

Capítulo 8

Web de la aplicación TRAMOS

8.1.- Introducción.

En este capítulo 8 se diseña y estudia la página Web que se ha creado para la aplicación TRAMOS. En primer lugar se hace una introducción genérica del mundo de Internet, comentando en qué consiste la red, los protocolos de comunicación, las aplicaciones y lenguajes web, el lenguaje de programación PHP y las bases de datos. Para concluir el capítulo se pasa al diseño realizado para la aplicación TRAMOS, comentando las posibilidades que presenta dicha página web.

8.1.1.- La red de Internet.

8.1.1.1.- Introducción.

El gran crecimiento experimentado por Internet ha permitido interconectar millones de ordenadores y ha hecho posible aumentar de forma extraordinaria el desarrollo de aplicaciones informáticas para la gestión de Sistemas de Información fuera del ámbito interno de las empresas.

Hasta la década de los 80 las redes informáticas de ámbito empresarial se restringían a redes propiedad de los distintos constructores hardware como IBM con su red SNA (System Network Architecture) o Digital y su red DECNet, que interconectaban un gran número de ordenadores homogéneos. Sin embargo existía una gran dificultad para conectar en una misma red ordenadores de distinto tipo o bien interconectar diferentes redes.

La necesidad de intercambiar información motivó una serie de avances en el campo de las comunicaciones que hicieron posible interconectar diferentes tipos de ordenadores. Así, la creación del protocolo TCP/IP, que permite asignar a cualquier ordenador de la red un nombre único (dirección IP), propició las bases para la comunicación de máquinas heterogéneas, creando lo que sería la red Internet: un gran conjunto de redes conectadas entre sí.

A finales de los 80 el CERN (Centro Europeo de investigaciones Nuclear) se planteó la necesidad de un método que permitiera intercambiar la gran cantidad de información generada por la comunidad científica dentro del campo de la energía nuclear. Así nació el proyecto de la World Wide Web.

La invención por Tim Berners Lee en el CERN de un lenguaje de códigos para la representación de la información: HTML (HyperText Mark-Up Language), y de un protocolo a nivel de aplicación: HTTP (HiperText Transport Protocol) que permitiera la transmisión de los documentos codificados en HTML a través de la red Internet; ha revolucionado el mundo de las TI/SI (Tecnologías y Sistemas de información), haciendo posible el estado actual de la red Internet.

Este hecho ha permitido intercambiar información codificada en lenguaje HTML. Dicha información está contenida en páginas ubicadas en determinados puntos de la red Internet (Web Sites) de forma que cualquier ordenador conectado a la red podrá tener acceso a ella, conociendo su dirección, la URI (Uniform Resource Identifier), y visualizarla mediante un agente usuario (navegador).

En el marco operativo que establece la WWW (HTTP, HTML, y TCP/IP), las aplicaciones Web están restringidas a estas tecnologías de forma que la información utilizada en las mismas está elaborada mediante el lenguaje HTML. El lenguaje usuario, que constituye la otra pieza en estas clave estas aplicaciones, es el encargado de interpretar la información codificada en HTML para que el usuario pueda visualizarla. Para mejorar la funcionalidad de las aplicaciones Web se han ido incorporando nuevas tecnologías cuyo resultado desde el punto de vista del desarrollador ha sido la inclusión en los documentos HTML de bloques de códigos escritos en otros lenguajes (Java, Java Script, CGI, ASP, PHP,...) que el agente usuario también tiene que interpretar. Así podemos hablar de un gran número de aplicaciones que pueden ser ejecutadas en la Red, desde pequeñas aplicaciones estáticas que solamente utilizan el lenguaje HTML hasta aplicaciones interactivas que incorporan para su construcción una serie de lenguajes de programación de propósito general, lenguajes de scripting y tecnologías diversas que permiten dotar a las aplicaciones de una mayor funcionalidad y una visualización más atractiva.

El hecho de que el desarrollo de aplicaciones basadas en la utilización de tecnologías Web resulte bastante económico por su facilidad de implantación y por los bajos costes de transmisión de información, ya que la llamada al centro proveedor de información suele realizarse mediante una llamada local, ha propiciado un aumento espectacular de estas aplicaciones y con ellas el desarrollo de negocios dentro de Internet estableciendo las bases del comercio electrónico.

8.1.1.2.- Orígenes y evolución de Internet.

Internet nace básicamente como consecuencia de la evolución tecnológica de las técnicas de comunicación. Hasta entonces, las redes construidas para la transmisión de voz y utilizadas posteriormente para transmitir datos, funcionaban utilizando la conmutación de circuitos para establecer una conexión, es decir, constituía un circuito eléctrico que unía un emisor y un receptor por el que pasaban los diferentes bits de una forma sincronizada. La evolución del sistema de comunicaciones que utilizaría la red de Internet, se inició con la investigación de un nuevo sistema de transmisión: la conmutación de paquetes. La necesidad de comunicación de los diferentes estamentos sociales: universidades, gobiernos y empresas, influyó en el desarrollo de una red global.

El origen de Internet se remonta a 1957. En ese año la Unión Soviética lanza el primer satélite artificial, el Sputnik. En respuesta a este hecho Estados Unidos crea el ARPA (Organismo de Proyectos de Investigación Avanzada) dentro del Ministerio de Defensa a fin de establecer su liderazgo en el área de la ciencia y la tecnología aplicada a las fuerza armadas. A continuación se describen una serie de hechos cronológicos destacables en el largo proceso de creación de la red Internet:

Años 60:

- L.Kleinrock del MIT (Massachusetts Institute of Technology) publica en julio de 1961 los primeros trabajos de la teoría de *comunicación de paquetes* como primer paso para el diálogo entre ordenadores.
- J.C. Licklider del MIT publica en agosto de 1962 el primer trabajo relacionado con la creación de una *red global* (Galactic Network - Red Galáctica) realizando programas de investigación en ordenadores de la red DARPA.
- En 1968 se inician los estudio para el diseño de la arquitectura de la red ARPANET utilizando como elemento clave la *conmutación de paquetes* para la transmisión de información.
- En 1969 se estableció el primer *nodo* de ARPANET en la UCLA's Network Measurements Center (Universidad de California).
- A finales del 1969 se conectaron cuatro ordenadores *host* a la red ARPANET Y los trabajos se centraron en el avance hacia el desarrollo de un protocolo de comunicación entre diferentes *hosts*.

Años 70:

- En 1970 el grupo de trabajo NWG (Network Working Group-Grupo de Trabajo de la Red) finalizó el desarrollo de un protocolo para la comunicación de *host* a *host* llamado NCP (Network Control Protocol - Protocolo de Control de Red).La evolución del protocolo siguió hasta 1972, año en que se inició el desarrollo de diferentes aplicaciones para la red, entre las cuales destacó el *correo electrónico*. Posteriormente se desarrollaron diferentes utilidades para facilitar su uso.
- En 1972, R. Kahn empezó el desarrollo de una nueva versión del protocolo NCP para incorporar las necesidades en un entorno de *redes abiertas*, a las que se podrían conectar diferentes redes sin necesidad de cambiar su estructura interna. Este programa recibió el nombre de *Internetting*. Para su conexión se utilizarían bloques cerrados que transmitirían los paquetes de información a través de ellos, sin almacenar información sobre los mismos. También se tendrían que evitar las pérdidas de paquetes en funciones de redireccionamiento, controlar los extremos de la comunicación y reensamblar los paquetes a partir de su fragmentación. Así se inició el desarrollo del protocolo TCP/IP (Transport Control Protocol/ Internet Protocol- Protocolo de Control de Transporte/ Protocolo de Internet) que constituyó el *lenguaje* de programación entre ordenadores de la red de Internet.
- En 1973, V. Cerf de la Universidad de Stanford se unió a R. Kahn en el desarrollo del protocolo TCP. Este protocolo debería suministrar todos los servicios de transporte y reenvío en Internet. Debido a los problemas que surgieron en su desarrollo, se vieron en la necesidad de dividir TCP en dos protocolos: un protocolo TCP dedicado al control de flujo y la recuperación de los paquetes perdidos, y un protocolo más sencillo que se encargara de asignar la dirección a los paquetes y reenviarlos. En ese año ARPANET se conecta a Gran Bretaña y Noruega.
- A mediados de la década de los 70 proliferó la construcción de diferentes redes, cada una de ellas con fines de investigación específicos: HEPNet- energía nuclear, CSNET- informática, SPAN-física, etc.

- A finales de la década de los 70 con la aparición de los ordenadores personales se tuvo que adaptar el protocolo TCP ya que su tamaño y complejidad impedía que este fuera utilizado en microordenadores, puesto que sólo se habían desarrollado para *mainframes*. La adaptación corrió a cargo del MIT, de forma que la red pasó a estar formada por *hosts* y ordenadores personales.

Años 80:

- En 1980, el ejército americano adoptó el protocolo TCP para su red de ordenadores. Tres años más tardes se realizó la transición del protocolo NCP de la red ARPANET a TCP/IP. Esta red estaba siendo usada por muchas organizaciones dedicadas a la investigación, mientras que otras se centraban en proyectos militares de defensa de los EE.UU., lo cual provocó la división de la red en dos: MILNET para la conexión de las redes con fines militares y ARPANET para las necesidades de investigación.
- En 1986 se inició el desarrollo de la NSFNet, la red de la NSF (National Science Foundation - Fundación Nacional para las Ciencias) que utilizó el protocolo TCP/IP y en su financiación participaron diferentes agencias federales. Para fomentar su desarrollo se creó la FNC (Federal Networking Council- Consejo Federal de Redes) que también participaba en otras organizaciones internacionales para coordinar el apoyo de la comunidad científica mundial a Internet. La red conectaba cinco centros con superordenadores a 56 Kb/s.
- En 1988 se inició un proceso de privatización de la NFS que finalizó en 1995. el *backbone* (segmento principal que unía las diferentes redes) había crecido hasta 21 nodos y conectaba a más de 50.000 redes en todo el mundo. Su velocidad se incrementó hasta 1,544 Mb/s.
- En 1989 T. Berners-Lee presentó un proyecto interno en el CERN (Centre European de la Recherche Nucleaire - Centro Europeo de Investigación Nuclear) para desarrollar un sistema cuyo objetivo era compartir la información de la comunidad científica que trabajaba en el campo de la energía nuclear. Este proyecto se llamó "*World Wide Web*" (*WWW*). Utilizó la red de transporte de Internet para la transmisión de información mediante la creación de un *lenguaje de códigos de formato* (HTML) y un *protocolo de aplicación* (HTTP) para su transmisión. Este hecho revolucionó completamente la propia red de Internet.

Años 90:

- En 1990 se disolvió ARPANET y TCP pasó a sustituir a la mayor parte de los protocolos de las redes de comunicaciones y el protocolo IP pasaría a ser el protocolo de direccionamiento de Internet. La red NSFNet reemplaza a ARPANET y se pone en marcha la aplicación *Archie* para acceder a las bases de datos de la red. España (RedIris) junto a otros países como Argentina, Chile, Brasil, Corea del Sur, India, Bélgica, Irlanda, Grecia y Suiza, se conecta a la red NSFNet.
- En 1991 B. Kahle inventa la WAIS (Wide Area Information Servers-Servidores de Información del Área Extensa) que permite el acceso a documentos específicos en una base de datos. Tim Berners- Lee finaliza el proyecto WWW y P. Lindner y M. McCaibución de paquetes y bytes en la NSFNet.hill de la Universidad de Minnesota ponen a disposición de la comunidad de la red, la

aplicación *Gopher*, que permite la conexión a bases de datos. En este mismo año la red central (backbone) de NSFNET aumenta su velocidad de transmisión a 44,736 Mb/s.

- En 1992 se funda la organización Internet Society (ISOC) compuesta por profesores, usuarios, corporaciones y agencias gubernamentales, para gestionar el desarrollo y control de la red. En este año se alcanza la cifra de 1.000.000 de *hosts* conectadas a la red.
- La NSF (National Science Foundation) crea la organización InterNIC en 1993 para la gestión de los *dominios* de la red.
- En 1994 el flujo de la NSFNet supera los 10.000 millones de bytes/mes. La WWW se transforma en el segundo servicio más popular de la red (después de ftp-data) basándose en el porcentaje de flujo y distribución de paquetes y bytes en la NSFNet.
- En octubre del mismo año se forma TERENA (Trans-European Research and Education Network Association - Asociación de Redes Transeuropeas de Educación e Investigación) con el objetivo de promover y participar en el desarrollo de infraestructura en información y telecomunicaciones internacionales para el beneficio de la investigación y la educación.
- NSF establece en 1995 el vBNS (very high-speed Backbone Network Services-Servicio de base de red de alta velocidad) que conecta centros de alto procesamiento: NCAR, NCSA, SDSC, CTC, PSC.
- También en 1995, WWW se convierte en el servicio de mayor flujo en Internet.
- La guerra del navegador WWW, principalmente entre Netscape y Microsoft, da origen en 1996 a una nueva era en el desarrollo de programas en la que se lanzan nuevas versiones trimestralmente con la ayuda de los usuarios de Internet ávidos por nuevos productos.
- En 1997 se establece el ARIN (American Registry for Internet Numbers - Registro Estadounidense de Registros de Internet) a fin de manejar la administración y registro de los números IP para las áreas geográficas anteriormente controladas por InterNIC.
- El Departamento de Comercio Estadounidense suscribe en 1998 un contrato con ICANN (Empresa de Internet para Números Asignados) a fin de establecer un proceso para el paso de DNS de manos del Estado a manos industriales.

Actualmente el tamaño estimado de la Red sobrepasa los mil millones de páginas susceptibles de ser catalogadas.

8.1.1.3.- Los componentes de Internet.

La red de Internet está formada por un conjunto de múltiples redes interconectadas por un eje central vNBS (very high-speed Backbone Network Services-backbone), que conecta una serie de centros computadores a una velocidad de 155Mb/s.

Las redes forman diferentes estructuras y son de distintos tipos: WAN (Wide Area Network- Red de Área Extensa), MAN (Metropolitan Area Network- Red de Área Metropolitana) y LAN (Local Area Network- Red de Área Local). Estas redes se conectan en un punto del eje central (backbone) llamado NAP (Network Access Point- Punto de Acceso a la Red).

