INSTALACIÓN

Y

ADMINISTRACIÓN

DE

SERVICIOS

WEB

MARÍA ÁNGELES PEÑASCO SÁNCHEZ- 2º ASIR-SRI-TEMA 4

- **INTRODUCCIÓN**
  - WWW
  - W3C Y ESTÁNDARES WEB

- **CARACTERÍSTICAS GENERALES DE UN SERVICIO WEB.**
  - COMPONENTES Y FUNCIONAMIENTO.
  - NOMBRES Y DIRECCIONES (URIS Y URLS)
  - PÁGINAS WEB, SITIOS WEB Y APLICACIONES WEB.

- **PROTOCOLO HTTP.**
  - FUNCIONAMIENTO BÁSICO.
  - MENSAJES HTTP.
  - MÉTODOS DE PETICIÓN: GET , POST, HEAD, PUT, DELETE Y TRACE.
  - CABECERAS.
  - CÓDIGOS DE ESTADO Y ERROR.
  - ALMACENAMIENTO EN CACHE.
  - REDIRECCIONES.
  - COMPRENSIÓN.
  - COOKIES.
  - AUTENTICACIÓN
  - CONEXIONES PERSISTENTES.

- **CONFIGURACIÓN DE UN SERVIDOR WEB.**
  - INSTALACIÓN, CONFIGURACIÓN Y USO.
  - AUTENTICACIÓN Y CONTROL DE ACCESO.
  - REGISTRO Y MONITORIZACIÓN DEL SERVICIO WEB.
  - TIPOS MIME.
  - WEBDAV.

- **NAVEGADORES WEB.**
  - PARÁMETROS DE APARIENCIA Y USO.

- **SEGURIDAD DEL PROTOCOLO HTTP:**
  - PROTOCOLO HTTPS.
  - CONEXIONES SEGURAS: SSL, TSL.
  - GESTIÓN DE CERTIFICADOS Y ACCESO SEGURO CON HTTPS

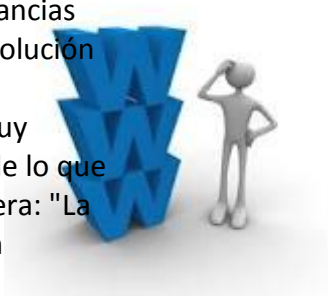
- **ALMACENAMIENTO VIRTUAL DE SITIOS WEB: «HOSTS» VIRTUALES.**
  - ALOJAMIENTO VIRTUAL BASADO EN IPS.
  - ALOJAMIENTO VIRTUAL BASADO EN NOMBRES.
  - ALOJAMIENTO VIRTUAL BASADO EN PUERTOS.
  - ALOJAMIENTOS HÍBRIDOS.

- **INTRODUCCIÓN**

WWW

Web o la web, la red o www de World Wide Web, es básicamente un medio de comunicación de texto, gráficos y otros objetos multimedia a través de Internet, es decir, la web es un sistema de hipertexto que utiliza Internet como su mecanismo de transporte o desde otro punto de vista, una forma gráfica de explorar Internet.

La World Wide Web o simplemente la Web, tuvo sus orígenes en 1989 en el CERN (Centro
Europeo para la Investigación Nuclear) ubicado en Ginebra (Suiza), en circunstancias en que el investigador británico Tim Berners-Lee se dedicaba a encontrar una solución efectiva al problema de la proliferación y la heterogeneidad de la información disponible en la Red. Integrando servicios ya existentes en Internet (como el muy utilizado Gopher por esa época) Berners-Lee desarrolló la arquitectura básica de lo que actualmente es la Web. El mismo Berners-Lee la describía de la siguiente manera: "La WWW es una forma de ver toda la información disponible en Internet como un continuo, sin rupturas.
Utilizando saltos hipertextuales y búsquedas, el usuario navega a través de un mundo de información parcialmente creado a mano, parcialmente generado por computadoras de las bases de datos existentes y de los sistemas de información".



En 1990 se desarrolló un primer prototipo, pero sólo a partir de 1993, cuando el NCSA (Centro Nacional de Aplicaciones de Supercomputadoras) de la Universidad de Illinois introdujo el primer "cliente" gráfico para la WWW, denominado Mosaic, la comunidad de usuarios de Internet comenzó su empleo en forma exponencial. A partir de allí y hasta nuestros días, es usual que la gente no dedicada al tema confunda, y con razón, a Internet con la Web.

W3C Y ESTÁNDARES WEB

W3C son las siglas de World Wide Web Consortium, un consorcio fundado en 1994 para dirigir a la Web hacia su pleno potencial mediante el desarrollo de protocolos comunes que promuevan su evolución y aseguren su interoperabilidad.

El consorcio está compuesto por un grupo de programadores, desarrolladores web, ejecutivos de la industria y usuarios que ayudan a definir las especificaciones para el desarrollo de la tecnología web.

¿Qué son los Estándares Web?

Un estándar es un conjunto de reglas normalizadas que describen los requisitos que deben ser cumplidos por un producto, proceso o servicio, con el objetivo de establecer un mecanismo base para permitir que distintos elementos hardware o software que lo utilicen, sean compatibles entre sí.

El W3C, organización independiente y neutral, desarrolla estándares relacionados con la Web también conocidos como Recomendaciones, que sirven como referencia para construir una Web accesible, interoperable y eficiente, en la que se puedan desarrollar aplicaciones cada vez más robustas.

Algunos de los estándares Web más conocidos y ampliamente utilizados son: HTML (Hypertext Markup Language), para definir la estructura de los documentos; XML (extensible Markup Language), que sirve de base para un gran número de tecnologías; y CSS (Cascading Style Sheets), que permite asignar estilos para la representación de los documentos.

¿Para qué sirven?

La finalidad de los estándares es la creación de una Web universal, accesible, fácil de usar y en la que todo el mundo pueda confiar. Con estas tecnologías abiertas y de uso libre se pretende evitar la fragmentación de la Web y mejorar las infraestructuras para que se pueda evolucionar hacia una Web con la información mejor organizada.

Acceso Universal

El W3C se guía por los principios de accesibilidad, internacionalización, e independencia de dispositivo, entre otros. Esto facilita que el acceso a la Web sea posible desde cualquier lugar, en cualquier momento y utilizando cualquier dispositivo.

No importa si se utiliza hardware, software, o una infraestructura de red específica. Además de las posibles restricciones técnicas, se tiene en cuenta la existencia de múltiples idiomas, las diversas localizaciones geográficas, y las diferencias culturales o tradiciones, así como las posibles limitaciones físicas, psíquicas o sensoriales de los usuarios.

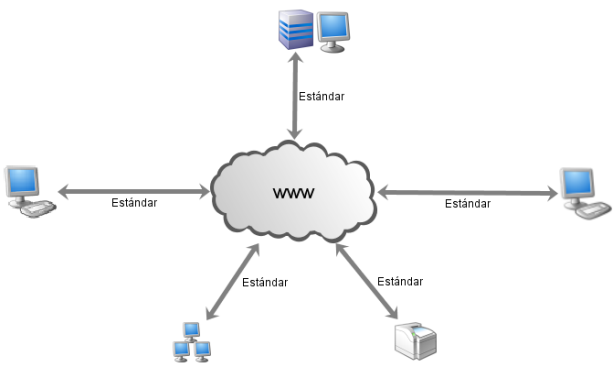
El avance de las tecnologías inalámbricas, así como la gran variedad de dispositivos con acceso a la Web presentes en sectores como el de la telefonía móvil, en el de automoción (navegadores en los salpicaderos de automóviles), en los electrodomésticos (refrigeradores con pantallas táctiles) o en los televisores, fomenta la ubicuidad de la Web. Esto pone de manifiesto la necesidad de utilizar tecnologías y lenguajes unificados, libres y gratuitos, cuyo uso no esté limitado por patentes comerciales.

¿Cómo funcionan?

La creación de un estándar Web requiere un proceso controlado, que consta de varias etapas que aseguran la calidad de la especificación. Este proceso permite la intervención de todos los usuarios de las tecnologías, con el objetivo de que puedan aportar su conocimiento y opiniones para la mejora de los documentos.

Tras este proceso, elaborado por especialistas en la materia, se obtienen unos estándares de calidad, y al estar disponible para todo el mundo, las especificaciones se depuran exhaustivamente antes de ser consideradas como Recomendación.

Estos estándares, están sujetos a la Política de Patentes del W3C, lo que permite que sean utilizados libremente por toda la comunidad Web. Al utilizar las mismas tecnologías, las máquinas se entienden entre sí y cualquier usuario puede interactuar con el resto.



Para ayudar a los desarrolladores que deseen utilizar sus Recomendaciones, el W3C ofrece una serie de herramientas que permiten verificar si se hace una correcta aplicación de las especificaciones. Manuales de directivas o buenas prácticas de tecnologías concretas, y los validadores sintácticos de los lenguajes, son ejemplos de estas ayudas.

- **CARACTERÍSTICAS GENERALES DE UN SERVICIO WEB.**

COMPONENTES Y FUNCIONAMIENTO.

