World Wide Web
Navegar Internet con W^3 .

Director
Lic. Javier F. Díaz

Alumnos
Andrea M. Keiliff
Gabriel Bonnefon Gil

A nuestros padres.

INDICE

PREFACIO

El trabajo de grado, responde al interes por presentar las bases fundacionales de una tecnología que aún no es bien conocida y que cuenta con mucha propaganda.

Si bien en la carrera de informática la tecnología cliente-servidor y el uso de recursos de redes para acceder a información multimedial no es atendido, el auge de Internet nos motivó a investigar en estas áreas y a realizar una recopilación selectiva de información, facilitando la incorporación de esta temática a los cursos de grado. Por tratarse de un área tan actual se fueron tomando desiciones para acotar una versión completamente consistente y que ilustre todas las posibilidades de la tecnología actual.

En principio se describen brevemente (pags. 4 a 8) los conceptos en que se basa el servicio, usualmente denominado World Wide Web.

Se dan claras definiciones y referencias de los estandares HTML y HTTP (pags. 9 a 30).

Se analizan las características de los servidores pioneros y que aún tienen gran parte del parque instalado (pags. 31 a 65).

El capítulo de browsers describe brevemente los clientes más difundidos tanto como sus características particulares (pags. 66 a 71).

Los tópicos más avanzados (seguridad, robots y proxies) son descriptos en detalle en las páginas 73 a 89.

Por último (pags. 90 a 94) se ilustran los conceptos antes descriptos en los desarrollos de distintos sitios de U.N.L.P. y una herramienta automatizada para recuperar páginas de distintos sitios.

En principio los desarrollos de la última parte constituyeron la real motivación del trabajo y permitieron consolidar los conceptos teóricos y afianzar los distintos temas.

Nuestra propuesta presenta las características de un primer trabajo en un área nueva y posibilita que a partir de él se profundize en distintos aspectos como :

- índices de búsqueda.
- aplicaciones en Java.
- generación de gateways a distintos servicios.


www

1. ¿ QUÉ ES WORLD WIDE WEB ?

WWW es un ambiente global, en el cual toda la información (texto, imágenes, audio, vídeo, servicios computacionales) que está disponible en Internet puede ser accedida en forma consistente y simple usando un conjunto de convenciones de acceso y de nombres.

Se puede conectar a cualquier servidor sobre la red simplemente clickeando sobre una selección (palabras subrayadas) o bien entrando la dirección IP específica. El acceso a distintos servicios de información no requieren trabajo extra.

Las características más importantes que hacen popular a WWW son las siguientes :

- fácil de usar.
- fácil ir de un lugar a otro.
- combina cualquier tipo de datos, texto, imágenes, sonido, movimiento, etc.
- hay muchas herramientas que facilitan su uso, semejante a los browsers.
- es fácil publicar información.

WWW fue desarrollado en CERN (Ginebra, Suiza). La tarea de desarrollo comenzó en 1989, la parte inicial fue el desarrollo del protocolo cliente/servidor HTTP, la definición de un servidor simple y una librería llamada wwwlib. En 1992 CERN puso en dominio público el soft. Al WWW también se lo denomina Web.

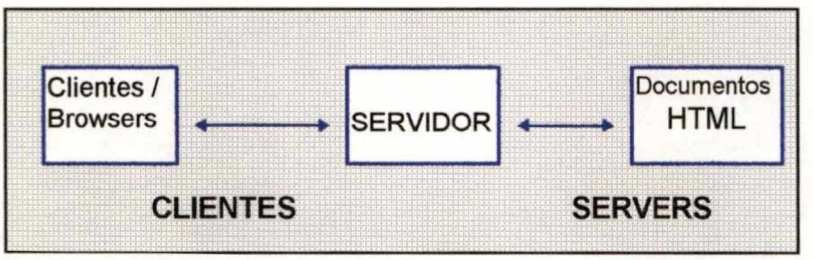
En Web no existe un control central, funciona y crece porque la gente produce documentos y servidores en la forma pautada (convenida), así cualquiera puede publicar algo y cualquiera lo puede leer ,esto es parte de la ética Web. Todos los servidores Web deben usar el mismo protocolo, HTTP es rápido, no está orientado a conexión, es un mecanismo de transporte usado para comunicarse en el ambiente Web; httpd, o daemons http, son la base de un servidor Web recibiendo mensajes y retornando datos como respuesta. El direccionamiento usado para localizar los recursos también debe ser común a todos, se hace por medio de URLs (Universal Resource Location) [10]. Por último todos los browsers deben usar el mismo lenguaje básico, por convención es HTML [9], HyperText Markup Languaje. En Web se soporta la negociación de formato de los datos a comunicar entre el servidor y el cliente [2].

1.1 COMO FUNCIONA WEB ?

Según [3], Web tiene una arquitectura simple, clientes que envían mensajes a servidores Web, el servidor es responsable de responder con la

información requerida al cliente (browsers), quien es responsable de presentar el documento al usuario.

Son mensajes cortos, el cliente envía un requerimiento al servidor, este responde lo que fue requerido y la conexión finaliza. Esto simplifica las comunicaciones, pero hace dificultoso manejar transacciones de larga vida, el servidor no recuerda los mensajes que recibió antes.



Esquema básico de cliente-servidor para HTTP.

1.2 ¿ PORQUÉ ES IMPORTANTE ?

Las siguientes características lo hacen importante:

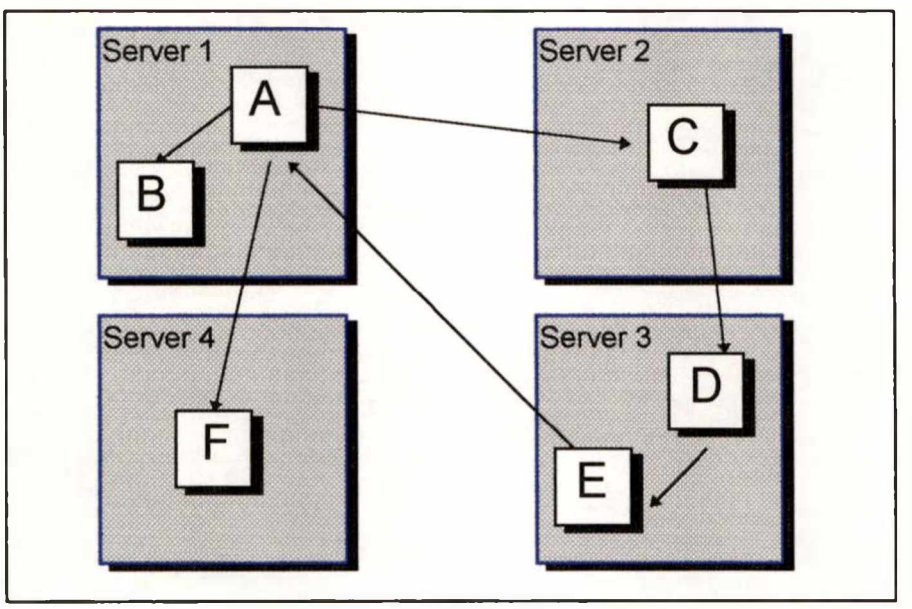
- La facilidad con que entrega información en diferentes formatos, a un ancho rango de plataformas.
- Es un potencial ambiente cliente-sevidor. Los browsers Web son clientes testeados sobre los cuales se construyen muchas aplicaciones cliente-servidor.
- Simplifica el acceso a Internet. Permite acceso a muchos de los recursos de Internet, provee interface gráfica, soporte multimedia, está basado en convenciones y estándares entonces resulta fácil compartir cosas, facilita el acceso a información tanto como proveerla.
- El crecimiento rápido en cantidad de información disponible. El volumen de información disponible es muy grande, las principales compañias del mundo, universidades y organizaciones de investigación están sobre Internet.

1.3 CONCEPTOS BÁSICOS

1.3.1 HYPERLINK

Los links en los documentos HTML [8] son las palabras que aparecen subrayadas (característica básica), estos lazos se activan presionando la tecla <ENTER> (trabajando con browsers no gráficos) o haciendo un click con el mouse sobre él. Al activar un link podemos ir a otra sección del mismo documento, a otro documento sobre el mismo servidor, a otro documento de cualquier servidor sobre Internet, o a una sección de otro documento sobre otro servidor. Al llegar a otro documento por medio de un link activado nos encontramos con que el nuevo documento tiene también otros lazos.

El siguiente gráfico ilustra el concepto :



Servidor 1:

El documento A tiene links apuntando al documento B del mismo servidor, al documento F del servidor 4 y al documento C del servidor 2.
El documento B es alcanzado por un link desde el documento A.

Servidor 2:

El documento C tiene un link hacia el documento D sobre el servidor 3.
El documento C es alcanzado por un link desde el documento A en el servidor 1.

Servidor 3:

El documento D tiene un link hacia el documento E sobre el mismo servidor .
El documento D es alcanzado por un link desde el documento C en el servidor 2.
El documento E tiene un link hacia el documento A sobre el servidor 1.
El documento E es alcanzado por un link desde el documento D en el mismo servidor.

Servidor 4:

El documento F se alcanza por un link desde el documento A en el servidor 1.

1.3.2 URL [10]

Cada lazo en un documento tiene 2 componentes, el anchor (texto o gráfico) que es donde se clickea y entonces se dispara el hyperlink y el otro elemento es el URL que dice que hacer cuando el hyperlink es activado. El URL indica el protocolo usado para alcanzar el server destino, el nombre del servidor donde se encuentra el documento, el directorio y el nombre del documento requerido, ej:

http://www.unlp.edu.ar/secyt/notas.html

este URL indica que va a usar el protocolo HTTP [12] para ubicar el servidor "www.unlp.edu.ar" y en el directorio "/secyt" buscar el documento "notas.html".

Cada archivo en Internet es únicamente direccionable por su URL, el concepto de URL también soporta otros protocolos importantes como FTP, Gopher,Telnet, Wais.

El ejemplo anteriormente visto es de un URL absoluto por que indica completamente como localizar al archivo, también existen URL relativos, es una forma corta de referenciar documentos sobre el mismo servidor Web, se necesita indicar el directorio relativo a donde atiende el servidor y el nombre del documento requerido, ej :

/secyt/notas.html

Para mayor detalles y ejemplos, también ver [4] en el capítulo 3.

1.3.3 LENGUAJE DE TAGS HTML [8] [9]

En WWW los documentos son texto ASCII marcado con tags HTML, tales documentos son llamados documentos HTML y generalmente tiene la extensión *.html* . El lenguaje de tags es usado para describir la estructura del documento y la información de lazos, pero no el exacto formateo, esto queda a criterio del browser.

El lenguaje de tags HTML esta basado en SGML (Standard Generalized Markup Language) que es un standard ISO usado para marcar documentos para publicación impresa u online.

Este punto se desarrolla más extensamente en el próximo capítulo.

HTML

2. HTML

2.1 INTRODUCCIÓN

El lenguaje de tags para hypertexto (HTML) es un formato de datos simple usado para crear documentos de hypertexto que son portables a cualquier plataforma. Los documentos HTML son documentos SGML con semántica genérica que es apropiada para representar información en un ancho rango de dominios.

HTML es usado por WWW desde 1990, en un principio la documentación era informal, la primer especificación formal de HTML fue en junio de 1994, en noviembre de 1995 se publico la especificación formal de HTML 2.0 agregando a lo anterior gran cantidad de características nuevas, y ya esta en draft la especificación de HTML 3.0, siempre compatible con las versiones anteriores.

HTML es una aplicación de SGML (ISO Standart 8879:1986, "Information Processing Text y Office Systems; Standart Generalized Markup Languaje"). La definición del tipo de documento HTML (DTD) es una definición formal de la sintaxis HTML en términos de SGML.

También se define HTML como un Internet Media Type [IMEDIA] y un Content-Type MIME llamado " text/html ", y así queda definida la semántica de la sintaxis HTML y cómo esta sintaxis será interpretada por los agentes usuarios. El Content-Type acepta el parámetro "versión" que será de gran ayuda en el futuro para problemas de compatibilidad, este indicará la versión de HTML usada en el documento.

2.2 HTML 3.0 [9]

La especificación de HTML 3.0 [9] se construye sobre HTML 2.0 [8] y da compatibilidad completa con esta versión. Las tablas han sido una de las principales características requeridas junto a otras como texto flotando alrededor de imágenes o correr operaciones matemáticas.

HTML 3.0 introduce un nuevo elemento, el FIG, para imágenes en línea, este provee manejo de zonas sensitivas del lado del cliente, además el texto puede flotar alrededor de las figuras y se puede controlar cuando saltar al comienzo de un nuevo elemento.

Incluye también soporte para ecuaciones y fórmulas relativamente de poca complejidad para el browser. El formato propuesto esta fuertemente influenciado por TeX. Semejante a tablas, el formato usado es un estilo liviano de tags,

bastante simple de tipear a mano, aunque es aún más fácil desde un editor WYSIWYG HTML 3.0 (por ej : Nestcape Gold, HotMetal).

Además se agregan listas clientizadas, con un fino control del posicionado, tabulación horizontal y alineación de los headers (encabezamientos o títulos) y párrafos de texto.

Los forms han sido extendidos a soportar menúes de selección gráfica, nuevos tipos de campos han sido incluidos por ejemplo : controles de rango, carga de un archivo, entre otros. El scripting del lado del cliente es posible por medio del atributo SCRIPT. Los forms y las tablas hacen una combinación poderosa ofreciendo rica variedad para crear interfaces a sistemas de información remoto.

Agregando más características de presentación HTML 3.0 esta diseñado para usar style sheet (no es obligación su uso), esto da más control sobre la presentación del documento.

2.3 CARACTERÍSTICAS DE ELEMENTOS HTML

La mayoría de los elementos html son identificados en el documento como un tag inicial al cual le sigue el nombre del elemento y atributos, luego el contenido, seguido por el tag de cierre.

Los tag de inicio están delimitados por "<" y ">", y los tags de cierre por "</" y ">". Por ejemplo :

<H1>Este es el título</H1>

Y esto es un párrafo <P>

Algunos elementos aparecen solo con un tag inicial, por ejemplo
, además los tags finales de algunos elementos como por ejemplo <LI> <DT> <DL> pueden ser omitidos dado que la posición del tag final esta claramente implícito por el contexto.

El contenido de un elemento es una secuencia de caracteres y elementos anidados, algunos elementos tales como anchores no pueden ser anidados. Anchores y caracteres resaltados pueden ser puestos dentro de otros constructores. El modelo de contenido de un tag define la sintaxis permitida para el contenido.

- **NOMBRE DE LOS TAG :**

El nombre del elemento seguido inmediatamente por los delimitadores de tag inicial. Un nombre de elemento consiste de una letra seguida por 72 otras

letras, números, o períodos. Los nombres no son sensitivos a mayúsculas o minúsculas, el límite de 72 caracteres esta dado por el parámetro NAMELEN en la declaración SGML para html 3.0.

- **ATRIBUTOS DE UN TAG :**

Están en el tag inicial, luego del nombre y un espacio en blanco y antes del ">". Un atributo consiste de un nombre de atributo, un signo "=" y un valor (aunque algunos atributos pueden ser un valor). Un espacio en blanco es permitido alrededor del signo igual.

El valor del atributo puede ser:

- un string literal, delimitado por comillas simples o dobles.
- un nombre token (una secuencia de caracteres, dígitos o períodos).

- **COMENTARIOS :**

Para incluir comentarios en un documento HTML se debe escribir :

<!--.....--> El texto dentro de estos delimitadores será ignorado, los comentarios no pueden ser anidados.

2.4 ESTRUCTURA DE UN DOCUMENTO HTML

Los documentos HTML están escritos en formato de texto plano, y pueden ser creados usando cualquier editor de texto o editores HTML. Un documento tiene formalmente la siguiente estructura :

<HTML>

<HEAD> Elementos del Head

<BODY> Elementos del Body

</HTML>

<HTML> Marca el comienzo del documento, puede ser omitido.

2.4.1 <HEAD> Y ELEMENTOS RELACIONADOS

El elemento HEAD no tiene atributos y el tag inicial y final pueden ser omitidos. La información del HEAD corresponde a la parte inicial de un mensaje memo o mail. Este describe propiedades del documento tales como el título, la barra del HEAD del documento y meta-información adicional.

El elemento <TITLE> debe estar siempre, ·en efecto el documento HTML mínimo consiste solo del elemento TITLE.

En el elemento HEAD solo ciertos elementos son permitidos, son los siguientes en orden alfabéticos :

BASE : permite que el URL [10] de este documento sea recordado en situaciones en las cuales leerá fuera de contexto

<base href = "http://www.acme.com/docs/mydoc.html>
<img scr = "images/me.gif>

lo resuelve como http://acme.com/docs/images/me.gif

ISINDEX : informa al agente usuario que el documento es un documento indexado. El documento puede ser consultado con una búsqueda por palabra clave <isindex> -
<isindex href = "xxxxx.db" PROMPT= Ingrese su nombre ">

El URL usado para procesar la consulta puede ser sobreescrito por el atributo HREF. También se usa el atributo PROMPT para cambiar el prompt default del agente usuario.

LINK : indica relación entre el documento y algún otro objeto, pueden existir atributos :

REL define la relación definida por el link
REV define la relación inversa. Usado para la dirección e-mail del autor.

Un uso importante del elemento LINK es definir barras de menú con botones de navegación o un mecanismo equivalente tal como menú de ítems.

REL= Home, el link referencia a una HomePage o al tope de alguna jerarquía
REL= Toc, referencia a un documento sirviendo como tabla de contenidos
REL= Index, provee un index para el documento.
REL= Glossary, referencia a un documento con glosario determinado también Copyrigth / Up / Next / Previous / Help / Bookmarks / Banner.

<LINK REL= Banner HREF = banner.html> para un banner compartido por varios documentos entonces así lo cachea en forma separada del documento.

META : este es usado en el HEAD para embeber información no definida por otros elementos HTML. Tal información puede ser usada por clientes/servers para usar en identificación, indexación o catalogación. Tiene 3 atributos :

Name : Usado para especificar autor, fecha de publicación ,etc

Content : Usado para dar un valor a una propiedad nombrada.

Http-Equiv : Este atributo liga el elemento a una respuesta HTTP del header

Si la semántica de la respuesta HTTP header nombrada por este atributo es conocido, entonces los contenidos pueden ser basados sobre mapeo sintáctico bien definido.

Si el documento tiene :

<Meta Http-Equiv = "Expires" CONTENT = "04 Dec 1993">
<Meta Http-Equiv = "Keywords" CONTENT ="Nanotecnología, Bioquímica">
<Meta Http-Equiv = "Reply-to" CONTENT = "wwwgab@isis.unlp.edu.ar">

El server incluirá la siguiente respuesta al header :

Expires = 04 Dec 1993
Keywords = Nanotecnología, Bioquímica
Reply-to = wwwgab@isis.unlp.edu.ar

Cuando HTTP-EQUIV está ausente el server no genera respuesta para meta-información

RANGE : es usado para marcar un rango del documento (útil para que regiones iluminadas hagan maching por criterios de búsqueda).

Soporta los siguientes atributos :

ID : un identificador SGML usado para nombrar el elemento rango

CLASS : un string usado para subclases del elemento rango

FROM : refiere a un identificador SGML para un elemento en el cuerpo del documento, inicio del rango marcado.

UNTIL : refiere a un identificador SGML para un elemento en el cuerpo del documento, fin del rango marcado.

STYLE : el elemento style provee una manera de incluir una forma de mostrar información usando un estilo de notación especificado. La información en el elemento Style sobreescribe los default del cliente.

TITLE : todo documento HTML debe contener un elemento TITLE. El título debe identificar el contenido del documento en un contexto global y debe ser usado en una lista de historia. A diferencia de los Headings, los Titles no son normalmente desplegados en el texto de los documentos.

Debe estar siempre dentro de la parte HEAD del documento HTML y no puede contener anchores, <P>, etc. Solo debe existir un title en el documento, la longitud es limitada, si es largo, su longitud puede ser truncada en algunas aplicaciones.

BODY : dentro de este elemento, se puede estructurar texto en párrafos, listas, también resaltar frases y crear links a otros documentos. Tiene los siguientes atributos, donde todos son opcionales : ID, LANG, CLASS, BACKGROUND no es necesario que exista este tag a menos que se quieran incluir estos atributos.

El cuerpo del documento está compuesto por nada o algunos de los siguientes elementos :

- Div - Usado para jerarquía y barreras estáticas.
- Headings - títulos de distinta importancia.
- bloque de elementos como párrafos, listas, forms, tablas, figuras, etc.
- Horizontal rules.
- Texto y caracteres.

2.4.2 <BODY> Y ELEMENTOS RELACIONADOS

Los siguientes elementos pueden formar parte del cuerpo del documento, esta es una breve descripción :

BANNER : este elemento es usado para logo de corporaciones, ayuda de navegación, disclaimers y otra información la cual no puede ser scrolleadas con el resto del documento. Este elemento va al principio del Body. Esto también es una alternativa a usar el elemento link en el head del documento para referenciar un banner definido externamente. Los atributos son ID, LANG, CLASS.

HEADING : HTML define 6 niveles de títulos. Un elemento heading implica el cambio de todos los fonts. Salta de párrafos antes y después, también salta cualquier espacio en blanco. Los tags son <H1>, <H2>, <H3>, <H4>, <H5>, <H6>, donde H1 es el de mayor importancia y H6 el de menor relevancia.

PARRAFOS : el elemento <P> es usado para definir un párrafo. La forma en que será mostrado no está definida, solo delimita párrafos.

SALTO DE LINEA : el elemento
 es usado para forzar un salto de línea.

HIPERLINKS : el elemento anchor <A> es usado para definir un link hipertexto. El texto entre el tag inicial y el tag de cierre es la referencia del link, puede ser texto o bien una imagen. Al clickear sobre la referencia, invocamos al documento indicado en el atributo HREF. También se puede usar el atributo NAME para marcar un determinado punto dentro de un documento.