Las redes pueden estar conectadas entre sí por diferentes elementos:

- Hubs: Conectan grupos de ordenadores para formar una LAN de diferentes topologías.
- Repetidores: Sirven para amplificar la señal cuando esta se debilita debido a la distancia entre los puntos de conexión y el medio de transmisión.
- Puentes (Bridges): Se utilizan para conectar LANs entre sí.
- Pasarelas (Gateways): Son similares a los puentes y se utilizan para conectar diferentes tipos de redes.
- Encaminadores (Routers): Realizan la gestión del tráfico en Internet para asegurar que la información llegue a su destino.

Para el transporte de la información existen diversos medios que transmiten a diferentes velocidades según la tecnología utilizada. Pueden utilizarse:

- Líneas telefónicas (56 Kb/s).
- Líneas telefónicas ISDN (Integrated Services Digital Network o RDSI).
- Líneas T1 (1.544 Mb/s).
- Líneas T3 (44.746 Mb/s).
- Enlaces por satélite.

El acceso a Internet se puede realizar mediante un ordenador personal a través de un módem o una línea ISDN que conecta a un proveedor de Internet, o bien desde cualquier terminal de una red que a su vez está conectada a la red de Internet mediante algunos de los elementos anteriores de conexión hasta alcanzar el punto de destino.

8.1.1.4.- La World Wide Web (WWW).

Internet, que se originó en el ámbito militar durante la guerra fría, se había desarrollado más allá de los propósitos originales como resultado de su uso por parte de la comunidad científica internacional, que necesitaba nuevos sistemas de distribución de la información. En 1989 la red mundial de datos ya existía en potencia pero requería de unas vías de acceso sencillas y homogéneas. Este era uno de los objetivos que Tim Berners-Lee se planteó en 1989 cuando presentó a sus superiores del CERN la propuesta original para el proyecto *World Wide Web*. Otro era la posibilidad de gestionar conexiones no lineales.

World Wide-Web (abreviado 'Web'; escrito también 'WWW' o incluso 'W3') significa algo así como 'red o telaraña global'. En principio se pensó como un medio para la distribución de la información entre equipos de investigadores geográficamente dispersos; concretamente se dirigía a la comunidad de físicos de altas energías vinculados al CERN.

Se pretendía que los recursos disponibles en formato electrónico, que residen en ordenadores distintos conectados a la red, fuesen accesibles para cada investigador desde su terminal, de un modo transparente y exento de dificultades, sin necesidad de aprender a utilizar varios programas distintos. Además, debería posibilitarse el salto entre elementos de información conexos. Los recursos existentes deberían integrarse en una red hipertextual distribuida gestionada por ordenadores.

El éxito del WWW, el crecimiento de la telaraña, ha sido espectacular. Durante 1993 se pasó de 50 a 500 nodos. En 1994 se contabilizan ya miles de servidores en el WWW que distribuyen todo tipo de información (de ellos, trece en España; el primero fue el del Departamento de Educación de la Universitat Jaume I, en septiembre de 1993).

La arquitectura del World Wide Web.

El diseño del World Wide Web sigue el modelo cliente-servidor: un paradigma de división del trabajo informático en el que las tareas se reparten entre un número de clientes que efectúan peticiones de servicios de acuerdo con un protocolo, y un número de servidores que las atiende. En la Web, nuestras estaciones de trabajo son clientes que demandan hipertextos a los servidores. Para poner en marcha un sistema como éste ha sido necesario:

a) Diseñar e implementar un nuevo protocolo que permitiera realizar saltos hipertextuales, esto es, de un nodo o lexía de origen a uno de destino, que podría ser un texto o parte de un texto, una imagen, un sonido, una animación, fragmento de vídeo, etc. Es decir, cualquier tipo de información en formato electrónico. Este protocolo se denomina HTTP (Hyper Text Transfer Protocol) y es el "lenguaje" que "hablan" los servidores del WWW.

b) Inventar un lenguaje para representar hipertextos que incluyera información sobre la estructura y el formato de representación y, especialmente, indicar origen y destino de saltos hipertextuales. Este lenguaje es el HTML o (HyperText Markup Language).

c) Idear una forma de codificar las instrucciones para los saltos hipertextuales de un objeto a otro de Internet. Dada la variedad de protocolos, y por tanto, formas de almacenamiento y recuperación de la información, en uso en Internet, esta información es vital para que los clientes puedan acceder a dicha información.

d) Desarrollar aplicaciones cliente para todo tipo de plataforma y resolver el problema de cómo acceder a información que está almacenada y es accesible a través de protocolos diversos (FTP, NNTP, Gopher, HTTP, X.500, WAIS, etc.) y representar información multiformato (texto, gráficos, sonidos, fragmentos de vídeo, etc.). A este fin se han desarrollado diversos clientes, entre los que destaca la familia Mosaic, del NCSA (National Center for Supercomputer Applications) de la Universidad de Chicago, y su sucesor Netscape Navigator, de Netscape Communications Corporation.

Veamos con cierto detenimiento los rasgos más sobresalientes de estos elementos clave del sistema.

HTTP: HyperText Transfer Protocol.

El HTTP (HyperText Transfer Protocol) es el protocolo de alto nivel del World-Wide Web que rige el intercambio de mensajes entre clientes y servidores de la Web. Un protocolo es una descripción formal de los formatos de los mensajes y las reglas que deben seguir dos ordenadores para intercambiar dichos mensajes. Los protocolos pueden describir detalles de bajo nivel de los interfaces de máquina a máquina (por ejemplo, el orden en el cual deben enviarse bits y bytes a través de un cable) o intercambios de alto nivel entre programas (por ejemplo, la forma en que dos programas transfieren un fichero a través de Internet).

El HTTP es un protocolo genérico orientado a objetos que no mantiene la conexión entre transacciones (Berners-Lee, 1993). Ha sido especialmente diseñado para atender las exigencias de un sistema hipermedia distribuido como es el World-Wide Web. Sus características principales son:

- **Ligereza:** reduce la comunicación entre clientes y servidores a intercambios discretos, de modo que no sobrecarga la red y permite saltos hipertextuales rápidos.
- **Generalidad:** puede utilizarse para transferir cualquier tipo de datos, según el estándar para la transmisión de objetos multimedia MIME (*Multipurpose Internet Mail Extensions*). Esto incluye también los que desarrollen en el futuro ya que el cliente y el servidor pueden negociar en cualquier momento el modo de representación de los datos: el cliente notifica al servidor una lista de formatos que entiende, y en adelante el servidor sólo remitirá al cliente datos que este sea capaz de manejar. El cliente debe aceptar al menos dos formatos: text/plain (texto normal) y text/html (hipertexto codificado en HTML).
- **Extensibilidad:** contempla distintos tipos de transacción entre clientes y servidores ("métodos", en la jerga HTTP), y la futura implementación de otros nuevos. Esto abre posibilidades más allá de la simple recuperación de objetos de la red: búsquedas, anotaciones, etc.

El esquema básico de cualquier transacción HTTP entre un cliente y un servidor es el siguiente (Berners-Lee, 1993):

- Conexión: El cliente establece una conexión con el servidor a través del puerto 80 (puerto estándar), u otro especificado.
- Petición: El cliente envía una petición al servidor.
- Respuesta: El servidor envía al cliente la respuesta (esto es, el objeto demandado o un código de error).
- Cierre: Ambas partes cierran la conexión.

La eficiencia del HTTP posibilita la transmisión de objetos multimedia y la realización de saltos hipertextuales con una rapidez razonable.

HTML: HyperText Markup Language.

HTML (HyperText Markup Language) es el lenguaje en el que se escriben los hipertextos del World-Wide Web. Cumple la norma SGML (Standard Generalized Markup Language) que es la norma ISO 8879-1986 para la definición de lenguajes que describen estructuras de documentos. Un lenguaje SGML especifica la estructura lógica del documento (las partes que lo componen, por ejemplo), pero no su aspecto en la pantalla o en el papel (márgenes o tipos de letra, por ejemplo). HTML permite añadir a un documento de texto:

- La especificación de estructuras del texto. Por ejemplo, títulos, encabezamientos, límites de los párrafos, listas de elementos.
- Estilos: texto enfatizado, citas, etc.
- Objetos multimedia: imágenes o sonido.
- Conexiones hipertextuales a otros objetos de la red.

Todo este "valor añadido" al texto se codifica como etiquetas ("*tags*", en la jerga) que se insertan en el propio texto. Las etiquetas del HTML se delimitan por medio de los signos < y >. Los links se abren y cierran con las etiquetas <A> y . El objeto de la red a donde nos lleva el link se codifica en la etiqueta de apertura por medio de una notación que se ha convertido de hecho en un estándar de Internet: los llamados URL.

URL: Uniform Resource Locator.

Los URL (Uniform Resource Locator) son una notación estándar para la especificación de recursos presentes en Internet. Constituyen la piedra angular de la Web, ya que hacen posible que un link de HTML se refiera a cualquier objeto de la red.

Un URL representa de un modo compacto la localización y el método de acceso de cualquier recurso de la red (Berners-Lee, Masinter y McCahill, 1994). No sólo hay más de dos millones de ordenadores conectados a los varios miles de redes que forman Internet, sino que existen múltiples protocolos o formas diferentes de acceder a la información (ftp, gopher, http, etc.). Los URL aportan esos dos datos esenciales: dónde se encuentra un recurso y cómo se puede acceder a él.

La sintaxis de los URL es la siguiente:

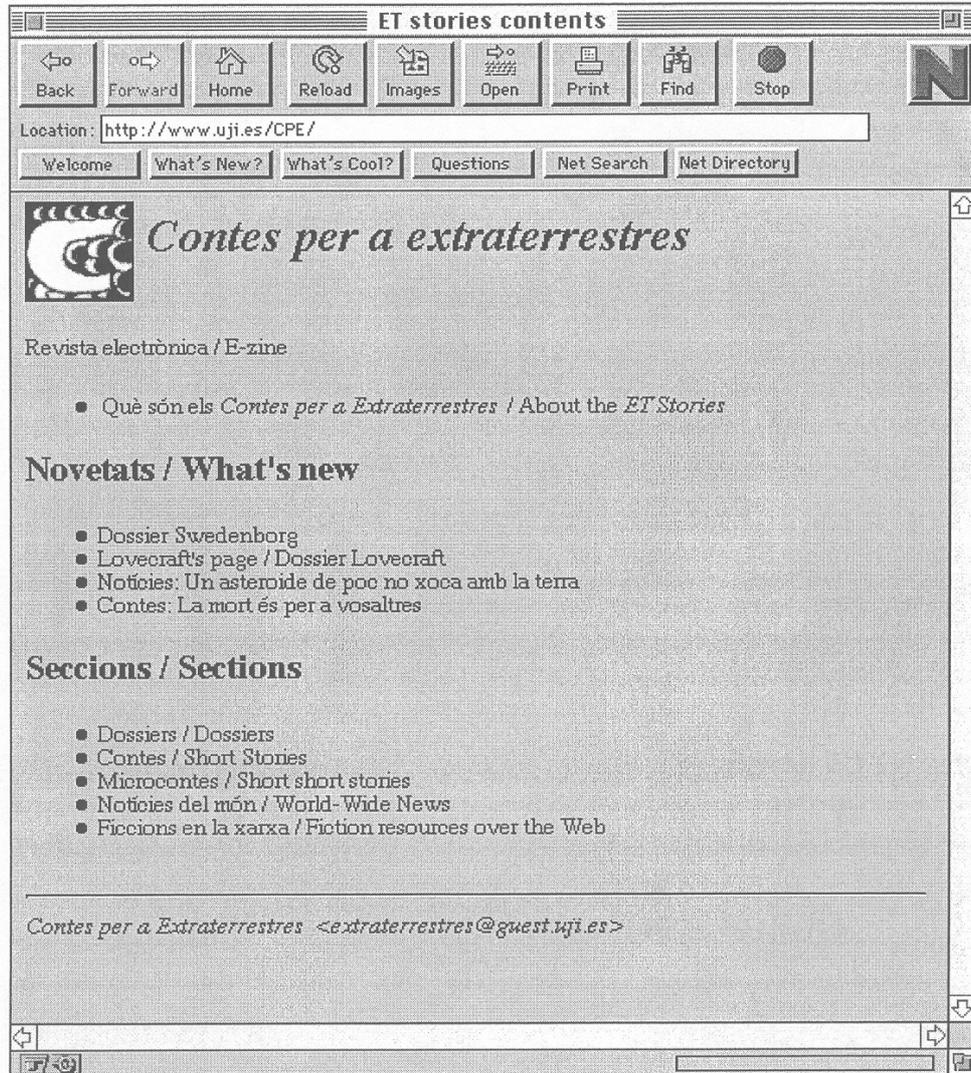
URL: <esquema>: <parte-especifica-del-esquema>

La utilidad y la necesidad de una notación que, como ésta, introduzca algo de orden en el caos de la red es obvia. Los URL se idearon para un proyecto concreto y limitado, el del WWW, pero ha cundido el ejemplo. Ahora mismo se está produciendo un amplio debate en el seno de Internet, concretado en un grupo de trabajo de la IETF (Internet Engineering Task Force) para el desarrollo de sistemas universales de designación y caracterización de objetos persistentes de la red, inspirados en los URL pero que irían más allá: debería ser posible, por ejemplo, asignar un URN (Uniform Resource Name) invariable para un objeto, aunque cambiara su path e incluso su método de acceso. Un sistema distribuido (similar al DNS o Domain Name System) resolvería un URN en uno o varios URL aplicando criterios de optimización de recursos (como proximidad al solicitante).

El interface de usuario del World Wide Web.

