

6

Gestión de servidores web

vamos a conocer...

1. Estructura de la *World Wide Web* (WWW)
2. Direcciones URL (*Uniform Resource Locator*)
3. Protocolo HTTP
4. Protocolo HTTPS (*HyperText Transfer Protocol Secure*)
5. Aplicaciones web
6. Servidores web en sistemas operativos libres y propietarios
7. Navegadores web

PRÁCTICA PROFESIONAL

- Instalación y configuración de un servidor web en Linux y un cliente en Windows
- Instalación y configuración de un servidor web en Windows XP y un cliente en Windows

MUNDO LABORAL

Los contenidos se han merendado al e-mail

y al finalizar esta unidad...

- Describirás los protocolos necesarios en un servidor web.
- Instalarás sitios virtuales.
- Crearás usuarios y verificarás su acceso.
- Configurarás los parámetros de un servidor web (seguridad, código...).
- Instalarás otros módulos sobre el servidor web.



situación de partida

Álex acaba de terminar de estudiar «Sistemas microinformáticos y redes» y se ha asociado con un par de amigos para crear una cooperativa de diseño y programación web.

Uno de ellos, Alberto, es Diseñador gráfico, tiene el título de la Escuela de Bellas Artes y Oficios y ha realizado un curso de actualización en Photoshop, Adobe Illustrator e InDesign.

La tercera socia, Rosa, es programadora (PHP, ASP, Java...), empezó de forma autodidacta y al final ha estudiado el Ciclo Formativo de Grado Superior.

Álex se encarga tanto de la configuración de la LAN de la empresa como de la de los servidores.

Rosa le ha pedido a Álex que le configure en su ordenador un pseudoservidor para poder probar la programación en modo local antes de subirla a internet. También le ha pedido que prepare

el servidor para que pueda contener bases de datos MySQL y soportar programación PHP.

Alberto, en cambio, está más preocupado por crear sitios web virtuales y así poder probar varios diseños a la vez para cada uno de los dominios que ahora ha alquilado para la empresa, y para los futuros que puedan venir.

La cooperativa se va a llamar «3colegas» y han alquilado los siguientes dominios:

- 3colegas.com
- 3colegas.info
- 3colegas.eu

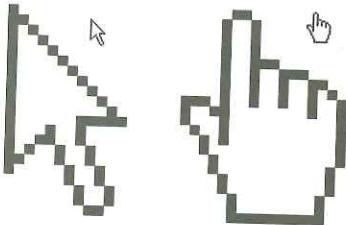
Álex es partidario de los servidores Linux, pero Rosa quiere instalar el de Microsoft, que ha estudiado en la Universidad, y Alberto prefiere consultarlo en los foros antes de decidirse.

estudio del caso

Analiza cada punto de la Unidad de Trabajo, con el objetivo de contestar las preguntas de este caso práctico.

1. ¿Qué servidor es más interesante para Álex: Apache, IIS...?
2. ¿Qué paquetes debería instalar Álex para la programación en PHP con SQL?
3. ¿Cómo puede Álex controlar la seguridad, la autenticación y evitar virus con el servidor elegido?
4. ¿Cómo se pueden crear los sitios virtuales que necesita Alberto?
5. ¿Cada sitio puede tener una configuración diferente?
6. ¿Qué clientes de páginas web (navegadores) le aconsejas instalar a Álex?
7. ¿Existe alguna forma barata de probar las páginas web en diferentes sistemas informáticos?
8. ¿Les interesa un servidor propio o alquilar uno virtual o físico?
9. ¿Deberían configurar los valores por defecto para una universalización o deberían variar para ofrecernos mayor seguridad?

1. Estructura de la World Wide Web (WWW)



La World Wide Web (Red Global Mundial, WWW) es un sistema de documentos enlazados entre sí, accesibles a través de internet. Estos documentos pueden ser solo de texto, llamados entonces **hipertextos**, o multimedia y/o interactivos, en este caso llamados **hipermedios**. Están enlazados entre ellos (de forma transparente para el usuario) con **hiperenlaces**, **hipervínculos**, **links**, vínculos o lo que simplemente llamamos enlaces. Los enlaces son visibles porque el puntero del ratón cambia de forma (de una flecha a una mano con un dedo extendido) al pasar por encima.

→ Primera página web.

Inicialmente, los hiperenlaces eran zonas de texto sobre las que se pulsaba con el ratón y se visualizaba otro documento distinto, pero relacionado en cuanto a contenido (podía estar en la misma carpeta, en el mismo ordenador o en otro computador en cualquier lugar del mundo). Hoy en día son enlaces de documentos electrónicos a recursos de internet (como otro texto electrónico, pero también podría tratarse de un correo electrónico o un FTP). Con un **navegador web** (cliente HTTP) el usuario visualiza los **sitios web** (*site*) compuestos por páginas web, en un mismo dominio o subdominio, con texto ASCII (de forma nativa) e imágenes en formato GIF y JPG (ahora pueden ser imágenes de cualquier formato [PNG], vídeos, sonido, contenido multimedia, etc.).

Fue Berners-Lee quien unió el hipertexto e internet, y quien planteó la WWW en 1990 como un prototipo (en <http://nxoc01.cern.ch/hypertext/WWW/TheProject.html>, que ya no existe), con el primer navegador y el primer servidor. Tres años más tarde, en 1993, se anunció la gratuidad de este servicio.