Ej : <A HREF= "http://www.unlp.edu.ar/index.html">Universidad Nacional de la Plata</A>

Este ejemplo nos está referenciando a un URL [10] completo

<A HREF= "lp_ciudad.html">Ciudad de La Plata</A>

Este ejemplo nos está referenciando un URL relativo al servidor local del documento.

<A HREF= "lp_ciudad.html#mapa">Mapa de La Plata</A>

Con el tag anterior estamos especificando que al clickear sobre el link invocaremos al archivo "lp_ciudad.html" y dentro de este no ubicaremos en el punto referenciado por "mapa" como indica el tag de abajo.

<A NAME= "mapa">Mapa</A>

ELEMENTOS A NIVEL CARACTER : estos son usados para especificar un significado estructural, o bien una apariencia física dentro del texto marcado sin causar un salto de línea. Tienen un tag inicial y otro final, se colocan solo alrededor de los caracteres afectados.

<EM> : apariencia Itálica
<CITE> : apariencia Itálica

<STRONG> : apariencia Bold

<CODE> : letras monoespaciadas

ELEMENTOS A NIVEL ESTILO : estos pueden ser anidados unos con otros, los browsers deben poder combinar distintos tipos de resaltados como se necesite.

<B> : Bold

<Y> : Italic

<TT> : Teletipo

<U> : Subrayado

<S> : Tachado

<BIG> : Texto en tamaño mas grande

<SMALL> : Texto en tamaño mas pequeño

<SUB> : Subíndice

<SUP> : Superíndice

IMÁGENES : el tag <IMG> es usado para mostrar gráficos en línea (íconos o gráficos). Los browsers que no pueden mostrar imágenes, ignoran este elemento a menos que el atributo ALT esté presente.

El atributo SRC es usado para indicar la locación de la imagen, WIDTH y HEIGHT indican el tamaño que ocupará la imagen, también se puede indicar la ubicación dentro de la página con el atributo ALIGN (top, middle, bottom, left, right, absmiddle). Con ALT se indica un texto alternativo que se muestra en browsers no gráficos. Por último el atributo ISMAP identifica la imagen como un mapa sensitivo, esto es un mapa gráfico y tiene la ventaja de ser sensitivo a las zonas donde se clickea, llamando a distintos URLs según esté indicado en el servidor.

2.4.2.1 LISTAS

2.4.2.1.1 LISTAS SIN ORDEN

Las listas sin orden son típicamente puntos para denotar ítems. HTML 3.0 brinda la posibilidad de personalizar los puntos (bullets), sacar los puntos, organizarlos horizontal o verticalmente para listas multicolumnas.

Debe estar el tag de apertura de la lista <UL> seguido por un encabezado opcional <LH></LH> y luego las listas de ítems cada una con <LI>.

Atributos del elemento <UL>

ID, LANG,CLASS, CLEAR, SRC, WRAP, COMPACT, LEFT, ALL, RIGHT

Atributos del elemento <LH>

ID, LANG, CLASS

El tipo de contenido de <LI> puede ser texto, imágenes, tablas, etc.

Atributos del elemento <LI>

ID, LANG, CLASS, CLEAR, SRC, MD, DINGBAT, LEFT, RIGHT, ALL

2.4.2.1.2 LISTAS CON ORDEN

Las listas con orden tiene la particularidad de numerar los ítems. HTML 3.0 le permite controlar el número de secuencia. El estilo de numerado está asociado con el style sheet.

El tag de inicio de la lista es <OL> y el de cierre </OL>. El tag de inicio está seguido por un <LH> opcional y luego vienen los <LI> para referenciar a los ítems.

Atributos del elemento <OL>

ID, LANG, CLASS, CLEAR, CONTINUE, SEQNUM, COMPACT, LEFT, RIGHT, ALL.

2.4.2.1.3 LISTAS DE DEFINICIÓN

Una lista de definición es una lista de términos y definiciones correspondientes. Están formateados generalmente con el término alineado a la izquierda y la definición en la próxima línea a la derecha, indentado respecto del término.

El tag inicial es <DL> y el de cierre es </DL> seguido por un tag opcional <LH> y luego viene pares de <DT><DD>

Atributos del elemento <DL>

ID, LANG, CLASS, CLEAR, LEFT, RIGHT, ALL, COMPACT

<DT> - Nombre del término (para la lista de definición).

El tag <DT> especifica el nombre del término y puede haber varios términos por elemento DD

Los nombres de los términos son restrictos a tags de nivel carácter solo incluyen énfasis, imágenes en línea, y notas al pie. Los tags de párrafos y otros equivalentes de bloques tales como headers no son permitidos.

Atributos del elemento <DT>

ID, LANG, CLASS, CLEAR.

<DD> Definición del término.

El tag <DD> especifica la definición del término y permite uno o más elementos <DT>.

Dentro del tag DD puede haber párrafos, listas, texto preformateado, forms, tablas e imágenes. Los headers no son permitidos.

Atributos del elemento <DD> son ID, CLASS, LANG, CLEAR.

2.4.2.2 TABLAS

El modelo de tabla HTML ha sido elegido por su simplicidad y flexibilidad. Por default el tamaño de la tabla se acomoda automáticamente de acuerdo al tamaño del contenido de las celdas y el tamaño de la ventana en que se presenta. El atributo COLSPEC puede ser usado cuando se necesita controlar el ancho de las columnas seteandolo explícitamente o bien en porcentajes relativos.

La tabla se forma por un título opcional seguido por una o más filas, cada fila está formada por una o más celdas con título y datos propios. Las celdas pueden ser mezcladas entre las filas y columnas e incluyen atributos que ayudan a su apariencia.

El modelo provee poco control sobre su apariencia, pueden contener gran variedad de tipo de dato tales como headers, listas, párrafos, forms, imágenes, texto y aún tablas anidadas.

Cuando una tabla esta a la derecha o izquierda, los elementos siguientes flotarán alrededor de la tabla según el lugar que exista. Este comportamiento queda anulado con el atributo NOFLOW o cuando la tabla está alineada al centro o justificada.

Atributos del elemento <TABLE>

ID, LANG, CLASS, CLEAR, NOFLOW, ALIGN, COLSPEC, DP, WIDTH, BORDER, NOWRAP

2.4.2.2.1 FILA DE LA TABLA

El elemento <TR> actúa como contenedor para una fila de celdas de la tabla con los elementos <TH> <TD>. Se puede setear el default de la alineación vertical y horizontal de los datos de las celdas para la fila.

Atributos permitidos ID, LANG, CLASS, ALIGN, DP, VALIGN, NOWRAP.

2.4.2.2.2 CELDAS DE LA TABLA

Los elementos <TH>, <TD> son usados para las celdas de la tabla, <TH> es usado para celdas de título, y <TD> para datos. Esto distinción da al browser una seña para el mostrado de tales celdas.

La alineación horizontal y vertical de los contenidos de la celda son determinados por los atributos ALIGN y VALIGN respectivamente.

Los atributos permitidos son ID, LANG, CLASS, COLSPAN, ROWSPAN, ALIGN, DP, VALIGN, NOWRAP.

2.4.2.3 TEXTO PREFORMATEADO

El texto preformateado entre el tag inicial y el de cierre es mostrado usando un font fijo. Los espacios en blanco son tratados literalmente.

 - causa un salto al principio de la línea siguiente.

<P> - debe ser evitado.

Pueden ser usado los anchores y resaltados, forms.

Elementos de bloque como headers, listas, imágenes, tablas, deben ser evitados .

Los atributos son ID, CLASS, LANG, CLEAR, WIDTH

2.4.2.4 ELEMENTO FORM

El elemento FORMS puede ser usado para cuestionarios, reservaciones de hoteles, ordenes, entrada de datos y otra gran variedad de aplicaciones. El FORM es especificado como parte del documento HTML. El usuario completa el FORM y luego lo submite, el browser envía los contenidos del form al servidor. Los FORMS son creados poniendo campos de entrada de datos en párrafos, textos preformateados, listas y tablas. Esto da una considerable flexibilidad en el diseño del FORM.

El form esta encerrado por los tags <FORM> </FORM>, puede haber varios forms en el mismo documento HTML, pero no pueden estar anidados. El browser es el responsable de manejar los datos entrados, no provee soporte para validación. Los contenidos submitidos del form consisten de una lista de pares nombre-valor donde los nombres están dados por el atributo NAME de los campos en el FORM, cualquier campo con valor nulo es omitido cuando se submite el form.

Atributos del elemento FORM :

ACTION : es un URL que especifica la locación a donde los valores del form serán enviados solicitando una respuesta. Si el atributo está ausente el mismo documento es asumido. La manera de submitir los datos varia con el protocolo de acceso al URL, y con los valores de los atributos METHOD y ENCTYPE.

METHOD : este especifica variación en el protocolo usado para enviar los datos, se usa GET por default o bien el POST. Este atributo indica al browser los métodos que el servidor soporta.

ENCTYPE : este atributo especifica el contenido tipo MIME a ser usado para codificar los datos. El default es " aplication / x-www-urlencoded ".

SCRIPT : puede ser usado para dar un URI para un script. El lenguaje para escribir scripts y la interface con el browser no es parte de HTML.

2.4.2.4.1 INPUT

El campo INPUT es usado para una gran variedad de diferentes clases de entradas, el atributo TYPE determina que tipo de campo es :

TYPE=text es texto en una línea, string cortos.

TYPE=password es equivalente a text pero los caracteres son mostrados con asterisco.

TYPE=checkbox tiene dos estados, seleccionado y no seleccionado, necesita los atributos name y value.

TYPE=radio puede tomar un solo valor entre varios, todos los botones radio del mismo grupo tienen el mismo nombre, necesita los atributos name y value.

TYPE=file permite al usuario ligar un archivo para que sea submitido junto al contenido del form. El atributo ACCEPT debe estar presente para indicar el tipo de archivo que se puede enviar.

TYPE=hidden este campo no esta presente para el usuario pero será enviado cuando se envíe el form.

TYPE=submit este botón cuando es presionado submite el form. Este campo no es enviado, salvo si el atributo NAME está presente.

TYPE=image actúa semejante al botón submit, pero incluye la ubicación donde el usuario clickeo sobre la imagen.

TYPE=reset cuando este botón es presionado, los campos del form toman sus valores iniciales.

Atributos del elemento INPUT :

ID, LANG, CLASS

TYPE define al tipo de campo.

NAME define el nombre del campo.

VALUE valor con el que es inicializado el campo.

DISABLE el campo se muestra pero no puede ser modificado.

ERROR indica que se despliegue un mensaje de error cuando el valor no corresponda.

CHECKED indica que campos RADIO o CHECKBOX están seleccionados al inicio.

SIZE ancho del texto.

MAXLENGTH el número máximo de caracteres para un campo texto o password.

ACCEPT contenidos de tipo MIME permitidos para un archivo.

SRC el URI para una imagen.

2.4.2.4.2 TEXTAREA

Este elemento es usado para entrar más de una línea de texto en un form. La sintaxis es la siguiente:

<TEXTAREA NAME="EJEMPLO" ROWS=64 COLS=6>
.....
.....
</TEXTAREA>

Los atributos son los siguientes : ID, LANG, CLASS, NAME, ROWS, COLS, DISABLED, ERROR, ALIGN.

2.4.2.4.3 SELECT

Es usado para menú de simple o múltiple elección. Esto es generalmente mostrado como un menú pop-up o drop-down y ofrece una alternativa más compacta que usar botones radio o checkbox.

Para un menú de simple elección si nada esta marcado la primer opción aparece marcada como seleccionada, como esto no conviene para menú múltiple elección se recomienda marcar OPTION SELECTED alguna de las opciones a mostrar.

Menú Gráfico : HTML 3.0 extiende el elemento SELECT para soportar menúes gráficos. Permite especificar una imagen para el elemento select y marcar zonas para cada elemento OPTION, de esta manera el mismo menú puede ser mostrado como texto para browsers no gráficos. La imagen es especificada de la misma manera que para IMG especificado WIDTH y HEIGHT, y las zonas marcadas para OPTION son indicadas por SHAPE.

Atributos admitidos : ID, LANG, CLASS, NAME, MULTIPLE, DISABLED, ERROR, VALUE, SELECTED, SHAPE.

PROTOCOLO HTTP

3. PROTOCOLO HTTP

El Hypertext Transfer Protocol (HTTP) [12] es un protocolo a nivel aplicación con la agilidad y la velocidad necesaria para sistemas de información de hypermedia distribuidos. Este es un protocolo orientado a objetos, genérico, el cual puede ser usado para muchas tareas, tales como name server y sistema de manejo de objetos distribuidos, a través de la extensión de sus métodos de requerimientos (comandos). Una característica saliente de HTTP es el tipo de representación de los datos, permitiendo a los sistemas ser construidos independientemente de los datos a ser transferidos.

HTTP ha sido usado por World Wide Web desde 1990.

3.1 PROPÓSITO

Los sistemas de información prácticos requieren más funcionalidad que una simple recuperación, incluyendo búsqueda, actualización y anotación. HTTP permite un conjunto de métodos de apertura y cierre a ser usados para indicar el propósito de un requerimiento. Este se construye sobre una disciplina de referencia provista por Uniform Resource Identifier (URI), como una locación (URL [10]) o nombre (URN), para indicar el recurso sobre el cual un método será aplicado. Los mensajes son pasados en formato similar al usado por Internet Mail y Multiproposit Internet Mail Extension (MIME [11]).

HTTP también es usado como un protocolo genérico para comunicación entre browsers y gateways proxies para otros protocolos de Internet tales como SMTP, NNTP, FTP, Gopher y WAIS permitiendo acceso de hypermedia básico para recursos disponibles desde diversas aplicaciones y simplificando la implementación del browser.

3.2 TERMINOLOGÍA

Brevemente describiremos los términos usados en una comunicación HTTP :

- **conexión :** un circuito virtual en la capa transparente establecida entre dos programa de aplicación con el propósito de comunicarse.

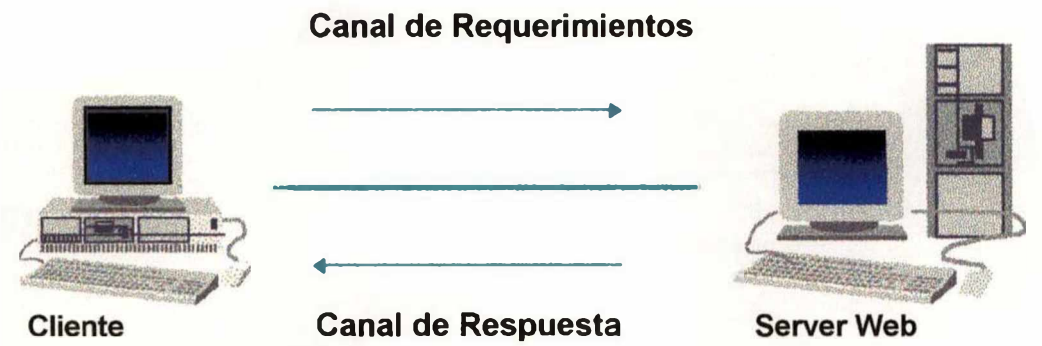
- **mensaje :** la unidad básica de comunicación HTTP, consiste de una secuencia estructurada de octetos con una sintaxis definida que son transmitidos vía la conexión.
- **requerimiento :** un mensaje de requerimiento HTTP.
- **respuesta :** un mensaje de respuesta HTTP.
- **recurso :** un objeto, dato o servicio de la red el cual puede ser identificado por un URI.
- **entidad :** una representación particular de un dato o de una réplica de un servicio, que puede estar dentro de un mensaje requerimiento o en un mensaje respuesta. Una entidad consiste de metainformación en el encabezamiento de la entidad y contenido en el cuerpo de la entidad.
- **cliente :** un programa aplicación que establece conexiones con el propósito de enviar requerimientos.
- **agente usuario :** clientes que inician un requerimiento, están desde los browser, editores, etc.
- **servidor :** un programa de aplicación que acepta conexiones a fin de servir requerimientos enviando las respuestas correspondientes.
- **servidor origen :** el servidor sobre el cual un recurso dado reside o será creado.
- **proxy :** un programa intermediario que usualmente corre sobre una máquina firewall. Un proxy server acepta requerimientos desde los clientes y sirve ellos internamente (si están cacheados) o pasa el requerimiento al servidor correspondiente actuando de esta manera como cliente.
- **gateways :** un servidor que actúa como un intermediario con otros servicios. A diferencia del proxy, este recibe requerimientos como si él fuese el servidor orígen para el requerimiento. El cliente que hace la consulta no se entera que se está comunicando con un gateway.
- **túnel :** es un programa que actúa como intermediario entre dos conexiones. Una vez activo, no es considerado como parte de la comunicación con HTTP, aunque haya sido iniciado por un requerimiento HTTP . Su función termina y es cerrado cuando la comunicación finaliza.
- **cache :** es un programa de almacenamiento local de mensajes de respuesta y además un subsistema que controla el almacenamiento, la recuperación y el borrado. Una cache almacena respuestas cacheables a fin de reducir el tiempo de respuesta y el ancho de banda de la red

para requerimientos equivalentes. Cualquier cliente o servidor puede incluir una cache.

NOTA : las clasificaciones de un programa como un cliente o un servidor no es excluyente, el uso de estos términos corresponde solo al rol que es realizado por el programa para una conexión particular, más que para las capacidades del programa en general. Cualquier servidor puede actuar como un server origen, proxy, gateway, o túnel dependiendo el comportamiento que asume de acuerdo al tipo de requerimiento.

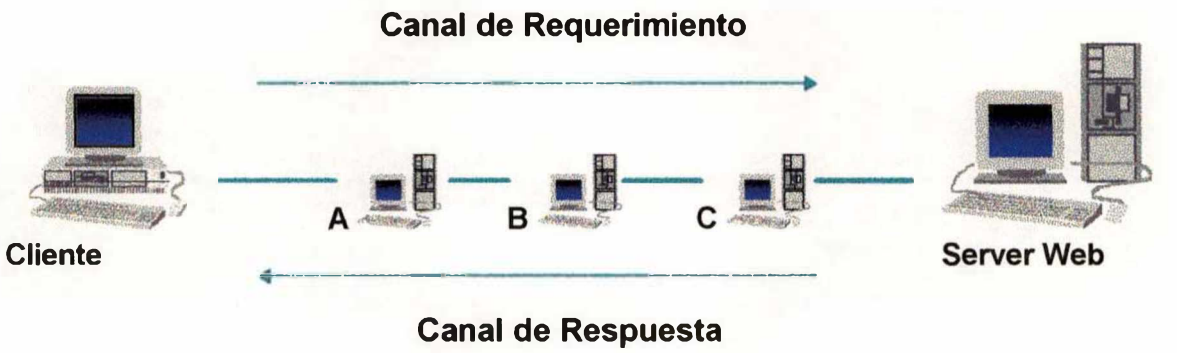
3.3 OPERACIÓN

El protocolo HTTP está basado sobre un paradigma requerimiento/respuesta. Un cliente establece una conexión con un servidor y envía un requerimiento al servidor en la siguiente forma : método de requerimiento, URI, y versión del protocolo seguido por un mensaje tipo MIME conteniendo modificadores del requerimiento, información del cliente y posible cuerpo del contenido. El servidor responde con una línea de estado incluyendo versión del protocolo del mensaje y código de éxito o error, seguido por un mensaje tipo MIME conteniendo la información del servidor, metainformación de la entidad y posible cuerpo del contenido.



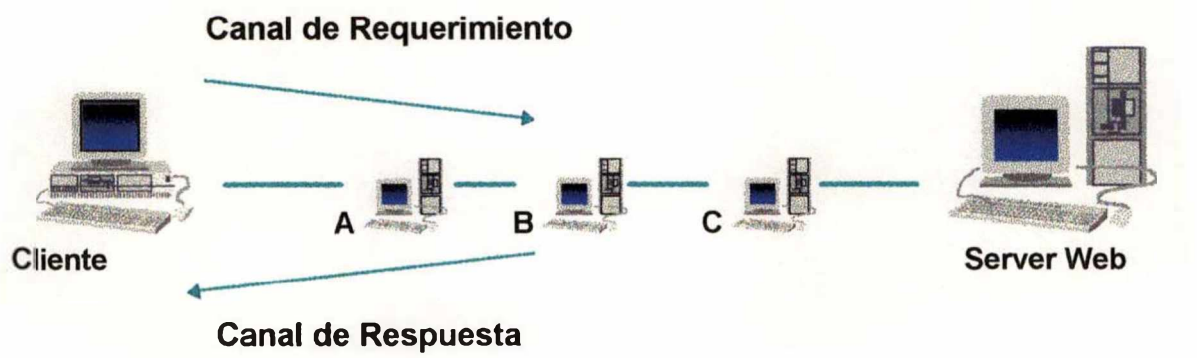
Una situación más complicada ocurre cuando uno o más intermediarios están presentes en el canal (requerimiento/respuesta). Hay tres formas comunes de intermediarios: proxy, gateways y túnel. Un proxy es un agente que envía, recibiendo requerimientos para un URI en su forma absoluta y reescribe todo o parte del mensaje y lo envía hacia el servidor identificado por el URI. Un gateway es un agente receptor, actuando como una capa anterior a algún otro servidor y si es necesario traslada el requerimiento a una capa inferior para el protocolo del servidor. Un túnel actúa como una articulación (relay) entre dos

conexiones sin cacheo de mensajes, los túneles son usados cuando la comunicación necesita pasar a través de un intermediario (tal como un firewall) aún cuando el intermediario no puede entender el contenido de los mensajes.



El gráfico muestra tres intermediarios (A, B, C) entre el cliente y el servidor origen. Un mensaje de requerimiento o respuesta que viaja a través de todo el canal debe pasar a través de cuatro conexiones separadas. Esta distinción es importante porque algunas operaciones de comunicación HTTP se deben aplicar solo a la conexión con el más cercano, un vecino, solo al punto final del canal o a todas las conexiones a lo largo del canal. Aunque el diagrama es lineal, cada participante puede estar en múltiples y simultáneas comunicaciones. Por ejemplo B puede estar recibiendo requerimientos desde otros clientes a parte de A y/o enviando requerimientos a otros servidores además de a C al mismo tiempo que esta manejando requerimientos desde A.