Dado que los nodos que forman el Web atienden peticiones en protocolos distintos, los programas cliente de la Web (también llamados "Web browsers") deben ser lo más parecido a un cliente universal capaz de presentar al usuario cualquier recurso de la red, dado su URL. Actualmente existe un número de "web browsers" para distintos sistemas y plataformas que satisfacen aquel requisito en mayor o menor medida. El más popular ha sido quizás el NCSA Mosaic, del National Center for Supercomputing Applications de la Universidad de Illinois, con versiones para X Window, Macintosh y MS Windows, sucedido recientemente por Netscape Navigator, de Netscape Communications Corporation.

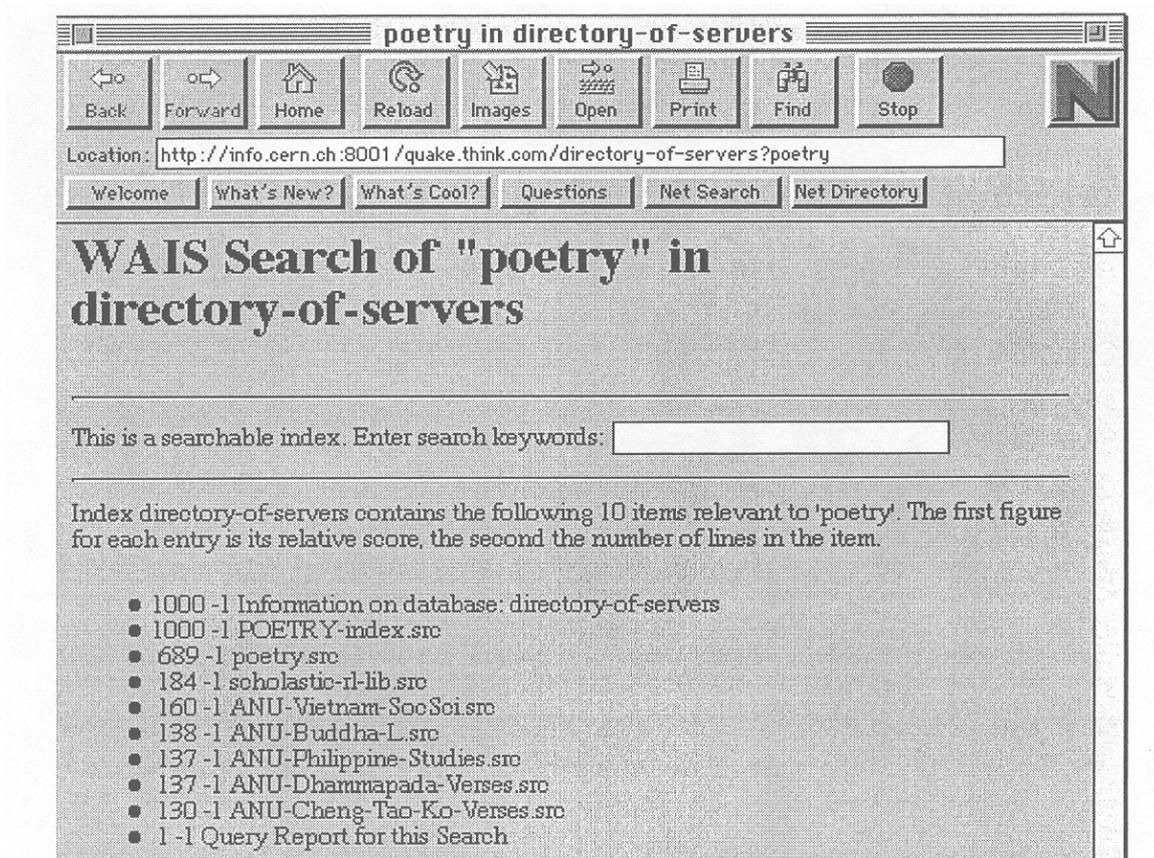
Una página típica del Web, vista con Netscape, tiene el aspecto que se aprecia en la figura siguiente:



El cliente universal.

Un cliente de la Web debería ser capaz de gestionar cualquier URL. Netscape Navigator nos permite acceder a recursos de los tipos siguientes: ftp, gopher, news y, por supuesto, http. Cuando se accede con Netscape a un recurso que no es un hipertexto, el software lo presenta igualmente en formato hipertextual. Por ejemplo, un directorio gopher aparece como una lista de su contenido en la que cada entrada es un link. Un clic con el ratón sobre un elemento que representa un documento de texto hará que Netscape nos presente el contenido del documento. Un grupo de noticias de Usenet se presenta como una lista de las cabeceras de los mensajes; cada cabecera es un link que nos lleva al contenido del mensaje.

Los mensajes se presentan como se ve en la figura siguiente:



Pasear por el hiperespacio.

Un clic sobre un link nos lleva al nodo al que el link apunta. Netscape lleva un registro de nuestros pasos, de modo que en cualquier momento podemos movernos hacia atrás y hacia adelante por el sendero que habíamos seguido gracias a los botones "back" y "forward" de la parte superior de la ventana. El botón "home" nos permite volver al punto de partida original, la llamada "home page". "View History", en el menú "Go" nos permite saltar directamente a cualquier punto del sendero.

Una característica útil para desplazarnos a puntos arbitrarios de la red es la posibilidad de acceder directamente a una página dando su URL. Otra posibilidad aún más importante es la de mantener "bookmarks", listas personalizadas de páginas de nuestro interés, a las que podemos volver en cualquier momento.

Aplicaciones auxiliares.

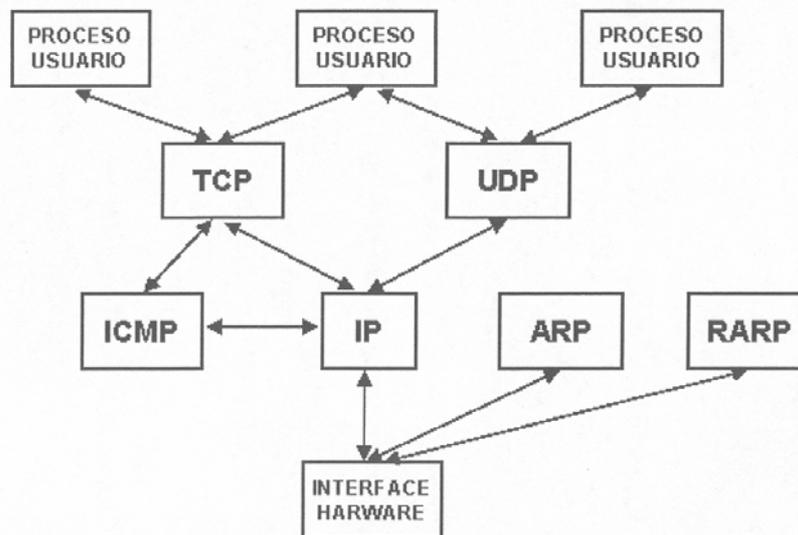
La información que se encuentra en el Web es accesible mediante protocolos distintos, como habíamos visto, pero los formatos de codificación también son diversos. Un cliente ideal de la Web debería poder gestionar todos esos formatos actuales y futuros. Netscape soluciona este problema mediante aplicaciones auxiliares. Cuando el tipo MIME de un objeto no es uno de los que puede gestionar por sí mismo, consulta una tabla (que el usuario puede configurar) que le dice qué aplicación se encargará de él. Por ejemplo, una imagen en formato TIFF (tipo MIME image/tiff) puede visualizarse en un Macintosh mediante la aplicación JPEG View (u otra determinada por el usuario).

Protocolo de transporte en la red TCP/IP.

TCP/IP (Transport Control Protocol/Internet Protocol) es un protocolo ARPA de comunicación que proporciona transmisión fiable de paquetes de datos sobre redes de ordenadores.

Su funcionamiento en líneas generales consiste en la transferencia de datos mediante el desensamblaje y ensamblaje de bloques de datos o paquetes. Cada paquete comienza con una cabecera que contiene información de control (como dirección de destino) seguida de los datos. Cuando se envía un archivo con TCP/IP su contenido se divide en varios de estos paquetes que son enviados de forma individual con la misma dirección de destino, de forma que cuando lleguen todos se ordenan, utilizando la información de control, y se reconstruye el archivo original. Su funcionamiento se dice que es fiable porque en destino se comprueban los paquetes que se van recibiendo y si se detecta algún error, bien que falte alguno o que estén mal, se vuelven a pedir, asegurándose de que lo que se recibe es lo que se quería enviar. Si por cualquier circunstancia no se puede conseguir una transmisión fiable se corta la conexión. De esta forma, una vez que se ha recibido cualquier cosa, se tiene la seguridad de que está bien.

Un esquema orientativo de una arquitectura TCP/IP puede ser el siguiente:



TCP: Transmisión Control Protocol.

UDP: User Datagrama Protocol.

ICMP: Internet Control Message Protocolo IP: Internet Protocol.

ARP: Adresse Resolution Protocol.

RARP: Reserve Adresse Resolution Protocol.

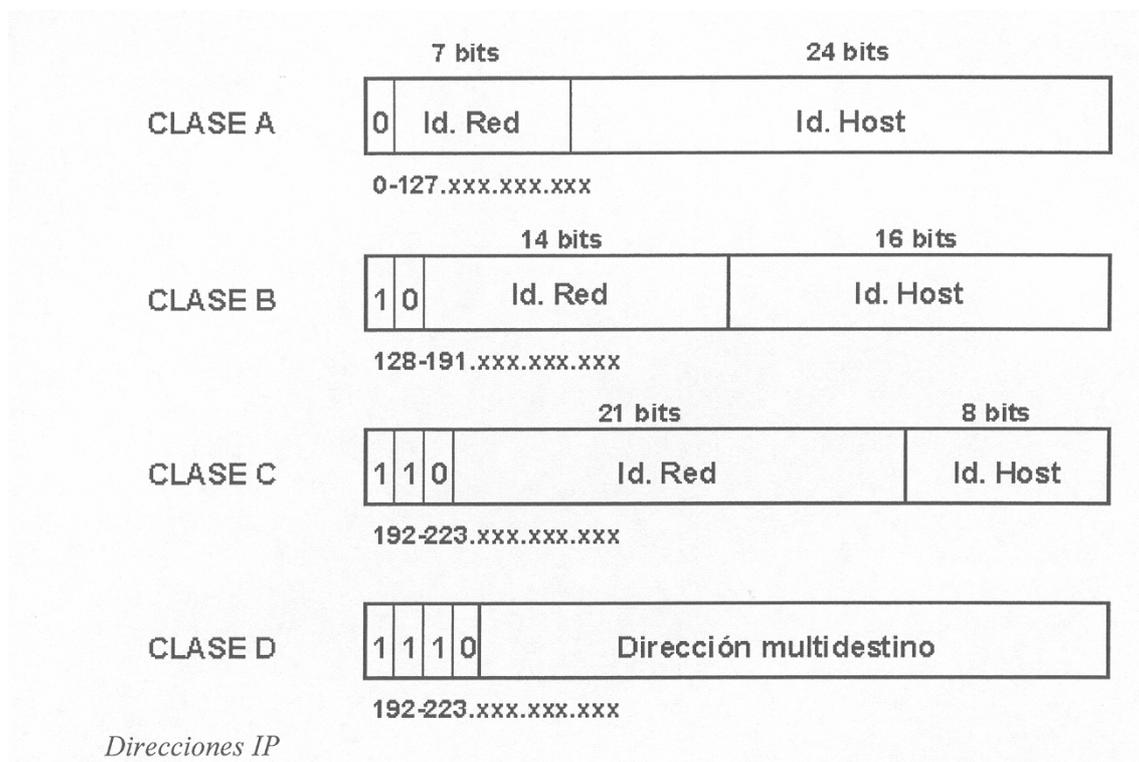
Dirección IP.

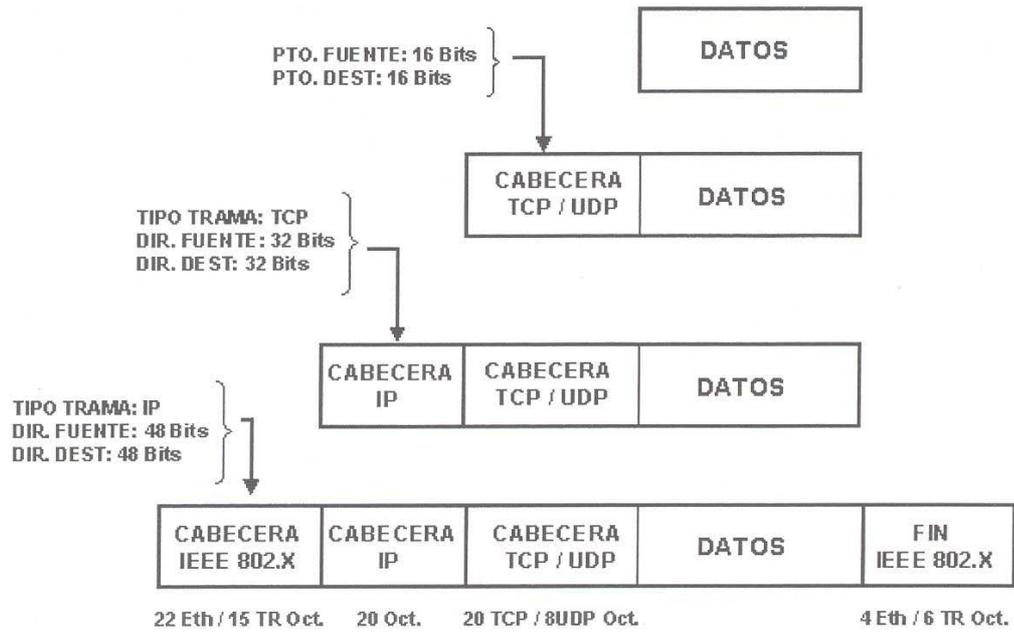
Para realizar un envío es necesario que el emisor identifique al receptor con la máxima exactitud. Esto se realiza asignando una dirección a cada punto de la red. Estas direcciones son identificaciones lógicas que se denominan direcciones IP.

TCP/IP a partir de la dirección IP identifica unívocamente a cada nodo de la red. Estas direcciones son independientes de las direcciones físicas de la red y permite que los cambios que se realizan a nivel de hardware de la red, no influya en TCP/IP ya que la dirección IP solo afecta a las capas superiores.