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

Componentes y funcionamiento

El Servidor web se ejecuta en un ordenador manteniéndose a la espera de peticiones por parte de un cliente (un navegador web) y que responde a estas peticiones adecuadamente, mediante una página web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error. A modo de ejemplo, al teclear www.wikipedia.org en nuestro navegador, éste realiza una petición HTTP al servidor de dicha dirección. El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo exhibe en pantalla. Como vemos con este ejemplo, el cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

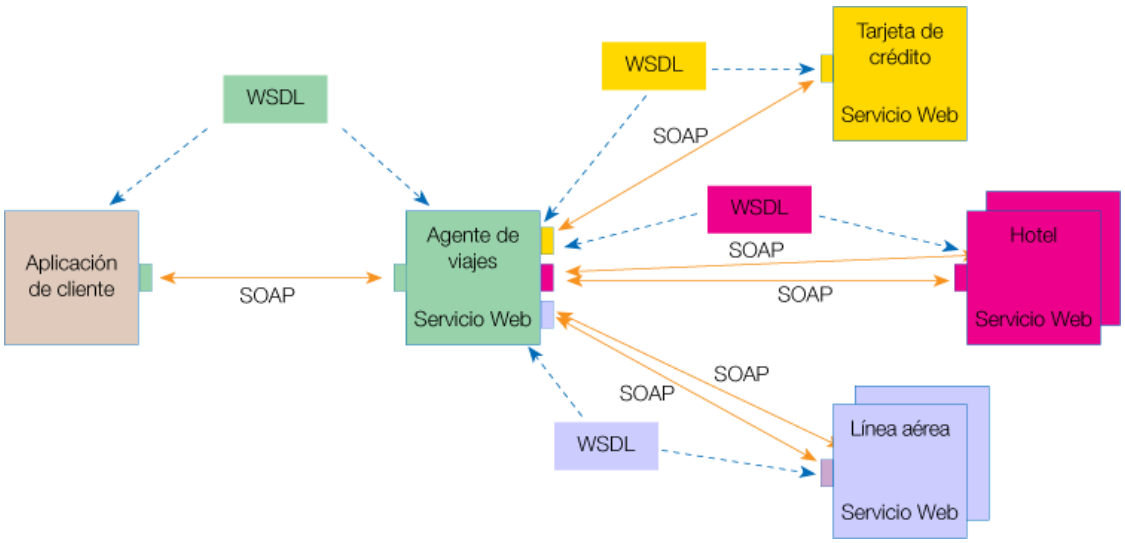
Además de la transferencia de código HTML, los Servidores web pueden entregar aplicaciones web. Éstas son porciones de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

- Aplicaciones en el lado del cliente: el cliente web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java "applets" o Javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts).

Comúnmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje javascript y java, aunque pueden añadirse más lenguajes mediante el uso de plugins.

- Aplicaciones en el lado del servidor: el servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

Las aplicaciones de servidor muchas veces suelen ser la mejor opción para realizar aplicaciones web. La razón es que, al ejecutarse ésta en el servidor y no en la máquina del cliente, éste no necesita ninguna capacidad añadida, como sí ocurre en el caso de querer ejecutar aplicaciones javascript o java. Así pues, cualquier cliente dotado de un navegador web básico puede utilizar este tipo de aplicaciones.



NOMBRES Y DIRECCIONES (URIS Y URLS)

Algunos ejemplos de nombres e identificadores son las URL, los nombres de dominio de Internet, los nombres de archivos… etc.

Podemos distinguir entre nombres puros (patrones de bits sin interpretar) y no puros (contienen información sobre el objeto al que nombran (p. ej.: la ubicación del objeto)). En el otro extremo de un nombre puro se sitúa la dirección de un objeto, la cual es eficaz para acceder a éste, pero está el problema de que un objeto puede cambiar de localización.

Se dice que un nombre está resuelto cuando está traducido a datos relacionados con el recurso en cuestión. La asociación entre un nombre y un objeto se llama enlace. Los nombres suelen enlazarse a los atributos de los objetos y no a su implementación. Un atributo es una propiedad de un objeto.

Identificadores de Recurso Unificados (URI): Un ejemplo de URI son los URL, que son direcciones únicamente de recursos web, a los que se puede acceder con facilidad (nombre DNS más un camino hacia el recurso). Pero si un recurso se mueve o se borra, el URL no apuntará a nada (se dice comúnmente que está roto) o apuntará a otro objeto (si ha sido referenciado igual que el anterior).

Otro tipo de URI son los Nombres Uniformes de Recurso (URN), que tratan de resolver los anteriores problemas. Un servicio de búsqueda URN relaciona los URN con su URL correspondiente, la cual puede variar en el tiempo (sin que varíe el URN). Si un administrador cambia la URL, debe registrar la nueva en el servicio de búsqueda.

URIs relativas

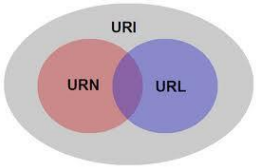
Las URIs relativas son URIs parciales, utilizadas para referirse a un documento desde otro en la misma computadora. De esta forma, podemos definir una URI relativa como la ruta que se debe seguir desde la ubicación del documento actual (ruta de directorios) a la ubicación del recurso referido, además del nombre de archivo.

Supongamos que el documento actual, localizado en "http://servidor.es/documentos/index.asp", necesita apuntar a un documento ubicado en "http://servidor.es/documentos/nuevos/mejores/dos.asp". La URI relativa para referirse a ese recurso desde el documento actual será: "nuevos/mejores/dos.asp"

El directorio especial ".." provee una forma de ir hacia atrás al directorio "padre". De modo que para apuntar desde "http://nuevoservidor.mil/documentos/nuevos/mejores/rec.htm" a "http://nuevoservidor.mil/documentos/antiguos/mejores/junio.htm", la URI relativa será: "../../antiguos/mejores/junio.htm"

¿Qué es un URL?

Los URLs (Uniform Resource Locator) son identificadores que permiten acceder a recursos (páginas) web. En la misma forma en que los humanos utilizamos direcciones para identificar y encontrar ubicaciones, los URLs le sirven al navegador (y otros sistemas) para encontrar una página o recurso Web en el vasto mundo del Internet.



PÁGINAS WEB, SITIOS WEB Y APLICACIONES WEB.

PAGINAS WEB: Una página web es el nombre de un documento o información electrónica adaptada para la World Wide Web y que puede ser accedida mediante un navegador para mostrarse en un monitor de computadora o dispositivo móvil. Esta información se encuentra generalmente en formato HTML o XHTML, y puede proporcionar navegación a otras páginas web mediante enlaces de hipertexto. Las páginas web frecuentemente incluyen otros recursos como hojas de estilo en cascada, guiones (scripts) e imágenes digitales, entre otros.

Las páginas web pueden estar almacenadas en un equipo local o un servidor web remoto. El servidor web puede restringir el acceso únicamente para redes privadas, p. ej., en una intranet corporativa, o puede publicar las páginas en la World Wide Web. El acceso a las páginas web es realizado mediante su transferencia desde servidores utilizando el protocolo de transferencia de hipertexto (HTTP).

SITIOS WEB: Un sitio web es una colección de páginas web relacionadas y comunes a un dominio de Internet o subdominio en la World Wide Web en Internet. Una página web es un documento HTML/XHTML que es accesible generalmente mediante el protocolo HTTP de Internet. Todos los sitios web públicamente accesibles constituyen una gigantesca World Wide Web de información (un gigantesco entramado de recursos de alcance mundial).

A las páginas de un sitio web se accede frecuentemente a través de un URL raíz común llamado portada, que normalmente reside en el mismo servidor físico. Los URL organizan las páginas en una jerarquía, aunque los hiperenlaces entre ellas controlan más particularmente cómo el lector percibe la estructura general y cómo el tráfico web fluye entre las diferentes partes de los sitios.

Algunos sitios web requieren una subscripción para acceder a algunos o todos sus contenidos. Ejemplos de sitios con subscripción incluyen muchos portales de pornografía en Internet, algunos sitios de noticias, sitios de juegos, foros, servicios de correo electrónico basados en web, sitios que proporcionan datos de bolsa de valores e información económica en tiempo real, etc.

APLICACIONES WEB: En la ingeniería de software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. Existen aplicaciones como los webmails, wikis, weblogs, tiendas en línea y la propia Wikipedia que son ejemplos bien conocidos de aplicaciones web.

Es importante mencionar que una página Web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.

- **PROTOCOLO HTTP.**

El Protocolo de Transferencia de HiperTexto (Hypertext Transfer Protocol) es un sencillo protocolo cliente-servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP. La especificación completa del protocolo HTTP 1/0 está recogida en el RFC 1945. Fue propuesto por Tim Berners-Lee, atendiendo a las necesidades de un sistema global de distribución de información como el World Wide Web.

Desde el punto de vista de las comunicaciones, está soportado sobre los servicios de conexión TCP/IP, y funciona de la misma forma que el resto de los servicios comunes de los entornos UNIX: un proceso servidor escucha en un puerto de comunicaciones TCP (por defecto, el 80), y espera las solicitudes de conexión de los clientes Web. Una vez que se establece la conexión, el protocolo TCP se encarga de mantener la comunicación y garantizar un intercambio de datos libre de errores.

HTTP se basa en sencillas operaciones de solicitud/respuesta. Un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Todas las operaciones pueden adjuntar un objeto o recurso sobre el que actúan; cada objeto Web (documento HTML, fichero multimedia o aplicación CGI) es conocido por su URL.

FUNCIONAMIENTO BÁSICO.

Etapas de una transacción HTTP.

Para profundizar más en el funcionamiento de HTTP, veremos primero un caso particular de una transacción HTTP; en los siguientes apartados se analizarán las diferentes partes de este proceso.

Cada vez que un cliente realiza una petición a un servidor, se ejecutan los siguientes pasos:

- Un usuario accede a una URL, seleccionando un enlace de un documento HTML o introduciéndola directamente en el campo Location del cliente Web.
- El cliente Web descodifica la URL, separando sus diferentes partes. Así identifica el protocolo de acceso, la dirección DNS o IP del servidor, el posible puerto opcional (el valor por defecto es 80) y el objeto requerido del servidor.
- Se abre una conexión TCP/IP con el servidor, llamando al puerto TCP correspondiente.
  Se realiza la petición. Para ello, se envía el comando necesario (GET, POST, HEAD,…), la dirección del objeto requerido (el contenido de la URL que sigue a la dirección del servidor), la versión del protocolo HTTP empleada (casi siempre HTTP/1.0) y un conjunto variable de información, que incluye datos sobre las capacidades del browser, datos opcionales para el servidor,…
- El servidor devuelve la respuesta al cliente. Consiste en un código de estado y el tipo de dato MIME de la información de retorno, seguido de la propia información.
- Se cierra la conexión TCP.

MENSAJES HTTP.

Una solicitud HTTP es un conjunto de líneas que el navegador envía al servidor.

Incluye:

- Una línea de solicitud: es una línea que especifica el tipo de documento solicitado, el método que se aplicará y la versión del protocolo utilizada. La línea está formada por tres elementos que deben estar separados por un espacio:
  - el método
  - la dirección URL
  - la versión del protocolo utilizada por el cliente (por lo general, HTTP/1.0)

- Los campos del encabezado de solicitud: es un conjunto de líneas opcionales que permiten aportar información adicional sobre la solicitud y/o el cliente (navegador, sistema operativo, etc.). Cada una de estas líneas está formada por un nombre que describe el tipo de encabezado, seguido de dos puntos (:) y el valor del encabezado.