Cualquier parte de la comunicación la cual no esta actuando como un túnel puede emplear una cache interna para manejar requerimientos. El efecto de una cache es que el canal de requerimientos/respuesta se acorta si uno de los participantes a lo largo del canal tiene una respuesta cacheada aplicable para el requerimiento. El siguiente gráfico ilustra el canal resultante si B tiene una copia cacheada de una respuesta reciente desde el Servidor Origen vía C para un requerimiento el cual no ha sido cacheado por el cliente o por A.



Históricamente HTTP/1.0 [12] y las aplicaciones no han definido adecuadamente que es y que no es una respuesta cacheable.

Sobre Internet las comunicaciones HTTP se dan sobre las conexiones TCP/IP [1], el port default es TCP 80 pero otros ports pueden ser usados. HTTP solo presume un transporte seguro, cualquier protocolo que provea tales garantías puede ser usado.

La práctica corriente requiere que la conexión sea establecida primero por el cliente para cada requerimiento y cerrada por el servidor luego de enviar la respuesta. Ambos, clientes y servidores deben ser capaces de manejar casos donde ellos cierran la comunicación prematuramente por acción del usuario, time out, o falla del programa. Siempre el cerrado de la conexión termina con el requerimiento actual.

SERVIDORES HTTP

4. SERVIDORES HTTP

El software servidor usado con WWW es llamado servidor Web, este se conecta al port 80 por default, cuando un cliente consulta una página específica el servidor toma la página y se la envía al cliente.

Los servidores Web pueden ejecutar scripts especiales que les permite funcionar como gateway con otros recursos de información sobre su sistema o Internet.

Los clientes y servidores Web se comunican usando el protocolo HTTP.

4.1 QUE ES CGI ?

La interface gateway común es un standard de interface para aplicaciones externas con servidores de información tales como HTTP o servidores WEB. Un documento HTML que un servidor Web recupera es algo estático que no lo afecta el transcurso del tiempo. Un programa CGI de otra manera es ejecutado en tiempo real y la salida es información dinámica.

No hay límite sobre las aplicaciones que pueden unirse al Web. Lo único que se debe recordar es que el programa no debe tomar demasiado tiempo de proceso.

Un programa CGI es un ejecutable, esto equivale a dar la posibilidad de correr un programa sobre su sistema, aunque hay que tomar algunas precauciones de seguridad para evitar accesos no autorizados al filesystem.

Los programas CGI necesitan residir en un directorio especial, el servidor conoce donde debe ejecutar el programa. Este directorio esta usualmente bajo el control directo del webmaster, prohibiendo al usuario común crear un programa CGI, un usuario que quiera crear su propio CGI debe comunicarse con el webmaster.

Un programa CGI [5] puede ser escrito en cualquier lenguaje que permita ser ejecutado sobre el sistema, tales como C, C++, Fortran, VisualBasic, Perl, TCL, Shell UNIX, Apple Script, que estén disponible sobre el sistema.

En un lenguaje como C, C++ o Fortran el programa debe ser escrito y compilado, en lenguajes interpretados como Perl [6], TCL o Shell UNIX solo se escribe el script y este debe residir en el mismo directorio que los ejecutables compilados. La mayoría de la gente prefiere lenguajes interpretados en vez de programar y compilar, dado que es más fácil la puesta a punto, la modificación y mantenimiento.

4.2 TESTEO DEL SERVIDOR

Para testear que el servidor corra correctamente en algún browser se deberá abrir un URL mínimo (por ej : http://www.unlp.edu.ar/)y se ve que sucede. Si existe un archivo index.html este deberá ser retornado y si no existe aparecerá un índice de archivos.

Otra prueba es haciendo telnet al port donde WWW sirve los documentos.

telnet www.unlp.edu.ar 80
Trying 163.10.xx.xx
Connected to anubis.unlp.edu.ar.
Escape character is '^]'
HEAD / HTTP/1.0<CR><CR>

HTTP/1.0 200 OK
Date : 23-Sep-96 19:42:02 GMT
Server : CERN/3.0
MIME-version : 1.0
Content-type : text/html
Content-Length: 3169
Last Modified : Monday, 09-Sep-96 20:50:04 GMT
Connection closed by foreign host

Esto muestra que el servidor esta respondiendo. Si no sucede así entonces se deberá mirar los archivos de logging para ver que dicen y analizar los archivos de configuración.

4.3 ARMADO DE HOME PAGE

Se denomina al punto principal de entrada de un sitio como Home Page. Debe existir solo una Home Page al tope del árbol de documentos. Si el servidor tiene múltiples usos, podrán existir muchas Home Pages en diferentes partes del árbol de documentos, un servidor puede almacenar varios sitios.

Para delegar el manejo de los documentos HTML de las distintas organizaciones que soporta el servidor se puede dar la posibilidad a cada organización de manejar sus documentos HTML restringiendo el acceso a otros subdirectorios. Para esto no hay que modificar ninguna directiva de configuración solo con el manejo de permisos para directorios y archivos brindados por UNIX puede hacerse. Si cualquiera de las organizaciones desea manejar scripts o

mapas sensitivos debe coordinar con el webmaster, también se puede asignar manejos especiales de los scripts a cada organización por separado.

4.4 CONVENCIONES PARA ACCESO PÚBLICO

Una convención es setear el nombre del servidor, no es recomendado que el nombre del servidor sea igual al de la máquina, la forma canónica de los nombres es identificarlo con www. Esto da flexibilidad para mover el servidor de máquina sin problema.

Se debe establecer un alias email para que los usuarios puedan enviar comentarios, requerimientos de links y reporte de errores de ejecución al administrador, por convención este alias es webmaster, usualmente es la persona que hace la instalación, configuración y mantenimiento.

4.5 ANUNCIO DEL SERVIDOR

Una vez instalado y corriendo el servidor debe ser anunciado a la comunidad. No existe un proceso de registración formal pero hay normas bien establecidas : decir que organización persona o entidad es dueña del servidor, que contenido tiene, el estado si está listo para su uso o bajo construcción y el URL de la Home Page.

El anuncio conviene enviarlo a los principales servidores WWW de Internet: WHAT´S NEWS de NCSA, CERN o W3C [http://www.w3.org/], Yahoo [http://www.yahoo.com/] y otros índices.

SERVIDOR NCSA

5. SERVIDOR NCSA

El programa ejecutable como servidor Web se llama httpd, el código fuente está escrito en C y para varias plataformas.

NCSA brinda un kit con el soft de instalación para distintas plataformas como Unix o Windows [7], este contiene una versión pre-compilada de httpd, archivos de configuración por default, scripts pre-compilados y un conjunto de íconos.

Al desempaquetar este kit, se generan una serie de directorios, los cuales son necesarios para el buen funcionamiento del servidor. Por default el servidor reside en el directorio : /usr/local/etc/httpd/ , esto se puede modificar en los archivos de configuración. De aquí colgarán los siguientes directorios :

cgi-bin : este directorio contiene ejemplos de gateways scripts y sus versiones binarias pre-compiladas. Si se construyen gateways propios, aquí es donde usualmente deben estar. El directorio es llamado cgi-bin porque los gateways usan una interface standard con el server Web llamado Common Gateway Interface (CGI)

conf : este directorio contiene todos los archivos de configuración usados para controlar la operación de su server Web.

icons : este directorio contiene todos los íconos que son usados por el server.

logs : directorio donde se van almacenando los archivos de control de acceso (access.log) y los de errores (error.log)

support : este directorio contiene programas que son usados para control de acceso global y por directorio.

5.1 CONFIGURACIÓN DE UN SITIO ESPECÍFICO

Según [2], se puede setear la configuración del servidor de acuerdo a las necesidades del ambiente editando los archivos de configuración :

httpd.conf, principal archivo de configuración del servidor.

srm.conf, archivo de configuración de los recursos del servidor.

access.conf, archivo de control de acceso global.

Estos archivos se encuentran en el directorio /conf .

Los dos primeros archivos son los principales que se deben ajustar, también se debe modificar el último respecto de las modificaciones hechas en el primero. Existe un archivo de configuración secundario que es el Mime.types que es usado para controlar el mapeo de las extensiones de los archivos a tipos de datos MIME [11], éste no necesita ser tocado a menos que se desee setear una correlación entre tipo de archivo y aplicación standard.

Al modificar cualquiera de estos archivos hay que tener en cuenta que :

- todas las directivas y texto no son sensitivos a mayúsculas/minúsculas excepto los pathname y los URLs.
- todas las líneas de comentarios comienzan con #.
- debe haber solo un directorio por línea.
- espacios en blanco extras, son ignorados.

Veremos el seteo básico de un servidor describiendo las combinaciones en los archivos de configuración que hay que realizar.

httpd.conf : este archivo controla como corre su servidor, pero no contiene detalle de los archivos servidos. Para setear un servidor NCSA como daemon standard, con acceso público irrestricto se debe :

- modificar las directivas User y Group con los nombres que correspondan. Esto define el identificador de usuario (UID) y el identificador de grupo (GID), que son usados por el servidor cuando corre, se recomienda User http, Group www.
- asegurarse que los nombres existan en los archivos /etc/passwd y /etc/group respectivamente.
- modificar la directiva ServerAdmin como webmaster@su.dominio (y setear su alias e-mail para que los mail lleguen a usted). La dirección e-mail está dada para usuarios así ellos pueden reportar problemas de run-time con su webserver.
- cambiar la directiva ServerRoot al path absoluto del directorio donde usted planea poner el httpd binario (ejecutable). Este también determina donde otros archivos relacionados deben ir, el default es /usr/local/etc/httpd
- especificar el ServerName como www.su.dominio. Luego setear el alias CNAME como este hostname.

- cambie ServerType si desea correr el server bajo inetd en vez de standalone, este es el default.

srm.conf : este archivo de configuración controla donde su servidor htpd busca sus documentos y scripts. Para inicializar el servidor con acceso público irrestricto usted debe :

- modificar la directiva DocumentRoot, que apunte al directorio que es el tope del árbol de documentos. Este es el principal directorio desde el cual se sirven los documentos, el default es /htdocs.
- modificar la directiva UserDir a DISEABLE mientras se setea el servidor. Esto previene de servir documentos al público desde su directorio home.
- modificar Alias o ScriptAlias para poner el directorio icons o cgi-bin en directorios no standard.

Estas directivas permiten hacer que algunos documentos aparezcan para los usuarios en un directorio y en realidad están en otro, también se conoce como directorios virtuales.

access.conf : este archivo controla que tipo de browsers web acceden al servidor o a ciertos directorios.

El seteo default habilita todas las características para todos los clientes Web, sin ninguna precaución de seguridad.

5.2 ARRANCAR EL SERVER

Hay diferentes maneras de arrancar el server, standalone o inetd. Correr bajo inetd es bueno para una etapa de testeo y prototipeo, mientras que correr standalone es mejor para un uso intensivo.

Este servidor tiene pocas opciones para seteos en línea de comando al levantar, porque la mayoría de las cosas son controladas por medio de los archivos de configuración.

Las opciones disponibles son :

-d directorio este controla donde httpd mira para sus archivos de configuración.

-f archivo el archivo que http busca.

-n despliega el menú del servidor.

5.3 MAPEO DE URL A DOCUMENTOS

Para abrir un documento HTML sobre el servidor se indica un URL. Cuando un URL no incluye un camino de directorio o nombre de archivo, el servidor retorna el contenido del archivo index.html en el directorio indicado por DocumentRoot, en httpd.conf y si no existe este archivo genera una lista de los archivos en el directorio.

El nombre virtual identifica el camino para encontrar el documento. Este no es el pathname absoluto del documento sobre el sistema, el nombre virtual es directorio virtual más el nombre del archivo real.

El servidor httpd traslada los URLs a directorios reales como sigue :

- mira el comienzo del camino del URL en los directorios virtuales definidos en Alias o ScriptAlias en srm.conf. Si lo encuentra reemplaza el directorio virtual por el directorio real y procesa el requerimiento.
- mira si existe el prefijo /-username/ (la barra seguida por nombre válido de un usuario del sistema) y si es así pone el subdirectorio público de HTML del usuario especificado para el archivo indicado. Si el directorio del usuario no existe retorna un error. Este proceso es salteado si la directiva UserDir es DISEABLE en srm.conf.

5.4 MAS RASGOS

Teniendo el servidor básico andando se puede agregar soporte para nuevos tipos MIME o expandir los tipos soportados.

También se puede habilitar includes del lado del servidor o seleccionar las directivas que el servidor muestra.

5.4.1 AGREGANDO TIPOS MIME

Los browsers y servidores Web usan httpd 1.0 pasando tipos MIME.

Se puede agregar soporte para nuevos tipos MIME sobre el servidor. Esto requiere que el cliente soporte también estos tipos. Del lado del cliente conoce como usar un tipo MIME que el servidor entrega; usan el archivo .mailcap para decidir que viewers o players externos invoca para cada tipo MIME.

Del lado del servidor, este conoce como mapear tipos de archivos y extensiones de archivos para tipo de datos MIME standard. Se controla por medio del archivo de configuración mime.types que es bastante completo para la mayoría de la gente.

Agregando las siguientes directivas en el archivo de configuración srm.conf se pueden agregar extensiones MIME especiales para su server Web.

AddType permite sobreescribir definiciones tipo MIME encontradas en el archivo de configuración mime.types

AddEncoding agrega codificación de MIME única para mensaje cliente/servidor

DefaultType usado para establecer un tipo MIME default que sea retornado al browser Web.

5.4.2 INCLUDES SERVER-SIDE

Son archivos include similar a los de cualquier otro lenguaje de programación pero además de archivos, pueden incluir valor de variables de ambiente. Proveen una manera fácil de incluir en forma consistente fecha, tamaño del archivo, nombre de autores o un html en un número de HTML.

Esta opción no está habilitada por default en el servidor NCSA y puede habilitarse para todo el servidor o por directorio especificando la cláusula Includes o IncludesNoExec en la directiva Options en los archivos de configuración access.conf o htaccess.

El uso de Includes toma muchos ciclos de cpu y por consiguiente disminuye mucho el tiempo de respuesta.

La tendencia es usar solo includes en algunos archivos.

5.4.3 ÍNDICE DE DIRECTORIO AUTOMÁTICO

Cuando un URL apunta a un directorio en vez de a un archivo, el servidor primero intenta retornar el archivo default (configurado en el servidor), en ese directorio pero si no existe, el servidor automáticamente genera un texto HTML describiendo el contenido del directorio.

En el archivo de configuración access.conf puede habilitarse o deshabilitarse esta característica, por default está habilitada.

Puede generarse de dos maneras, texto plano o " fancy ". En texto plano, se ve de manera similar a hacer ls -l con la facilidad de hacer click sobre un archivo para recuperarlo. Con fancy hay un ícono asociado, el tamaño del archivo, una descripción opcional del archivo útil para browsers no gráficos. Puede existir un ícono distinto por cada tipo de archivo.

El índice de directorios automáticos es controlado con directivas en el archivo de configuración srm.conf .

5.4.4 MULTIHOME / VIRTUALHOST

NCSA httpd soporta esto para reconocer desde que interface el servidor fue llamado. Se puede tener www.foo.org y www.bar.org mapeando dos direcciones IP y el servidor responder a la que llamo.

La razón más usual para esta funcionalidad es el caso de una compañía que provea presencia WEB sin aumentar el costo por medio de distintos Proveedores a Servicios Internet, con esto se puede proveer diferentes servicios a usuarios sobre la máquina local .

En el archivo de configuración httpd.conf luego de los parámetros normales para el servidor, comienza la sección VirtualHost con la sintaxis semejante al archivo access.conf

```
<VirtualHost nombre>
  Directiva
  Directiva
</VirtualHost>
```

donde *nombre* es una dirección IP o hostname correspondiente a una simple dirección IP. Las directivas que se usan son las mismas que en la otra parte del archivo de configuración httpd.conf. Ejemplo:

```
<VirtualHost 127.0.0.1>
  ServerName localhost.foo.bar.org
  ResourceConfig conf/localhost_srm.conf
  TransferLog   logs/localhost_access_log.conf
</VirtualHost>
```

acá cualquier acceso al servidor por localhost tendrá las opciones de configuración dadas y las sobreescritas , el servidor parseará localhost_srm.conf y seteará los default para el servidor localhost. Todos los accesos serán logeados en logs/localhost_access_log y todos los errores estarán en el standard logs/error_log.

La cantidad de VirtualHost que se pueden definir no está limitada por NCSA httpd, pero si lo está por la memoria de la máquina y las posibilidades que brinda el k-shell. La configuración de VirtualHost se debe hacer al final del archivo httpd.conf, de otra manera los resultados son impredecibles.

5.4.5 IMAGEMAP

Para generar un imagen imagemap debe estar NCSA httpd instalado y corriendo. Luego se crea una imagen, se crea una archivo mapa para el imagemap donde se definen regiones de la imagen y el URL al cual apuntan. Las regiones se definen por medio de la siguiente sintaxis :

metodo URL coord1, coord2, ..., coordn

Los métodos pueden ser:

default, luego irá el URL y no lleva coordenadas;

circle, luego irá el URL y se indica el centro y radio como coordenadas;

poly, luego irá el URL y se indican las coordenadas de cada vertice;

rect, luego irá el URL y las coordenadas de la esquina superior izquierda y de la esquina inferior derecha;

point, luego irá el URL y las coordenadas del punto.

El URL puede ser relativo o absoluto y las coordenadas deben tener la forma de pares (x,y) . La primer línea del mapa indicará la respuesta default con el método default, las líneas siguientes especificarán las regiones y los URLs respectivos.

Para hacer referencia al imagemap en el archivo HTML, se debe construir un URL apuntando a este.

```
<A HREF="http://www.unlp.edu.ar/cgi-bin/imagemap/laplata.map>
<IMG SRC="/pictures/laplata.gif" ISMAP>
</A>
```

Lo que está después de /cgi-bin/imagemap es el directorio y el nombre del archivo mapa.

5.5 SEGURIDAD EN NCSA httpd

Por medio del control de a quién se deja acceder a los documentos del servidor se busca evitar accesos no autorizados.

- El control de acceso por hostname y autenticación de usuarios básico son facilidades provistas por httpd relativamente seguras pero no completamente. La autenticación de usuarios envía la password a través de la red en texto plano, haciendo esto fácilmente legible. El control de acceso basado en DNS es solo tan seguro como DNS.

- Se puede usar la directiva Options para deshabilitar los Server Side Includes, se debe deshabilitar la característica exec en access.conf.

- Siempre que sea posible use AllowOverrride None en access.conf. esta directiva previene la sobreescritura de los seteos de cualquier directorio no autorizado, también se gana en performance.

- Proteger los directorios home de los usuarios. Con la directiva Directory cada usuario tiene su directorio home en por ejemplo /home, entonces

```
<Directory /home>
        AllowOverride None
        Options Indexes
</Directory>
```

5.6 DIRECTIVAS DE httpd.conf

AccessConfig nombre del archivo

determina la ubicación del archivo de control de accesos (ACF). El *nombre del archivo* es o bien un nombre parcial relativo a ServerRoot o un nombre completo. Este archivo controla los accesos al servidor completo o por directorios.

Se permite sólo una de estas directivas.

ErrorLog nombre del archivo

determina la ubicación del archivo de los log de errores (*nombre del archivo* puede ser en forma parcial relativo a ServerRoot o completa). Este archivo da la información de los errores ocurridos:

-Time Out causado por los clientes.

- Scripts que causen errores de ejecución y no generan una salida.

- archivos de acceso por directorio (.htaccess) que intenten sobre escribir los permisos globales.

- Bugs del servidor

Se permite sólo una de estas directivas.

Group [*nombre* | # *número*]

esta directiva indica el Group ID que tendrá el servidor cuando corra standalone. El nombre o número indicado deberá existir en el archivo "/etc/group". Esto no cambia el grupo de httpd pero sirve para controlar como correrán los procesos hijos, el servidor deberá ser root.

Se permite sólo una de estas directivas.

IdentityCheck [*on* | *off*]

determina si la identidad del usuario remoto será loggeada en el archivo indicado por la directiva TransferLog. Setearla en on afecta la performance del servidor, si no piensa usarla seteela en off.

Se permite sólo una de estas directivas.

PidFile nombre del archivo

especifica donde el ID de los procesos del servidor, solo es importante cuando el servidor corre standalone es usado, se usa cuando se quiere cancelar al servidor o rearrancarlo.

Se permite sólo una de estas directivas. El default es :

PidFile logs/httpd.pid

Port número

define el número de port a donde el servidor entregará la información. Puede ser cualquiera entre 0 y 65536, el port standard para http es el port 80, se debe tener en cuenta cuando se usa un port no standard que muchos browsers solo pueden acceder al port 80 (del otro lado de un firewall, corriendo proxy server).

Se permite sólo una de estas directivas. El default es : Port 80

ResourceConfig nombre del archivo

identifica la ubicación del archivo de configuración de recursos (el *nombre del archivo* puede ser parcial relativo a ServerRoot o completo). Las directivas de tal archivo indican como será manejado el indexado automático de directorios.

Se permite sólo una de estas directivas. El default es :

ResourceConfig conf/srm.conf

ServerAdmin dirección

es usada para indicar la dirección e-mail del administrador que es mostrada cuando hay problemas de ejecución, se usa el alias *webmaster*.

Se permite sólo una de estas directivas. No hay default.

ServerName nombre del host

define el nombre del host que el servidor usa cuando crea redirección de URL, además se debe setear en forma correspondiente la variable CNAME.

Se permite sólo una de estas directivas. No hay default.

ServerRoot directorio

define el camino (path) absoluto que será el directorio raíz donde está ubicado el servidor.

Se permite sólo una de estas directivas. El default es :

ServerRoot /usr/local/etc/httpd

ServerType [inetd | standalone]

define la forma en que se ejecutará el servidor, si como un proceso inetd o como un daemon standalone, para un uso intenso es recomendado que corra como standalone.
Se permite sólo una de estas directivas. El default es :

ServerRoot standalone

TimeOut tiempo

indica la cantidad de tiempo (en segundos) que el servidor esperará por un requerimiento una vez que el browser se conecto.