Las direcciones IP tienen una longitud de 4 bytes cuyas cifras se separan mediante puntos para identificar a la red y al host: xxx.xxx.xxx.xxx.

Para adaptar los diferentes tipos de redes existentes (relación con el número de hosts que componen las diferentes redes) a la dirección IP, se establecen unas clases: A, B, C, D, E, para cada una de las cuales se asigna un número de bits determinado. La mayoría de las direcciones IP utilizadas son de las clases A (24 bits), B (16 bits) y C (8bits). Las clases D y E están reservadas. Véanse los siguientes gráficos orientativos:





Arquitectura TCP/IP

0		16		32	
VERSION	LONG. CABECERA	TIPO DE SERVICIO	LONGITUD TOTAL		
IDENTIFICADOR			FLAGS	OFFSET FRAGMENTO	
TIEMPO DE VIDA	PROTOCOLO		CODIGO REDUNDANCIA (CABECERA)		
DIRECCION FUENTE					
DIRECCION DESTINO					
OPCIONES					
DATOS					

Formato datagrama IP

El sistema de nombres de dominios DNS.

Para conectar una red a Internet es necesario disponer de las direcciones IP correspondientes, para ello la organización InterNIC se encarga de gestionar las peticiones de las diferentes empresas o entidades para realizar el registro.

Inicialmente la lista de hosts se recogía en un archivo de texto en cada ordenador de Internet. Con el crecimiento de la Red, este sistema fue cada vez más costoso ya que al ser un directorio único, su tamaño creció considerablemente y el espacio físico ocupado en cada ordenador era considerable, además cada nombre del ordenador debía ser único para poder identificar unívocamente cada nombre de hosts con su dirección IP.

En 1984 se desarrolló un sistema conocido como DNS (Domain Name System Servidor de Nombres de Dominio) que consistía en una base de datos de estructura jerárquica con todos los nombres y direcciones de los hosts conectados a Internet. Esta base de datos era accesible desde la red, de tal forma que para los usuarios bastaba saber el nombre del ordenador al cual se querían conectar.

La idea del DNS consistía en gestionar el espacio de direcciones de Internet en áreas separadas e independientes llamadas dominios, en lugar de un directorio común. A cada dominio le corresponde una organización que se responsabiliza de la asignación de nombres dentro de su dominio. Dentro de cada dominio se pueden crear subdominios de forma que el nombre de host puede alcanzar hasta cuatro subdominios.

En el nivel superior de la organización encargada de la gestión de los nombres de dominio se encuentra InterNIC.

Los dominios se agrupan en cuatro bloques divididos en dos niveles:

- Dominio de primer nivel (TLD - Top Level Domain). El nivel superior está reservado y en él existen dos grupos de dominios:
 - COM para las empresas comerciales.
 - EDU para las instituciones educativas.
 - NET para las organizaciones de desarrollo y comercialización de Internet.
 - ORG para los casos que no corresponda con las categorías anteriores.
 - INT para organizaciones de tratados internacionales.

El segundo grupo está reservado para los países mediante un código formado por dos o tres letras de acuerdo con la normativa ISO 3166. Existen dos dominios en este nivel reservados para el gobierno de EE.UU:

- GOV para las diferentes agencias del gobierno federal.
- MIL para el ejército.

Recientemente han sido aprobados nuevos dominios. El más importante es el nuevo dominio .BIZ destinado a páginas Web con finalidades comerciales de empresas y particulares. La relación nueva es la siguiente:

- BIZ para los negocios.
 - INFO para todos los usos.
 - AERO para las industrias del transporte aéreo.
 - COOP para cooperativas.
 - MUSEUM para museos.
 - NAME para usos individuales.
 - PRO para profesionales.
- Dominio de segundo nivel. Sirve para identificar el nombre del hosts. En él se indican los dominios y subdominios que identifican al host. Por tanto a cada dominio el DNS asigna su dirección IP. Por ejemplo:

www.sun.com-----192.18.97.241

Si desde un ordenador se quiere acceder por ejemplo al dominio www.sun.com, primero nos debemos conectar con nuestro proveedor de Internet (IPS) para tener acceso a la red, a continuación se deberá obtener la dirección **IP** correspondiente al dominio al cual queremos acceder. Para ello, el DNS del proveedor buscará la dirección **IP**, si no la encuentra la requerirá a un DNS superior y así hasta obtener la dirección **IP**, que se enviará al ordenador que realiza la conexión. Este sin embargo, no sabrá de qué DNS procede la dirección. A partir del **IP** se accederá al hosts que corresponde al dominio indicado.

8.1.2.- Los protocolos de comunicación.

8.1.2.1.- Elementos de la comunicación.

En una comunicación intervienen tres componentes: el emisor, el medio y el receptor. El emisor generará una información que enviará a un receptor y para que la transmisión sea posible será necesario un medio por el que fluya la información.

El emisor generará la información de acuerdo a un determinado sistema de codificación. La mayoría de los ordenadores utilizan el sistema ASCII para la codificación de la información, es decir cada símbolo del alfabeto utilizado viene determinado por una combinación de valores de 8 bits.

Una vez determinado el sistema de representación de la información, esta se debe propagar mediante una señal de una determinada naturaleza: analógica o digital. La información generada por un ordenador es siempre digital de acuerdo con la arquitectura de construcción de los circuitos que integran los procesadores. Si el medio por el cual se propaga la información es de naturaleza analógica habrá que convertir la señal de digital a analógica, este proceso se realiza mediante un módem.

El medio utilizado para enviar la información influirá en la eficiencia de la transmisión, en la velocidad, en la calidad de la señal y en el volumen de información que es capaz de manejar. Así, en función del tipo de cable (par trenzado, coaxial o fibra óptica) o el tipo de ondas electromagnéticas utilizadas, se obtendrán unas prestaciones determinadas.

Finalmente, el canal de transmisión determina la forma en cómo se realiza el intercambio de información, es decir el protocolo de comunicación. Dicho protocolo establece el conjunto de reglas para que la comunicación entre el emisor y el receptor sea posible.

La comunicación se implementa mediante un sistema teleinformático formado por:

- Un sistema central. Se trata de un ordenador que realiza las funciones de emisor y receptor con diferentes capacidades de proceso.
- Un conjunto de interfaces .Garantizan la conexión entre ordenadores y la red de comunicaciones.
- Una red de comunicaciones. Establece la conexión entre el emisor y el receptor.

En una red podemos distinguir tres conceptos:

- Tipo de enlace: *líneas conmutadas*-requieren que emisor y receptor establezcan una conexión, o *líneas dedicadas*-no necesitan establecer ningún enlace previo.
- Topología de la red. Según la forma de establecer el enlace entre estaciones: *punto a punto*-con diferentes estructuras (estrellada, arborecente,...) según el número de terminales, o *multipunto*-diferentes terminales comparten la misma línea de transmisión.

- Elementos de conmutación. Establecen el encaminamiento de la información según tres sistemas básicos de conmutación: *conmutación de circuitos*-el terminal que inicia la comunicación solicita la autorización de llamada y a continuación llama al terminal destinatario, si está libre se establece un enlace físico que dura hasta que finaliza el proceso, *conmutación de mensajes*- el emisor incorpora a la información a transmitir la dirección de destino, el mensaje se transmite de nodo a nodo hasta llegar a su destino final, *conmutación de paquetes*-la información se divide en unidades llamadas paquetes.

8.1.2.2.- Los protocolos de comunicación.

En una comunicación entre dos sistemas, además de un conjunto de elementos físicos (sistema teleinformático), son necesarios un conjunto de sistemas lógicos para conseguir el entendimiento total. Un protocolo es un conjunto de reglas que permiten la comunicación entre equipos informáticos. Los objetivos que persiguen los protocolos son:

- Permitir la comunicación entre ordenadores heterogéneos.
- Adoptar estándares entre fabricantes y usuarios.
- Dividir el proceso de la comunicación para hacer mas fácil la misma.

Los protocolos realizan las siguientes funciones:

- Establecen la comunicación, de la siguiente forma:
 - La estación origen dialoga con la red para indicarle la estación de destino.
 - La red acepta o rechaza en función de los medios disponibles.
 - Las peticiones aceptadas progresan hasta el terminal destino.
 - El terminal destino acepta o rechaza la petición.
- Supervisan la comunicación mediante las acciones siguientes:
 - Comprueban que no se pierde ninguna información.
 - Que los datos lleguen correctamente.
 - Proporciona los medios para que exista una buena sincronización entre los extremos.

Los protocolos se estructuran en capas o niveles. Cada capa tiene un conjunto de funciones para comunicarse, de forma que cada nivel provee servicios al nivel superior. Las capas permiten fraccionar el desarrollo de los protocolos y facilitan el entendimiento en el comportamiento global del protocolo.

El modelo de referencia de la ISO.

Los protocolos de comunicación eran normas internas de cada fabricante, de forma que era casi imposible interconectar equipos de fabricantes diferentes. Para resolver el problema, la organización ISO (Internacional Standards Organization - Organización Internacional de Normas) adoptó un modelo de referencia estructurado en capas:

1. Nivel Físico.

Es prácticamente todo hardware y define el medio de comunicación (tipo de cable y conectores), los niveles eléctricos de la señal, velocidad de la transmisión con el fin de que un bit enviado con un determinado valor (0 ó 1) sea reconocido en el extremo receptor.

2. Nivel de Enlace.

Se refiere a la conexión entre máquinas adyacentes. Debe asegurar la transmisión sin errores, para ello divide los datos emitidos en tramas. Este nivel asegurará la recepción sin problemas de cada trama con las tramas de asentimiento devueltas por el receptor que analiza el campo de la trama que asegura su integridad (normalmente un código de redundancia).

3. Nivel de Red.

Se encarga de encaminar los paquetes desde su origen a su destino. Las rutas pueden basarse en tablas estáticas o encaminarse dinámicamente en forma diferente en cada paquete. Una de las tareas del nivel de red es ocuparse de evitar la congestión por exceso de paquetes en alguna rama de la subred. Si el nivel de red recibe una trama que no es para la máquina en que reside, no la pasará a los niveles superiores y la reenviará hacia la máquina destino.

4. Nivel de Transporte.

Realiza una conexión extremo a extremo entre los niveles de transporte de las máquinas origen y destino. Los protocolos de los tres niveles inferiores trabajan entre máquinas adyacentes y los paquetes pasan por esos niveles de todas las máquinas de la ruta. En una comunicación, sólo se ven involucrados los niveles de transporte del origen y destino sin intervenir los de las máquinas intermedias.

En una comunicación a través de varios nodos puede suceder que se pierdan las tramas de asentimiento con lo que el nodo que ha enviado la trama volverá a hacerlo creyendo que se ha perdido por lo que tendremos tramas duplicadas en la red. Además si empleamos un encaminamiento dinámico utilizando diferentes rutas puede ser que una trama enviada más tarde de un mismo conjunto de datos llegue antes que las enviadas anteriormente. El nivel de transporte será el encargado de eliminar las tramas repetidas y ponerlas todas en el orden correcto. El nivel de transporte será el encargado de subsanar las posibles deficiencias del nivel de red.

5. Nivel de Sesión.

Gestiona el control de diálogo entre los usuarios de diferentes máquinas, mejorando los servicios entre ellos y permitiendo que el tráfico vaya en un solo sentido o evitando que ambos extremos ejecuten una misma operación a la vez. También realiza tareas de sincronización. Por ejemplo el nivel de sesión en una transferencia de archivo va insertando puntos de verificación. Si tenemos una red insegura cada vez que exista una interrupción de las comunicaciones el nivel de sesión se encargará de proseguirlas a partir del último punto de verificación y no tendrá que empezar de nuevo otra vez.

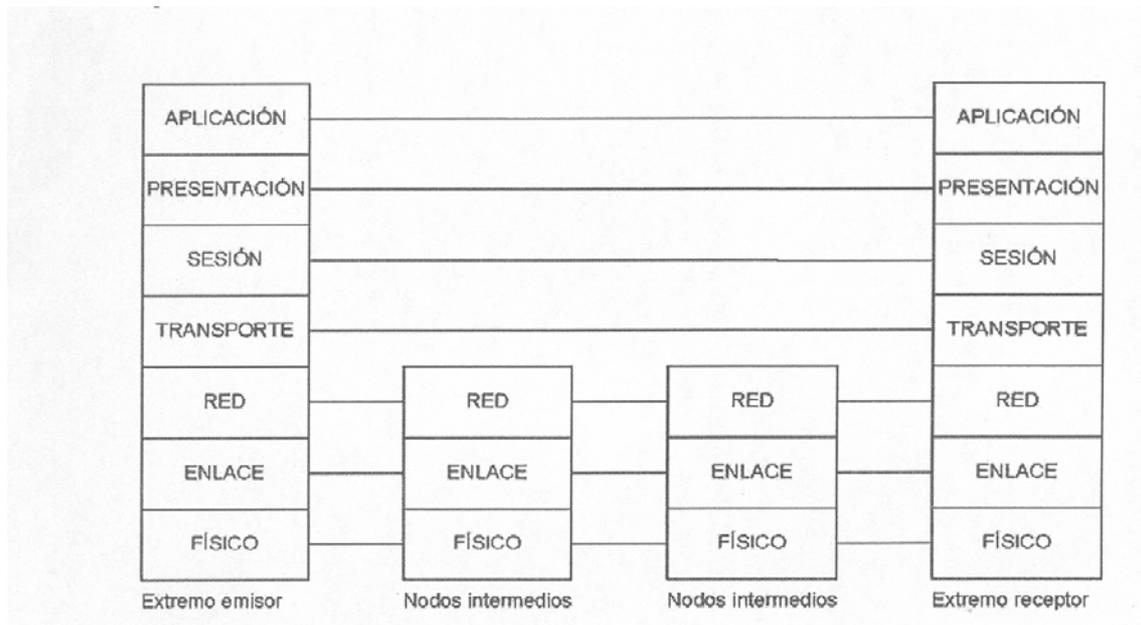
6. Nivel de Presentación.