- El cuerpo de la solicitud: es un conjunto de líneas opcionales que deben estar separadas de las líneas precedentes por una línea en blanco y, por ejemplo, permiten que se envíen datos por un comando POST durante la transmisión de datos al servidor utilizando un formulario.

MÉTODOS DE PETICIÓN: GET, POST, HEAD, PUT, DELETE Y TRACE.

Método	Significado
GET	Devuelve el recurso identificado en la URL pedida.
HEAD	Funciona como el GET, pero sin que el servidor devuelva el cuerpo del mensaje. Es decir, sólo se devuelve la información de cabecera.
POST	Indica al servidor que se prepare para recibir información del cliente. Suele usarse para enviar información desde formularios.
PUT	Envía el recurso identificado en la URL desde el cliente hacia el servidor.
TRACE	Inicia un ciclo de mensajes de petición. Se usa para depuración y permite al cliente ver lo que el servidor recibe en el otro lado.
DELETE	Solicita al servidor que borre el recurso identificado con el URL.

CABECERAS.

Connection (conexión)
Permite especificar diferentes opciones para la conexión. Por ejemplo:
Connection: close
Indica que la conexión debe cerrarse una vez transmitido el mensaje completo

Content-Language (idioma del contenido)
Esta cabecera indica el idioma de los destinatarios del recurso. Si no existe, se entiende que el recurso está orientado a todos los usuarios, independientemente del idioma. Esta cabecera permite listar varios idiomas.
Por ejemplo, una herramienta on-line de traducción inglés-francés, podría incluir en sus páginas la cabecera: Content-Language: es, fr
Es importante recalcar que esta cabecera no indica necesariamente el idioma del documento, sino del público al que objetivamente se orienta.
Un texto sencillo de inglés para estudiantes hispanoparlantes incluiría la cabecera: Content-Language: es
Aunque el contenido pueda estar en inglés (y, por tanto, las metaetiquetas HTML indiquen que se trata de un documento en inglés).

Content-Length (longitud del contenido)
Indica la longitud del cuerpo del recurso, expresada en número de octetos.

Content-Location (localización del contenido)
Dirección complementaria que ofrece el servidor en su respuesta. Esta nueva dirección (una URI absoluta o relativa) no corrige la dirección original del recurso solicitado por el cliente, sino que ofrece una ruta a un recurso que complementa al solicitado originalmente.

Content-Type (tipo de contenido)
Indica, como su nombre indica, el tipo de contenido del recurso. Así, la cabecera
Content-Type: text/html; charset=ISO-8859-1
Indica que el recurso es de tipo texto, concretamente código HTML, y codificado según la especificación ISO-8859-1.

Date (fecha)
Indica la fecha de creación del recurso. Tiene la forma:
Date: Tue, 12 Jul 2005 09:32:25 GMT

Expect (espera)
Mediante esta cabecera, el cliente indica qué tipo de respuesta espera del servidor. Si el servidor no está preparado para responder como el cliente espera, debe indicarlo mediante el envío de un código de estatus 417 (Expectation Failed).

Expires (expiración)
Indica la fecha a partir de la cual el recurso debe considerarse obsoleto. Un ejemplo: Date: Tue, 12 Jul 2005 09:32:25 GMT

From ("desde")
Dirección de correo electrónico del usuario (humano) autor de la solicitud.

If-Match ("si cuadra")
Se usa junto con la cabecera de método para hacerlo condicional. Esto permite actualizaciones eficientes de la caché. Si el cliente guarda en su caché alguna entidad (algún elemento distinguible) del recurso solicitado puede verificar gracias a esta cabecera si esta entidad sigue estando en vigor, es decir, si la copia guardada en la caché sigue siendo válida.

If-Modified-Since ("si se ha modificado desde")
Igual que la cabecera If-Match, If-Modified-Since se usa con la cabecera que indica el método para expresar una condición. Si el recurso no ha variado desde la fecha indicada por el cliente, el servidor no debe enviarlo. Enviará, en cambio, un código de estatus 304, confirmándole al cliente (navegador, por ejemplo, o robot de un buscador) que la copia que tiene en caché sigue siendo una copia fiel del recurso guardado en el servidor.

If-None-Match ("si no cuadra")
Igual que las cabecera If-Match e If-Modified-Since, se usa junto con la cabecera de método para someterlo a una condición. Funciona de forma inversa a if-Match. El servidor no debe ejecutar la solicitud (expresada mediante la cabecera de método) si la entidad expresada por la condición de If-None-Match se cumple.

IP (remote adress)
No es estrictamente una cabecera del protocolo HTTP, sino del protocolo TCP/IP. Expresa la identificación numérica de una máquina.

Host (servidor)
Nombre del servidor.

Last-Modified (última modificación)
Mediante esta cabecera el servidor informa de la fecha y hora en que el recurso fue modificado por última vez.

Location (localización)
Mediante este campo el servidor indica la dirección (la URL) de un recurso cuando no se encuentra en la dirección en que se ha solicitado. De esta forma, el servidor invita al navegador (o al software del cliente en general) a que se redirija a la nueva localización.

Referer (remitente)
Documento desde el cual se ha realizado la solicitud actual. Si desde la URL www.cibernetia.com/index.php clicamos el enlace que lleva a www.cibernetia.com/headers_manual/index.php, la primera URL figurará como referer en la solicitud de la segunda URL.

Request (solicitud))
Indica el fichero (el documento) solicitada y el método y versión del protocolo que se van a emplear para realizar la conexión.

Status Code (código de estado)
Mediante el código de estado el servidor informa al navegador sobre cómo ha resuelto la solicitud de un documento. Esta cabecera nos indicará, por ejemplo, si se ha servido el documento con éxito o se ha producido algún problema, como un error interno del servidor, o alguna incidencia, como una redirección hacia otra URL diferente.

User-Agent (agente de usuario)
El user-agent identifica el software de la máquina cliente (es decir, se refiere al software instalado en el ordenador que solicita una página web). La identificación se realiza, normalmente, mediante una combinación de sistema operativo y navegador.

Solicitud (Request)
Esta cabecera indica:

- El método utilizado por el cliente para solicitar el documento.
  Método GET
  Es el más habitual. Permite, además de indicar la página que se solicita, "pasar" variables en la solicitud.
  Método POST
  Se utiliza habitualmente cuando se envían datos a través de un formulario. Permite, igual que GET, enviar variables, pero lo hace de forma distinta.
- El fichero que solicita el cliente
- La versión del protocolo HTTP que se emplea en la conexión. Las versiones actualmente en uso son la 1.0 y la 1.1. No todos los servidores (aunque sí la mayoría) soportan la versión 1.1.

CÓDIGOS DE ESTADO Y ERROR.

Cuando se solicita al servidor una página de su sitio (por ejemplo, cuando un usuario accede a su página a través de un navegador o cuando Googlebot rastrea la página), se muestra un código de estado de HTTP en respuesta a la solicitud.

Este código, que proporciona información acerca del estado de la solicitud, ofrece a Googlebot datos acerca del sitio y de la página solicitada.

A continuación se muestran algunos de los códigos de estado más frecuentes:

- **200** - El servidor ha mostrado la página correctamente.
- **404** - La página solicitada no existe.
- **503** - El servidor está temporalmente fuera de servicio.

A continuación se muestra una lista completa de códigos de estado de HTTP.

1xx (Respuesta provisional)

Códigos de estado que indican una respuesta provisional y requieren que el solicitante realice una acción para poder continuar.

Código	**Descripción**
100 (Continuar)	El solicitante debe continuar con la solicitud. El servidor muestra este código para indicar que ha recibido la primera parte de una solicitud y que está esperando el resto.
101 (Cambiando de protocolos)	El solicitante ha pedido al servidor que cambie los protocolos y el servidor está informando de que así lo hará.

2xx (Correcto)

Códigos de estado que indican que el servidor ha procesado la solicitud correctamente.

Código	**Descripción**
200 (Correcto)	El servidor ha procesado la solicitud correctamente. Generalmente, esto implica que el servidor ha proporcionado la página solicitada. Si aparece este estado al solicitar su archivo robots.txt, significa que Googlebot lo ha recuperado correctamente.
201 (Creado)	La solicitud se ha procesado correctamente y el servidor ha creado un nuevo recurso.
202 (Aceptado)	El servidor ha aceptado la solicitud, pero todavía no la ha procesado.
203 (Esta información no concede autorización)	El servidor ha procesado la solicitud correctamente, pero muestra información que puede proceder de otra fuente.
204 (Sin contenido)	El servidor ha procesado la solicitud correctamente, pero no muestra ningún contenido.
205 (Restablecer contenido)	El servidor ha procesado la solicitud correctamente, pero no muestra ningún contenido. A diferencia de la respuesta 204, esta requiere que el solicitante restablezca la vista del documento (por ejemplo, borrar los datos de un formulario para introducir nueva información).
206 (Contenido parcial)	El servidor ha procesado una solicitud GET parcial correctamente.

3xx (Redirigido)

Es necesario llevar a cabo acciones adicionales para completar la solicitud. A menudo, estos códigos de estado se utilizan para el Redireccionamiento. Google recomienda utilizar menos de cinco Redireccionamiento en cada solicitud. Puede utilizar Herramientas para webmasters de Google para verificar si Googlebot tiene problemas para rastrear sus páginas redireccionadas. En la página Errores de rastreo, dentro de **Diagnósticos**, se muestran las URL que Googlebot no pudo rastrear debido a errores de Redireccionamiento.