Se permite sólo una de estas directivas. El default es:

TimeOut 1800

TransferLog nombre del archivo

indica el nombre del archivo para los tranfer log, este puede ser parcial relativo al ServerRoot o completo. Este archivo guarda el acceso al servidor por host, fecha y archivo.

Se permite sólo una de estas directivas. El default es:

TranferLog logs/access_log

TypesConfig nombre del archivo

identifica la ubicación del archivo de configuración de los tipos MIME. El nombre del archivo puede ser parcial relativo o completo. Esta configuración es usada para controlar como mapear la extensión de los archivos a los tipos MIME correspondientes al retornar la información a los browsers.

Se permite sólo una de estas directivas. El default es:

TypesConfig conf/mime.types

User [nombre | número]

es usada para definir el user ID cuando el servidor corre standalone, puede ser diferente al del arranque. El user ID debe ser una cuenta válida, debe existir en /etc/passwd.

Se permite sólo una de estas directivas. El default es:

User #-1

5.7 DIRECTIVAS DE srm.conf

AccessFileName nombre del archivo

controla el nombre de archivo usado para el control de acceso por directorio específico.
Se permite sólo una de estas directivas. El default es :

AccessFileName .htaccess

AddDescription texto descriptivo tipo del archivo

esta directiva asocia una frase con un tipo particular de archivo. Esto es usado cuando se genera un índice del directorio en forma automática
Puede haber muchas de estas directivas, pero si es omitida no hay default.
Ejemplo:

AddDescription "(archivo tar comprimido por gzip)" *.tar.gz

AddEncoding type extensión

agrega una directiva de codificación única para el manejo de cliente/servidor, *type* es el tipo de documento a codificar, *extensión* define el tipo de archivo que debe usar el nuevo codificado. Esta directiva es usada para por ejemplo archivos comprimidos que pueden ser descomprimidos en el cliente, para que esto sea útil es necesario que el cliente soporte el método de codificación dado.
Puede haber varias de estas directivas, si no hay ninguna no hay default.

AddIcon camino-iconos nombre1 nomre2 ...

esta directiva indica que iconos pueden ser mostrados para un documento por su nombre.
Puede haber varias de estas directivas, si no hay ninguna no hay default.

AddIconbyEncoding camino-iconos tipo1 tipo2 ....

esta directiva es similar a la anterior , pero asocia los iconos de acuerdo a la codificación.
Puede haber varias de estas directivas, si no hay ninguna no hay default.

AddIconType camino-iconos tipo1 tipo2 ...

es similar a la directiva AddIcon excepto que asocia iconos con tipos MIME.
Puede haber varias de estas directivas, si no hay ninguna no hay default.

AddType *tipo/subtipo extensión*

permite sobreescribir la definición de tipos MIME encontrada en el archivo de configuración mime.types.
Puede haber varias de estas directivas, el default es encontrado en el archivo de configuración mime.types.

Alias virtual camino

esta directiva crea un nombre virtual o directorio, es apropiada si se tiene varias organizaciones sobre el mismo servidor y desea tener URL cortos. *virtual* es el camino o nombre de archivo con el que se quiere mapear y *camino* es el path completo que se remapeará.
Puede haber varias de estas directivas, si no hay ninguna no hay default pero la configuración inicial da:

Alias /icons/ /usr/local/etc/httpd/icons/

DefaultType tipo/subtipo

es usada para devolver un tipo MIME default al browser si el mapeo encontrado en mime.types no es adecuado.
Solo es permitida una directiva, si no está presente el default es:

DefaultType text/html

DefaultIcon camino

esta especifica el ícono a ser mostrado cuando FancyIndexing está en on y no hay un ícono asociado.
Solo es permitida una directiva, si no está presente el default es:

DefaultIcon /icons/unknown.xbm

DirectoryIndex archivo

indica el archivo que será devuelto cuando un URL indica un directorio y no un archivo sobre el servidor.
Solo es permitida una directiva, si no está presente el default es:

DirectoryIndex index.html

DocumentRoot path absoluto

esta directiva setea desde donde serán servidos los documento en el servidor. Para servir archivos fuera de este directorio se debe usar la directiva Alias o crear links simbólicos.
Solo es permitida una directiva, si no está presente el default es:

DocumentRoot /usr/local/etc/httpd/htdocs

HeaderName archivo

identifica el archivo que será usado como encabezado del directorio durante el indexado automático.
Solo es permitida una directiva, si no está presente no hay default, el seto inicial es :

HeaderName HEADER

IndexIgnore patron1,patron2

identifica archivos que serán ignorados durante el indexado del directorio automático, tales archivos será aquellos que coincidan con el patrón indicado en esta directiva.

IndexOptions option1,option2

especifica donde se quiere un indexado de directorio fancy o standard y que opciones desea activar para el indexado. Las opciones son :

FancyIndexing, lo setea en on
IconsAreLinks, hace un ícono parte del anchor para el nombre del archivo
ScanHTMLTitles, causa que el titulo sea mostrado como un comentario
SuppressLastModified, no muestra la última fecha de actualización
SuppressSize, no muestra el tamaño
SuppressDescription, no muestra descripción para ningún archivo.

Solo es permitida una directiva, si no está presente todas las opciones se setean en off.

ReadmeName archivo

será incluido el archivo al tope del índice del directorio.
Solo es permitida una directiva, si no está presente no hay default pero la configuración incial es:

ReadmeName README

Redirect *Virtual Url*

es usada para mapear el path y nombre de un documento existente en el servidor a un nuevo URL. *URL* es el url al que será trasladado.
Puede haber varias de estas directivas.

ScriptAlias Virtual Path

crea un nombre virtual o directorio sobre el servidor para scripts. Virtual es el nombre dado al url externamente y Path el camino completo del directorio que contiene los scripts.
Puede haber varias directivas indicadas , no hay default pero la configuración inicial es:

ScriptAlias /cgi-bin/ /usr/local/etc/httpd/cgi-bin/

UserDir [camino] DISABLED]

agrega al árbol de documentos los directorios homes de usuarios para que esten disponibles para httpd. Si se setea a DISABLED no se servirán documentos desde directorios de los usuarios.
Solo es permitida una directiva, si no está presente el default es :

UserDir public_html

5.8 DIRECTIVAS DE access.conf

allow from host1 host2 ...

esta directiva controla que host pueden acceder a un directorio dado con un metodo dado. Host puede ser :

domain-name, nombre de dominio solo desde este dominio podrán acceder
host-name, nombre del host completo
full-IP-addess, una dirección IP de un host
parcial-IP-address, un dirección IP parcial para restricción de subredes
all, esto significa que todos los host tendran el acceso denegado

Solo es permitida una directiva por sección Limit, si no está presente no hay default.

AllowOverride opt1, opt2, ...

está directiva controla los archivos de control de acceso por directorio local que están permitidos para sobreescribir los seteos definidos por access.conf. Las entradas válidas son :

None, no esta permitido en este directorio un archivo de control de acceso.
All, es irrestricto un archivo de control de acceso en este directorio.
Options, permite usar la directiva Options.
FileInfo, permite usar las directivas AddType y AddEncoding.
AuthConfig, permite usar las directivas: AuthName, AuthType, AuthUserFile, AuthGroupFile.
Limit, permite usar la directiva de sección Limit.
Solo es permitida una directiva, si no está presente el default es :

```
AllowOverride All
```

AuthGroupFile path

setea el archivo a usar como lista de grupos usuarios para autenticación de usuarios. Path es un camino absoluto para el archivo group a usar en este directorio.
Está directiva debe ir acompañada por las directivas AuthName, AuthType y AuthUserFile. Se aplica tanto a los archivos de control de acceso global como a los archivos de control por directorio. No hay default.

AuthName nombre

setea el nombre de la autorización para este directorio. Este nombre es dado a los usuarios que conocen su username y password. *Nombre* es un nombre corto describiendo está autorización.
Está directiva debe ir acompañada por las directivas, AuthType y AuthGroupFile. Se aplica tanto a los archivos de control de acceso global como a los archivos de control por directorio. No hay default.

AuthType tipo

setea el tipo de autorización usado en este directorio. *Tipo* es el tipo de autenticación a usar para este directorio. Solo Basic esta implementado.
Está directiva debe ir acompañada por las directivas, AuthName, AuthUserFile, y AuthGroupFile. Se aplica tanto a los archivos de control de acceso global como a los archivos de control por directorio. No hay default.

AuthUserFile path

setea el archivo a usar como lista de usuarios y passwords usadas para autenticación de usuarios. Path es un camino absoluto para el archivo de usuarios creado con el programa de soporte htpasswd.

Está directiva debe ir acompañada por las directivas AuthName, AuthType y AuthGroupFile. Se aplica tanto a los archivos de control de acceso global como a los archivos de control por directorio. No hay default.

deny from host1, host2,

decide que hosts tiene el acceso denegado a este directorio. *Host* es uno de los siguientes:

domain-name, nombre de dominio solo desde este dominio podrán acceder.
host-name, nombre del host completo.
full-IP-addess, una dirección IP de un host.
parcial-IP-address, un dirección IP parcial para restricción de subredes.
all, esto significa que todos los host tendran el acceso denegado.

Solo es permitida una directiva por sección Limit, si no está presente no hay default.

<Directory dir>

es una directiva seccionada, identifica el directorio o los directorios a donde otras directivas de control de acceso se pueden aplicar. *Dir* es el camino completo del directorio. Debe existir luego la directiva </Directory>.

<Limit metodo1, metodo2, ...>

es una directiva seccionada, que identifica que clientes pueden acceder a directorio, se aplica a archivos de control de acceso global y por directorio. *Metodo1* es uno de los siguientes:

GET permite a los clientes recuperar documentos y ejecutar scripts.
POST permite usar scripts con POST.

Solo order, deny, allow y require están permitidas dentro de la sección Limit. El tag </Limit> cierra la directiva correspondiente Limit.

Options option-list

controla el grado de que características avanzadas son permitidas en el servidor.
Las entradas válidas son :

None, nada está habilitado en este directorio.
Indexes, permite a los usuarios requerir índices sobre este directorio. Deshabilitar esto implica que se deshabilita solo los índices generados por el servidor.
Includes, habilta los Server-Side Includes.
IncludesNoExec, habilita los includes pero deshabilita su ejecución.
ExecCGI, la ejecución de cgi está permitida.
FollowSymLinks, el servidor seguirá los links simbólicos en este directorio.
SymLinksIfOwnerMatch, seguirá los links simbólicos solo si el archivo destino tiene el mismo User ID que el link.
All, todo esta habilitado en este directorio.

order ord

indica el orden en el cual deny y allow serán evaluadas en la sección Limit, solo en esta sección tiene sentido.

require entity nombre1, nombre2

esta directiva causa que usuarios autenticados puedan acceder a este directorio con un método dado. *entity* puede ser:

user solo los usuarios nombrados pueden acceder con los métodos dados.
group, solo los usuario en los grupos nombrados pueden acceder con los métodos dados.
Valid-user solo los usuarios listados en AuthUserFile pueden acceder dando su password.

nombre es un nombre especifico en el contexto del tipo de la entidad.

Solo es permitida una directiva de esta y no hay default.

Además de las directivas explicadas las siguientes también pueden ser usadas :

DefaultType	solo archivos de control de accesos por directorio
AddEncoding	solo archivos de control de accesos por directorio
AddDescription	solo archivos de control de accesos por directorio
AddIcon	en todos los archivos de control de accesos
IndexIgnore	en todos los archivos de control de accesos
DefaultIcon	en todos los archivos de control de accesos
ReadmeName	en todos los archivos de control de accesos

SERVIDOR CERN

6. SERVIDOR CERN

El servidor implementado por CERN [http://www.w3.org/] se llama "httpd_3.0", esta escrito en lenguaje C y para varias plataformas.

Al igual que NCSA, este brinda el soft de instalación para distintas plataformas.

La instalación del soft genera un conjunto de directorios donde son ubicados los iconos, archivos de configuración, archivos scripts, ejecutables y una versión precompilada del servidor.

En el directorio **/opt/cern_httpd/bin/** se encuentra el servidor **(httpd_3.0)** y otros archivos binarios de gran importancia como por ejemplo el programa **htimage** que es usado para manejar imágenes sensitivas. También se generan los siguientes directorios :

- **/opt/cern_httpd/icons** : aquí se encuentran archivos con extensión .xbm, que son iconos usados por el servidor cuando genera páginas html.

- **/var/opt/cern_httpd** : en este directorio encontramos el archivo de configuración httpd.conf, donde están todas las directivas que setean el servidor, el archivo httpd_pid que guarda el número de proceso asignado al servidor en ejecución y otros directorios que vemos a continuación.

- **/var/opt/cern_httpd/cachedir** : debajo de este directorio están todos los archivos cacheados por el servidor indicando el árbol por niveles de directorio de la paginas html accedidas. El servidor no cacheará lo que le sea indicado por las directivas NoCaching. La directiva CacheRoot indica que este será el directorio de cacheo. Estas directivas las encontramos en el archivo de configuración httpd.conf.

- **/var/opt/cern_httpd/cgi-bin** : este directorio contiene archivos que se ejecutarán como CGI.

- **/var/opt/cern_httpd/htdocs** : directorio raíz donde se guardarán todas las páginas html.

- **/var/opt/cern_httpd/logs** : en este se almacenan los archivos de control de acceso y errores. El nombre de estos archivos se especifica en las directivas de configuración.

6.1 CONFIGURACIÓN DE UN SITIO ESPECÍFICO

Para setear la configuración del servidor teniendo en cuenta las necesidades del sitio, se debe editar el único archivo de configuración "httpd.conf", a diferencia de NCSA donde veíamos 3 archivos de configuración. En este archivo se define como el servidor traslada un requerimiento a un nombre de documento, como controlar el funcionamiento del servidor y también como define la protección.
Para escribir o modificar este archivo hay que recordar que :

- todas las directivas y texto no son sensitivos a mayúsculas/minúsculas excepto los URLs y los pathnames.
- las líneas que comienzan con el símbolo # son líneas de comentarios.
- solo debe haber una directiva por línea.
- los espacios en blanco son ignorados.

Para que el servidor corra como un daemon standard con acceso público irrestricto, se debe:

- indicar el User y Group que el servidor usará para correr, estos deben existir en los archivos /etc/passwd y /etc/group.
- indicar Port 80 por default
- indicar ServerRoot con el pathname completo donde irá el binario.
- modificar la directiva PidFile con el nombre del archivo pidfile.
- indicar la directiva UserDir con public_html.

Estos valores son para los seteos generales del servidor.
El manejo de URLs se controla usando las directivas Exec y Pass básicamente. Exec indicará donde ejecutar los CGI y con la regla Pass se indicará donde buscar los html.
Indicar los archivos logs, el nombre y el path por medio de las directivas AccessLog, ErrorLog y también LogFormat.
El servidor se puede correr como un servidor proxy y entonces también se deben setear directivas relacionadas con el caching de la información y directivas para el manejo del proxy.

6.2 ARRANCAR EL SERVIDOR

Al igual que en NCSA el servidor se puede arrancar en forma standalone o inetd.
Al tener un solo archivo de configuración se agrega la posibilidad de setear determinados aspectos del servidor por medio de opciones en la línea de comando al levantar el servidor.
La sintaxis de línea de comando para 'httpd' permite un número de opciones y opcionalmente un argumento directorio, por ejemplo :

httpd [-opt -opt ...] [directorio]

Si el argumento "directorio" esta presente nos indica que ese directorio será usado como raíz del servidor, de no estar presente se toma del archivo de configuración lo que indica la directiva ServerRoot y si esta no esta seteada exportará por default el directorio /public/.

Cuando el argumento directorio es pasado, el archivo de configuración no se carga.

Las opciones de la línea de comando son :

-r archivo : toma el 'archivo' como archivo de configuración.
-p port : indica el port a usar para la comunicación.
-l archLog : se debe usar el archLog como archivo de log.
-restart : vuelve a arrancar cualquier httpd que esta corriendo, cargando nuevamente el archivo de configuración.
-Gc_only : solo para proxies y corriendo como inetd, realiza un garbage collection sobre la cache y finaliza.

Las siguientes tres directivas pueden setearse en el archivo de configuración con la directiva DirAccess :

-dy: habilita el mostrado de directorios generado el servidor un documento html con el contenido del directorio.
-dn: deshabilita en mostrado de directorios.
-ds: permite el mostrado selectivo, solo se permite en los directorios que contienen el archivo .www-browseab.

Es común poner dentro de los directorios un archivo README conteniendo noticias o instrucciones. Esto se puede setear en el archivo de configuración a través de las directivas DirReadme o bien en la línea de comando con las siguientes opciones :

-dt : para un directorio habilitado a mostrar al tope de la lista de archivos.
-db : pone este archivo al final de la lista.
-dr : deshabilita la inclusión del archivo README.

6.3 MAPEO DE URLS [10]

En el archivo de configuración httpd.conf existe un conjunto de directivas que posibilita el mapeo de URLS virtuales. Las principales directivas son :

Map : mapea URLs a archivos actuales.
Pass : acepta un requerimiento.
Fail : rechaza un requerimiento.

Redirect : redirecciona un requerimiento.
Protect : setea protección.
DefProt : setea la protección por default.
Exec : setea donde se ejecutarán los scripts.

6.4 CARACTERÍSTICAS PARTICULARES

6.4.1 PROXY [Cap 10]

Esta es una de las características más importantes del servidor. Un proxy es un servidor http corriendo sobre una maquina firewall, permitiendo el acceso al mundo a usuarios que están dentro del firewall. Además permite realizar cacheo de documentos, consiguiendo con esto un mejor tiempo de respuesta en un futuro acceso al mismo documento. [Ver cap. 10]

El servidor httpd de CERN corre como proxy si en su archivo de configuración tiene seteados los métodos de acceso que pueden ser aceptados, se determina esto con la directiva Pass, por ejemplo :

```
Pass http:*
Pass ftp:*
Pass gopher:*
Pass wais:*
```

Existen formas de proteger un proxy de accesos no autorizados. Las directivas Enable, Disable habilitan y deshabilitan los métodos de acceso. También se puede proteger a nivel de hosts con la combinación de las directivas Protection y Protect, por ejemplo:

```
Protection Protname{
            Mask @ (*.unlp.edu.ar)}

Protect  http: * protname
Protect  ftp: * protname
Protect  gopher: * protname
Protect  news: * protname
```

Tenemos la facilidad de configurar un Proxy para conectarse a través de otro proxy, esto se setea mediante las directivas :

```
http_proxy    http://otro.proxy.server
ftp_proxy     http://otro.proxy.server
gopher_proxy  http://otro.proxy.server
wais_proxy    http://otro.proxy.server
```

6.4.2 CACHING

El servidor httpd, corriendo como proxy, puede realizar un cacheo de los documentos recuperados desde algún host remoto, permitiendo esto agilizar la respuesta para un futuro requerimiento del mismo documento. Para habilitar el caching, se deben setear las directivas correspondientes en el archivo de configuración httpd.conf.

6.4.3 CONTROL DE ACCESO

El servidor tiene la facilidad de registrar todos los requerimientos entrantes en un archivo de registro de acceso (log), también se registran los errores de ejecución internos del servidor en orto archivo de registro de errores.

Las principales directivas en el archivo de configuración para setear el formato de los archivos de registros e indicar que cosas registrar son las siguientes :

AccessLog : setea el nombre del archivo de registro de acceso.
ProxyAccessLog : setea el nombre del archivo de registro de proxy.
CacheAccessLog : setea el nombre del archivo de registro de cache.
ErrorLog : setea el nombre del archivo de registro de errores.

El formato de los archivos de registro es común a todos los archivos y en general es usado por todos los servidores así da la posibilidad de poder correr programas de estadísticas libres. Tal formato es el siguiente :

hostRemoto Logname Authuser [fecha y hora] requerimiento estado bytes
del usuario

6.4.4 PROTECCIÓN A NIVEL USUARIOS

El acceso puede ser restringido de acuerdo a la dirección internet (IP) o al nombre del usuario, o a ambos. El control de acceso puede ser a nivel de árbol, de archivo, o por usuario. Para esto se controlan archivos de Password y Group que contienen a los usuarios permitidos y se tiene la directiva Protection que define máscaras y la directiva Protect que indica a quien aplicar esas máscaras. Cada directiva Protect lleva asociado un archivo de seteos en el cual se debe indicar el esquema de autenticación a usar, el archivo de grupo y el archivo de password a usar, y la identificación del servidor. Por ejemplo :

Protect	/exactas/*	/httpd.setup1	user.group
	⇓	⇓	⇓
	que proteger	como	a quien

httpd.setup1 tiene el siguiente contenido :

AuthType **basic**
ServerId **xxxxxxxxxx**
PasswordFile **/...../passwd** ⎫
⎬ **Propios del servidor no de unix**
GroupFile **/...../group** ⎭

Para proteger todo el árbol se agrega la regla :

GetMask group, user, group@address,..... en el archivo de seteos.

Pero si se quiere proteger archivos individuales se usan archivos ACL donde se indican todos los archivos permitidos para acceso (aquí no es necesario GetMask) pueden existir ambas cosas para permitir el acceso solo a un grupo y restringirlo en el futuro.

Indicando en el archivo de configuración **ACLOverRide On** permite sobreescribir las máscaras definidas en el archivo ACL, inhibe el chequeo de las máscaras cuando existe un archivo ACL. Un archivo ACL es un archivo llamado .www_acl en el mismo directorio de los archivos que se les controla el acceso. Un archivo ACL no puede ser creado sin definir un témplate en el archivo de configuración usando Protection o DefProt, el motivo es que la protección ACL necesita conocer que UserId, password, etc., debe usar a fin de determinar donde un usuario tiene acceso para un archivo específico.