Se ocupa de los aspectos de representación de la información. Por ejemplo se ocupa del tipo de codificación de los datos previamente establecido. También se ocupa de la compresión de los datos y de su encriptación.

7. Nivel de Aplicación.

Se ocupa de emulación de terminales, transferencia de ficheros, correo electrónico y otras aplicaciones.

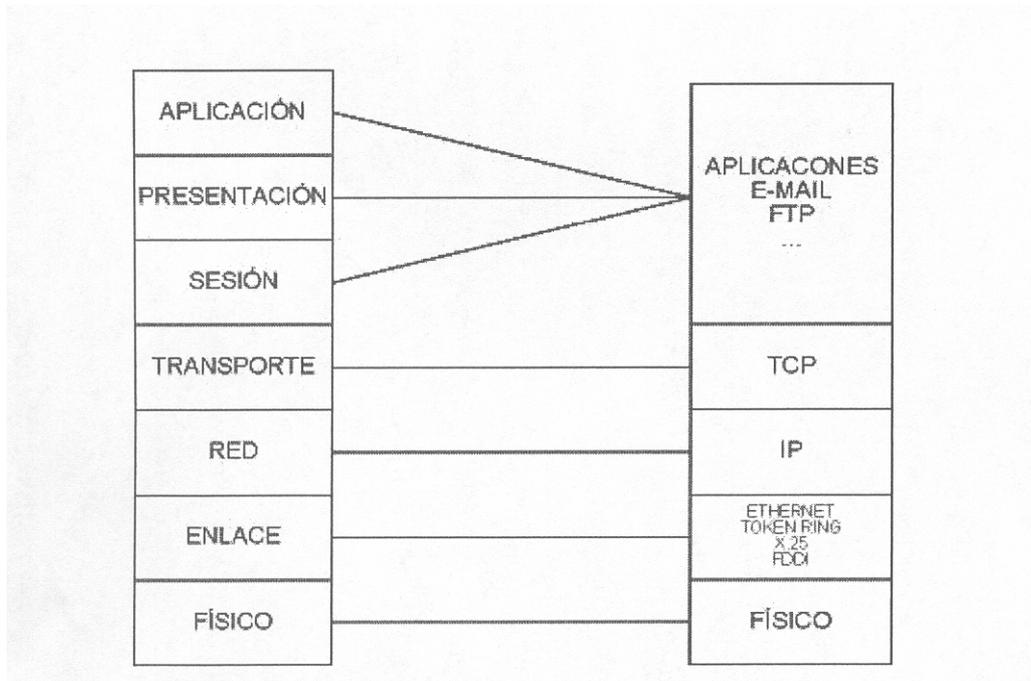
A continuación se muestra un esquema del modelo de referencia de la ISO:



El modelo de Internet.

Aunque el modelo OSI estaba pensado para que fuera seguido por la mayoría de las redes futuras, el auge de Internet ha hecho que la pila de protocolos basada en TCP/IP sea la más utilizada - con mucho - en la actualidad.

En la figura podemos ver la comparación entre ambos modelos (OSI e Internet):



El modelo Internet divide el proceso de comunicación en cuatro capas:

1. Aplicación. Cubre las funciones que realizan las capas de aplicación, presentación y sesión del modelo OSI. Dentro de cada capa existen diferentes protocolos relacionados con las aplicaciones y servicios de la red:

- **FTP** (Protocolo de Transferencia de Archivos-File Transfer Protocol). Permite la transferencia de archivos entre hosts.
- **Telnet** (Protocolo de Terminal Remoto). Permite la conexión de terminales con hosts remotos para ejecutar aplicaciones que residen en el host.
- **SMTP** (Protocolo Simple de Transferencia de Ficheros- Simple Mail Transfer Protocol). Realiza servicios de gestión de mensajes.
- **NSP** (Protocolo de Servidor de Nombres- Simple Network Management Protocol). Permite el acceso a unidades situadas en hosts remotos como si fueran locales.

- **SNMP** (Protocolo Simple de Gestión de Red- Simple Network Transfer Protocol). Permite la obtención de información de los dispositivos de la red.
- **HTTP** (Protocolo de Transferencia de Información Hipertexto- HyperText Transfer Protocol). Se utiliza para la transmisión de información en la Web (WWW).

2. Transporte. Se encarga de la integridad de los datos entre los extremos de la comunicación. Dentro de esta capa existen los siguientes protocolos:

- **TCP** (Protocolo de Control de Transmisión). Permite la conexión simultánea de diferentes hosts con control de errores.
- **UDP** (Protocolo de Datagramas de Usuario- User Datagram Protocol). Sirve para aquellas aplicaciones que no necesitan TCP y da acceso directo a los servidores básicos del IP.

3. Red. Realiza el encaminamiento de los mensajes a través de las redes:

- **IP** (Protocolo de Internet). Permite la transmisión de los paquetes de una red a otra a partir de la ruta de la siguiente subred. Durante el encaminamiento puede desechar algunos datagramas.

4. Acceso a red. Incluye el nivel de enlace y el nivel físico del modelo OSI. Se encarga del intercambio de información entre un host y la red.

8.1.3.- Las aplicaciones web.

8.1.3.1.- Comercio electrónico.

De la misma forma que la WWW utiliza la red de Internet mediante el protocolo TCP/IP para la transmisión de información, las aplicaciones se desarrollan en el marco de la WWW. Desde el punto de vista funcional, las aplicaciones Web son como cualquiera de las demás aplicaciones informáticas, únicamente cambia el entorno de explotación y las herramientas empleadas en su construcción.

Inicialmente, el proyecto de desarrollo de la WWW estaba concebido para el intercambio de la información entre máquinas heterogéneas de forma que independientemente de la arquitectura de la máquina y el sistema operativo utilizados, se pudiera establecer la comunicación de una forma fácil. Un usuario, cliente, podía acceder a una determinada información mediante el conocimiento de la ubicación de dicha información, su URL, y visualizarla mediante su agente usuario o navegador. Sin embargo la información solicitada era estática ya que existía previamente. La llegada del protocolo CGI (Common Gateway Interface- Interfaz de Pasarela Común) hizo posible la petición de información que debería ser elaborada en el momento, en función de los datos enviados por el cliente.

La interactividad entre los clientes y los servidores Web ha potenciado aún más la aparición de nuevas tecnologías para mejorar las posibilidades de comunicación en el entorno de la WWW propiciando el comercio a través de la red y dando lugar a lo que se conoce de forma general como *comercio electrónico*.

Se entiende por *comercio electrónico* al intercambio de información utilizando las IT (Tecnologías de la Información) entre compradores y vendedores. Se suele indicar *comercio electrónico (e-Commerce)* al proceso de la venta de un producto o servicio entre un comprador y un vendedor a través de la red. Mientras que se utiliza el término *negocio electrónico (e-Business)* para especificar el proceso que se realiza de forma electrónica en todas o la mayoría de las actividades derivadas de este comercio.

La facilidad de desarrollo en el ámbito de la WWW ha facilitado el aumento espectacular de las empresas que, a través de sus sitios Web, operan en el amplio marco de la red de Internet. La venta de productos a través de la red se ha convertido en uno de los grandes negocios en aquellos productos cuyos compradores son los usuarios finales y por tanto todos los usuarios de Internet constituyen los clientes potenciales de sus empresas.

Otro ámbito de negocio en la red está formado por los diferentes proveedores de servicios que a través del escaparate de la red se ofrecen a los internautas. Esto ha propiciado el desarrollo de pequeñas pero útiles aplicaciones Web que con pocos recursos en su desarrollo pueden proporcionar ganancias considerables.

En otros casos, este medio de gran difusión ha hecho posible aumentar la oferta en la comercialización o marketing de los productos o servicios. Agencias de viajes, editoriales, casas de discos, etc. forman parte del amplio grupo de empresas que operan en la red. Otro servicio es el que ofrecen los Bancos y Cajas de Ahorros que mediante agencias virtuales permiten a sus clientes realizar operaciones bancarias a través de la red.

8.1.3.2.- B2B (BUSSINESS-TO-BUSSINESS).

Las grandes ventajas aportadas por Internet a la empresa no han hecho más que vislumbrar el futuro de posibilidades para la adopción de nuevos modelos empresariales basados en Internet. Es de la experiencia de la pequeña y mediana empresa en Internet, donde esta se plantea el uso de este medio no solo como herramienta para el ensanchamiento del mercado potencial, sino también como herramienta para la mejora de los procesos de negocio, tanto en su operativa interna (poco o nada planteada por la empresa hasta ahora), como en su operativa externa (cubriendo no sólo los procesos de negocio en Internet sino toda la red de venta, distribución y logística que rodean a la empresa).

Entramos en este punto en el terreno del e-business y las soluciones B2B. Esta transición del e-commerce al e-business, implica en los directivos encargados de evaluar estas actuaciones un cambio de mentalidad: de estrategias centradas en el producto y la venta, a estrategias centradas en el cliente, el valor añadido y la optimización de todos los procesos de negocio. La perspectiva empresarial de Internet se agranda y pasamos de verlo como un medio para comprar, vender, comunicar y publicitarse, a verlo como una herramienta de gestión empresarial global, que nos acerca a la realidad de una empresa que interactúa con proveedores, clientes, procesos productivos y con otras empresas al mismo tiempo. Si el comercio electrónico o e-commerce ha sido la evolución del propio comercio, el e-business es la evolución lógica de los modelos de gestión tradicionales.

De la urgencia para la comprensión y adopción de modelos de e-business por parte de la empresa, nos da buena cuenta el hecho de que la industria del software, Microsoft y Oracle entre ellas, haya comenzado a implantar en todas sus plataformas estándares como XML, en búsqueda de mejoras en los sistemas transaccionales de interacción con terceros. Esta tendencia, nos llevará al uso generalizado de soluciones empresariales de gestión global basadas en Internet para todo tipo de empresas, con independencia del tamaño y sector en el que desenvuelvan su actividad.

La ventaja competitiva de soluciones e-business basadas en Internet sobre los sistemas EDI tradicionales (de implantación demasiado costosa debido a la tecnología en la que están basados), ha venido propiciada por la propia evolución de Internet y de los lenguajes de desarrollo de aplicaciones como el ya mencionado XML.

El e-business engloba a toda una serie de modelos de negocio basados en tecnología Internet encaminados a mejorar las relaciones comerciales entre empresas, cadenas de aprovisionamiento, mercados verticales y un largo etcétera de posibilidades. En última instancia un sistema de e-business puede tomar múltiples formas y es la empresa quien debe decidir la mejor o más adecuada según sus necesidades. A continuación se describen brevemente los modelos que con toda seguridad más tendrán que digerir las empresas en los próximos años:

- **ERP** (Enterprise Resource Planning). Podemos considerar este software como la tecnología subyacente de gestión interna sobre la cual basar el resto de modelos de negocio de e-business. El término ERP deriva de MRP (Material Requirement Planning) herramienta para el control de procesos productivos. Los sistemas ERP administran los procesos internos del negocio para la optimización de la cadena de valor que sirve a todos los departamentos dentro de la empresa. El software ERP incluye diversas funcionalidades: facturación, contabilidad, compras, producción, transporte, informes de gestión y recursos humanos entre otras.
- **SCM** (Supply Chain Management). Gestiona los procesos de negocio tanto internos como externos de la empresa implicando a todos los agentes que directa o indirectamente están involucrados, desde la producción a la distribución. El SCM incluye el aprovisionamiento de materias primas, proveedores, la atención al cliente, la logística y en general toda la cadena de valor de la empresa, optimizando los procesos más que automatizándolos, como es el caso del ERP.

- **CRM** (Customer Relationship Management). Dirigido a todos los aspectos relacionados con la atención y el servicio al cliente, coordina a todos los departamentos involucrados en esta atención: departamentos de ventas, marketing y relaciones con los clientes. Las soluciones CRM gestionan conjuntamente el servicio de reclamaciones, la gestión de incidencias, vendedores y seguimiento de ventas. Al funcionar sobre sistemas de Datawarehouse permiten obtener perfiles de usuario, preferencias y hábitos de compra.
- **Marketplaces**. Mercados virtuales para la venta, compra e intercambio de información entre múltiples participantes. Básicamente consiste en un directorio de empresas con información sobre los productos de cada una donde compradores y vendedores buscan productos o servicios, solicitan ofertas y procesan pedidos.
- **E-procurement**. Abastecimiento electrónico de productos y servicios vía Internet. Bajo estas plataformas se gestionan los procesos de compra a proveedores bien sean compras de productos directos (implicados en el proceso de producción del producto final): materias primas, o indirectas (no implicadas en el producto final): papelería, informática, servicios varios. La principal ventaja del uso de estas plataformas radica en el ahorro de tiempo en la gestión de compras, la comodidad y la reducción de los precios de adquisición de productos y la posibilidad de acceder a nuevos proveedores.
- **BI** (Business Intelligence) centrado en el apoyo a la toma de decisiones y la evaluación de indicadores de negocio.
- El **KM** (Knowledge Management) para la gestión del conocimiento y cuyo objetivo es lograr que la información dentro de una organización llegue a todo aquel que la necesite, procesada de forma tal que sea posible llevarla a la práctica.

8.1.4.- Los lenguajes web.

8.1.4.1.- La arquitectura de las aplicaciones web.

Los lenguajes de código de formato (lenguajes derivados del SGML) y los lenguajes de scripting (lenguajes de guiones) desarrollados en el entorno de WWW (Java Script, JScript, VBScript, etc.), junto con los lenguajes de programación tanto de alto nivel (Pascal, C, etc.) como los orientados a objetos (C++, Java, etc.) y los lenguajes shell (Unix) constituyen las herramientas mediante las que se construyen las aplicaciones Web. Las aplicaciones generadas constituirán el denominado *software de aplicación*: conjunto de páginas creadas con los lenguajes de código de formato (HTML, CSS, DHTML, etc.), los scripts y los programas desarrollados en otros lenguajes (C, C++, Java, Perl, etc.). Además del software elaborado para implementar las funciones de las aplicaciones, hay que almacenar los datos que van a utilizar dichas aplicaciones. Estos residirán en ficheros o bases de datos, junto con el software desarrollado.