Código	**Descripción**
300 (Varias opciones)	El servidor puede realizar varias acciones de acuerdo con la solicitud. Puede elegir una acción definida por el solicitante (user agent) o bien presentar una lista para que el solicitante elija una acción.
301 (Movido permanentemente)	La página solicitada se ha movido definitivamente a una ubicación nueva. Cuando el servidor muestra esta respuesta (como respuesta a una solicitud GET o HEAD), dirige automáticamente al solicitante a la ubicación nueva. Debe utilizar este código para comunicar a Googlebot que una página o un sitio se han movido a una ubicación nueva de forma definitiva.
302 (Movido temporalmente)	El servidor responde a la solicitud con una página de otra ubicación, pero el solicitante debe seguir utilizando la ubicación original para solicitudes futuras. Este código es similar al 301 en que para una solicitud GET o HEAD, el sistema dirige automáticamente al solicitante a una ubicación diferente. Sin embargo, no se debe utilizar para comunicar a Googlebot que una página o un sitio se han movido, ya que el robot continuará rastreando e indexando la ubicación original.
303 (Ver otra ubicación)	El servidor muestra este código cuando el solicitante debe realizar una solicitud GET independiente a una ubicación diferente para poder obtener la respuesta. Para todas las solicitudes distintas de HEAD, el servidor dirige automáticamente al usuario a la ubicación nueva.
304 (No modificado)	La página solicitada no ha sufrido cambios desde la última solicitud. Cuando el servidor muestra esta respuesta, no devuelve el contenido de la página. Cuando una página no ha cambiado desde la última solicitud, debe configurar su servidor para que muestre esta respuesta (denominada cabecera "HTTP If-Modified-Since"). Esta función le ahorra ancho de banda y otros gastos, ya que su servidor puede comunicar a Googlebot que una página no ha cambiado

	desde la última vez que se rastreó
305 (Usar proxy)	El solicitante sólo puede acceder a la página solicitada mediante un proxy. Cuando el servidor muestra esta respuesta, también indica el proxy que debe utilizarse.
307 (Redireccionamiento temporal)	El servidor responde a la solicitud con una página de otra ubicación, pero el solicitante debe seguir utilizando la ubicación original para solicitudes futuras. Este código es similar al 301 en que para una solicitud GET o HEAD, el sistema dirige automáticamente al solicitante a una ubicación diferente. Sin embargo, no se debe utilizar para comunicar a Googlebot que una página o un sitio se han movido, ya que el robot continuará rastreando e indexando la ubicación original.

4xx (Error de solicitud)

Los códigos de estado siguientes indican que puede haberse producido un error en la solicitud que impidió al servidor procesarla.

Código	**Descripción**
400 (Solicitud incorrecta)	El servidor no ha entendido la sintaxis de la solicitud.
401 (No autorizado)	La solicitud requiere autenticación. El servidor puede mostrar esta respuesta para una página que requiera información de acceso.
403 (Prohibido)	El servidor ha rechazado la solicitud. Si Googlebot recibe este código de estado al intentar rastrear las páginas válidas del sitio (puede comprobarlo en la página **Rastreo web** de la pestaña **Diagnósticos**, en las Herramientas para webmasters de Google), es posible que el servidor o el host esté bloqueando el acceso del robot.
404 (No se encuentra)	El servidor no encuentra la página solicitada. El servidor a menudo muestra este código cuando, por ejemplo, se realiza una solicitud de una página que no existe en el servidor. Si no dispone de un archivo robots.txt en su sitio y aparece este estado en la página de robots.txt de la pestaña "Diagnósticos" en Herramientas para webmasters de Google, este será el estado correcto. Sin embargo, si dispone de un archivo robots.txt y aparece este estado, su archivo podría presentar un nombre incorrecto o bien encontrarse en la

	ubicación equivocada (el archivo debe encontrarse en el nivel superior del dominio y denominarse robots.txt). Si aparece este estado para las URL que Googlebot intentó rastrear (en la página de errores de HTTP de la pestaña "Diagnósticos"), es posible que Googlebot haya seguido un enlace de otra página que no es válido (obsoleto o con algún error ortotipográfico).
405 (Método no permitido)	No se permite el método especificado en la solicitud.
406 (Inaceptable)	No se puede ofrecer la página solicitada con las características de contenido requeridas.
407 (Se requiere autenticación de proxy)	Este código de estado es similar al 401 (No autorizado), aunque en este caso se especifica que el solicitante debe autenticarse mediante un proxy. Cuando el servidor muestra esta respuesta, también indica el proxy que debe utilizarse.
408 (El tiempo de espera de la solicitud ha caducado)	Se ha excedido el tiempo de espera de respuesta de la solicitud.
409 (Conflicto)	El servidor ha detectado un conflicto al llevar a cabo la solicitud, por lo que debe incluir la información correspondiente en la respuesta. El servidor podría mostrar este código como respuesta a una solicitud PUT que entre en conflicto con una solicitud anterior junto con una lista de diferencias entre ambas.
410 (No disponible permanentemente)	El servidor muestra esta respuesta cuando el recurso solicitado se ha eliminado definitivamente. Es similar al código "404 (No se encuentra)", aunque en ocasiones se utiliza en su lugar para identificar aquellos recursos que existieron anteriormente. Si el recurso se ha movido permanentemente, debe utilizar un código 301 para especificar su nueva ubicación.
411 (Requiere longitud)	El servidor no aceptará la solicitud sin el campo válido "Content-Length" (longitud del contenido) en la cabecera.
412 (Error de condición previa)	El servidor no cumple con una de las condiciones previas que el solicitante ha especificado en la solicitud.
413 (Entidad de solicitud demasiado larga)	El servidor no puede procesar la solicitud porque es demasiada larga.
414 (URI solicitada demasiado larga)	La URI solicitada (generalmente una URL) es demasiado larga para que el servidor la procese.
415 (Tipo de soporte incompatible)	La solicitud se encuentra en un formato que la página solicitada no admite.
416 (Intervalo solicitado	El servidor muestra este código de estado cuando se realiza

no válido)	una solicitud de un rango que no se encuentra disponible para la página.
417 (Error de expectativa)	El servidor no puede cumplir los requisitos del campo de expectativa de solicitud en la cabecera.

5xx (Error del servidor)

Los códigos de estado siguientes indican que se ha producido un error interno del servidor al intentar procesar la solicitud. Estos errores suelen afectar al servidor, no a la solicitud.

Código	**Descripción**
500 (Error interno del servidor)	Se ha producido un error en el servidor y no puede completar la solicitud.
501 (No implementado)	El servidor no dispone de las funciones necesarias para completar la solicitud. Este código puede mostrarse, por ejemplo, cuando el servidor no reconozca el método de solicitud.
502 (Pasarela incorrecta)	Al actuar como pasarela o proxy, el servidor ha recibido una respuesta no válida del servidor ascendente.
503 (Servicio no disponible)	El servidor no está disponible en estos momentos, debido a tareas de mantenimiento o a una sobrecarga. Generalmente, este es un estado temporal.
504 (El tiempo de espera de la pasarela ha caducado)	Al actuar como pasarela o proxy, el servidor no ha recibido una solicitud puntual del servidor ascendente.
505 (Versión de HTTP no compatible)	El servidor no es compatible con la versión del protocolo HTTP utilizada en la solicitud.

ALMACENAMIENTO EN CACHE

Se llama caché web a la caché que almacena documentos web (es decir, páginas, imágenes, etcétera) para reducir el ancho de banda consumido, la carga de los servidores y el retardo en la descarga. Un caché web almacena copias de los documentos que pasan por él, de forma que subsiguientes peticiones pueden ser respondidas por el propio caché, si se cumplen ciertas condiciones.

Tipos de cachés web

Las cachés web pueden utilizarse de diversas formas. Las cachés de agente de usuario (User-Agent), como las presentes en los navegadores web, son cachés privados, que funcionan solo para un único usuario. También existen paquetes específicos que se instalan como proxy local y actúan como caché además de realizar otras tareas, como por ejemplo Proxomitron.

Los intermediarios en la comunicación cliente-servidor también pueden implementar cachés compartidos (también llamadas proxy-cachés directos) que sirvan páginas a varios usuarios. Los proxy-cachés suelen ser usados por los proveedores de servicios de Internet (ISP), universidades y empresas para ahorrar ancho de banda. La intermediación de estos proxy-cachés difiere de la de los privados en que los clientes no necesitan ser explícitamente configurados para usarlos. Algunos paquetes que pueden ser usados como proxy-cachés son Squid, Microsoft ISA Server y Blue Coat.

Las cachés pasarela (llamadas también proxy-cachés inversos o aceleradores web) funcionan a cargo del propio servidor original, de forma que los clientes no distinguen unos de otros. Puede hacerse funcionar conjuntamente varias cachés pasarela para implementar una Content Delivery Network (CDN), como es el caso de Akamai. Paquetes como Varnish Cache pueden usarse para este propósito.

Los intermediarios que funcionan como caché realizan con frecuencia otras tareas, tales como la autenticación de usuarios y el filtrado de contenidos. Varios cachés pueden ser coordinados entre sí con las ayuda de protocolos específicos tales como ICP o HTCP.

Control de los cachés web

El protocolo HTTP define tres mecanismos básicos para controlar las cachés:

- Frescura, que permite que una respuesta sea usada sin comprobar de nuevo el servidor origen, y puede ser controlada tanto por el servidor como el cliente. Por ejemplo, la cabecera de respuesta Expires facilita una fecha en la que el documento caduca, y la directiva Cache-Control: max-age informa al caché del número de segundos durante los que la respuesta será válida.

- Validación, que puede usarse para comprobar si una respuesta cacheada sigue siendo buena tras caducar. Por ejemplo, si la respuesta tiene una cabecera Last-Modified, un caché puede hacer una petición condicional usando la cabecera If-Modified-Since para saber si la página cambió.
- Invalidación, que normalmente es un efecto secundario de otra petición que pasa por la caché. Por ejemplo, si la URL asociada con una respuesta cacheada es solicitada posteriormente mediante una petición POST, PUT o DELETE, la respuesta cacheada quedará invalidada.

REDIRECCIONES.

¿Cuándo se necesita una redirección web?

Existen diferentes casos de real necesidad para los cuales se debe de usar la redirección: por ejemplo en caso de cambio en la URL de nuestro portal, variación del nombre de un fichero, o cambio de carpeta en la arborescencia de nuestro sitio web.