6.4.5 MANEJO DE TIPOS MIME [10]

El servidor usa sufijos para descubrir el content-type, content-encoding, y el content-language de un archivo. Los valores default que el servidor conoce son los tipos de archivos usuales, pero para agregar nuevos sufijos ligados a un tipo de archivo o sobreescribir los default existentes, las siguientes directivas pueden usarse en el archivo de configuración :

AddType: mapea un sufijo de archivo a un Content-Type MIME.
AddEncoding: mapea un sufijo de archivo a un Content-Encoding MIME.
AddLanguage: soporta multilenguaje. Mapea sufijos a distintos Content Language.
SuffixCaseSense: setea el sufijo sensitivo a mayúsculas/minúsculas.

6.4.6 MANEJO DE SCRIPTS

Los servers scripts son usados para manejar búsquedas, imágenes sensitivas, forms, etc., y como resultado del script pueden producir documentos en el aire.

El servidor conoce que un requerimiento es un requerimiento script mirando al comienzo del URL, el cual debe coincidir con lo seteado en la directiva

Exec /url-xxx/* /camino físico completo/*

si el requerimiento comienza con **/url-xxx/** entonces el script se ejecutará en el directorio **/camino físico completo/** que es un pathname completo del filesystem donde están los script.

El paso de información a los scripts se hace por medio de variables de ambiente, standard input o argumentos en línea de comando. Las variables de ambiente son:

QUERY_STRING : es todo lo que sigue al signo de interrogación cuando se invoca al script. Esto va codificado en forma particular todos los caracteres especiales como +, -, *, etc. se representan en notación hexadecimal.

PATH_INFO : información extra del path es guardada aquí, por ejemplo :

Exec /htbin/* /../../cgi-bin/*

y el URL del script es

/htbin/ejemplo/mas/ info

donde el script es "ejemplo" y en PATH_INFO se guarda "/mas/info".

Se debe recordar que el archivo script debe tener permiso de ejecución, el shell a ejecutar debe estar escrito en la primera línea y que todos los script son ejecutados desde ServerRoot. El resultado de un script debe ser o bien un documento standard o una redirección. Para devolver un documento el resultado debe comenzar con una línea **Content-Type : xxx/xxx** seguida por una línea en blanco y luego el documento, para devolver una locación la respuesta debe comenzar con:

Location: http://xxx.xxxx.xx/aaa/aa URL siempre completo.

6.5 PRINCIPALES DIRECTIVAS

6.5.1 DIRECTIVAS GENERALES

ServerRoot "path"

especifica el directorio raíz donde encontraremos toda la información a mostrar.

HostName nombre.host.completo.

provee el nombre del host, esto es útil cuando el nombre real del host es distinto al nombre que el cliente ve.

Port 80

esta directiva solo se usa cuando el servidor corre en modo standalone, indica el port con el cual se establece la comunicación.

ServerType [standalone | inetd]

define la forma en que se ejecutará el servidor.

PidFile path

define el nombre del archivo donde se guarda el número identificador de proceso del servidor en ejecución.

UserId webmast

setea el usuario default para ejecutar el servidor. Esta directiva solo es útil cuando el servidor corre como root.

GroupId

setea el grupo default para ejecutar el servidor. Al igual que la directiva anterior solo es importante cuando el servidor se corre como root.

ParentUserId cualquiera

esta directiva causa que el servidor setee la identificación de usuario indicada inmediatamente después de conectarse al port.

ParentGroupid

esta directiva causa que el servidor setee la identificación de grupo indicada inmediatamente después de conectarse al port.

Enable METHOD

habilita la ejecución del método indicado. Los métodos pueden ser POST, PUT, entre otros.

Disable METHOD

deshabilita la ejecución del método indicado. Los métodos pueden ser POST, PUT, entre otros.

Welcome archivo.html

identifica el archivo por default a leer cuando solo se especifica el nombre del directorio en el URL

6.5.2 DIRECTIVAS DE CONTROL DE LOGGING

AccessLog path/archlog

el archivo archlog contiene los logs de todos los requerimientos, esta directiva indica donde se encontrará este archivo. Donde path puede ser un path absoluto o un path relativo a ServerRoot.

ProxyAccessLog path/archlog

si estamos corriendo el servidor como proxy y queremos tener los logs de transacciones proxy separados de transacciones normales, especificamos el archivo de logs del proxy con esta directiva. Donde path puede ser un path absoluto o un path relativo a ServerRoot.

CacheAccessLog path/archlog

para guardar los logs de los accesos a la cache en un archivo separado se usa esta directiva. Donde path puede ser un path absoluto o un path relativo a ServerRoot.

ErrorLog path/archlog

en archlog se guardarán los logs de los errores causados por el servidor. Donde path puede ser un path absoluto o un path relativo a ServerRoot.

LogFileDateExt %dd - %mm - %aa

especifica una extensión común para todos los archivos logs basados en un formato fecha : hora

LogFormat modo

puede ser common o propio, es el formato en que se guardarán los logs, por default se usa el common.

NoLog template

indica que los requerimientos coincidentes con el template no se registren como log.

6.5.3 DIRECTIVAS DE MAPEO DE URLs

Map template result

si la dirección coincide con template, usa el string result para futuros mapeos.

Pass template result

si la dirección coincide con template, usa el string result como si fuera este.

Fail template

si la dirección coincide con template, prohibe el acceso.

Redirect template result

los documentos que coinciden con template son redirigidos a result, el cual debe ser un URL completo.

Exec template script

mapea las direcciones que coinciden con template a script, y ejecuta el programa deseado.

6.5.4 DIRECTIVAS DE DEFINICION DE SUFIJOS

AddType .sufijo representación codificación [calidad]

- agrega una definición de matching de un tipo de datos con un sufijo de terminación, con esto se agranda el conjunto de sufijos predefinidos en la directiva.
- *.sufijo* indica la última parte del nombre del archivo. Existen dos casos especiales : *.* y * que indican matching para archivos que no han matcheado con ningún otro sufijo.
- *representación* indica una descripción del estilo de la representación del tipo MIME.
- *codificación* indica el tipo de codificación que se usará para transferir el tipo MIME. Básicamente en archivos ASCII (7 u 8 bits) o binario. Otras pocas codificaciones son permitidas.
- [calidad] es opcional, esta representado por un número real entre 0.0 y 1.0, el valor default es 1.0.

AddEncoding .sufijo codificación

se usa también los sufijos para determinar la codificación del contenido de un archivo. Por ejemplo un archivo .z es un archivo x-compressed.

AddLanguage .sufijo codificación

el soporte de multilenguaje es construido con sufijos que se usan para determinar el lenguaje del documento. Por ejemplo :
AddLanguaje .en en
AddLanguaje .uk en_UK

SuffixCaseSense On

indica si el sufijo será sensitivo a las mayúsculas/minúsculas. El valor por default es Off

6.5.5 DIRECTIVAS DE SETEOS DE TIMEOUTS

Todos los tiempos son expresados de las siguientes posibles formas:

45 secs
10 mins
2 mins 30 secs
1 hour

InputTimeOut tiempo

especifica el tiempo de espera del cliente para enviar el requerimiento. El valor default es 2 mins.

OutputTimeOut tiempo

especifica el tiempo para enviar una respuesta. El valor default es 20 mins.

ScriptTimeOut tiempo

especifica el tiempo para permitir al servidor finalizar un script. Si un script no termina en el tiempo indicado entonces el servidor envía una señal TERM y una KILL al script. El valor default es 5 mins.

6.5.6 DIRECTIVAS DE PROXY-CACHING

Caching On

el cacheo de documentos es puesto implícitamente en On por medio de la directiva CacheRootDirectory pero puede ser explícitamente puesto en Off con esta directiva.

CacheRoot /camino/completo/directorio

setea el directorio raíz para usar de cache para el servidor corriendo como proxy, este debe ser un path absoluto.

CacheSize 20 M

setea el tamaño de la cache en Megabytes, este es el máximo espacio que podrá ocupar.

NoCaching url

los url que coinciden con el indicado en la directiva NoCaching nunca serán cacheados.

CacheOnly url

los url que coinciden con el indicado en la directiva CacheOnly serán cacheados.

CacheClean url tiempo

todos los documentos que coincidan con el url especificado en esta directiva y sean más viejos que lo que se indica, serán removidos.

CacheUnUsed url tiempo

todos los documentos que coincidan con el url especificado en esta directiva y no hayan sido usados desde determinado tiempo, serán removidos.

CacheDefaultExpiry url tiempo

este tiempo es dado si el servidor remoto no lo indica. El default es 0.

CacheLastModifiedFactor factor

este factor es usado para aproximar la fecha de expiración. El valor default es 0.1, lo que significa que si el documento ha sido modificado hace 20 días expirará en 2 días.

KeepExpired [On | Off]

normalmente el garbage colector removerá todos los archivos que han expirado inmediatamente. Esto salva espacio de disco, pero decrementa la eficiencia del GET condicional.

CacheNoConnect [On | Off]

solo los documentos encontrados en la cache son retornados y si se requiere uno que no esta en la cache retornará error. Esto es útil para realizar una demo. Por default esta seteada en Off.

CacheExpiryCheck [On | Off]

esta directiva es útil si el propósito es hacer una demostración y recuperar solo páginas de la cache sin importar si han expirado o no, entonces se setea en Off.

Gc [On | Off]

cuando funciona la cache el garbage collection esta activado por default con esta directiva se puede inhabilitar.

GcDailyGc tiempo

especifica el horario para correr el garbage collection, normalmente durante la noche.

CacheAccessLog archivo

registra el acceso a la cache en un archivo de logging distinto.

Existen más directivas para controlar la Cache como por ejemplo GcMemUsage, CacheLimit_1, CacheLimit_2, CacheLockTimeOut.

6.5.7 DIRECTIVAS PARA CONFIGURAR UN PROXY CONECTADO A OTRO PROXY

Se indica esto por medio del seteo de las siguientes variables :

http_proxy http://servidor.proxy.saliente/

ftp_proxy http://servidor.proxy.saliente/
gopher_proxy http://servidor.proxy.saliente/
wais_proxy http://servidor.proxy.saliente/

no_proxy url

indica que no se conecte a otro proxy para recuperar tal url.

6.5.8 DIRECTIVAS PARA BROWSEO DE DIRECTORIOS

DirAccess [On | Off | Selective]

en ON habilita el browseo de directorios en todos los directorios, en OFF lo deshabilita y en Selective permite que se browseen solo los directorios que contienen el archivo .www_browseable.

DirReadme [Top | Bottom | Off]

para cualquier directorio browseable que contenga un archivo readme incluye su texto arriba con la opción Top, abajo con la opción Bottom y no lo muestra con la opción Off.

FTPDirInfo [Top | Bottom | Off]

los menajes Ftp de los servidores Ftp son puestos arriba con la opción Top, abajo con la opción Bottom y no lo muestra con la opción Off.

DirShowIcons [On | Off]

muestra una imagen en el frente de cada línea, los iconos visualizan el Content-Type del archivo y son definidos por la directiva AddIcon.

DirShowDate [On | Off]

muestra la fecha de la ultima actualización.

DirShowSize [On | Off]

muestra el tamaño del archivo.

DirShowDescription [On | Off]

muestra la descripción si esta es disponible. Si el archivo es HTML muestra la información dentro del tag TITLE.

6.5.9 DIRECTIVAS PARA MANEJO DE ICONOS

IconPath full-path

esta directiva debe ser usada si el directorio no se encuentra debajo de ServerRoot.

AddIcon url-icono texto-alternativo template

esta directiva liga un ícono con un tipo MIME, url-icono es el url del ícono, texto-alternativo es el texto que se despliega sobre una terminal de caracteres, y template indica el tipo de archivo MIME al cual esta ligado por ejemplo : text/html.
Usando el servidor como proxy, el url debe ser indicado en forma completa.

Seteo de iconos especiales

AddUnknownIcon : indica el url del ícono usado para tipos de archivos no conocidos.
AddDirIcon : url del ícono para directorios.
AddParentIcon : url del ícono para directorios padres.

BROWSERS

7. BROWSERS

A fines del 94 el Web era un servicio poco conocido sobre internet, y NCSA Mosaic [http://www.ncsa.uiuc.edu] fue el browser gráfico (de distribución libre) que servía para ver documentos académicos distribuidos en Internet sobre Web. Mosaic fue seguido por Netscape Navigator, un browser que tuvo gran aceptación por parte de los usuarios. Viendo el auge de estas herramientas, grandes empresas como IBM [http://www.ibm.com/], ORACLE [http://www.oracle.com/] y MICROSOFT [http://www.msn.com/] presentaron productos del mismo nivel como el de los recién mencionados.

Texto plano con algunos hiperlinks fue uno de los primeros diseños de páginas para Web, avanzando en el tiempo observamos sofisticados diseños de páginas ricas en contenidos gráficos, integrando además audio, video, y animación.

La caracteristica mas importante de un browser es la rapidez en mostrar el código fuente HTML de una página. Actualmente los browsers mas populares soportan el standard HTML 2.0 [8] y muchos tags de la propuesta HTML 3.0 [9] (como ser tablas, imágenes sensitivas del lado del cliente, entre otras). También deben proveer soporte on-line de formatos gráficos tipo bitmap, usados comunmente en Web incluyendo las extenciones .BMP, .GIF, .JPEG, .PCX, .TIFF. Algunos browsers proveen soporte para GIF animados y JPEG progresivos, pero su uso no es generalizado.

Otras características a resaltar son el completo conjunto de controles de navegación claramente identificables, tales como la capacidad de tener un **history** (donde se guarda la dirección URL de todos los sitios visitados desde que se invocó al browser), **bookmark** (lugar donde se guardan los sitios que nos interesan, estos perduran mas allá de la ejecución del browser).

Es deseable también que el browser mantenga un conjunto de páginas recientemente visitadas en una caché. De esta manera si el usuario quiere retornar a una de las páginas ya consultadas, no tenga que conectarse con el servidor. Asimismo debería proveer la posibilidad de determinar el tamaño de disco que destinará como caché.

La gran variedad de formatos para audio, animación y video se conoce colectivamente como tipo de datos MIME [11], es muy común encontrarlos en sitios Web actualizados. La mayoría de los browsers ofrecen capacidades básicas de multimedia, pero todavía hay poca standarización sobre cómo se deben manejar los distintos contenidos multimediales. En ciertos casos se usan aplicaciones **"helpers"** que son componentes de software para una tarea especifica, generalmente son desarrolladas y distribuidas por terceras partes (por ejemplo software para playback de archivos RealAudio o para clips de video QuickTime). Estas aplicaciones se ejecutan automáticamente cuando el browser

recibe una página html con el tipo apropiado de archivo multimedia dentro de el. En un browser fuertemente orientado a multimedia es fácil configurar y manejar estas aplicaciones.

Los **plug-ins** son una alternativa mas sofisticada, al igual que las aplicaciones helpers son escritas por terceras partes pero son especificamente escritas para la API del browser, por tal motivo trabajan en forma transparente con el browser correspondiente y tienen la habilidad de mostrar los contenidos multimediales directamente dentro de la ventana del browser.

La capacidad de los browser para soportar contenidos multimediales integrados es cada vez más comun y los tipos de datos multimediales son ejecutados sobre la misma página html o en el caso del audio en background de manera que se escucha mientras se mira el texto.

El problema de estos avances es que el browser debe estar preparado para manejar todos los tipos de datos necesarios. Este rasgo es cubierto por el desarrollo más radical sobre Web hoy en día : lenguaje **JAVA** de Sun Microsystem. Los programas java pueden ser almacenados como parte de una página HTML. Cuando se requiere la página el programa es bajado y ejecutado por un interprete java en su browser. Java está basado en C++, puede ser usado para crear cualquier aplicación, incluyendo juegos multimediales.

Un browser para ser usado en **Intranet** debe tener otras consideraciones, mientras que el soporte multimedia on-line no es una de las caracteristicas mas deseadas, la programación en Java se debe considerar seriamente. La estrategia de Intranet hoy día está orientada a transmitir documentos para publicación, entonces la necesidad es poder moverse rapidamente de un lugar a otro sin tener las demoras que implica el soporte de multimedia.

Analizaremos tres de los browsers más populares ellos son Netscape Navigator, Microsoft Internet Explorer y NCSA Mosaic, de ellos describiremos las características particulares que ofrecen, tanto como las que lo diferencian.

Basándonos en estudios realizados recientemente por empresas reconocidas en Internet sobre la utilización de exploradores para consultar la red, hemos decidido incluir en este trabajo, el análisis de las potencialidades y características de tres exploradores importantes: el **Netscape Navigator 2.x**, el **Microsoft Internet Explorer 2.x**, el **NCSA Mosaic** que son considerados los más populares.

Las estadísticas consideradas que informan sobre los porcentajes de popularidad de los exploradores actuales, fueron extraídas de:

- "BenLo Park Research", en Agosto de 1995, encontrándose esta información en la página http://emporium.tumpike.net/j/jc/public-html/stats.html,
- "Web Trends Report", en Abril de 1996, con información en la página http://www.webtrends.com .

- "OnLine Business Today (OB9T). Un reporte sobre los browsers más populares", en Abril de 1996, en la página http://www.hpp.com.

  Y arrojaron los siguientes resultados :

Encuesta de WebTrends

	Browser	% del total
1	Netscape	88.70
2	Microsoft Internet Explorer	7.92
3	IBM WebExplorer	0.40
4	IWENG	0.35
5	Spry AIR Mosaic	0.34
6	NCSA Mosaic	0.32
7	WebWhacker 32 v1.0	0.23
8	SpyGlass Mosaic	0.18
9	Lynx	0.14
10	Otros	1.42

Encuesta de BenLo Park

	Browser	% del total
1	Netscape	83.5
2	Mosaic	2.9
3	IWENG	2.8
4	NetCruiser	1.8
5	AIR Mosaic	1.6
6	IBM WebExplorer	1.6
7	PRODIGY	1.1
8	Microsoft Internet Explorer	0.7
9	InternetMCI	0.6
10	Otros	3.4

Encuesta de Online Business Today (OBT)

	Browser	% del total
1	Netscape	83.3
2	Microsoft Internet Explorer	4.9
3	Lynx	2.6
4	NCSA Mosaic	1.3
5	Otros	7.9

En estas tablas estadísticas se consideraron las últimas versiones de los softwares en cuestión y coincidieron en que el explorador notablemente más utilizado en la actualidad es el Netscape Navigator. WebTrends y OBT presentan al Microsoft Internet Explorer en el segundo puesto mientras que BenLoPark lo marca octavo. El Mosaic se encuentra generalmente entre los primeros ocho puestos.

Los restantes exploradores fueron descartados de este informe, principalmente por cuestiones de incompatibilidad, como es el caso de Lynx que debe montarse únicamente sobre Unix, IWENG que requiere fundamentalmente el uso de módems, y otros, debido a que su porcentaje de utilización no superaban el 2% en ninguna encuesta.

7.1 NCSA Mosaic [http://www.ncsa.uiuc.edu/]

El Centro Nacional de Aplicaciones de Supercomputación (NCSA) desarrollo el primer browser gráfico llamado Mosaic.

Este Centro mantiene la política de implementar y respetar sobre su browser solo los estándares aceptados como HTML 2.0, por esta causa no está capacitado para visualizar páginas con técnicas avanzadas de diseño.

Este soft es de libre distribución y se encuentra on_line en Internet, provee la facilidad de desinstalarse, no requiere librerías Winsock propietarias y es compatible con las librerias Winsock provistas por Windows '95. Permite deshabilitar la transferencia de imágenes y es totalmente compatible con HTML 2.0, dado que solo soporta los estándares; y como las extensiones de HTML 3.0 son solo propuestas no las implementa.

No ofrece soporte a macros, Java y Plugs-ins. Permite cambiar el tamaño de la cache local y tiene la caracteristica de que el boton derecho del mouse es sensible al contexto actual. Está disponible para Windows '95, NT y 3.1x este ultimo con win32s.

Respecto a las facilidades de navegación que provee son excelentes, brindando un completo manejo de bookmarks tal como agruparlos, anidar estos grupos y editar el título de las páginas guardadas. También permite salvar las imágenes localmente, salvar las páginas como HTML o como TXT. Otra característica importante es el regsitro de páginas visitadas durante la ejecución actual del navegador.

El soporte en el área de multimedia es pobre ya que solo soporta aplicaciones helpers/MIME clientizables, no brinda nada integrado al browser.

Respecto de las herramientas integradas encontramos E-mail, FTP, Gopher y News, como documentación de ayuda sobre el funcionamiento del browser existe documentación impresa, on-line local y on-line sobre Internet.

7.2 Microsoft Internet Explorer [http://www.msn.com/]

Este browser es de libre distribución, se puede encontrar on-line en Internet o bien adquiriendo Windows 95.

El Internet Explorer es fácil de instalar y configurar, y además no requiere librerías Winsock propietarias, trabaja con cualquiera y en particular con las que provee Windows 95

Es compatible con HTML 3.0 dado que soporta los tags más populares de la propuesta HTML 3.0, permite definir una caché local y modificar su tamaño, también se puede deshabilitar la transferencia de imágen a fin de lograr mayor velocidad en bajar las páginas permite salvar las páginas en formato HTML o como TXT, las imágenes también se pueden salvar en el disco local.

Por estos días no provee soporte para aplicaciones Java, permite el uso de plugs-in y si existen aplicaciones definidas para distintos tipos de archivos definidos en Windows 95 usa esas aplicaciones en forma automática para mostrar en el browser tales archivos.

Muchas de las ventajas de Internet Explorer se deben a su integración con Windows 95 dado que usa muchas herramientas del S.O para realizar las operaciones del browser tales como la generación de carpeta para bookmarks, permitiendo así anidar distintas carpetas. Los bookmarks en sí los almacena como referencias rápidas o direcciones favoritas.

Otra característica importante es el registro de páginas visitadas durante la ejecución actual del navegado, este registro se mantiene de sesión a sesión si el usuario así lo desea, guardando las direcciones URL´s como una referencia rápida de Windows 95.

Para el control en la ejecución de compras por medio de Internet, está integrando la tecnología SSL para permitir una comunicación segura.

En el área multimedia el soporte que brinda es bueno, ya que provee soporte integrado al browser para aplicaciones de sonido, animación y video. La empresa planea incluir en el browser el uso de ACTIVE VRML para integrar realidad virtual al ambiente. También provee la capacidad de clientizar distintas aplicaciones para diferentes tipos MIME.