Las aplicaciones informáticas constituyen el software que permite realizar de una forma mecanizada el conjunto de procedimientos que deben seguirse en el desarrollo de las actividades de una empresa o institución (facturación, gestión de pedidos, etc.). Una aplicación informática consta por tanto de un conjunto de programas cada uno de los cuales realiza una determinada función.

Las diferentes arquitecturas de construcción de aplicaciones determinarán donde tienen que residir los datos, el software de la aplicación y donde se van a ejecutar los procesos. Estas arquitecturas son:

- Arquitectura centralizada. Las aplicaciones y bases de datos residen en el host (mainframe) y los procesos se ejecutan en él. Los accesos a las aplicaciones serán *locales*, si se ejecutan desde terminales conectadas directamente con el host, o bien *remotos*, si estos se realizan a través de una red de teleproceso.
- Arquitectura descentralizada. Las aplicaciones y las bases de datos se distribuyen en diferentes hosts a los cuales se accede local o remotamente. Entre las diferentes aplicaciones se establecen funciones de transferencia de datos o conexiones de igual a igual (peer-to-peer). El acceso a la aplicación puede ser local o remoto.
- Arquitectura con procesos cooperativos. Los procesos se ejecutan en dos hosts. En uno de ellos, que será el host o el mainframe, se realiza la mayor parte del proceso (lógica del negocio y acceso a la BD) y en otro, que es el terminal que realiza la petición, se realizan las funciones de representación de datos. En este caso, el host se descarga de los procesos de representar la información y las validaciones formales de los datos de entrados por la estación de trabajo ya que solo debe enviar los datos que obtiene de la BD que han sido solicitados y por tanto disminuye la comunicación entre ambos.
- Arquitectura Cliente-Servidor. En este caso los procesos se ejecutan en más de un host, en los cuales se puede distribuir la lógica del proceso, la lógica de los datos y las funciones de representación de los datos (interfaz de usuario). El acceso a la aplicación puede ser local o remoto.

8.1.4.2.- La arquitectura Cliente/Servidor.

La arquitectura Cliente/Servidor es un modelo para construir sistemas de información partiendo de una diversidad de tecnologías que distribuyen aplicaciones, datos y servicios a través de múltiples procesadores. El término Cliente/Servidor implica la relación entre unos procesos que inician solicitudes de servicios (clientes) y otros procesos que responden a las peticiones ejecutando procesos que se realizan en el mismo procesador o en otros cualesquiera de la red.

El modelo C/S permite una clara separación de funciones basada en el concepto de servicio, situando a cada aplicación en la plataforma más adecuada dentro del entorno de sistemas abiertos, estableciéndose entre el cliente y el servidor un protocolo de acceso.

El objeto de las aplicaciones C/S es conseguir una distribución de funciones entre los procesadores de una red y/o WAN que permita el uso óptimo de los recursos. De acuerdo con las funciones que se distribuyan se distinguen tres modelos: presentación distribuida, datos distribuidos, y función distribuida.

Una aplicación C/S se compone de un número determinado de procesos cliente y servidor. Generalmente los procesos cliente se ejecutan en una estación de trabajo y solicitan servicios de múltiple procesos al servidor, local o remoto, utilizando cualquiera de los tres modelos.

Estudiemos a continuación los tres modelos:

a) Presentación distribuida. En este modelo la parte de la aplicación que se ocupa de la presentación de los datos está situada remotamente respecto de la lógica de negocio y de los datos de la aplicación. La lógica de la aplicación se desarrolla generalmente con ayuda de las herramientas del desarrollo C/S. La presentación distribuida en su forma más simple puede consistir en una interfaz gráfica de usuario para acceder a programas de proceso de transacciones ya existentes.

b) Datos distribuidos. Cuando se distribuyen los datos, su acceso y manipulación quedan separados del resto de la aplicación:

- **Ficheros distribuidos.** Se utilizan principalmente para compartir los recursos de una red. Es la forma más sencilla de proceso distribuido y consiste en una redirección física de las solicitudes de lectura y escritura. Lo más común en este tipo de servidores es la lógica de la aplicación, que siempre se ejecutará en el proceso cliente.
- **Bases de datos distribuidas.** Permiten utilizar de forma más eficiente el procesador que controla la BD. El proceso cliente envía las peticiones SQL en forma de mensajes al servidor de la BD. Todas las operaciones de búsqueda se realizan en el servidor y únicamente devuelve el resultado a la aplicación que realiza la petición.

c) Función distribuida. Este modelo proporciona la máxima flexibilidad y permite a los departamentos de desarrollo un control total sobre la ubicación de las funciones en la red. Un proceso cliente invoca a un proceso servidor que reside en el mismo procesador o en otro procesador de la red. El servidor puede realizar funciones relativas al negocio y acceder a ficheros o BD, devolviendo el resultado al proceso que lo ha invocado. El cliente recibe el resultado sin saber si procede del mismo procesador o de otro distinto. Un servidor de aplicaciones puede, simultáneamente, actuar como cliente, solicitando funciones de otros servidores de la red.

8.1.4.3.- La arquitectura de un sitio web.

Las aplicaciones Web son aplicaciones informáticas que se realizan en el entorno de la World Wide Web. Sus componentes fundamentales son:

- El **lenguaje de códigos de formato HTML** como elemento central de construcción de páginas Web que incorporan datos, objetos multimedia y programas desarrollados en lenguajes scripting y de alto nivel.
- El protocolo de comunicación entre el cliente y el servidor a nivel de aplicación: **protocolo HTTP**.
- El **servicio de transporte** a través de la red Internet mediante el protocolo **TCP/IP** que permite transportar la información e identificar los recursos de la red mediante una dirección IP que indica la ubicación de la aplicación Web, su URI o sitio Web.

La mayor parte de las aplicaciones están estructuradas alrededor de documentos HTML que mediante una estructura hipertexto permiten recorrer los diferentes documentos de que se compone. Todas las mejoras a partir de este punto han consistido en añadir lenguajes de programación y de scripting a los documentos HTML de diferentes formas y utilizando diferentes técnicas.

Desde el punto de vista funcional podemos aplicar las aplicaciones Web como:

- **Aplicaciones estáticas.** En este caso los clientes realizan determinadas peticiones de recursos al servidor. Dichos recursos ya existen en algún sitio Web y el servidor se encarga de enviarlos al cliente que los ha solicitado. Los documentos enviados se visualizan a través del agente usuario o navegador y finaliza la transacción.
- **Aplicaciones dinámicas.** Las aplicaciones dinámicas son idénticas a las aplicaciones estáticas en el sentido de que no pueden intercambiar información del cliente al servidor. Sin embargo, estas aplicaciones pueden dotarse de formas variables de representación de la información (información dinámica) en función de la acción realizada por el usuario.
- **Aplicaciones interactivas.** El cliente puede establecer un dialogo con el servidor. El cliente introduce unos datos que envía al servidor para que se incorporen a los ficheros o bases de datos de la aplicación o para solicitar una determinada información que el servidor tendrá que elaborar en función de los datos recibidos y remitirlos al cliente de forma dinámica. Como vemos, en este caso no existe previamente la página solicitada sino que la crea el servidor a partir de los datos recibidos del cliente.

8.1.4.4.- Lenguajes web.

Uno de los principales objetivos del desarrollo de las aplicaciones Web es lograr el máximo de interactividad entre los clientes y el servidor. A continuación se detallan las diferentes tecnologías que se han desarrollado en este aspecto:

a) Lenguajes del **lado del cliente.**

- **Hojas de estilo** (Cascade Style Sheets). La utilización de las hojas de estilo permite variar la visualización de una página Web. En realidad es una forma de separar la información del formato utilizada en su representación.
- **Lenguajes de scripting.** Mediante estos lenguajes (JavaScrip, VBScrip, y JSCript) podemos elaborar programas que se incrustan en las páginas Web y que se ejecutan en el cliente cuando se activa un evento. El cliente ejecuta los scripts mediante un motor de scripts que está incluido en el agente usuario, de forma que este, además de convertir la información a formato HTML, deberá también ejecutar las instrucciones contenidas en dicho script.
- **DOM** (Document Object Model-Modelo de Objetos de Documento). Es un interfaz que permite a los scripts o programas acceder dinámicamente a la estructura y estilo de los documentos para cambiar la presentación de los mismos. Permite la representación de objetos en los documentos y establece estándares de interfaces para su acceso y tratamiento.

- **Applets de Java.** Java es un lenguaje de programación orientado a objetos que permite elaborar *applets* que se ejecutan dentro del navegador para proporcionar una determinada funcionalidad. El funcionamiento es similar a los lenguajes scripting pero su funcionalidad mucho mayor. Los applets se incrustan en las páginas Web que se ejecutan en el cliente cuando se activa un evento. El cliente ejecuta los scripts mediante un motor de scripts que está incluido en el agente usuario, de forma que este, además de convertir la información a formato HTML, deberá también ejecutar las instrucciones contenidas en dicho script.
- **Activex.** Esta tecnología permite incorporar una fracción de código y datos basados en objetos que proporciona el modelo COM (Component Object Model). Mediante esta tecnología se pretende proporcionar una interfaz estándar que facilite la comunicación entre los diferentes componentes del software.

b) Lenguajes del **lado del servidor**.

- **SSI (Server Side Includes).** Esta tecnología se basa en incluir en el documento HTML pequeños bloques de programas de forma que, una vez que el servidor ha recibido la petición de una página Web por parte del cliente, debe analizar el documento HTML solicitado y realizar un preprocesado de los bloques de programa existentes y sustituirlos por su resultado antes de enviarlos al cliente.
- **CGI (Common Gateway Interface).** Este método consiste en un enlace ejecutable de forma que desde un documento HTML se activa un enlace que ejecuta un proceso en el servidor Web realizado por un programa que se denomina script CGI. Tras la ejecución el cliente recibe una página Web. Este método permite la máxima interacción entre el cliente y el servidor. CGI es un estándar que define cómo los servidores Web utilizan los programas externos, es decir, como los servidores pasan los datos enviados por el cliente a los programas CGI y como estos devuelven el resultado al cliente.
- **API (Application Programming Interface).** Para solventar el problema del rendimiento se utilizó, por parte de Netscape (NASAPI) y posteriormente Microsoft (ISAPI), un sistema que permitía extender las funciones del servidor mediante la inclusión de librerías que contenían módulos para realizar la ejecución de los procesos que incluían las páginas Web. Al no existir un proceso suplementario el rendimiento es superior.
- **Lenguajes de scripting: JavaScript.** Java Script es un lenguaje de guiones orientado a objetos. Está formado por un conjunto de objetos y elementos que incluyen, operadores, control de estructuras y sentencias. Mediante este lenguaje se pueden crear página HTML dinámicamente a partir de las peticiones de los clientes utilizando la tecnología NASAPI y además separar el código del lenguaje del documento HTML generado.
- **ASP (Active Server Page).** Esta tecnología consiste en ejecutar páginas ASP según la tecnología ASAPI. Las páginas ASP (ficheros con extensión .asp) están formadas por código HTML en el que se incrustan bloques de programa de scripting, en general se utilizan VBScript y JScrip pero pueden utilizarse otros lenguajes: Perl, Pitón,...

- **JSP** (lava Servlets). Esta tecnología es similar a la de API pero utilizando el lenguaje Java, lo cual proporciona mayor estabilidad. El resultado es una página HTML en el cual está incrustado código Java. JSP evita mezclar código con la salida de código HTML mediante la utilización de plantillas HTML, mejorándose el mantenimiento de las aplicaciones.
- **Enterprise Java Beans**. Esta tecnología está formada por componentes Java que son ejecutados en servidores de aplicaciones. Estos compuestos son independientes de la plataforma y constituyen "piezas" para formar aplicaciones distribuidas.
- **PHP**. Es un lenguaje de scripting que permite generar páginas dinámicas en el servidor y puede ejecutarse en múltiples plataformas (Linux, Win32, etc.). Es un lenguaje de libre disposición por lo que es muy utilizado. Desde el punto de vista del lenguaje es parecido al lenguaje C y contienen conceptos de otros lenguajes como Perl y Java. A continuación lo desarrollaremos más detalladamente.

8.1.5.- Lenguaje de programación PHP.

8.1.5.1.- Introducción a la programación en PHP.

PHP es uno de los lenguajes de lado del servidor más extendidos en la red. Nacido en 1994, se trata de un lenguaje de creación relativamente reciente que ha tenido una gran aceptación debido sobre todo a su potencia y simplicidad.

PHP nos permite embeber los fragmentos de su código dentro de páginas HTML y realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas programados íntegramente en un lenguaje distinto al HTML. Por otra parte, y es aquí donde reside su mayor interés con respecto a lenguajes pensados para los CGI, PHP ofrece un sinfín de funciones para la explotación de bases de datos de una manera sencilla.

Otro conocido lenguaje de programación es ASP. Si bien PHP y ASP son lenguajes parecidos en cuanto a potencia y dificultad, su sintaxis difiere sensiblemente. Algunas de sus diferencias principales son:

- PHP, aunque multiplataforma, ha sido concebido inicialmente para entornos UNIX y es en este sistema operativo donde se pueden aprovechar mejor sus prestaciones. ASP, siendo una tecnología Microsoft, está orientado hacia sistemas Windows, especialmente NT.
- Las tareas principales que puede realizar directamente el lenguaje son definidas en PHP como funciones mientras que ASP invoca más frecuentemente los objetos. Por supuesto, esto no es más que una simple cuestión de forma ya que ambos lenguajes soportan igualmente ambos procedimientos.
- ASP realiza numerosas tareas sirviéndose de componentes (objetos) que deben ser comprados a empresas especializadas o programados por el servidor. PHP presenta una filosofía totalmente diferente, es progresivamente construido por los usuarios que implementan nuevas funciones en nuevas versiones del lenguaje.

8.1.5.2.- Breve historia de PHP.