Su funcionamiento:

- Se necesita que el encabezamiento enviado por la página consultada corresponda a su estatus. Por ejemplo, si una página ha cambiado de lugar en nuestro portal, es de vital importancia que la antigua URL haga un Redireccionamiento hacia la nueva, utilizando un encabezamiento HTTP que precise que esta página ha cambiado de manera definitiva de dirección (código 301) – Esto permitirá al robot el no volver a indexar nunca la antigua URL, poniendo al día su base de datos aplicando la nueva URL a la página en cuestión.

Si no aplicamos la redirección desde la antigua URL, el robot y los visitantes obtendrán un error 404, lo cual no será una buena señal, ya que de este modo el encontrar la nueva dirección se convertiría en una misión complicada.

I- Redirecciones web compatibles con el posicionamiento:

A continuación, presentamos un resumen de las técnicas de re direccionamiento más usadas, y compatibles con el posicionamiento.

1. Redirección permanente por .htaccess (Redirect Permanent)
2. Redirección por .htaccess (URL Rewriting)

COMPRENSIÓN.

El protocolo de transferencia de documentos de hipertexto (HTTP), utilizado en la web, provee la poderosa pero poco conocida habilidad de trabajar con información comprimida utilizando algoritmos de compresión estándares en la industria.
Se trata entonces de comprimir la información enviada por el servidor del sitio web, dejando al navegador del visitante el trabajo de descomprimirlo. Esto se realiza automáticamente, sin que el visitante lo perciba ni deba intervenir.

Ventajas

Al comprimir información, esta se envía mucho más rápido desde el servidor hasta el navegador del visitante, produciendo así una mejor experiencia en la visita del sitio y recortando la cantidad de ancho de banda --y sus costos-- utilizado por el sitio. En general se puede conseguir una compresión de entre 5:1 y 10:1 (y de hasta 50:1), logrando así una reducción del tamaño de las páginas de, en promedio, 65% a 85%. Esto resulta generalmente en una transferencia de entre 3 a 6 veces más rápido,
Google, Amazon, Yahoo, AT&T y una larga lista de gigantes utilizan esta tecnología. Por ejemplo la pagina principal de Google tiene apenas 1.412 bytes, que sin compresión hubiera tenido 3.873 bytes, logrando así un ahorro del 63.5%.

Desventajas

Como la compresión se realiza dinámicamente, esta requiere algo de procesamiento. Sin embargo, en nuestra experiencia esto no tiene un impacto significativo en la performance del servidor.

COOKIES

Los cookies son un conocido mecanismo que almacena información sobre un usuario de internet en su propio ordenador, y se suelen emplear para asignar a los visitantes de un sitio de internet un número de identificación individual para su reconocimiento subsiguiente. Sin embargo, la existencia de los cookies y su uso generalmente no están ocultos al usuario, quien puede desactivar el acceso a la información de los cookies.

Sin embargo, dado que un sitio Web puede emplear un identificador cookie para construir un perfil del usuario y éste no conoce la información que se añade a este perfil, se puede considerar a los cookies una forma de spyware.

Por ejemplo, una página con motor de búsqueda puede asignar un número de identificación individual al usuario la primera vez que visita la página, y puede almacenar todos sus términos de búsqueda en una base de datos con su número de identificación como clave en todas sus próximas visitas (hasta que el cookie expira o se borra).

Estos datos pueden ser empleados para seleccionar los anuncios publicitarios que se mostrarán al usuario, o pueden ser transmitidos (legal o ilegalmente) a otros sitios u organizaciones.

AUTENTICACIÓN

La autenticación es el proceso de identificar si un cliente es elegible para tener acceso a un recurso. El protocolo HTTP soporta la autenticación como un medio de negociar el acceso a un recurso seguro.

La solicitud inicial de un cliente es normalmente una solicitud anónima, que no contiene ninguna información de autenticación. Las aplicaciones de servidor HTTP pueden denegar la solicitud anónima indicando que se requiere la autenticación. La aplicación de servidor envía encabezados de la autenticación de WWW para indicar los esquemas de autenticación soportados.

Existen varios tipos de autenticación:

- Autenticación básica: soportado por todos los servidores web y navegadores, así como terminales móviles.
- Autenticación mediante resúmenes ó digest: soportada por todos los servidores y en algunos navegadores.
- Autenticación de Windows integrada: evolución de la antigua autenticación por desafío respuesta de Windows. Solamente en plataforma Windows para navegador Internet Explorer.
- Autenticación http: es una combinación del protocolo HTTP y protocolos criptográficos

AUTENTICACIÓN BÁSICA

Cuando el usuario accede a un recurso del servidor web protegido mediante autenticación básica, tiene lugar el siguiente proceso:

1. El navegador presenta al usuario la ventana de autenticación, para que introduzca su nombre y contraseña.
2. El navegador intenta establecer una conexión con el servidor utilizando esta información.
3. Si el servidor rechaza la información de autenticación, el navegador le presenta nuevamente la ventana al usuario hasta que éste introduce por fin una contraseña válida o cierra la ventana.
4. Cuando el servidor web verifica con éxito los datos de autenticación, se establece la conexión de acceso al recurso protegido.

Ventana de autenticación que presenta el navegador cuando se pretende acceder a un recurso protegido.

El gran inconveniente de este método es que la contraseña viaja en claro desde el navegador del usuario hasta el servidor, por lo que cualquier atacante armado con un sniffers podrá interceptarla inmediatamente. Su mayor ventaja, en cambio, es que forma parte del protocolo HTTP 1.0 y posteriores versiones, estando por tanto universalmente aceptado en la práctica totalidad de navegadores. Cuando se salta a otra página o recurso igualmente protegidos por este método, en lugar de presentar al usuario una nueva ventana de autenticación, lo cual resultaría muy engorroso, el navegador recuerda la información tecleada por el usuario la primera vez y se la envía al servidor automáticamente. Nótese que en las sucesivas autenticaciones, esta información continúa viajando en claro, aunque el usuario no sea consciente de ello.

AUTENTICACIÓN MEDIANTE RESÚMENES O DIGEST

Dado que el método anterior envía las contraseñas en claro, no resulta muy adecuado cuando las exigencias de seguridad son elevadas. Para paliar este inconveniente, además de cifrar el canal con SSL, otra alternativa consiste en enviar un resumen criptográfico de la contraseña (un hash) en vez de la propia contraseña, de la siguiente forma:

1. El servidor envía al navegador cierta información que será utilizada en el proceso de autenticación.
2. El navegador añade esta información a su nombre de usuario y contraseña, junto con otra información adicional, y crea un resumen del conjunto. Esta información adicional persigue el cometido de impedir ataques de reactuación, en los que un atacante intercepta y copia el resumen, volviéndolo a utilizar para autenticarse él mismo ante el servidor.
3. Se envía en claro tanto el resumen como la información adicional al servidor a través de la red.
4. El servidor añade esta información adicional a una copia en claro de la contraseña del cliente y crea el resumen del conjunto.
5. El servidor compara el resumen que ha creado con el que le ha llegado del navegador.
6. Si ambos números coinciden, se le concede acceso al usuario.

La autenticación mediante resúmenes ha sido incorporada al estándar HTTP 1.1, pero desgraciadamente la mayoría de navegadores no la soportan. Se puede encontrar una descripción sobre su funcionamiento y consideraciones sobre su seguridad en un borrador de Internet.

AUTENTICACIÓN DE WINDOWS INTEGRADA

Este tipo de autenticación, exclusivo de NT, ha pasado a llamarse Autenticación Integrada de Windows y constituye una variante de la autenticación mediante resúmenes criptográficos. Se trata igualmente una forma segura de autenticación en la medida en que no se envían ni la contraseña ni el nombre de usuario a través de la Red. En su lugar, el navegador tiene que demostrarle al servidor que conoce la clave por medio de un corto intercambio de datos, pero sin revelar nunca la clave. No obstante, debido a los detalles de implantación, resulta incompatible con la autenticación por resúmenes, por lo que su uso se circunscribe a servidores NT.

Funciona de la siguiente manera:

1. A diferencia de la autenticación básica o mediante resúmenes, no se le presenta al usuario una ventana para que introduzca su nombre y contraseña, sino que se utiliza la información de la sesión abierta por el ordenador del cliente, es decir, se utiliza el nombre de usuario, contraseña y dominio que se utilizó al entrar al ordenador desde el que se está navegando.
2. Si este intercambio inicial falla, entonces se le presenta al usuario la ventana de identificación, donde introduce su nombre, contraseña y además el dominio.
3. Si los datos proporcionados no son correctos, se le presenta esta ventana repetidamente hasta que el usuario introduce la información correcta o la cierra.

Aunque este método presenta la ventaja de no transmitir las contraseñas a través de la Red, superando el inconveniente de la autenticación básica, posee dos importantes limitaciones para su uso en Internet:

- Sólo está soportado por Microsoft Internet Explorer, versión 2.0 o posterior y servidores NT.
- No funciona para conexiones con proxy.

Estas limitaciones hacen que la autenticación integrada de Windows sea más adecuada para intranets, en las que se puede exigir a los usuarios que el navegador que utilicen sea Internet Explorer y en las que tanto los servidores como los clientes se encuentran detrás del mismo proxy. Es muy importante que las cuentas de los usuarios que se autentiquen de esta forma posean el derecho de Acceder a este equipo desde la red.

AUTENTICACIÓN HTTPS

El uso del formato HTTPS para enviar mensajes garantiza la autenticación de los usuarios que necesitan acceso a los recursos de Message Queue Server por medio de un servidor Web estableciendo una conexión de nivel de sockets seguro (SSL) para conseguir una comunicación segura entre un remitente y un destinatario. El emisor es siempre considerado como cliente SSL y el destinatario como servidor SSL independientemente de si el equipo está ejecutando Message Queue Server o software de cliente. Tenga en cuenta que la autenticación para establecer una sesión de SSL no es la misma que la autenticación de mensajes, que confirma que un mensaje no se ha manipulado y se puede utilizar para comprobar la identidad del remitente. Para obtener información acerca de la autenticación de mensajes, consulte Administrar la autenticación de mensajes.

En la autenticación HTTPS se utilizan dos tipos de certificados:

- Certificados de servidor. Este certificado contiene información sobre el servidor que permite a un cliente identificar el servidor antes de compartir información confidencial.
- Certificados de cliente. Este certificado contiene información personal sobre el usuario e identifica el servidor al cliente de SSL (el remitente).