En cuanto a extensiones propietarias presenta colores y tamaños para fonts, alineación de tablas, colores para las celdas, archivos de sonido .WAV para background, videoanimación .AVI que se ven cuando aparece la página.

Un bug que se detectó es que cuando se visitan varios sitios con demasiadas imágenes disminuye la velocidad en la construcción de las pantallas y la solución por el momento es cerrar el browser e invocarlo otra vez.

Al momento de localizar una página y traerla lo hace en un tiempo aceptable, permitiendo mientras localiza el sitio realizar otras tareas sobre la página actual que estamos consultando tal como consultarla haciendo scroll.

En cuanto a otros servicios internet que integrados sobre el browser se encuentran FTP, GHOPER y MAIL recién en la versión 3.0

La documentación de ayuda sobre el funcionamiento del browser ofrece un completo índice y claras explicaciones en forma local, característica que es muy importante ya que no es necesario conectarse a la red para obtener ayuda.

7.3 Netscape Navigator [http://www.nestcape.com/]

El browser desarrollado por Netscape Comunications no es de libre distribución, tiene un costo de u$s 49.00, aunque se encuentra disponible on-line en su sitio Internet. Es muy fácil su seteo e instalación, además no requiere librerías winsock propietarias y es compatible con las librerías de Windows'95. Está disponible para Windows 3.xx, NT, '95, Mac.

Respecto de las características generales provistas por los browsers brinda soporte a todas tales como : deshabilitar la transferencia de imágenes, modificar el espacio en disco para la cache, es compatible con la propuesta HTML 3.0 dado que implementa los tags más novedosos y útiles.

Una de los rasgos más importantes y que lo ponen delante del resto de los browser es el soporte integrado para aplicaciones JAVA y ser interprete de código JavaScript escrito dentro de las páginas HTML.

Permite al igual que el resto de los browsers salvar las páginas y las imágenes en disco local. Ofrece facilidades para la navegación tal como guardar un HISTORY donde se registran todos los lugares visitados desde la invocación del Netscape Navigator; ofrece también un manejo muy completo de BOOKMARKS permitiendo armar una organización tipo árbol de los URLs o direcciones de páginas según un criterio adoptado por el usuario puede ser por temas por ejemplo. Estas marcas se almacenan en un archivo .htm y con solo marcar con el mouse la dirección, arrastrarla y dejarla caer en una rama del árbol de marcas ya queda insertada la dirección como una marca. Permite crear alias para los URLs y estos pueden aparecer repetidos en distintas ramas del árbol.

Con el objetivo de acelerar la bajada de una página desde la red Netscape Navigator provee soporte para imágenes con formato JPEGs progresivos con lo cual pueden ser identificadas con solo el 10% del archivo bajado. Otra característica que contribuye con la velocidad de consultas es el soporte de mapa sensitivo del lado del cliente con lo que acelera el proceso de bajado de páginas.

Una de las características más innovadoras es la capacidad integrada de ver objetos multimediales en una página a través de los Plugs-Ins, quienes deben ser seteados la primera vez (inicializados). Permitiendo ver documentos Adobe Acrobat, archivos Macromedia Director, movies QuickTime y otros objetos multimediales como sonido y VRML sin la necesidad de cargar aplicaciones aparte ("helpers"), dando la posibilidad de jugar con los objetos multimediales antes de finalizar la bajada de la página.

Respecto a las extensiones propietarias propuestas por la empresa, la más novedosa fue la inclusión de frames (cuadros) dentro de las páginas Web permitiendo dividir la página en varios sectores estableciendo en un sector (o cuadro) un menú de opciones que según su selección varía la información HTML en otro sector. En cada sector se muestran distintas páginas (distintos URLs). Esta herramienta es muy potente pero a su vez agrega otra ventana e control al ambiente que manejamos por lo que su ventaja es relativa. No obstante la empresa esta trabajando sobre opciones para darle más facilidad y utilidad a los frames en su versión 3.0 beta.

Dentro de las extensiones propietarias tenemos texto parpadeante, distinto color para las letras, imágenes sensitivas del lado el cliente, gráficos background y tablas.

En el aspecto seguridad Netscape Navigator incluye mecanismos de seguridad completos, basados en la tecnología Secure Sockets Layers 3.0 y en el sistema de codificación con clave pública RSA. Estos estandares encriptan los datos ingresados por los usuarios para ser transmitidos sobre Internet. los sitios comerciales que permiten compras con tarjetas de crédito o realizar negocios usan estos estándares para evitar conflictos.

Netscape Navigator provee una aplicación mail/news completamente integrada que soporta attachments MIME y HTML embebidos. No solo se puede enviar un mail a cualquier dirección sobre Web sino que también puede chequear su cuenta mail POP3 a intervalos regulares para saber si recibió un nuevo mail.

SEGURIDAD

8. SEGURIDAD

Internet no fue concebida como un entorno con seguridad propia, la información transmitida es susceptible a fraude o malos usos por parte de los usuarios, por esto es necesario incorporarle esquemas de seguridad. La información que viaja entre una computadora y un servidor usa un algoritmo de ruteo que puede determinar que deba pasar por varias otras computadoras, cualquiera de estas representa a un intermediario con potencial acceso al flujo de información, entonces es necesario evitar que los intermediarios alteren en forma alguna los datos.

Según [4], la seguridad en internet se compone de dos áreas distintas :

- **seguridad en accesos** - se dice de la capacidad de restringir el uso de equipos (computadora, disco, memoria, impresora, etc.) dentro de una organización. El control es generalmente una combinación de técnicas de autenticación por "Kerberos", instalación de proxies en internet firewalls, control de acceso estricto con password, entre otras.

- **seguridad en transacciones** - se refiere a la capacidad de dos entidades sobre Internet de realizar una transacción en forma privada con la ayuda de sistemas de criptografía y autenticación de firma digitalizada.

Los aspectos fundamentales de seguridad en una transacción Internet son los siguientes : **privacidad, autenticación e integridad.**

- **Privacidad :** el propósito de la privacidad es asegurar que la información esté oculta para cualquiera que no deba verla. Esto es más importante cuando se transmiten datos particulares de relevancia como el número de una tarjeta de crédito, información confidencial, transacciones bancarias, etc. Los algoritmos de encriptados pueden ser usados para comunicaciones privadas sobre un canal inseguro. El mensaje original es encriptado por el enviador con una password secreta, para luego ser transmitido. Con este tipo de algoritmo se asegura que el mensaje original no puede ser recuperado sin usar la password secreta.

- **Autenticación :** el web tiene una capacidad limitada para la identificación de usuarios y autenticación de clientes y servidores. Para el uso comercial del web, los clientes y servidores necesitan verificar y validar la identidad del otro a fin de asegurar que la información que fluye entre ellos es auténtica. Es por esto que se ha estudiado la creación de una firma digitalizada. La firma es una pieza de datos no

falsificable y asegura que la persona nombrada la escribió o ligó el documento. El receptor puede verificar que el documento proviene de quién lo firma. Un sistema de firma digital segura consiste de dos partes:

✓ un método de firmar un documento tal que sea imposible falsificarla.
✓ un método para verificar que una firma ha sido generada por quién representa.

- **Integridad :** con el comercio electrónico sobre internet la integridad de los datos es crítica. Una firma digital sobre un mensaje asegura que este no ha sido alterado desde que fue firmado. También posibilita que el receptor verifique matemáticamente la identidad de la persona que firmó el mensaje, además permite detectar si el mensaje ha sido alterado luego de ser firmado.

8.1 DOS PROPUESTAS DE SEGURIDAD EN WEB

Un número de características de seguridad se establecen entre los clientes y los servidores Web, se implementan con el fin de garantizar la privacidad, integridad y autenticación de los datos.

Analizaremos dos soluciones :

✓ Secure Sockets Layer (SSL)
✓ Secure HTTP

8.1.1 SSL

Es un protocolo que usa una abstracción de sockets[1] mejorada en seguridad, puntualizando la atención sobre transacciones seguras a nivel capa de transporte. Las propiedades de seguridad están ligadas al canal de comunicaciones que es establecido entre dos puntos y no a los documentos que son transportados.

SSL esta sobre la capa de transporte confiable (TCP) que asegura la integridad de los datos durante la transmisión, permite que las aplicaciones cliente/servidor se comuniquen en privado. El protocolo esta diseñado para que los servidores estén siempre autenticados y los clientes pueden ser opcionalmente autenticados.

La ventaja de este protocolo es que es un protocolo de aplicación independiente. Un protocolo de aplicación de "alto nivel" (por ejemplo HTTP, FTP, TELNET) es un capa sobre el tope de SSL en forma transparente. El SSL puede negociar un algoritmo de encriptado y una sesión de clave tanto como autenticar un servidor antes que el protocolo de aplicación transmita o reciba los primeros bytes de datos.

El protocolo SSL provee un **"canal seguro"** que tiene tres propiedades básicas :

1. **El canal es privado**. La encriptación es usada por todos los mensajes hasta que un simple handshake es usado para definir una clave secreta. Para la encriptación de los datos se usa criptografía simétrica.
2. **El canal es autenticado**. El servidor final de la comunicación también es autenticado. Para el cliente final la autenticación es opcional. La criptografía asimétrica, también es usado para la autenticación.
3. **El canal es seguro**. El mensaje transportado incluye un chequeo de integridad.

Este protocolo esta compuesto por dos protocolos, en el nivel más bajo, encima de algún protocolo de transporte seguro esta el "SSL Record Protocolo", que es usado para la encapsulación de todos los datos transmitidos y recibidos; y el otro es el "SSL Handshake Protocol", que es usado para establecer parámetros de seguridad.

8.1.2 SECURE HTTP

La seguridad en las transacciones es un aspecto crítico en las aplicaciones comerciales de Internet, esto motivó el desarrollo Secure HyperText Transfer Protocol (S-HTTP), por EIT en 1994. S-HTTP provee servicios de seguridad en transacciones de propósito general que son necesarios para transacciones seguras en aplicaciones comerciales electrónicas, estos servicios son confidencialidad de la transacción, autenticación, integridad de mensajes y no repudiabilidad de mensajes.

S-HTTP soporta transacciones seguras entre un cliente y un servidor, incorporando manejo de criptografía para mensajes a nivel aplicación, esto en contraste con el mecanismo de autorización de HTTP, el cual requería que el cliente intente acceder y le sea denegado el acceso antes que el mecanismo de seguridad sea empleado. S-HTTP incorpora criptografía de claves públicas de RSA Data Security sumado al soporte tradicional de secreto compartido (password) y sistemas de seguridad basado en Kerberos.

EIT y Terisa Systems han implementado esto para distintas empresas comerciales.

S-HTTP provee un mecanismo de comunicación seguro entre pares de clientes y servidores HTTP a fin de habilitar transacciones comerciales espontáneas para un ancho rango de aplicaciones. Este diseño intenta proveer un protocolo flexible que soporte modos de operación múltiple ortogonal, mecanismos de manejo de claves, modelos confiables, algoritmos de criptografía

y formatos de encapsulación a través de la opción de negociación entre las partes para cada transacción.

8.1.2.1 RASGOS

S-HTTP soporta una variedad de mecanismos de seguridad para clientes y servidores HTTP proveyendo opciones de servicios de seguridad apropiados para el ancho rango de los potenciales usuarios finales en Web. También provee capacidades simétricas para clientes y servidores (igual tratamiento es dado para los requerimientos y las réplicas) mientras preserva el modelo de transacción y las características de implementación de HTTP.

Varios formatos de mensajes de criptografía standard son incorporados dentro de clientes y servidores S-HTTP, también soporta interoperación entre una variedad de operaciones y es compatible con HTTP, previene a los clientes que puedan comunicarse con servidores S-HTTP y viceversa, aunque tales transacciones no pueden usar las características de seguridad de S-HTTP.

S-HTTP no requiere certificación de clave pública del lado del cliente, soportando modo de operación de clave simétrica, esto es importante porque significa que transacciones privadas espontaneas pueden ocurrir sin requerir que usuarios individuales tengan una clave pública establecida.

Provee completa flexibilidad de algoritmos de criptografía, modos y parámetros. La opción de negociación es usada para permitir a los clientes y servidores ponerse de acuerdo sobre el modo de la transacción (¿debe el requerimiento ser firmado, encriptado, o ambos?, ¿qué hacer con la respuesta ?), los algoritmos de criptografía (RSA vs. DSA para firmar, DES vs. RC2 para encriptar) y la selección de certificación.

S-HTTP intenta evitar presumir un modelo confiable particular, aunque sus diseñadores admiten que su esfuerzo consistió en facilitar múltiples raíces jerárquicas confiables, y anticipar las principales cosas que deben tener muchos certificados de claves públicas.

8.1.2.2 MODOS DE OPERACIÓN

La protección de mensajes puede ser provista de tres maneras : ***firma, autenticación y encriptación.***

Los mecanismos de manejo de claves múltiples son provistos incluyendo el estilo de pasword, intercambio de claves públicas y la distribución de tickets kerberos. En particular esto ha sido hecho para sesiones de claves simétricas prearmadas a fin de enviar mensajes confidenciales a quienes no tienen un par de claves.

FIRMA : si el manejo de firma digital es aplicado, un certificado apropiado debe ligarse al mensaje o bien el enviador debe esperar que el receptor obtenga el certificado requerido independientemente.

INTERCAMBIO DE CLAVES Y ENCRIPTACION : S-HTTP define 2 mecanismos de transferencia clave, uno usando claves públicas ligadas y el otro con claves puestas externamente.

En el primer caso el parámetro "cryptosystem" de clave simétrica es pasado encriptado bajo la clave pública del receptor.

En el último modo, encriptamos el contenido usando una sesión de claves prearmadas, con la información de la identificación de clave especificada en una de las líneas header del mensaje. Las claves pueden también ser extractadas desde tickets kerberos.

INTEGRIDAD DE MENSAJES Y AUTENTICACIÓN DEL ENVIADOR : S-HTTP provee una manera para verificar la integridad del mensaje y la autenticidad del enviador para mensajes HTTP, vía la computación de un MESSAGE AUTHENTICATION CODE (MAC), computado como una clave sobre el documento usando un secreto compartido. Esta técnica no requiere el uso de criptografía de claves públicas o de encriptación.

Este mecanismo es útil para casos donde se permite a las partes identificarse de una manera segura en una transacción. La provisión de este mecanismo es motivada porque la acción de firmar una transacción debe ser explícita y concisa para el usuario, mientras muchas necesidades de autenticación pueden ser puestas con un mecanismo más liviano que conserva las ventajas de criptografía de claves públicas para el intercambio de claves.

ROBOTS

9. ROBOTS

El WWW es un servicio muy dinámico, diverso y descentralizado, lo que dificulta la navegación. Los usuarios para encontrar la información deseada deben atravesar un gran número de links. Debido a esto, se vio potenciada la necesidad de crear herramientas de búsqueda que faciliten la ubicación de información mediante palabras claves.

Para lograr esto, se implementaron robots que realizan esta tarea.

Los robots son programas que atraviesan el espacio Web recuperando un documento y recursivamente recuperando todos los documentos que son referenciados dentro de este. Estos programas muchas veces son llamados Spiders, Wanderers, Worms [4].

Estos robots hablan el lenguaje nativo de Web (HTTP) y usan éste para recuperar documentos web desde servidores remotos. El propósito de los robots es descubrir nuevos servidores, indexar la información del web para futuras búsquedas por palabras claves y hacer el mantenimiento de un servidor en forma automatizada.

La principal ventaja de implementar este tipo de robots es que no requieren la participación activa del hombre; además de ser más eficiente y obtener un caudal de información mayor a la que puede obtener un grupo de webmaster que actúan e intercambian información en conjunto.

Este tipo de robots tiene como función principal atravesar el espacio web recolectando información para ser posteriormente almacenada en bases de datos. Estos datos luego serán indexados, facilitando el acceso a ellos a través de una herramienta de búsqueda.

9.1 USO DE LOS ROBOTS

Pueden ser usados para distintas tareas :

- **Análisis Estadístico** : descubren y cuentan la cantidad de Servidores Web. Pueden también calcular el número medio de documentos por servidor; el tamaño de las páginas, el grado de interconectividad.
- **Mantenimiento :** el principal objetivo de robots desarrollados para mantenimiento es chequear links, para determinar si estos aún siguen activos, han sido movidos o bien si esas páginas se han eliminado. Otro

objetivo de este tipo de robots es chequear la estructura HTML de los documentos.

- **Descubriendo Recursos :** la mayoría de los robots implementados realizan esta tarea. Además resumen gran parte del espacio Web y lo almacenan en bases de datos, las cuales se pueden consultar por medio de herramientas de búsqueda. Estas bases de datos serán actualizadas en forma automática y en intervalos regulares de tiempo, solucionando así el problema de inconsistencia en los links.

El usuario Web puede combinar la navegación con el uso de las herramientas de búsqueda para encontrar la información deseada.

- **Mirroring :** es un técnica común para replicar una estructura de información a otro servidor. La complejidad de este tipo de robots es que los hiperlinks usados para apuntar a una página Web original deben ahora apuntar a la nueva página Web copiada. También los links relativos deben ser tenidos en cuenta ya que si no se espeja la página a la que están apuntando se deberá cambiar ese link relativo a un link absoluto apuntando a la página original.

9.2 COMO FUNCIONAN LOS ÍNDICES

Con una magnitud de páginas Web medida en millones de páginas se hace difícil encontrar la información que se busca, necesitándose herramientas de búsqueda automatizadas para poder encontrarla.

Al día de hoy una de las herramientas más populares (INFOSEEK GUIDE) dice que puede realizar hasta 7 millones de búsquedas diarias.

Las herramientas de búsqueda Web intentan crear un registro detallado del Web usando agentes de software automatizados que recorren de URL a URL, visitando todos los sitios en áreas públicas del Web y registran las direcciones. Todas las herramientas de búsqueda manejan estos pasos esencialmente de la misma manera.

Desde este punto en adelante las diferentes cosas que hacen las distintas herramientas marca la diferencia en calidad y cantidad de resultados de la búsqueda. Algunos envían un software robot a visitar todos los sitios y registrar el texto completo de toda la página. Otros primero analizan la dirección en una base de datos para determinar la popularidad del sitio (típicamente dada por la cantidad de links que apuntan al sitio) y luego envían un soft para registrar información solo de estos sitios construyendo un resumen de los contenidos de las páginas.

Además la base de datos debe ser reconstruida, refrescada o actualizada regularmente para tener la información corriente.

La lógica de búsqueda usada para extraer información de la base de datos es otra componente importante de estas herramientas.

Las herramientas deben poder encontrar sitios Web que coincidan con un criterio de búsqueda y ordenen los resultados acorde al grado de relevancia. El tiempo de búsqueda actual de un query típico es medido en fracción de segundos.

9.2.1 WEBCRAWLER [http://www.webcrawler.com/]

Es una herramienta para descubrir recursos en WWW, creada por Brian Pinkerton de la Universidad de Washington en Seattle.

Esta provee una rápida manera de encontrar recursos manteniendo un índice del Web que puede ser consultado por tópicos específicos.

Realiza las siguientes funciones :

- ✓ construye índices de Web
- ✓ navega automáticamente

Trabaja de la siguiente manera :

✓ usa un sistema de indexado full-text basado en contenidos para proveer un índice de alta calidad.

✓ usa la estrategia de búsqueda Breadth-First para crear un índice amplio, repartiendo la carga entre servidores y asegura que siempre un servidor con contenido útil tenga varias páginas representadas en el índice.

✓ intenta incluir tanta cantidad de servers web como sea posible, lo hace en una manera amigable tratando de no sobrecargar los servidores Web con requerimientos en ráfaga. También respeta el standard de exclusión de los robots.

WebCrawler descubre nuevos documentos aprendiendo sus URL's. Comienza con un conjunto conocido de documentos, examina los links salientes de él, siguiendo uno de los links que dá con un nuevo documento, y entonces repite todo el proceso. Simplemente explora el espacio Web como un gran grafo dirigido, usa un algoritmo para recorrer el grafo que realiza la siguiente secuencia de acciones :

a) descubre un nuevo documento.

b) marca el documento como que está siendo recuperado.

c) descubre los links salientes.

d) indexa el contenido del documento.

La arquitectura del WebCrawler está compuesta por cuatro componentes :

1. **Motor de Búsqueda :** dirige las actividades del WebCrawler y es el responsable de decidir cuales documentos nuevos explorar e iniciar su recuperación.
2. **Agentes :** son los responsables de recuperar los documentos desde la red a la dirección del motor.
3. **Base de Datos :** esta maneja el almacenamiento persistente de los datos, los links entre los documentos y el índice full-text.
4. **Servidor de Consultas :** implementa el servicio de consulta provisto por Internet. El usuario ingresa palabras claves y dispara la consulta.

9.2.2 LYCOS [http://www.lycos.com/]

Fue un proyecto encabezado por el Dr. Michael Mauldin del "Center for Machine Translation de la Universidad de Carnegie Mellon", anunciado al mundo en agosto de 1994.

Esta herramienta ayuda a los usuarios a localizar documentos Web conteniendo palabras claves especificadas por el usuario.

Lycos define el espacio de Web de esta manera :

- ✓ espacio http
- ✓ espacio ftp
- ✓ espacio gopher

ignorando los espacios de :

- ✓ bases de datos Wais
- ✓ usenet News
- ✓ espacio Mailto
- ✓ servicios Telnet
- ✓ espacio de archivos locales

ignorando también archivos con extensiones como : AU, BIN, DAT, EXE, FLI, GIF, GZ, HDF, JPG, MAC, MPEG, PS, TAR, TGA, TIF, UU, WAV, Z, ZIP.

Para reducir la cantidad de información que necesita almacenar, extrae de los documentos la información de : títulos, headers, subheaders, 100 palabras más importantes, primeras 20 líneas, tamaño en bytes, número de palabras.

Para seleccionar las 100 palabras más importantes usa un algoritmo de medición que considera la ubicación y frecuencia de las mismas, asignando cierto puntaje a cada una de ellas de acuerdo a su ubicación en el texto, (si está en el título, en el primer párrafo, etc).