PHP es un lenguaje creado por una gran comunidad de personas. El sistema fue desarrollado originalmente en el año 1994 por Rasmus Lerdorf como un CGI en C que permitía la interpretación de un número limitado de comandos. El sistema fue denominado Personal Home Page Tools y adquirió relativo éxito gracias a que otras personas pidieron a Rasmus poder utilizar los programas que desarrolló en sus propias páginas. Dada la aceptación del primer PHP y de manera adicional, su creador diseñó un sistema para procesar formularios al que le atribuyó el nombre de FI (Form Interpreter) y el conjunto de estas dos herramientas sería la primera versión compacta del lenguaje PHP/FI.

La siguiente contribución al lenguaje se realizó a mediados del 97 cuando se volvió a programar el analizador sintáctico. Se incluyeron nuevas funcionalidades como el soporte a nuevos protocolos de Internet y el soporte a la gran mayoría de las bases de datos comerciales. Todas estas mejoras sentaron las bases de PHP versión 3. Actualmente PHP se encuentra en su versión 4, que utiliza el motor Zend, desarrollado con mayor meditación para cubrir las necesidades actuales y solucionar algunos inconvenientes de la anterior versión. Algunas mejoras de esta nueva versión son su rapidez, gracias a que primero se compila y luego se ejecuta mientras que antes se ejecutaba mientras se interpretaba el código, su mayor independencia del servidor web, creando versiones de PHP nativas para más plataformas, y un API más elaborado y con más funciones.

En los últimos años el número de servidores que utilizan PHP se ha disparado logrando situarse por encima de 5 millones de sitios y más de 800.000 direcciones IP. El hecho de que PHP se haya convertido en una tecnología popular se debe, entre otras razones, a que es el complemento ideal para que el tándem Linux-Apache sea compatible con la programación del lado del servidor de sitios web. Gracias a la aceptación lograda y a los esfuerzos realizados por una creciente comunidad de colaboradores para implementarlo de la manera más óptima, podemos asegurar que este lenguaje se convertirá en un estándar de éxito entre los sistemas desarrollados en código abierto.

8.1.5.3.- PHP: lenguaje del lado del servidor.

El navegador es una especie de aplicación capaz de interpretar las órdenes recibidas en forma de código HTML fundamentalmente, y convertirlas en páginas que son el resultado de dicha orden.

Al pinchar sobre un enlace hipertexto lo que pasa es que establecemos una petición de un archivo HTML residente en el servidor (un ordenador que se encuentra continuamente conectado a la red) el cual es enviado e interpretado por nuestro navegador (el cliente).

Si la página que pedimos no es un archivo HTML, el navegador es incapaz de interpretarla y lo único que es capaz es de salvarla en forma de archivo. Es por ello que si queremos emplear lenguajes accesorios para realizar un sitio web, es absolutamente necesario que sea el propio servidor quien los ejecute e interprete para luego enviarlos al cliente en forma de archivo HTML totalmente legible por él.

Así pues tenemos lenguajes de lado servidor, que son aquellos reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible por él, y lenguajes de lado cliente (entre los que no sólo se encuentra HTML sino también Java y JavaScript los cuales son simplemente incluidos en el código HTML), que son directamente "digeridos" por el navegador y no necesitan un pretratamiento.

Cada tipo tiene sus ventajas e inconvenientes. Así un lenguaje de lado cliente es totalmente independiente del servidor lo cual permite, por ejemplo, que la página pueda ser albergada en cualquier sitio sin necesidad de pagar más ya que, por regla general, los servidores que aceptan páginas con scripts de lado servidor son en su mayoría de pago o sus prestaciones son muy limitadas. Inversamente, un lenguaje de lado servidor es independiente del cliente por lo que es mucho menos rígido respecto al cambio de un navegador a otro o respecto a las versiones del mismo. Por otra parte, los scripts son almacenados en el servidor de quien los ejecuta y traduce a HTML por lo que permanecen ocultos para el cliente. Este hecho puede resultar una forma legítima de proteger el trabajo intelectual realizado.

En el dominio de la red, los lenguajes de lado servidor más ampliamente utilizados para el desarrollo de páginas dinámicas son el ASP, PHP y PERL.

PERL es el lenguaje más rápido y potente por lo que obviamente requiere un aprendizaje más largo quedando más reservado para personas ya familiarizadas con la programación.

El ASP (Active Server Pages) es un lenguaje derivado del Visual Basic desarrollado por Microsoft. Evidentemente su empleo se realiza sobre plataformas funcionando bajo sistema Windows NT.

El PHP podría ser considerado como un lenguaje análogo al ASP utilizado en plataformas Unix y Linux.

Estos dos lenguajes resultan bastantes útiles para la explotación de bases de datos y su aprendizaje es accesible para una persona profana en programación. Cualquiera de ellos resultaría una buena opción a la hora de hacer evolucionar un sitio web realizado en HTML.

8.1.5.4.- Tareas principales del PHP.

En un principio diseñado para realizar un poco más que un contador y un libro de visitas, PHP ha experimentado en poco tiempo una verdadera revolución y, a partir de sus funciones, en estos momentos permite realizar una multitud de tareas útiles para el desarrollo web:

- **Gestión de bases de datos.** Resulta difícil concebir un sitio actual, potente y rico en contenido que no sea gestionado por una base de datos. El lenguaje PHP ofrece interfaces para el acceso a la mayoría de las bases de datos comerciales y por ODBC a todas las bases de datos posibles en sistemas Microsoft, a partir de las cuales podremos editar el contenido de nuestro sitio con absoluta sencillez.

- **Gestión de archivos.** Crear, borrar, mover, modificar,...cualquier tipo de operación más o menos razonable que se nos pueda ocurrir puede ser realizada a partir de una amplia librería de funciones para la gestión de archivos por PHP. También podemos transferir archivos por FTP a partir de sentencias en nuestro código, protocolo para el cual PHP ha previsto también gran cantidad de funciones.
- **Tratamiento de imágenes.** Evidentemente resulta mucho más sencillo utilizar Photoshop para el tratamiento de imágenes pero PHP es útil, por ejemplo, cuando tenemos que tratar gran cantidad de imágenes enviadas por nuestros internautas. Puede resultar muy tedioso uniformar en tamaño y formato miles de imágenes recibidas día tras día pero con PHP podemos automatizarlo eficazmente.
- **Funciones de correo electrónico.** Podemos, con gran facilidad, enviar un e-mail a una persona o lista parametrizando toda una serie de aspectos tales como el e-mail de procedencia, asunto, persona a responder, etc.

Muchas otras funciones pensadas para Internet (tratamiento de cookies, accesos restringidos, comercio electrónico,...) o para propósito general (funciones matemáticas, explotación de cadenas, de fechas, corrección ortográfica, comprensión de archivos, botones dinámicos,...) son realizadas por este lenguaje. A esta inmensa librería cabe ahora añadir todas las funciones personales que uno va creando por necesidades propias y que luego son reutilizadas en otros sitios y todas ellas intercambiadas u obtenidas en foros o sitios especializados.

8.1.5.5.- Instalación de PHP en nuestro servidor.

Como todo lenguaje del lado del servidor, PHP requiere de la instalación de un servidor en nuestro PC para poder trabajar en local. Este modo de trabajo es evidentemente más práctico que colgar los archivos por FTP en el servidor y ejecutarlos desde nuestro navegador.

Así pues, antes de crear programas en PHP, necesitamos:

- Convertir nuestro ordenador en un servidor. Esto se consigue instalando uno de los varios servidores disponibles para el sistema operativo de nuestra máquina.
- Introducir en nuestro servidor los archivos que le permitan la comprensión del PHP.

8.1.6.- Las bases de datos.

8.1.6.1.- Introducción a las bases de datos.

Las bases de datos constituyen uno de los componentes fundamentales de las aplicaciones puesto que en ellas se guardan los datos gestionados por la aplicación. Las bases de datos, dada su estructura lógica que asocia varios agregados de datos, disponen de lenguajes específicos para la creación y manipulación de estas estructuras de datos.

Existen tres modelos de datos: el modelo jerárquico, el modelo en red y el modelo relacional. Cada uno utiliza un sistema propio para establecer las estructuras de los registros de las BD (Base de Datos) y emplean una nomenclatura propia, si bien los conceptos básicos en el modelo de datos son válidos para todos ellos. Por su facilidad de diseño y capacidad de adaptación a los cambios de aplicaciones, el modelo relacional se utiliza principalmente, a pesar de los requerimientos de hardware para que los accesos sean rápidos cuando las BD tengan muchos registros.

8.1.6.2.- Las bases de datos relacionables.

Una relación, en el modelo relacional, está conceptualmente asociada a una *tabla* en la cual las columnas se denominan *atributos* y las filas *tuplas*. Más formalmente una tabla es un conjunto de tuplas y una tupla es un conjunto de valores de los atributos sobre los que se define la relación. Los conjuntos de valores que pueden tomar los atributos se denominan *dominios*: relación (atributo_1, atributo_2,..., atributo_n).

Para representar la información que gestiona una aplicación se utiliza un modelo basado en los siguientes conceptos:

- **Entidad.** Es un concepto del mundo real del cual queremos guardar información.
- **Atributo.** Es un concepto que corresponde a la información que caracteriza a una entidad. Una entidad tendrá diferentes atributos y uno de ellos definirá de forma unívoca a la ocurrencia de la entidad. Este atributo recibe el nombre de *clave* de la entidad.
- **Interrelación.** Es una asociación entre dos entidades (interrelación binaria). Esto significa que ocurrencias de una entidad están asociadas con otras ocurrencias de otra entidad.

8.1.6.3.- Los sistemas gestores de bases de datos relacionables.

Para la utilización de una BD se requiere un software específico de gran complejidad que se denomina Sistema Gestor de Base de Datos (SGBD). Este sistema actúa como intermediario en los programas de usuario y los datos físicos. Cuando un programa de aplicación pregunta si una determinada información (atributo) está en la BD, el SGBD realiza la búsqueda de la información solicitada.

El SGBD permite expresar las relaciones existentes entre los datos y facilita el diseño de nuevas aplicaciones sobre los mismos datos. Así mismo, proporciona un sistema de copias de seguridad y capacidad de reconstrucción para prevenir a la BD contra pérdidas o destrucciones de información, de esta forma se garantiza la integridad de la BD.

El enfoque a tres niveles en la modelización de una BD es el aspecto más significativo de este sistema. Se diseña para poder soportar los cambios y la evolución de la empresa, en tres esquemas:

- **Esquema conceptual.** El papel de este esquema es el de actuar como denominador común entre los datos almacenados según determinados criterios de optimización y las formas de ver estos datos por los usuarios de la BD.
- **Esquema externo.** Los diferentes usuarios del sistema de gestión de la BD trabajan con visiones particulares del modelo de la empresa, los cuales satisfacen sus necesidades particulares. Estas formas de ver los datos según cada usuario son representadas en la arquitectura del modelo con esquemas externos.
- **Esquema interno.** Las formas de ver los datos en el ordenador se describe en el esquema interno. En él se explica la forma de almacenar la información de la empresa.

El software que forma un SGBD es complejo y está formado por diferentes componentes para poder implementar las características del modelo de relación: Lenguaje de definición de datos (DDL), Lenguaje de manipulación de datos (DML), Lenguaje de descripción de la estrategia de almacenaje de los dispositivos (DSDL), Lenguaje de definición de los subesquemas (SDDL), etc. De entre todos ellos los dos fundamentales para el desarrollo de las aplicaciones Web que corresponden al lenguaje relacional son:

- **DDL (Data Description Lenguaje).** Es un lenguaje necesario para declarar la BD. Se utiliza para describir las estructuras lógicas y físicas de la BD y para su creación inicial. Es un lenguaje propio y autosuficiente.
- **DML (Data Manipulation Lenguaje).** Es un lenguaje mediante el cual el programador utiliza los datos que necesita de la BD. No es completo por sí solo y por tanto necesita de la ayuda de un lenguaje de propósito general que llamamos lenguaje host para poder manipular la estructura de datos.

8.1.6.4.- Los lenguajes relacionales: SQL.

El modelo relacional tiene diferentes enfoques de lenguajes relacionales entre los cuales destaca el lenguaje SQL. De este lenguaje, que estableció E. Codd siguiendo un modelo de operaciones algebraicas, existen diferentes versiones desde su creación que han ido aumentando la capacidad del lenguaje en función de las necesidades de las aplicaciones y las nuevas tendencias aparecidas en la programación.

El SQL no es más que un lenguaje estándar de comunicación con bases de datos. Hablamos por tanto de un lenguaje normalizado que nos permite trabajar con cualquier tipo de lenguaje de programación (ASP, PHP, ..) en combinación con cualquier tipo de base de datos (MS Access, SQL Server, MySQL,...).

Aparte de esta universalidad, el SQL posee otras dos características muy apreciadas. Por una parte presenta una potencia y versatilidad notables que contrasta, por otra parte, con su accesibilidad de aprendizaje.

Existen diferentes forma de acceder a una BD mediante el lenguaje SQL, entre ellas destacamos las siguientes:

- **Mediante programas de aplicación con SQL embebido.** Las sentencias SQL están embebidas dentro del lenguaje anfitrión (lenguaje host). Se utilizan unas sentencias SQL para recuperar los datos. Mediante un precompilador estas sentencias se traducen a códigos ejecutables.
- **Las API (Application Programming Interface) de SQL.** En este caso, el programa se comunica con la BD mediante un conjunto de sentencias que constituyen un interfaz de programa de aplicación (API). Las sentencias embebidas SQL acceden a la BD a través de las llamadas al módulo API para recuperar los datos, sin necesidad de un precompilador como en el caso anterior.

8.1.6.5.- Base de datos MySQL.

MySQL es un sistema gestor de bases de datos relacionado con el cliente-servidor de SQL. MySQL incluye un servidor SQL, programas cliente para acceder al servidor, herramientas administrativas y una interfaz de programación para escribir sus propios programas.

Los orígenes de MySQL se remontan a 1979, con la herramienta de base de datos UNIREG creada por Michael Widenius para la empresa sueca TcX. En 1994 TcX comenzó a buscar un servidor SQL para empleado en el desarrollo de aplicaciones Web.