CONEXIONES PERSISTENTES.

Las conexiones persistentes del HTTP, también llamadas HTTP guardar-vivo, o reutilización de la conexión del HTTP, son la idea de usar la misma conexión del TCP para enviar y para recibir múltiplo Peticiones del HTTP/responses, en comparación con abrir una nueva conexión para cada solo par de la petición/de la respuesta.

Ventajas

- menos CPU y uso de la memoria (porque pocas conexiones están abiertas simultáneamente)
- reducido congestión de red (menos Conexiones del TCP)
- reducido estado latente en las peticiones subsecuentes (no apretón de manos)
- los errores se pueden divulgar sin la pena de cerrar la conexión del TCP

Conexión HTTP persistente, también llamado mantenimiento de conexiones HTTP, o volver a usar la conexión HTTP, es la idea de usar la misma conexión TCP para enviar y recibir múltiples peticiones HTTP / respuestas, en lugar de abrir una nueva conexión para todos y cada uno de petición / respuesta par.

En HTTP 1.0, no hay especificación oficial para saber cómo funciona keepalive. Era, en esencia, añadido a un protocolo existente. Si el navegador es compatible con mantenimiento de conexión, se añade una cabecera adicional a la solicitud:
Entonces, cuando el servidor recibe la solicitud y genera una respuesta, sino que también agrega un encabezado a la respuesta:
Después de esto, la conexión no se cae, sino que se mantiene abierta. Cuando el cliente envía una nueva solicitud, que utiliza la misma conexión.

Esto continuará hasta que el cliente o el servidor deciden que la conversación ha terminado, y uno de ellos cae la conexión.

En HTTP 1.1 se consideran todas las conexiones persistentes menos que se declare lo contrario. Las conexiones HTTP persistentes no utilizan separar los mensajes de keepalive, que sólo permiten múltiples solicitudes para el uso de una única conexión. Sin embargo, el tiempo de espera de conexión por defecto de Apache http 2.0 es tan poco como 15 segundos y para Apache 2.2 a 5 segundos. La ventaja de un tiempo corto es la capacidad de ofrecer múltiples componentes de una página web de forma rápida sin atar varios procesos de servidor o discusiones durante mucho tiempo.

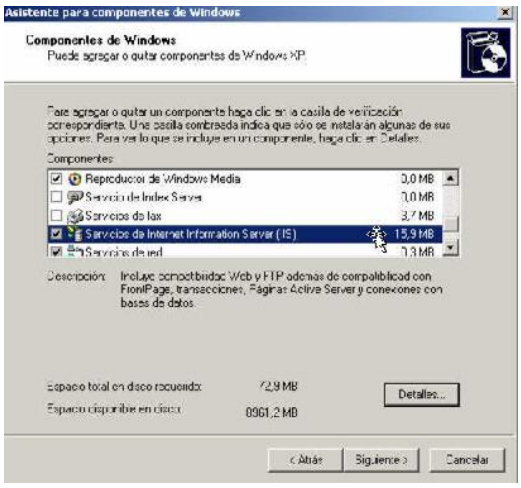
- **CONFIGURACIÓN DE UN SERVIDOR WEB.**

INSTALACIÓN, CONFIGURACIÓN Y USO.

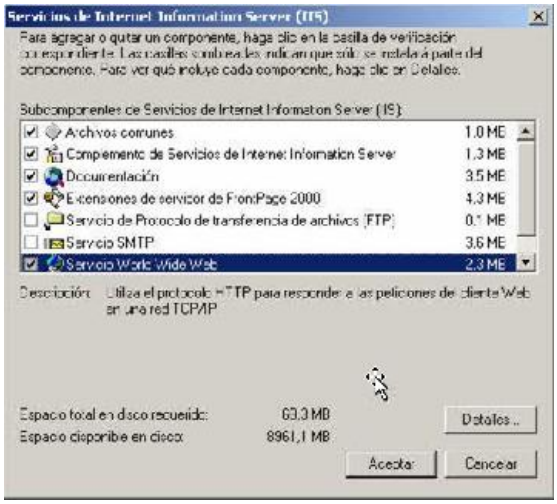
Primero debemos saber que Windows XP PRO solo nos permite montar un solo servidor de páginas web y también un solo servidor FTP. Otra limitación es que nos permite hasta un máximo de 10 conexiones TCP simultáneas.

Si el servidor de páginas web lo montamos para una red local solo deberemos conocer la dirección IP del ordenador en el cual instalaremos el servidor, si lo hacemos para dar servicio de páginas web a internet tendremos que tener una conexión a internet con una IP fija, esto normalmente sucede cuando nuestra conexión es del tipo de banda ancha (por ejemplo es el caso de ADSL).

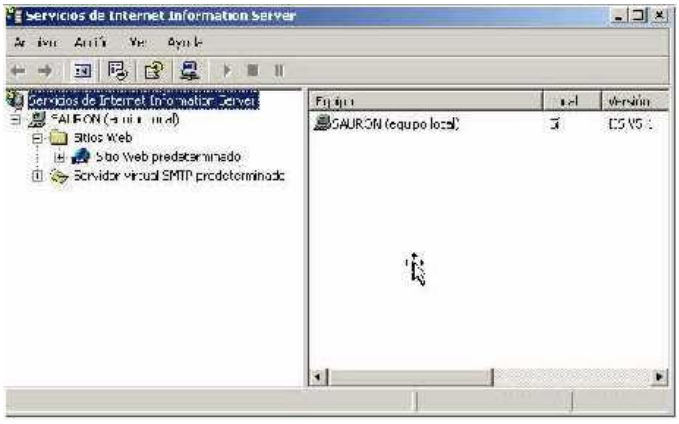
Primero tendremos que instalar el servidor en nuestro Windows XP PRO para ello hacemos lo siguiente: vamos a INICIO -> CONFIGURACION -> PANEL DE CONTROL -> AGREGAR O QUITAR PROGRAMAS y pinchamos en "Agregar o quitar componentes de Windows"



Tendremos que seleccionar la instalación de "Servicios de Internet Information Server o IIS", pichamos luego en detalles y veremos lo siguiente:



Una vez que hayamos terminado la instalación podemos ver la consola de administración de nuestro sitio WEB o FTP. Para abrir la consola vamos a INICIO -> CONFIGURACION -> PANEL DE CONTROL -> HERRAMIENTAS ADMINISTRATIVAS y pinchamos en "Servicios de Internet Information Server", veremos la siguiente pantalla:



Algunos consejos útiles:

Tener un Antivirus con las últimas actualizaciones en el ordenador que dará servicios de páginas web.

Es altamente recomendable que utilicemos un cortafuego para evitar visitas no deseadas ya que al tener el servidor constantemente encendido y conectado a internet/intranet puede ser objeto de ataques.

Conviene dar permisos de Lectura pero no así de Escritura o Examinar directorio para evitar que nos dejen programas o aplicaciones no deseadas, que pueden en algunos casos ejecutarse para recolectar información privada.

Ver el archivo de registros de visitas para ver que secciones de nuestra web son las más visitadas y cuáles no lo son y así mejorarlas. Para ver este archivo es tan fácil como abrir con un editor de texto lo que veamos en la siguiente dirección de nuestro ordenador \WINDOWS\System32\Log Files.

AUTENTICACIÓN Y CONTROL DE ACCESO.

La autenticación es el proceso de identificar si un cliente es elegible para tener acceso a un recurso. El protocolo HTTP soporta la autenticación como un medio de negociar el acceso a un recurso seguro.

La solicitud inicial de un cliente es normalmente una solicitud anónima, que no contiene ninguna información de autenticación. Las aplicaciones de servidor HTTP pueden denegar la solicitud anónima indicando que se requiere la autenticación. La aplicación de servidor envía encabezados de la autenticación de WWW para indicar los esquemas de autenticación soportados. Este documento describe varios esquemas de autenticación para HTTP y aborda su suporte en Windows Communication Foundation (WCF).

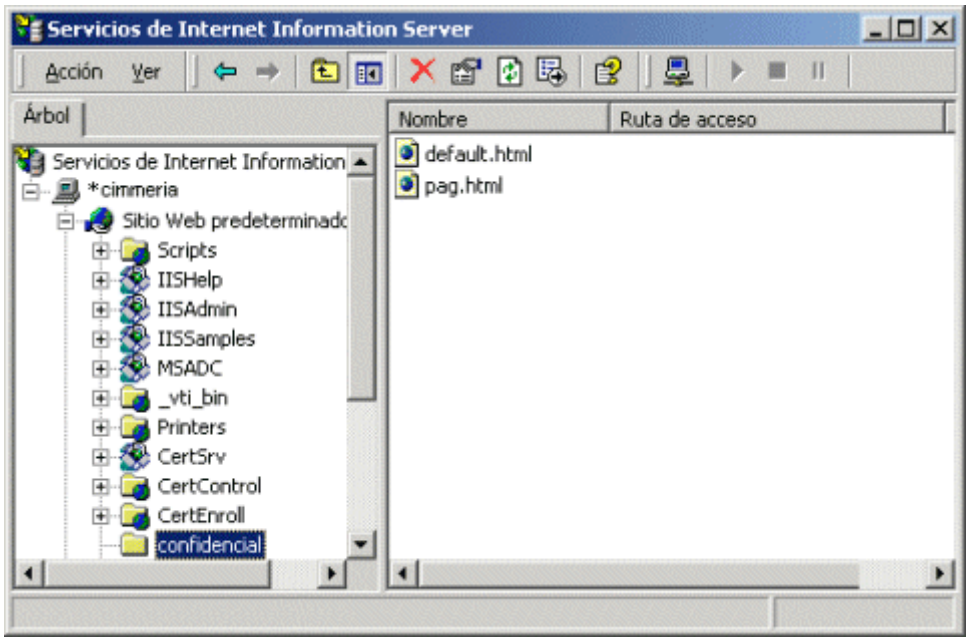
El control de acceso constituye una poderosa herramienta para proteger la entrada a un web completo o sólo a ciertos directorios concretos e incluso a ficheros o programas individuales. Este control consta generalmente de dos pasos:

- En primer lugar, la autenticación, que identifica al usuario o a la máquina que trata de acceder a los recursos, protegidos o no.
- En segundo lugar, procede la cesión de derechos, es decir, la autorización, que dota al usuario de privilegios para poder efectuar ciertas operaciones con los datos protegidos, tales como leerlos, modificarlos, crearlos, etc.