Para moverse en el espacio Web, Lycos usa un esquema probabilístico que salta de servidor en servidor. Esto evita sobrecargar cualquier servidor con una ráfaga de requerimientos y también permite dar preferencia a URL's más informativos. Los pasos básicos de este algoritmo son :

1. cuando un URL es alcanzado, y su documento recuperado, se analizan las nuevas referencias a URL's, generando una cola interna.
2. para elegir el próximo URL a explorar, se hace una elección random entre las referencias http, ftp y gopher de la cola, basada en la preferencia más alta.

Lycos tiene una preferencia por los URL's cortos, los cuales generalmente son la raíz del árbol de documentos. También respeta el standard de exclusión de robots, se identifica como "Lycos" indicando esto en el campo HTTP USER-AGENT del mensaje HTTP.

Dada la fenomenal tasa de crecimiento de Web, la corriente generación de Spiders está presionada por los requerimientos del hardware. Una nueva generación está sucediendo a la actual, enfocando hacia el incremento de performance y escalabilidad.

Estos Spiders usan una arquitectura sofisticada de diseminación y acumulación de información. Todos los Spiders de la nueva generación son fuertes, rápidos y activos que pueden sobrevivir bien al rápido crecimiento del dinámico y masivo Web.

Ejemplo de estos son : Altavista e InfoSeek Guide.

PROXY

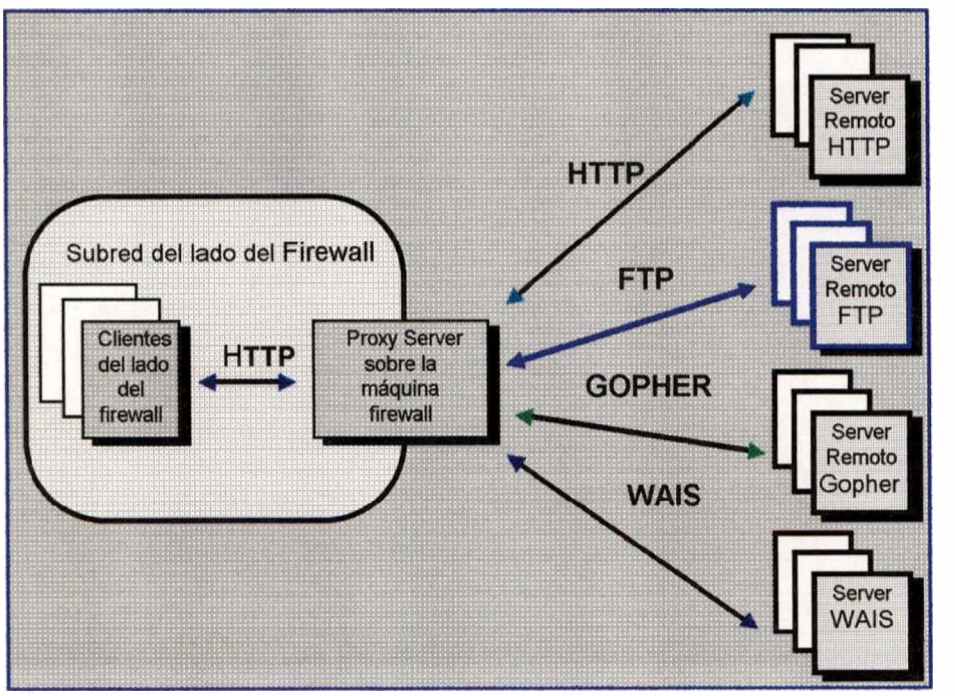
10. PROXY

Proxy Server es diseñado para proveer acceso al Web a personas que están sobre redes cerradas (ocultas), que solo pueden acceder a Internet a través de una máquina Firewall, también para mejorar la performance de las comunicaciones, estas son mejoradas porque el servidor almacena localmente páginas accedidas frecuentemente, generando un proceso llamado **"cacheo"**. Esto hace útil al proxy aún para usuarios que tienen acceso directo a Internet.

En la actualidad existen servidores de hipertexto que son capaces de correr como proxy, proveyendo acceso externo a HTTP, GOPHER, WAIS y FTP.

Los clientes no pierden funcionalidad pasando a través de un proxy, con excepción de procesos que puedan hacerse por protocolos de Web no nativos como Gopher y Ftp.

Las organizaciones muchas veces usan Proxy Server como parte de su estratégia. El mejoramiento de la performance con Proxy brinda además un beneficio de costo - eficiencia adquiriendo un ancho de banda adicional porque este posibilita que páginas remotas sean servidas localmente, por lo tanto el tráfico y las comunicaciones sobre la Internet pueden ser sustancialmente reducidas.



Un proxy es un servidor especial de HTTP que tipicamente corre sobre una máquina firewall. El proxy espera un requerimiento desde la red interna a la máquina firewal, y envía el mismo hacia el host remoto (red externa del otro lado del Firewall), lee la respuesta y la envía hacia el cliente que hizo la solicitud.

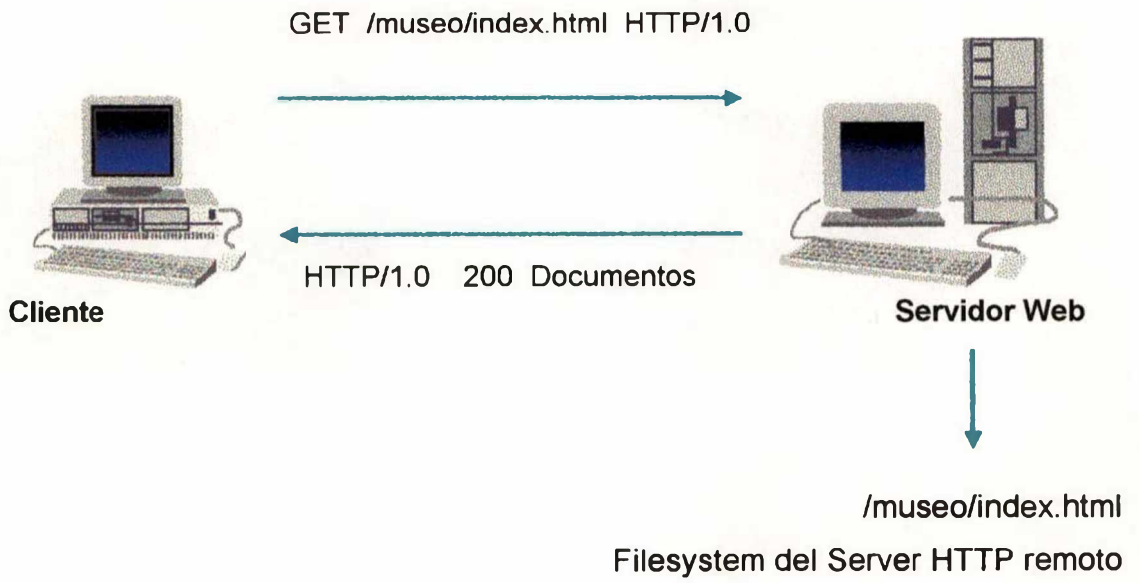
El proxy es usado por todos los clientes dentro de una subred dada, esto es posible por que el proxy hace un cacheo eficiente de los documentos que son solicitados por los clientes.

Los clientes Web más populares ya soportan proxy. Esto baja el tráfico sobre la red y costo ya que muchos de los documentos son tomados desde la cache local una vez que el requerimiento inicial ha sido realizado.

10.1 DETALLES TÉCNICOS

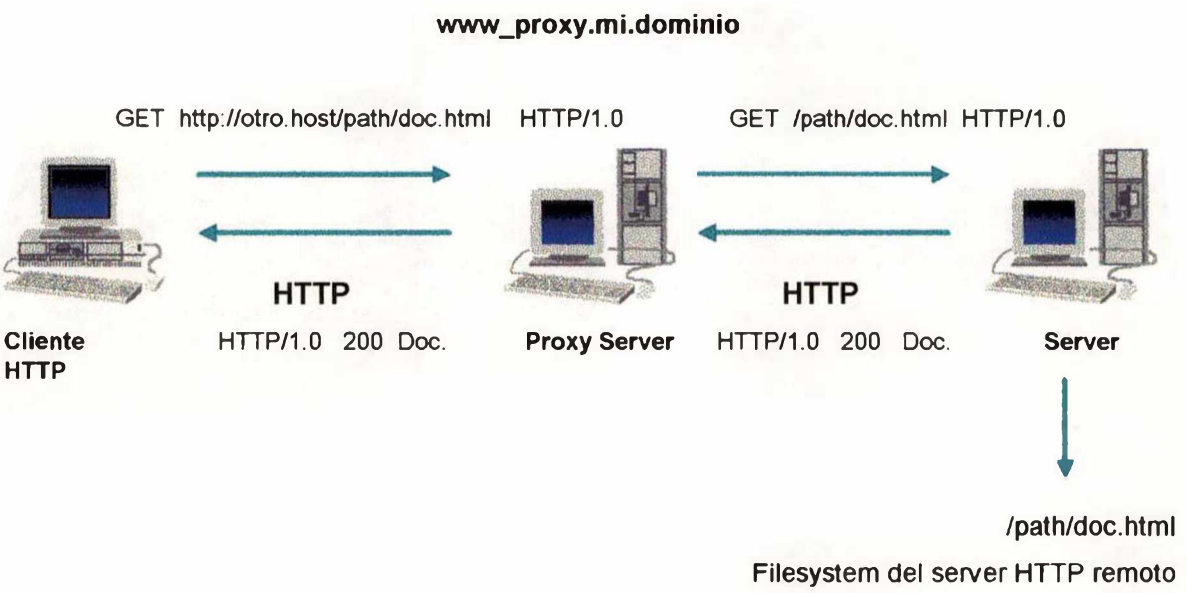
Cuando un requerimiento normal HTTP es realizado por un cliente, el servidor toma solo el path y la porción clave del URL requerido.

Otras partes como el nombre que especifica el protocolo " http:// " y el hostname son descartados por el servidor http, este conoce que es un server http, y conoce la máquina host sobre la cual esta corriendo. El path requerido especifica el documento sobre el filesystem local del server, o algún otro recurso disponible desde este server.



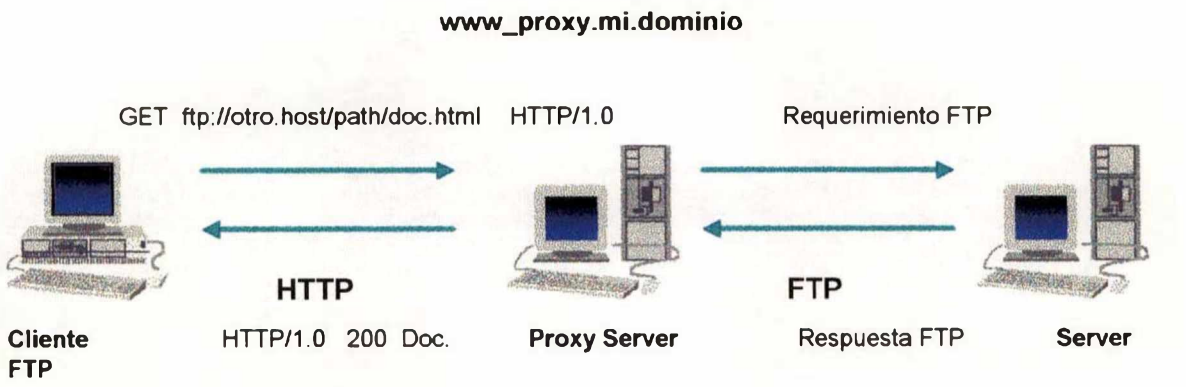
Cuando un cliente envía un requerimiento a un proxy server, la situación es ligeramente diferente. El cliente siempre usa http para transacciones con el

proxy server, aun cuando accede a un recurso servido por un server remoto usando otro protocolo como Gopher o FTP. Sin embargo en vez de especificar solo el camino y la posible clave de búsqueda al proxy server, se especifica el URL completo.



http://otro.host/path/doc.html

http_proxy= http://www_proxy.mi.dominio/



ftp://otro.host/path/doc.html

ftp_proxy= http://www_proxy.mi.dominio/

De esta manera el proxy server tiene toda la información necesaria para hacer el requerimiento al server remoto especificado en el URL, usando el protocolo también especificado en el URL.

10.2 RASGOS DEL LADO DEL CLIENTE

Muchos clientes son construidos en base a las libwww, las librerías comunes de WWW, que manejan los distintos protocolos de comunicación usados en el Web, llamados HTTP, FTP, GOPHER, NEWS, WAIS [3].

El soporte del proxy es manejado automáticamente por clientes usando las librerías libwww. Las variables de ambiente son usadas para controlar las librerías. Hay una variable de ambiente individual por cada acceso al protocolo : ej. http_proxy, ftp_proxy, gopher_proxy, y wais_proxy. Estas variables son seteadas para que el URL apunte al proxy server que va a servir el protocolo requerido.

```
Ej :  ftp_proxy = http://www_proxy.mi.dominio
      export ftp_proxy
```

Cuando una variable de ambiente para un protocolo dado es seteada, el código de las libwww causa que una conexión siempre sea hecha a través del proxy antes que con el servidor remoto.

Otra diferencia en el protocolo entre el cliente y el proxy es que el URL requerido tiene que ser un URL completo cuando este es requerido desde el proxy server. Esta es la única diferencia entre una transacción normal y una con proxy.

El soporte de proxy es implementado solo para HTTP/1.0, sobre el servidor, así que los clientes deben usar este protocolo. Actualmente esto no es un problema ya que casi todos los clientes usan HTTP/1.0 .

10.3 RASGOS DEL LADO DEL SERVIDOR

El proxy server tiene la virtud de actuar como server y como cliente. Es un server cuando acepta requerimientos HTTP [12] desde clientes conectados a este, pero actua como un cliente cuando tiene que conectarse con un server remoto a fin de conseguir los documentos solicitados por su cliente. Los headers de los campos pasados al proxy desde el cliente son usados sin modificarse cuando el proxy se conecta al server remoto así que el cliente no pierde funcionalidad cuando pasa a través del proxy.

Un proxy server completo puede hablar todos los protocolos Web, los más importantes son HTTP, FTP, GOPHER, WAIS. Existen proxies que manejan un solo protocolo como HTTP, pero el cliente Web podría necesitar acceder a otro proxy server para que maneje el protocolo deseado.

Cern_httpd, servidor de HTTP, tiene una arquitectura única, el es corrientemente el único server HTTP que está construido en base a las librerías comunes de WWW, que son usadas también por los clientes Web. A diferencia de otros servidores HTTP que solo entienden el protocolo HTTP, cern_httpd es capaz de hablar todos los protocolos Web igual que los clientes Web, que tienen todos los protocolos implementados por las libwww (www common library).

10.4 CACHING

La idea básica del cacheo es la de almacenar los documentos alcanzados, en archivos locales para uso futuro, de esta manera no sería necesario conectarse al servidor remoto para un próximo acceso.

El mecanismo de cacheo es basado en disco y persistente, que significa sobrevivir a procesos de arranque del proxy, así como también de la máquina donde esta corriendo el mismo.

Para que un documento sea cacheado deberá cumplir ciertas condiciones:

- el método del pedido debe ser GET [12], para otros métodos el cacheo no tiene sentido o no es importante el esfuerzo.

- la solicitud no sea un query. Esto es por que los query producen resultados dinámicos, no tienen un tiempo de expiración.

- que el documento no expire rapidamente o tenga un corto tiempo de vida.

- el recurso requerido sea servido por un servidor HTTP, FTP, GOPHER.

HTTP provee una manera de indicar que sitios o documentos pueden ser cacheados, seteando su URL, o bien que URL no cachear, esto se hace a través de directivas en el servidor.

10.4.1 TIEMPO DE VIDA EN LA CACHE

El protocolo HTTP especifica un objeto Expire, este es especificado por el server remoto, él es siempre usado, el cacheo es considerado para ser actualizado hasta que este tiempo sea alcanzado.

Especialmente si el documento expira en un corto tiempo, este nunca será cacheado, esto ahorra recursos, por que es poco probable que el mismo archivo sea requerido nuevamente en pocos minutos.

Los documentos con una línea invalida en el header Expire nunca serán cacheados. El uso del header Expire es la única manera correcta para determinar si un documento puede ser cacheado, los documentos sin este campo nunca serán cacheados. En la práctica es raro que los servidores HTTP den este dato,

entonces es necesario usar algoritmos para calcular alguna fecha de expiración para el documento que no lo tenga.

El campo Expire es parte solo del protocolo HTTP, no de otros protocolos WWW.

Muchos servidores de HTTP dan fecha de ultima modificación para archivos que son retornados desde su filesystem local. Esto puede ser usado para aproximar el tiempo de vida util; los archivos que han sido recientemente modificados pueden probablemente actualizarse nuevamente, de otra manera archivos que no han sido modificados por un largo tiempo, podrían cambiar repentinamente.

CERN maneja esto vía el factor LM (last modified), este especifica la fracción de tiempo por el cual el archivo permanecería válido. Generalmente se le aplica un factor 0.1 a la última modificación. El factor LM puede ser especificado diferentemente para diferentes URL´s macheando distintos patrones.

Cuando el servidor remoto no da la fecha de expiración o de la última modificación, un valor por default es usado, este valor es configurado por patrones de URL.

Los documentos producidos por scripts desde interfaces CGI, no tienen campo de última modificación, entonces el valor es puesto a cero, por eso es que casi nunca son cacheados, por que a una nueva consulta el contenido usualmente cambia.

10.4.2 INTERVALO GARANTIZADO DE REFRESCO EN LA CACHE

A veces es vital tener información actualizada de un determinado sitio, no importando el tiempo de expiración especificado por el server remoto o calculado por el proxy. En CERN es posible configurar un intervalo de actualización de cache para URLs que machean con un patrón dado. Esto causará que el proxy server chequee que el archivo esté aún al día sin mas que el máximo de tiempo permitido a pasar el último chequeo.

Como un caso especial, especificando el intervalo de refresco a 0 todo acceso a la cache causará que el chequeo sea hecho desde el server remoto. Esto es ideal para usuarios, quienes necesiten tener siempre la version al día absoluto.

Cuando es esencial que el documento recuperado sea actualizado, es necesario contactarse al server remoto por cada pedido GET. El protocolo HTTP ya contiene el método HEAD para pedir información sobre el documento, pero no el documento en sí mismo. Esto es útil para chequear si el documento ha sido modificado desde el último acceso.

Sin embargo, en caso de que el documento halla sido cambiado, este método resultará ineficiente por que tendría que hacer una segunda conexión al server remoto para recuperar el documento a través de un pedido GET.

10.4.3 CAPACIDAD MÁXIMA DE USO DE LA CACHE

CERN httpd da una cierta cantidad de espacio en disco para la cache. Si el límite especificado es alcanzado, httpd realiza un " garbage collection ", removiendo archivos cacheados que no hallan sido accedidos recientemente, o que halla expirado su tiempo.

Si el espacio en disco es un factor crítico, esto es, si es deseable que httpd ponga la cache en el mínimo, este siempre remueve todos los archivos expirados durante el " garbage collection " . Esto es el default. Sin embargo, es perjudicial. Muchas veces archivos que han expirado, no han cambiado y un simple requerimiento GET puede ser hecho para verificar esto y lo actualiza nuevamente.

U.N.L.P.

11. Aplicación de la tecnología
Web Server U.N.L.P.

Hemos desarrollado distintos trabajos sobre el Web Server de U.N.L.P., cada uno de ellos con diferentes perfiles.

11.1 Prototipo de cliente HTTP

El motivo principal que nos llevó a desarrollar una herramienta con estas características fue la de poder recorrer el espacio Web pero de una manera más automática, automatizando tareas para la recuperación de páginas.

Observando el ambiente Web vemos que la navegación es dificultosa ya que el Web es dinámico, descentralizado y diverso, encontrar información dentro de este medio se torna muy complejo si el usuario no tiene tanta experiencia. Usualmente los usuarios recorren el espacio Web en búsqueda de información, siguiendo los links de las páginas. El gran crecimiento de este espacio hace más complicada la búsqueda, por lo que se trató de agilizar la tarea de ubicar recursos dentro de la red.

Aprovechando el tendido de red dentro de la U.N.L.P., que conecta a distintas facultades en diferentes puntos geográficos dentro de nuestra ciudad, e integra distintos medios de comunicación (linea punto a punto, enlaces de radio, pares telefónico), creímos conveniente realizar un robot que tendría como función recorrer el espacio Web dentro de la Universidad.

Este robot tiene como finalidad conectarse a los distintos host que se encuentran bajo el dominio de la U.N.L.P. (unlp.edu.ar) y verificar si responde al servicio WWW.

Con este fin, desarrollamos en lenguaje C para entorno Linux, un prototipo de cliente que toma una dirección IP o nombre de host desde un archivo previamente generado con todas las direcciones que deseamos testear.

Como primera medida verificamos que la dirección sea una dirección válida de ser así verificamos si este host esta brindando el servicio de Web y de obtener una respuesta positiva averiguamos el port por el cuál atiende este servicio.

Teniendo todo esto verificado, creamos una interface socket y solicitamos conectar nuestro cliente al servidor en cuestión.

Una vez logrado esto le enviamos al servidor un requerimiento usando para este fin el protocolo HTTP y aguardamos la respuesta.

Lo que le solicitamos al servidor es la página principal, una vez obtenida esta se puede trabajar con ella.

Con este archivo se podría trabajar para poder hacer :

- chequeo y consistencia de links, esto ayuda a los creadores de la páginas y a los webmaster a mantener la estructura de información Web atravezando automaticamente los links, chequeando su respuesta y verificando su estado.
- descubrir y mantener estadísticas de la cantidad de servidores que existen en esta Intranet.
- recuperar y almacenar la información que se expone en cada sitio a fin de poder crear una base de datos centralizada para brindar el servicio de índice o búsqueda por palabra clave.
- para poder espejar otros sitios en horario sin tanto tráfico en la red para evitar la sobrecarga, todo esto a fin de que los clientes de esa Intranet consulten información en forma local. Consiguiendo con esto no sobrecargar la red.

