> - cabezado de tabla- arriba de la tabla

<tbody> - Cuerpo de la tabla – contenido

<tfood> - Pie de table- debajo de la tabla

<colgroup> - grupo de columnas

<th> - Table encabezado

Las celdas de la tabla que actúan como encabezados de columna o de fila deben usar el elemento <th> (encabezado de tabla).

**El marcado (código HTML) de una tabla siempre se basa en filas, nunca en columnas**.

<table *border*>

<tr>

<th>Encabezado 1</th>

<th>Encabezado 2</th>

<th>Encabezado 3</th>

<th rowspan=3>Encabezado 4</th>

<th rowspan=4>Encabezado 5</th>

</tr>

<tr>

<td>Columna 1</td>

<td>Columna 2</td>

<td>Columna 3</td>

</tr>

<tr>

<td>Columna 1</td>

<td>Columna 2</td>

<td>Columna 3</td>

<!-- <td rowspan=2 >Columna 3</td> -->

</tr>

<tr>

<td>Columna 1</td>

<td colspan=3>Columna 2</td>

</tr>

</table>

<! DOCTYPE> es obligatorio

El atributo xmlns en <html> es obligatorio

<html>, <head>, <title> y <body> son obligatorios

Los elementos siempre deben estar correctamente anidados

Los elementos siempre deben estar cerrados

Los elementos siempre deben estar en minúsculas

Los nombres de los atributos siempre deben estar en minúsculas

Los valores de atributo siempre deben estar entrecomillados

La minimización de atributos está prohibida

<link rel="shortcut icon" href="imagenes/icon.ico" type="image/x-icon">