Por defecto, todas las páginas y servicios del servidor web se pueden acceder anónimamente, es decir, sin necesidad de identificarse ante el servidor y sin ningún tipo de restricción. En máquinas NT, el usuario anónimo pertenece al grupo Invitados y tiene asignada la cuenta IUSR_nombremáquina, donde nombremáquina toma el valor del nombre del servidor: para una máquina llamada Mordor, la cuenta de acceso anónimo a Internet sería IUSR_MORDOR. Esta cuenta anónima debe tener permiso para conectarse localmente. En Linux, en cambio, no es necesario crear una cuenta en la máquina para los usuarios anónimos.

Análogamente, toda la información que viaja por las redes de comunicaciones lo hace en claro, de manera que puede ser fácilmente interceptada por un atacante. De ahí la necesidad de proteger los datos mientras se encuentran en tránsito por medio de un canal cifrado, para lo que se utiliza normalmente SSL, como se describirá más adelante.

Los diversos métodos de control de acceso presentados en este curso se han particularizado para dos servidores ampliamente difundidos en Internet: IIS 5.0 corriendo bajo Windows 2000, y servidores para Linux tipo Apache, como Stronghold 2.4. Aunque en otros servidores el proceso no será idéntico, sí es cierto que resultará muy parecido. Se puede consultar una completa comparativa de servidores web en webcompare.internet.com.



REGISTRO Y MONITORIZACIÓN DEL SERVICIO WEB.

Los archivos de registros o archivos log como se conocen comúnmente, son archivos en donde se van almacenando un registro de todos los eventos que ocurren en un sistema durante un periodo de tiempo en particular. Estos archivos son usados tanto por el sistema operativo como por las aplicaciones o demonios (procesos) para registrar datos o información sobre un evento en particular. En un sistema Linux podemos encontrar estos archivos de registro o logs en la carpeta /var/log En esta carpeta encontraremos casi todos los archivos de registros de un sistema, pero cabe destacar que muchas aplicaciones crean estos archivos en sus propias carpetas fuera de /var/log.

Ahora bien, ¿En qué nos sirve los logs para monitorear nuestro sistema? pues muy sencillo, los principales archivos logs que están en la carpeta /var/log van almacenando información de casi todos los eventos que ocurren en tu PC prácticamente desde que la enciendes y en ellos podremos ver por ejemplo que pasa internamente en Linux cuando conectas una Memoria USB, un Modem USB o cuando estas conectado a internet puedes ver los intentos de entrada bloqueados por tu firewall. En otras circunstancias podremos ser capaces de observar algún mensaje de error que se pueda producir cuando estas conectando algún hardware nuevo o si tienes un servicio web instalado podrás ver quienes están conectados a tu equipo.

TIPOS MIME.

Multipurpose Internet Mail Extensions

Multipurpose Internet Mail Extensions o MIME (en español "extensiones multipropósito de correo de internet") son una serie de convenciones o especificaciones dirigidas al intercambio a través de Internet de todo tipo de archivos (texto, audio, vídeo, etc.) de forma transparente para el usuario. Una parte importante del MIME está dedicada a mejorar las posibilidades de transferencia de texto en distintos idiomas y alfabetos. En sentido general las extensiones de MIME van encaminadas a soportar: G

- Texto en conjuntos de caracteres distintos de US-ASCII;
- adjuntos que no son de tipo texto;
- cuerpos de mensajes con múltiples partes (multi-part);
- información de encabezados con conjuntos de caracteres distintos de ASCII.

Prácticamente todos los mensajes de correo electrónico escritos por personas en Internet y una proporción considerable de estos mensajes generados automáticamente son transmitidos en formato MIME a través de SMTP. Los mensajes de correo electrónico en Internet están tan cercanamente asociados con el SMTP y MIME que usualmente se les llama mensaje SMTP/MIME.

En 1991 la IETF (Grupo de Trabajo en Ingeniería de Internet, Internet Engineering Task Force en inglés) comenzó a desarrollar esta norma y desde 1994 todas las extensiones MIME están especificadas de forma detallada en diversos documentos oficiales disponibles en Internet.

Los tipos de contenido definidos por el estándar MIME tienen gran importancia también fuera del contexto de los mensajes electrónicos. Ejemplo de esto son algunos protocolos de red tales como HTTP de la Web. HTTP requiere que los datos sean transmitidos en un contexto de mensajes tipo e-mail aunque los datos pueden no ser un e-mail propiamente dicho.

En la actualidad ningún programa de correo electrónico o navegador de Internet puede considerarse completo si no acepta MIME en sus diferentes facetas (texto y formatos de archivo).

WEBDAV.

El objetivo de WebDAV es hacer de la World Wide Web un medio legible *y* editable, en línea con la visión original de Tim Berners-Lee. Este protocolo proporciona funcionalidades para crear, cambiar y mover documentos en un servidor remoto (típicamente un servidor web). Esto se utiliza sobre todo para permitir la edición de los documentos que sirve un servidor web, pero puede también aplicarse a sistemas de almacenamiento generales basados en web, que pueden ser accedidos desde cualquier lugar. La mayoría de los sistemas operativos modernos proporcionan soporte para WebDAV, haciendo que los ficheros de un servidor WebDAV aparezcan como almacenados en un directorio local.

- Visión general acerca del protocolo WebDAV

WebDAV añade los siguientes métodos a HTTP:

- PROPFIND - Usado para recuperar propiedades, almacenadas como XML, desde un recurso. También está sobrecargado para permitir recuperar la estructura de colección (alias jerarquía de directorios) de un sistema remoto.
- PROPPATCH - Usado para cambiar y borrar múltiples propiedades de un recurso en una simple operación atómica (atomic commit).
- MKCOL - Usado para crear colecciones (alias directorio)
- COPY - Usado para copiar un recurso desde un URI a otro.
- MOVE - Usado para mover un recurso desde un URI a otro.
- LOCK - Usado para bloquear (lock) un recurso. WebDAV soporta tanto bloqueos compartidos como exclusivos.
- UNLOCK - Para desbloquear un recurso.

Recurso es el nombre HTTP para una referencia que está apuntada por un Identificador de Recursos Uniforme o URI (Uniform Resource Identifier).

El grupo de trabajo WebDAV está todavía trabajando en unas cuantas extensiones a WebDAV, incluyendo: control de redirecciones, enlaces, límites de espacio en disco y mejoras en la especificación base para que alcance el nivel de madurez del resto de estándares de Internet.

- **NAVEGADORES WEB.**

La funcionalidad básica de un navegador web es permitir la visualización de documentos de texto, posiblemente con recursos multimedia incrustados. Los documentos pueden estar ubicados en la computadora en donde está el usuario, pero también pueden estar en cualquier otro dispositivo que esté conectado a la computadora del usuario o a través de Internet, y que tenga los recursos necesarios para la transmisión de los documentos (un software servidor web).

Tales documentos, comúnmente denominados páginas web, poseen hipervínculos que enlazan una porción de texto o una imagen a otro documento, normalmente relacionado con el texto o la imagen.

El seguimiento de enlaces de una página a otra, ubicada en cualquier computadora conectada a la Internet, se llama navegación, de donde se origina el nombre navegador (aplicado tanto para el programa como para la persona que lo utiliza, a la cual también se le llama cibernauta).

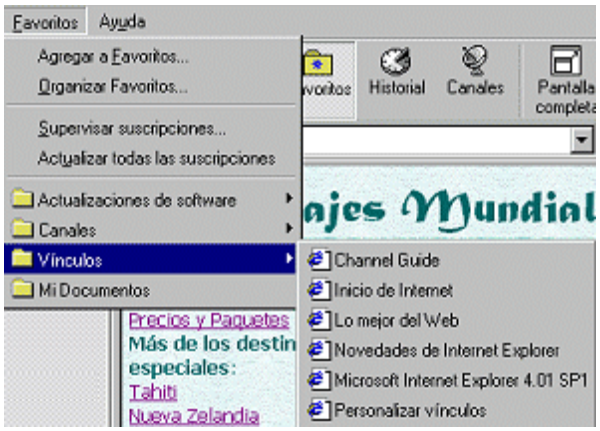
La función principal del navegador es descargar documentos HTML y mostrarlos en pantalla. En la actualidad, no solamente descargan este tipo de documentos sino que muestran con el documento sus imágenes, sonidos e incluso vídeos streaming en diferentes formatos y protocolos. Además, permiten almacenar la información en el disco o crear marcadores (bookmarks) de las páginas más visitadas.

Algunos de los navegadores web más populares se incluyen en lo que se denomina una Suite. Estas Suite disponen de varios programas integrados para leer noticias de Usenet y correo electrónico mediante los protocolos NNTP, IMAP y POP.

PARÁMETROS DE APARIENCIA Y USO.

Barra de Título
La barra de título muestra el título de la página y el nombre del navegador a la izquierda. A la derecha se hallan los botones estándar: Minimizar, Maximizar y Cerrar.

Barra de Menús
La Barra de Menús contiene las listas de comandos en cascada.



Barra de Herramientas La barra de herramientas tiene botones para los comandos utilizados con más frecuencia. Cuando el ratón pasa por encima de un botón, este se verá en colores y parecerá en relieve. Algunos botones no se verán, si el tamaño de la ventana es pequeño.

Barra de Direcciones La Barra de Direcciones muestra la URL (Universal Resource Location), también llamada dirección, (address), para las páginas web que se ven en la ventana del navegador. La barra de Vínculos generalmente se ve a la derecha de la barra de Direcciones.

Puede escribir una URL en la Barra de Direcciones y apretar la tecla ENTRAR para desplegar la página cuya ubicación ha escrito. El botón Ir es agregado a la derecha de la Barra de Direcciones en IE5. Si prefiere más usar el ratón que el teclado, puede hacer un clic en el botón Ir, en lugar de apretar la tecla ENTRAR para abrir la página en la dirección que figura en la Barra de Direcciones.