11.2 Museo [http://www.unlp.edu.ar/museo/]

El objetivo de desarrollar páginas para un sitio como el Museo de Ciencias Naturales de La Plata fue pensado como medio de difusión, al mismo nivel de los museos más destacados del mundo, a través de Internet, con mucha información, detallando y mostrando al mundo todo el material con que cuenta este lugar así como también la gran variedad de colecciones, ubicándolo de esta manera en uno de los más importantes a nivel mundial.

La posibilidad de una presentación fácil de usar y agradable para diversos usuarios hace de este sitio una herramienta muy potente que lo posicionan en un lugar de privilegio.

En este trabajo representamos la información de las 23 salas que pueden ser visitadas en el museo, cada una de estas salas está representada por una página html conteniendo un mapa esquemático que muestra la ubicación de la sala en el plano del museo, completa información sobre las piezas que se exhiben en ella y sobre la ciencia que en ella se expone (información textual).

La mayoría tiene fotos de las piezas que se exhiben, generando de esta manera un gran volúmen de imágenes, se observa en ellas un trabajo fundamental de reducción de tamaño en bytes conservando una buena definición, situación que nos permite lograr una mayor performance de la red. Este sitio demandó un gran trabajo de edición, compaginación, scaneo y tratamiento de

imágenes, generando así 32 archivos con extensión .jpg que totalizan 910.720 bytes y que representan las fotos de las piezas expuestas en las salas; 29 archivos con extensión .gif totalizando 150.200 bytes representando los mapas esquemáticos de cada sala e iconos, y 43 archivos .htm con información textual que totalizan 150.712 bytes.

El conjunto de páginas del Museo de La Plata representa un Web clásico, refleja una unidad independiente y completa de información. En este Web se implementa un recurso muy potente como lo son los mapas sensitivos. Esta característica permite definir zonas dentro de una imagen o gráfico las cuales referencian a distintos hiperlinks. Aplicamos el recurso de mapa sensitivo para representar el plano del edificio, dividido por sus diferentes plantas y salas, simulando el esquema de recorrido del museo. Este mapa sensitivo nos permite acceder a la información de cada una de las salas, simplemente clickeando sobre la zona referenciada por esta. Tal imágen sensitiva esta implementada por medio de un script que se ejecuta en el servidor.

La idea de este proyecto fue simular un recorrido semejante al que una persona realizaría visitando el museo, agregandole la posibilidad de un recorrido por área de investigación : Geología, Etnografía, Paleontología, Zoología, entre otras.

Creemos haber logrado el objetivo mencionado anteriormente al verificar que importantes museos de ciencias naturales del mundo han representado su información en forma similar. Otra medida que acredita el logro de nuestro objetivo es el análisis de los **logs** (que nos muestra la cantidad de accesos a este sitio), la cantidad de mail´s recibidos pidiendo más información a cerca de temas específicos, y buenas opiniones sobre las páginas e información brindada son parámetros a considerar a la hora de evaluar los resultados

Cabe aclarar que este desarrollo fue realizado entre los meses de agosto y octubre del año 1995, a partir de esta fecha se puso a consideración de las autoridades del Museo, que realizaron correciones sobre el texto y permitieron su publicación en internet a partir del mes de febrero de 1996.

11.3 Facultad de Ciencias Exactas [http://www.unlp.edu.ar/exactas/]

Respecto a la páginas generadas para la Facultad de Ciencias Exactas, se expresa una organización de la información similar al usado al generar los informes del Honorable Consejo Académico de la Facultad.

El principal motivo del desarrollo de este Web Site estuvo orientado a brindarle transparencia y amplia difusión a los informes del H.C.A., aplicando un recurso nuevo, poderoso y fácil de usar para consultas de información actual e histórica.

Cada página de este Web contiene información sobre los temas tratados en una reunión del H.C.A. Las reuniones tratan temas pendientes de reuniones anteriores, temas agendados y temas presentados sobre tablas. Dentro de la diversa variedad de temas encontramos :

- movimiento de personal
- renovación de cargos
- llamadas a concursos
- dictámenes de Comisión Asesora (por distintas carreras)
- informes del Sr. Decano
- toma de conocimiento de convenios, etc.

En estas páginas hemos implementado una herramienta de búsqueda que nos facilita la localización de información referente a los puntos tratados y a los resueltos en las reuniones. Esta herramienta fue desarrollada con un script implementado en AWK que es un lenguaje interpretado para UNIX.

Se ha realizado esta tarea sobre las reuniones ocurridas entre los meses de septiembre de 1995 a agosto de 1996, quedando como responsabilidad de la Facultada de Ciencias Exactas la continuación de esta tarea.

11.4 Ciencia y Técnica [http://www.unlp.edu.ar/secyt/]

El objetivo principal de este sitio es difundir las actividades de la Secretaría de Ciencia y Técnica. Fue construido con la premisa principal de ser útil a todos los docentes e investigadores de la Universidad Nacional de La Plata, y de otras Universidades. Se brinda información general sobre evaluadores, reglamento para el otorgamiento de becas, FOMEC, MUTIS, etc..

Consideramos que la utilidad de este sitio a docentes e investigadores de la U.N.L.P. está dada por la posibilidad de obtener el acceso a distintos formularios, cronogramas, convocatorias, etc.

El beneficio de investigadores de otras universidades es la posibilidad de contar con información de todos los investigadores categorizados con que cuenta nuestra universidad como posibles evaluadores de distintos proyectos a ser presentados.

Este sitio refleja en forma contínua la información sobre los vencimientos para la presentación de documentación, plazos para diferentes convocatorias y toda otra novedad de interés para los docentes e investigadores generada en el ámbito de la Secretaría.

También se diseñaron páginas apuntando al interés de los docentes e investigadores egresados que estén fuera del país y tengan interés de consultar que propuestas hay en la actualidad en la U.N.L.P. y generar a través de este Web un medio de comunicación.

Fue aprobado por el rector de la U.N.L.P., esto se está implementando en la actualidad y hay cuatro tésis de las Facultades de Física y de Astronomía preparándose para incorporarlas.

También se trató y aprobó el proyecto de generar una Revista Electrónica con información científica a partir de publicaciones realizadas por investigadores pertenecientes a la U.N.L.P.

El logro sobre este sitio es estar informatizando una gran base de información. El desarrollo es incremental, seleccionando los puntos más solicitados para ser explotados, por tal motivo este Web Site debe evaluarse separadamente por cada servicio que brinda.

GLOSARIO

GLOSARIO

ANSI

(American National Standards Institute) Esta organización es responsable de aprobar los estandares en muchas áreas, incluyendo las áreas de computadoras y comunicaciones. Los estandares que son aprobados por esta organización son llamados generalmente ANSI standard. ANSI es un miembro de la ISO.

Archie

Una herramienta para encontrar archivos almacenados en sitios FTP anónimo. Usted necesita conocer el nombre exacto del archivo o un substring del mismo.

ARPANET

(Advanced Research Proyects Agency Network) La red precursora de Internet. Desarrollada a fines de los 60's y principios de los 70's por el Departamento de Defensa de US. como un experimento de una WAN que sobreviviera a una guerra nuclear.

ASCII

(El Código Estándard Estadounidense para el Intercambio de Información) Es el standart de código númerico para representar en computadores todas las letras Latinas, números, signos de puntuación, etc. Hay 128 carácteres ASCII que pueden ser representado por un 7 dígitos binarios desde el dígito: 0000000 hasta el 1111111.

Backbone

Una serie o línea rápida de conexiones que forma un sendero importante dentro de una red. El término es relativo como un espinazo en una red pequeña probable será mucho más pequeño que muchos no - espinazo raya en una red grande.

Bit

(El Dígito Binario) Un dígito único numerado en base - 2, en otras palabras, un 1 o un cero. La unidad más pequeña de datos computerizados.

BITNET

Una red de sitios académicos que provee los servicios de correo electrónico y transferencia de archivos utilizando un protocolo de store-and-forward.

Bps

(Bits por segundo) Medida de velocidad de datos que se mueven de un lugar a otro. Un "28.8 modem" puede mover 28,800 bits por segundo.

Browser

Un programa cliente (software) usado para explorar diversos tipos de recursos Internet.

Bytes

Un conjunto de bits que representa un carácter único. Comúnmente hay 8 Bits en un byte, a veces más, dependiendo de como se haga la medida.

Cliente

Un programa de software usado para llamar y obtener datos desde un programa Servidor sobre otra computadora, frecuentemente a través de una gran distancia. Cada programa Cliente se diseña para trabajar con uno o más tipos de Servidores, y cada Servidor requiere un tipo específico de Cliente. Un "Web Browser" es un tipo específico de Cliente.

Cyberspace

Término usado actualmente para describir la gama entera de los recursos de información disponible mediante redes de computadora.

DNS

(Domain Name System) Nombre único que identifica un sitio Internet. Los Nombres de Dominio siempre tienen 2 o más partes, separado por puntos. La parte sobre la izquierda es el más específica, y la parte sobre el derecho es el más general. Una máquina determinada puede tener más de una de Name Domain pero un Name Domain determinado indica a una única máquina. Comúnmente, todas las máquinas sobre una Red determinada tendrán lo mismo en la porción derecha de su Name Domain, por ej.

anubis.unlp.edu.ar
isis.unlp.edu.ar
ayelen.fisica.unlp.edu.ar

Es posible también que exista un Name Domain sin estar conectado a una máquina real. Este se hace frecuentemente para que un grupo o empresa tenga una dirección e-mail sin tener establecido un sitio Internet. En este caso, algún máquina debe manejar el correo en nombre del Name Domain dado.

E-mail

(El correo electrónico) Mensajes, comúnmente texto, enviado desde una persona a otra por medio de la computadora. E-mail puede también ser enviado automáticamente a un número de direcciones (Mailing List).

Ethernet

Un método muy común de conectar computadoras en un LAN. Ethernet manejará sobre 10 Mbits por segundo y puede usarse con casi cualquier tipo de computadora.

FAQ

(Preguntas Frecuentemente pedidas) FAQS son los documentos que listan y contestan las preguntas más comunes hechas sobre un tema particular. Hay centenares de FAQS sobre diversos temas. FAQS son escritos comúnmente por la gente quien se ha cansado de contestar la misma pregunta una y otra vez.

Finger

Una herramienta Internet para ubicar gente sobre otro sitio Internet. Se usa comunmente para ver si una persona tiene una cuenta en un sitio Internet particular. Muchos sitios no permiten el uso de esta herramienta, pero muchos otros si.

Firewall

Una combinación de hardware y software que separa una LAN en dos o más partes con propósitos de seguridad.

FTP

(File Transfer Protocol) Un método muy común de mover archivos entre dos sitios Internet. FTP es una manera especial a hacer un login en otro sitio Internet con el propósito de recuperar y / o enviar archivos. Hay mucho sitios Internet que han establecido lugares públicamente accesibles con material que puede obtenerse usando FTP, usando el nombre de cuenta "anónymous" , así estos sitios se llaman FTP anonimos .

Gateway

El significado técnico es una pieza de hardware o de software que hace una traduccion entre dos protocolos disímiles.

Host

Cualquier computadora sobre una red que puede tener servicios disponibles a otras computadoras sobre la red. Es bastante común tener un HOST proveyendo varios servicios, tal como WWW y USENET.

HTML

(HyperText Markup Language) El lenguaje usado para crear documentos con Hypertexto, usado sobre el WWW.

HTTP

(HyperText Protocolo de Transporte) Protocolo para mover archivos de hipertexto a través del Internet. Requiere un programa Cliente y un programa Servidor , es el protocolo usado en WWW.

Hypertext

Generalmente, cualquier texto que contiene **links** a otros documentos.

Internet

(Con I mayúscula) Colección de redes interconectadas que usan el protocolo TCP/IP y que nació a partir de ARPANET.

Internet

(Con i minúscula) Dos o más redes interconectadas.

Intranet

Una red privada dentro de una compañía u organización que usa los mismos servicios que usted encontraría sobre Internet, pero que únicamente es para el uso interno.

IP Número

Un número único sobre la red, que consiste de 4 partes separado por puntos, p. ej.

165.113.245.2

Cada máquina que está sobre Internet tiene un único número IP, si una máquina no tiene un número IP no está sobre el Internet.

Java

Java es un nuevo lenguaje de programación inventado por Sun Microsystems que se diseñó específicamente para escribir programas que pueden sin riesgo transmitirse a su computadora mediante Internet e inmediatamente correrlos. Usando pequeño programas de Java (llamados" Applets"), las páginas Web pueden incluir funciones tales como animaciones, calculadores, y otros trucos.

LAN

(Local Area Network) Una red de computadora limitada al área inmediata, comúnmente el mismo edificio o piso de un edificio.

Login

El nombre de una cuenta de un usuario para poder acceder a computadoras con cierto nivel de seguridad.

MIME

(Multipurpose Internet Mail Extensions) El standard para incorporar archivos que no son de texto a los mensajes de correo. Estos archivos serían gráficos, hojas de cálculos, sonido, etc. Cuando uno de estos archivos se envían usando MIME ellos se convierten (se codifican) en texto - aunque el texto resultante no sea legible.
Muy usado por servidores Web para identificar archivos que son enviados hacia clientes Web.

MODEM

(MOdulador DEModulador) Un dispositivo que usted conecta a su computadora y a una línea de teléfono, esto permite a la computadora poder comunicarse con otras computadoras mediante el sistema de teléfono.

Mosaic

El primer browser WWW que fue disponible para Macintosh, Windows, y UNIX todos con la misma interfase.

Netscape

Un WWW browser y el nombre de una compañía. El browser se baso originalmente en el programa de Mosaic que desarrolló el Centro Nacional para Supercomputing de Aplicaciones (NCSA).

Netscape ha crecido rápidamente y se reconoce ampliamente como el mejor y más popular browser WWW. Netscape Corporation también produce el software de servidor web.

Netscape agrego mejoras importantes en la velocidad y en la interfase sobre el browser Mosaic, y también generó la discusión por crear nuevos elementos para el lenguaje HTML usado Web (pero las "extensiones" Netscape a HTML no son apoyadas universalmente.).

El autor principal de Netscape, Marc Andreessen, se alejo de NCSA junto a Jim Clark, y ellos fundaron la compañía que se llamó Netscape Corporation Comunications.

Network

Al conectar 2 o más computadoras para que puedan compartir recursos, se tiene una red de computadora. Conecte 2 o más redes juntas y usted tiene una internet.

NIC

(Networked Centro de Información) - - Generalmente, cualquier oficina que maneja información sobre una red. El más famoso de estas sobre Internet es el InterNIC, lugar donde los nombres de dominio sobre intenet se registran.

Nodo

Cualquier computadora conectada a una red.

Packet Switching

El método usado para mover datos sobre Internet. Así todos los datos que vienen de una máquina se dividen en pedazos, cada pedazo tiene la dirección de donde vino y a donde va. Esto permite pedazos de datos desde muchas fuentes diferentes se mezclen sobre las mismas líneas, y se clasifiquen y dirijan por rutas diferentes a través de máquinas especiales a lo largo de la red. Así mucha gente puede usar las mismas líneas a la vez.

Password

Un código usado para acceder a un sistema . Las password que son buenas contienen letras y números y no son combinaciones simples. Una pasword buena podría ser:

Hot$1-6

POP

Tiene dos significados: "El punto de Presencia" y "Protocolo de oficina de correos" . Un "el Punto de Presencia" comúnmente significa una ciudad o ubicación donde una red puede conectarse, frecuentemente con líneas de teléfono. El segundo significado, "el Protocolo de oficina de correos" refiere a la manera en que un software e-mail tal como Eudora consigue comunicarsecon el un servidor de correo.

Port

Tiene tres significados. El primero, un lugar donde la información entra o sale de una computadora, o ambos. P. ej. el "port serial" sobre una computadora de persona es donde un modem se conectaría. Sobre el Internet "port" frecuentemente se refiere al número que es la parte de un URL, apareciendo después de un signo dos puntos (:) a la derecha después del nombre de dominio. Cada servicio Internet sobre un servidor "escucha" sobre un número de port particular sobre ese servidor. La mayoría de los servicios tienen un número de port, p. ej. servidores de Web normalmente escuchan sobre el port 80. Finalmente, "port" también refiere a traducir un pedazo de software para traerlo desde un de tipo de sistema de computadora a otro, p. ej. para traducir un programa de Windows para que se corrá sobre un Macintosh.

RFC

(Request for Coments - Pedido Para Comentarios) - - El nombre del resultado y el proceso para crear una norma sobre el Internet. Las nuevas normas se proponen y son publicadas en línea, como un "RFC " . El Internet Engineering Task Force es un grupo que contruye y avala una norma, también facilita la discusión, y eventualmente una nueva norma se

establece con un número asignado por ellos, entonces la norma se llamará por ejemplo RFC 822.

Router

Una computadora de propósito especial (o el software) que maneja la conexión entre 2 o más redes. Los Routers estan todo el tiempo mirando las direcciones de destino de los paquetes que pasan por ellos y decidiendo sobre que ruta enviarlos.

Server

Una computadora, o un paquete de software, que provee un tipo específico de servicio al software de cliente que corre sobre otras computadoras. El término puede referir al pedazo particular de software, tal como un servidor de WWW, o a la máquina sobre la que el software corre. Una máquina única de servidor podría tener varios diferente paquetes de software de servidor que corren sobre ella.

SLIP

(Serial Line Internet Protocol) Una norma para usar una línea telefónica regular (una "línea serial") y un modem para conectar una computadora como un sitio verdadero de Internet. Esta siendo reemplazado gradualmente por PPP.

TCP/IP

(Transmission Control Protocol / Internet Protocol) esto es n conjunto de protocolos que definen la Internet. Originalmente estuvo diseñado para el sistema operativo UNIX, el software TCP/IP esta disponible ahora para la mayoría de los sistemas operativos en uso. Para que su computadora este sobre Internetdebe tener TCP/IP.

Telnet

El programa y comando usado para hacer un login sobre una computadora remota sobre Internet.

Terminal

Un dispositivo que permite enviar comandos a una computadora en cualquier otro sitio.Como minimo implica tener un teclado, un monitor y una conexión a la computadora.

Terminal Server

Una máquina de propósito especial que tiene de un lado conexión a muchos modems y una conexión LAN o una máquina Host del otro. Así un terminal server trabaja atendiendo los llamados y pasando la conexión sobre un nodo apropiado. la mayoría de los terminal servers proveen servicios de PPP o SLIP.

UNIX

Un sistema operativo de computadoras. UNIX fue diseñado para ser usado por mucha gente al mismo tiempo y tiene TCP/IP incluido. Es el sistema operativo más usado para servidores Internet.

URL

(Uniform Resource Locator) La manera standard de dar la dirección de cualquier recurso parte de WWW sobre Internet. Tiene la siguiente forma

http://www.unlp.edu.ar/lp_ciudad.html

También puede referenciar a otros servicios como por ejemplo

ftp://ftp.unlp.edu.ar/pub/....

Usenet

Un sistema para grupos de discusión, con comentarios pasados a través de cientos de miles de máquinas..

WAIS

(Wide Area Information Servers) Un paquete de software comercial que permite ordenar una gran cantidad de información, entonces hace índices buscables a través de redes tal como Internet. Una característica importante es que los resultados de las búsquedas son dados con puntaje.

WAN

Cualquier red que cubre una gran área, más que una facultad o una compañia

WWW

(World Wide Web). El servicio de hipertexto ofrecido en Internet, donde los servidores permiten archivos de testo, gráficos, de sonido, entre otros.

REFERENCIAS

REFERENCIAS

[1] COMER DOUGLAS E., [1995].
Internetworking with TCP/IP
Volume 1, Princiles, Protocols, and Architecture.

[2] CRICKET LIU, JERRY PEEK, RUSS JONES, BRYAN BUUS, ADRIAN NYE.
O'Reilly & Associates, Inc. [1994].
Managing Internet Information Services.

[3] KROL ED., [1994]
The Whole Internet User's Guide & Catalog

[4] FAH-CHUN CHEONG, [1996]
Internets Agents : Spiders, Wanderers, Brokers, and Bots.

[5] DEEP JOHN, and HOLFELDER PETER, [1996].
Developing CGI Aplications with Perl.

[6] WALL LARRY, and SCHWARTZ RANDAL, [1991].
O'Reilly & Associates, Inc.
Programming Perl.

[7] BEAN GREG, [1995].
Internet Server Construction Kit For Windows.

[8] BERNERS-LEE T., and CONNOLLY D.
Hypertext Markup Language 2.0
RFC 1866, MIT/W3C, Noviembre 1995.

[9] RAGGETT DAVID.
Hypertext Markup Language 3.0
DRAFT INTERNET, Marzo 1995.

[10] BERNERS-LEE T., MASINTER L., and McCAHILL M.
Uniform Resource Locators (URL).
RFC 1738, CERN, University of Minnesota, Diciembre 1994.

[11] BORENSTEIN N., and FREED N.
MIME (Multipurpose Internet Mail Extensions)
Part One : Mechanisms for Specifying and Describing the Format of Internet Mesage Bodies.
RFC 1521, Bellcore, Innosoft, Septiembre 1993.

[12] BERNERS-LEE T., and FIELDING R.
Hypertext Transfer Protocol - HTTP/1.0
RFC 1945, MIT/LCS, Mayo 1996.

[13] LOTTOR M.
Domain Administrators Operations Guide.
RFC 1033, SRI INTERNATIONAL, Noviembre 1987.

[14] MOCKAPETRIS P.
Domain Names - Concepts and Facilities.
RFC 1034, ISI, Noviembre 1987.

[15] MOCKAPETRIS P.
Domain Names - Implementation and Specification.
RFC 1035, ISI, Noviembre 1987.

DIRECCIONES INTERNET DE REFERENCIA

BROWSER WATCH - http://www.browserwatch.com
CERN - http://www.w3.org
CIO - http://www.cio.com
MICROSOFT - http://www.msn.com
NCSA - http://www.ncsa.uiuc.edu
NETSCAPE - http://www.nestcape.com
SERVER WATCH - http://www.serverwatch.com