Probaron algunos servidores comerciales pero resultaron demasiado lentos para las inmensas tablas de TcX. También probaron con mSQL pero carecía de ciertas características necesarias para TcX. Por tanto, Widenius comenzó a desarrollar un nuevo servidor. La interfaz de programación fue explícitamente diseñada para que fuera similar a la empleada por mSQL, ya que estaban disponibles varias herramientas gratuitas para mSQL y con un mínimo esfuerzo podían adaptarse a MySQL.

En 1995, David Axmark de detron HB comenzó a animar a TcX para que hiciera público MySQL en Internet. MySQL fue entregada en 1996 en forma de distribuciones binarias para Linux y Solaris. Hoy en día, MySQL funciona en muchas más plataformas y está disponible tanto en forma dinámica como en código fuente.

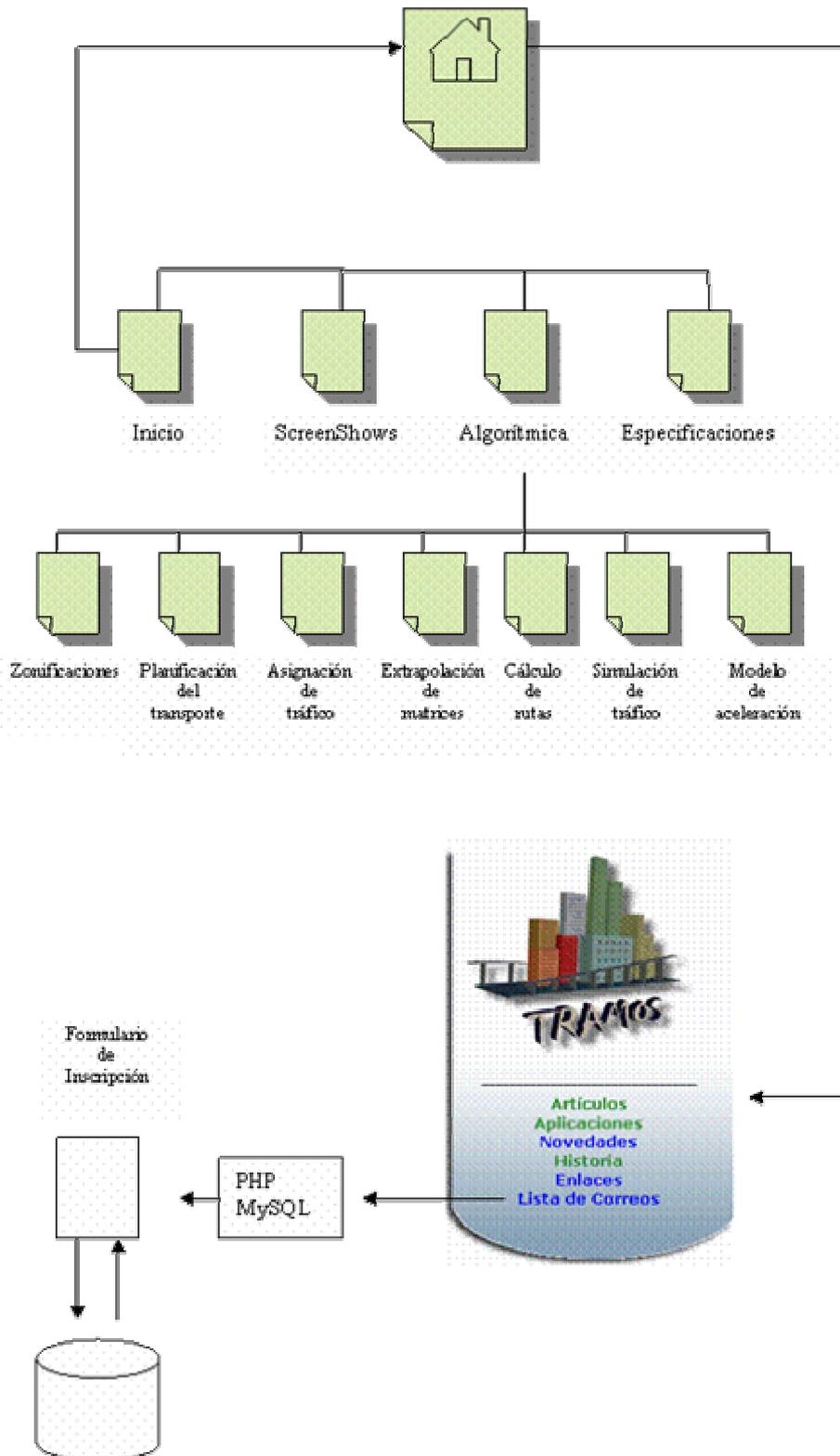
MySQL se sitúa en el sector de sistemas operativos gratuitos que se ejecutan en un potente, pero poco costoso, hardware, colocando un sustancial potencial de procesamiento y capacidad en manos de mucha gente y para una mayor variedad de sistemas.

Veamos algunas características atractivas que nos ofrece MySQL:

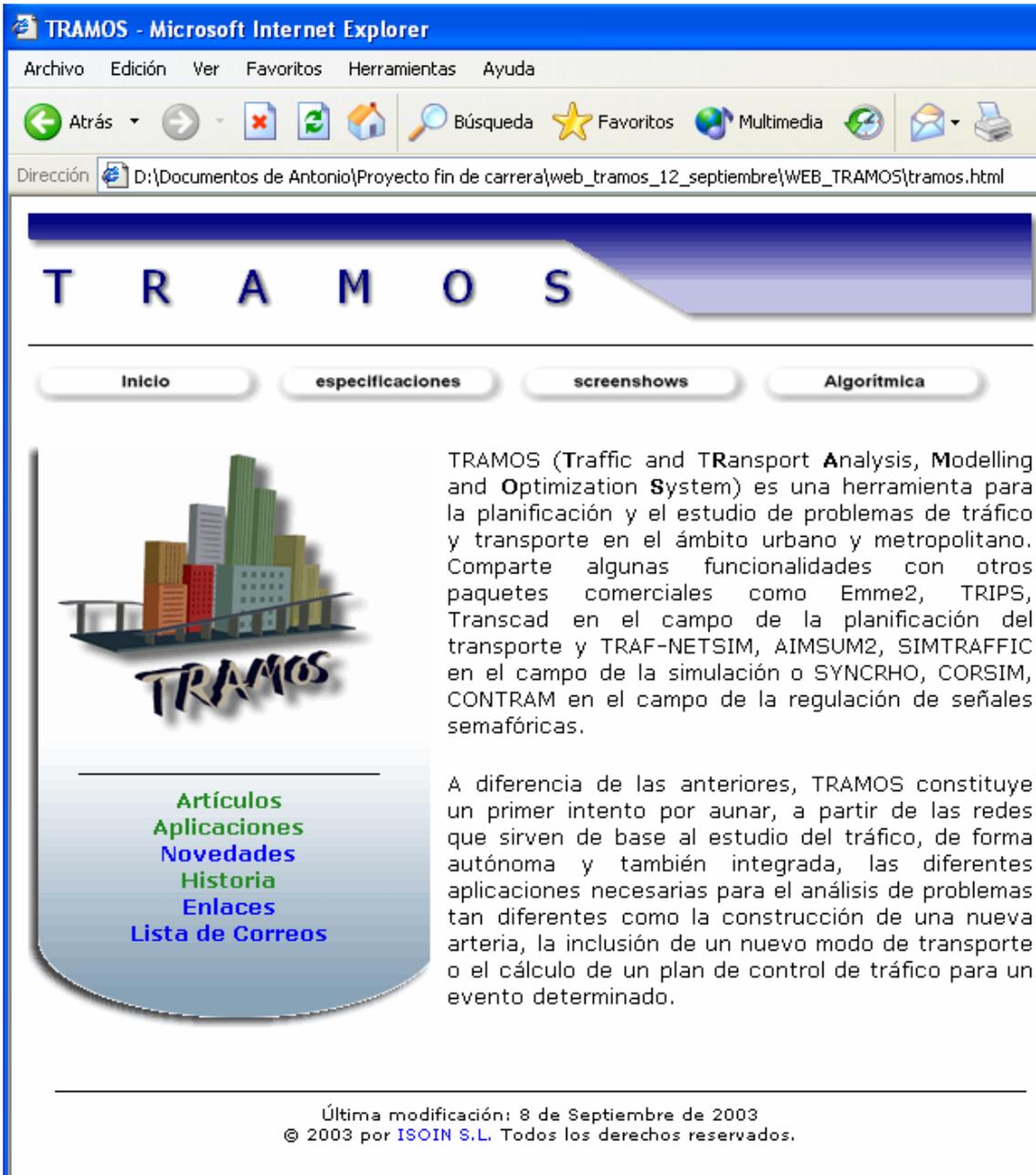
- **Velocidad.** MySQL es rápido. Los desarrolladores sostienen que MySQL es posiblemente la base de datos más rápida.
- **Facilidad de uso.** MySQL es un sistema de base de datos de alto rendimiento pero relativamente simple y es mucho menos complejo de configurar y administrar que sistemas más grandes.
- **Coste.** MySQL es gratuito para la mayoría de usos internos.
- **Capacidad de gestión de lenguajes de consulta.** MySQL comprende SQL (Structured Query Language- Lenguaje de Consulta Estructurado), el lenguaje elegido para todos los sistemas de BD modernos. También se puede acceder a MySQL empleando aplicaciones que admitan ODBC (Open Database Connectivity/Conectividad de Base de Datos), un protocolo de comunicación de bases de datos desarrollado por Microsoft.
- **Capacidad.** Pueden conectarse muchos clientes simultáneamente al servidor. Los clientes pueden utilizar varias BD simultáneamente. Pueden acceder de forma interactiva a MySQL empleando diferentes interfaces que le permiten introducir, consultar y visualizar resultados. Además, está disponible una amplia variedad de interfaces de programación para lenguajes como C, Perl, Java y PHP. Por tanto, tiene la posibilidad de elegir entre usar un software cliente pre-empaquetado o escribir sus propias aplicaciones a medida.
- **Conectividad y seguridad.** MySQL está completamente preparado para el trabajo en red y las BD pueden ser accedidas desde cualquier lugar de Internet, por lo que puede compartir sus datos con cualquiera, en cualquier parte. Pero MySQL dispone de control de acceso, de forma que aquellos que no deberían ver sus datos no los ven.
- **Portabilidad.** MySQL se ejecuta en muchas variantes de UNIX, así como en otros sistemas no-UNIX, como Windows. MySQL se ejecuta en hardware que va desde PC hasta servidores de alta capacidad.
- **Distribución abierta.** MySQL es fácil de obtener, simplemente usando el navegador Web.
- **Mantenimiento.** La comunidad MySQL, desarrolladores y no desarrolladores por igual, son muy participativos. Existe una lista de correo a la que cualquiera puede suscribirse y donde se cuelgan las preguntas. A menudo las respuestas aparecen en minutos. Cuando se detecta un error de programación generalmente los desarrolladores generan un parche disponible a través de Internet.

8.2.- Diseño de la Web.

A continuación se muestra el esquema representativo de la Web.



La página Web principal tiene el formato siguiente:



Se ordena en cuatro secciones principales que se pasan a describir a continuación:



En el menú inicio se describe la aplicación de forma genérica.

También se puede acceder al submenú siguiente:



- Artículos: artículos en donde se ha utilizado la aplicación TRAMOS.
- Aplicaciones: usos que puede haber tenido la herramienta TRAMOS.
- Novedades: Últimas noticias relacionadas con la aplicación.
- Historia: evolución de las versiones anteriores.
- Enlaces: links a páginas con información adicional.
- Lista de Correos: lugar donde se mantiene una base de datos de personas que previamente se han registrado y a las que se les envía información y todo tipo de documento. Esto se ha desarrollado en PHP y MySQL.

especificaciones

Con esta opción aparece una pantalla que describe las características técnicas de la aplicación. A continuación se muestra la página Web correspondiente:

ESPECIFICACIONES

- Herramientas de Planificación de transporte.
- Asignación multimodal.
- No existe limitación al número de líneas de autobus y otros modos de transporte público.
- No existe limitación en el número de zonas.
- No existe limitación al tamaño de las redes.
- Editor gráfico de redes.
- Importación de escenarios de Emme/2.
- Importación/exportación texto.
- Soporte para importación de archivos DXF.
- Digitalización de escenarios en pantalla.
- Amplio conjunto de informes gráficos

Esta página está todavía en proceso de desarrollo, pero su objetivo final será el de mostrar diversas pantallas de la aplicación TRAMOS.

Algorítmica

Con esta opción se obtiene una introducción a la simulación microscópica. A continuación se muestra la página Web correspondiente:



La simulación microscópica incorpora adelantamientos y cambio de carril así como aparcamientos. Se puede simular el comportamiento del tráfico de modo aleatorio, respetando porcentajes de giro o a partir de matrices O/D. EN este último caso los vehículos se comportan a nivel microscópico como se supone que lo hacen en los modelos macroscópicos de planificación.

En el apartado de simulación de tráfico, se ha desarrollado un modelo microscópico que aprovecha, si así se desea, información obtenida del modulo de asignación para obviar el proceso de introducción de porcentajes de giro en intersecciones.

TRAMOS incorpora diferentes algoritmos, de acuerdo a los estándares admitidos en el mundo del transporte como mejores prácticas.

En el campo de la planificación del transporte se utilizan algoritmos de asignación unimodal y multimodal con demanda fija. En la actualidad se trabaja en la incorporación de modelos con demanda variable. TRAMOS incorpora la posibilidad de calibración de matrices O/D a partir de datos de conteo. Incluye un novedoso algoritmo para asignar el tráfico imponiendo condiciones de contorno sobre los arcos que se consideren oportunos (observaciones de actuales de trafico).

Se dispone de un modulo para la extrapolación temporal de matrices O/D, que permiten actualizar antiguas matrices al flujo actual observado sobre la red.

Dentro de la sección algorítmica se puede acceder también a los siguientes contenidos:

- Zonificaciones
- Planificación del Transporte
- Asignación de tráfico
- Extrapolación de matrices
- Cálculo de rutas
- Simulación de tráfico
- Modelo de aceleración