- **SEGURIDAD DEL PROTOCOLO HTTP:**

PROTOCOLO HTTPS.

Hyper Text Transfer Protocol Secure (en español: Protocolo seguro de transferencia de hipertexto), más conocido por sus siglas HTTPS, es un protocolo de aplicación basado en el protocolo HTTP, destinado a la transferencia segura de datos de Hiper Texto, es decir, es la versión segura de HTTP.

Es utilizado principalmente por cualquier tipo de servicio que requiera el envío de datos personales o contraseñas.

El sistema HTTPS utiliza un cifrado basado en SSL/TLS para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente) más apropiado para el tráfico de información sensible que el protocolo HTTP. De este modo se consigue que la información sensible (usuario y claves de paso normalmente) no pueda ser usada por un atacante que haya conseguido interceptar la transferencia de datos de la conexión, ya que lo único que obtendrá será un flujo de datos cifrados que le resultará imposible de descifrar.

El puerto por defecto de este protocolo es el 443.

Se pueden autenticar mediante certificados digitales, ya sean propios o contratados de otra empresa.

CONEXIONES SEGURAS: SSL, TSL.

Secure Sockets Layer (SSL; en español «capa de conexión segura») y su sucesor Transport Layer Security (TLS; en español «seguridad de la capa de transporte») son protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet.

El protocolo SSL intercambia registros; opcionalmente, cada registro puede ser comprimido, cifrado y empaquetado con un código de autenticación del mensaje (MAC). Cada registro tiene un campo de content_type que especifica el protocolo de nivel superior que se está usando.

Cuando se inicia la conexión, el nivel de registro encapsula otro protocolo, el protocolo handshake, que tiene el content_type 22.

El cliente envía y recibe varias estructuras handshake:

- Envía un mensaje ClientHello especificando una lista de conjunto de cifrados, métodos de compresión y la versión del protocolo SSL más alta permitida. Éste también envía bytes aleatorios que serán usados más tarde (llamados Challenge de Cliente o Reto). Además puede incluir el identificador de la sesión.

- Después, recibe un registro ServerHello, en el que el servidor elige los parámetros de conexión a partir de las opciones ofertadas con anterioridad por el cliente.

- Cuando los parámetros de la conexión son conocidos, cliente y servidor intercambian certificados (dependiendo de las claves públicas de cifrado seleccionadas). Estos certificados son actualmente X.509, pero hay también un borrador especificando el uso de certificados basados en OpenPGP.

- El servidor puede requerir un certificado al cliente, para que la conexión sea mutuamente autenticada.

- Cliente y servidor negocian una clave secreta (simétrica) común llamada master secret, posiblemente usando el resultado de un intercambio Diffie-Hellman, o simplemente cifrando una clave secreta con una clave pública que es descifrada con la clave privada de cada uno. Todos los datos de claves restantes son derivados a partir de este master secret (y los valores aleatorios generados en el cliente y el servidor), que son pasados a través una función pseudoaleatoria cuidadosamente elegida.

TLS/SSL poseen una variedad de medidas de seguridad:

- Numerando todos los registros y usando el número de secuencia en el MAC.
- Usando un resumen de mensaje mejorado con una clave (de forma que solo con dicha clave se pueda comprobar el MAC). Esto se especifica en el RFC 2104).
- Protección contra varios ataques conocidos (incluyendo ataques man-in-the-middle), como los que implican un degradado del protocolo a versiones previas (por tanto, menos seguras), o conjuntos de cifrados más débiles.
- El mensaje que finaliza el protocolo handshake (Finished) envía un hash de todos los datos intercambiados y vistos por ambas partes.
- La función pseudo aleatoria divide los datos de entrada en 2 mitades y las procesa con algoritmos hash diferentes (MD5 y SHA), después realiza sobre ellos una operación XOR. De esta forma se protege a sí mismo de la eventualidad de que alguno de estos algoritmos se revelen vulnerables en el futuro.

GESTIÓN DE CERTIFICADOS Y ACCESO SEGURO CON HTTPS

Hypertext Transfer Protocol Secure (ó HTTPS) es una combinación del protocolo HTTP y protocolos criptográficos. Se emplea para lograr conexiones más seguras en la WWW, generalmente para transacciones de pagos o cada vez que se intercambie información sensible (por ejemplo, claves) en internet.

De esta manera la información sensible, en el caso de ser interceptada por un ajeno, estará cifrada.

El nivel de protección que ofrece depende de la corrección de la implementación del navegador web, del software y de los algoritmos criptográficos soportados. Además HTTPS es vulnerable cuando es aplicado a contenido estático públicamente disponible.

El HTTPS fue creado por Netscape Communications en 1994 para su navegador Netscape Navigator.

- **ALMACENAMIENTO VIRTUAL DE SITIOS WEB: «HOSTS» VIRTUALES.**

El término Hosting *Virtual* se refiere a hacer funcionar más de un sitio web (tales como www.company1.com y www.company2.com) en una sola máquina. Los sitios web virtuales pueden estar "basados en direcciones IP", lo que significa que cada sitio web tiene una dirección IP diferente, o "basados en nombres diferentes", lo que significa que con una sola dirección IP están funcionando sitios web con diferentes nombres (de dominio). El hecho de que estén funcionando en la misma máquina física pasa completamente desapercibido para el usuario que visita esos sitios web.

Apache fue uno de los primeros servidores web en soportar hosting virtual basado en direcciones IP. Las versiones 1.1 y posteriores de Apache soportan hosting virtual (vhost) basado tanto en direcciones IP como basado en nombres. Ésta última variante de hosting virtual se llama algunas veces *basada en host* o *hosting virtual no basado en IP*.

ALOJAMIENTO VIRTUAL BASADO EN IPS.

También llamado IP dedicado o virtual hosting, cada máquina virtual tiene una dirección IP diferente. El servidor Web está configurado con múltiples interfaces de red física, o interfaces de red virtual en la misma interfaz física. El software del servidor web, utiliza la dirección IP del cliente se conecta con el fin de determinar a qué sitio web para mostrar al usuario. La razón principal de un sitio para que utilice una IP dedicada debe ser capaz de utilizar su propio certificado SSL en lugar de un certificado común.

El hosting virtual basado en IPs usa la dirección IP de la conexión para determinar qué host virtual es el que tiene que servir. Por lo tanto, necesitará tener diferentes direcciones IP para cada host. Si usa hosting virtual basado en nombres, el servidor atiende al nombre de host que especifica el cliente en las cabeceras de HTTP. Usando esta técnica, una sola dirección IP puede ser compartida por muchos sitios web diferentes.

ALOJAMIENTO VIRTUAL BASADO EN NOMBRES.

El hosting virtual basado en nombres es normalmente más sencillo, porque solo necesita configurar su servidor de DNS para que localice la dirección IP correcta y entonces configurar Apache para que reconozca los diferentes nombres de host. Usando hosting virtual basado en nombres también se reduce la demanda de direcciones IP, que empieza a ser un bien escaso. Por lo tanto, debe usar hosting virtual basado en nombres a no ser que haya alguna razón especial por la cual tenga que elegir usar hosting virtual basado en direcciones IP. Algunas de estas razones pueden ser:

- Algunos clientes antiguos no son compatibles con el hosting virtual basado en nombres. Para que el hosting virtual basado en nombres funcione, el cliente debe enviar la cabecera de Host HTTP. Esto es necesario para HTTP/1.1, y está implementado como extensión en casi todos los navegadores actuales. Si necesita dar soporte a clientes obsoletos y usar hosting virtual basado en nombres, al final de este documento se describe una técnica para que pueda hacerlo.
- El hosting virtual basado en nombres no se puede usar junto con SSL por la naturaleza del protocolo SSL.
- Algunos sistemas operativos y algunos elementos de red tienen implementadas técnicas de gestión de ancho de banda que no pueden diferenciar entre hosts a no ser que no estén en diferentes direcciones IP.

ALOJAMIENTO VIRTUAL BASADO EN PUERTOS.

El número de puerto por defecto para HTTP es 80. Sin embargo, la mayoría de servidores web se puede configurar para funcionar en casi cualquier número de puerto, siempre que el número de puerto no está en uso por cualquier otro programa en el servidor.

Por ejemplo, un servidor puede alojar el sitio web www.example.com. Sin embargo, si el propietario desea operar un segundo sitio, y no tiene acceso a la configuración del nombre de dominio para su nombre de dominio y / o no posee otras direcciones IP que pueden ser utilizados para servir el sitio de, en su lugar podría utilizar otro número de puerto, por ejemplo, www.example.com:81 para el puerto 81, www.example.com:8000 para el puerto 8000, o www.example.com:8080 para el puerto 8080.

Sin embargo, este es un enfoque de usuario poco amigable. Los usuarios no se pueden esperar razonablemente que saber los números de puerto para sus sitios web y móvil de un sitio entre los servidores puede requerir cambiar el número de puerto. No se usen los números de puerto estándar también puede ser visto como poco profesional y poco atractivo para los usuarios. Además, algunos firewalls bloquear todos los puertos, pero la más común, provocando un sitio alojado en un puerto no estándar que no aparecen disponibles para algunos usuarios.

ALOJAMIENTOS HÍBRIDOS.

Por medio de un software simulamos dividir una computadora en cuatro o en cinco computadoras. Así, cada servidor virtual trabaja como si fuera una computadora independiente con un alojamiento dedicado. La diferencia con los servidores compartidos es que en éstos sólo abrimos carpetas en el disco duro para las diferentes páginas.

No son tan baratos como los compartidos, ni tan caros como los dedicados. Sin tantas ventajas técnicas como éstos últimos, pero sin tantos inconvenientes como los primeros. Una buena elección intermedia.

Puente de servidores privados virtuales la brecha entre los servicios de alojamiento web compartido y hosting dedicado, lo que la independencia de otros clientes del servicio de VPS en términos de software, pero a menor costo que un servidor dedicado físico. Como VPS ejecuta su propia copia de su sistema operativo, los clientes tienen superusuario nivel de acceso a esa instancia del sistema operativo, y se puede instalar casi cualquier software que se ejecuta en el sistema operativo.