Las diferencias básicas de esta tecnología con las existentes son:

- Los enlaces eran unidireccionales, en vez de bidireccionales (con el problema asociado de **enlaces rotos** a páginas que ya no existían).
- Era un **sistema no propietario**, con lo que se desarrollaron servidores, clientes y se añadieron extensiones sin coste.

En 1994 ya existían unos 1.500 servidores web en Europa, en los que se empezaron a colocar las primeras páginas web en español (hasta el momento solo la comunidad científica publicaba páginas y lo hacían en inglés). En agosto de 2009 se cifró el número de servidores de páginas web en 225.950.957, pero en el caso de las páginas web no se sabe con exactitud cuántas pueden existir, pues Altavista solo indexa un 8-12% de las mismas y el número supera los 27.000 millones, además existen algunos servidores que albergan más de 1.200 millones de páginas.

2. Direcciones URL (*Uniform Resource Locator*)

Los Localizadores Uniformes de Recursos, URL (*Uniform Resource Locator*) están formados por una secuencia de caracteres (normalmente en ASCII) de acuerdo a un formato estándar, que se usa para nombrar recursos en internet por y para su localización de forma única, es decir, nos dan su dirección en la web. Estos recursos pueden ser páginas web, documentos, imágenes, videos, sonidos, programas, etc. Un ejemplo completo sería:

EJEMPLO

```
http://alex:123456@info.cern.es:80/alex/index.html;lg=es
?l=34;c=es#info
```

Donde:

http:	//alex:	123456@	info.cern.es			:80	/alex	/index	.html	;lg=es	?l=34;c=es	#info					
scheme	user	password	subd	dom	TLD	port	path	fname	fext	params	query	fragment					
scheme	user	password	host			port	path	file		params	query	fragment					
scheme	net_loc						path			params	query	fragment					
NID	NSS																
URN																	
URL																	
scheme	userinfo		host			port	path			query							
scheme	authority						path			query							
URI																	

Cuyos elementos son:

- **Scheme:** protocolo con el que se negocia la transmisión o comunicación. Los más famosos y universales son http (para páginas web), mailto (para correo), news, telnet, file (archivo) y ftp; aunque se soportan otros como https, ftps, ff (finger), y para videoconferencia h323, callme (Skype), ils (Net-meeting), etc. Usa los separadores barra-barra (//) o dos puntos-barra-barra (://) si es un protocolo. No todos los navegadores soportan todos los scheme.
- **Userinfo:** que puede llevar el nombre de usuario solo, o el nombre de usuario y su contraseña. El userinfo precede al separador arroba (@) y utiliza dos puntos (:) entre usuario y contraseña.
- **Host:** puede ser una FQDN (info.cern.es), un nombre de dominio (cern.es), una IPv4 (192.168.0.1) o una IPv6 ([FEDC:BA98:7654:3210: FEDC:BA98:7654:3210]), en este último caso va entre corchetes ([]).
- **Port:** puerto de comunicación del protocolo. Puede ser oficial, oficioso o reconfigurado por el administrador del servidor. Le precede el separador dos puntos (:).
- **Path:** dentro de la computadora, es la ruta donde se aloja el recurso o documento (permite direccionamiento relativo con directorio raíz y puede usar punto-punto-barra para la carpeta padre [../]). El separador de directorios es la barra (/). Los archivos tienen un nombre y suelen llevar una extensión (separados por un punto [.]). También pueden llevar parámetros separados por punto y coma (;).

EJEMPLO

```
//servidor/impresora
//192.168.0.1/leeme.txt
http://google.es
```

saber más

Los URL están especificados en el RFC 1396, 1630, 1738, 1808, 2396 y 2732 y en el URN RFC 2141.

- Upgrade: HTTP/2.0, SHTTP/1.3, IRC/6.9 //Indica para qué versiones de protocolos está actualizado.
- Accept-Language: EN, ES //Indica las lenguas aceptadas, según normas ISO, en el caso EN es inglés y ES español.

El encabezado termina con un línea en blanco (literalmente, debe llevar solo un salto de línea, nada de espacios o tabulaciones); después de esta, se pueden añadir datos adicionales, a los que se les llama **cuerpo**.

En la cabecera se puede especificar su longitud (en bytes) con el parámetro **Content-Length**. Este cuerpo suele ser una cadena de caracteres que dependerá de las gramáticas y reglas de los lenguajes de programación (XML, ASP, PHP, AJAX, DHTML...).

Respuestas

La respuesta del servidor HTTP suele contener un encabezado y un cuerpo:

EJEMPLO

```
HTTP/1.1 200 OK
Date: Fri, 31 Dec 2009 23:59:59 GMT
Content-Length: 1221
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Etag: "3f80f-1b6-3e1cb03b"
Content-Type: text/html; charset=UTF-8
<html>
    <body> (Contenido de la página web) ... </body>
</html>
```

La línea principal, o línea de respuesta, contiene la versión de HTTP soportada por el servidor y un código y mensaje de «error» o estado (ver el apartado [3.2. Códigos de estado](#)).

Después (en la segunda y sucesivas líneas) aparecen los parámetros de respuesta (algunos coinciden con los de petición). En el ejemplo podemos ver:

- Content-Length: 1221 //Longitud del cuerpo en bytes.
- Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux) //Tipo de servidor.
- Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT //Fecha y hora de última modificación del archivo o recurso requerido.
- Content-Type: text/html; charset=UTF-8 //Tipo de contenido aceptado y el código aceptado.

A continuación le sigue una línea en blanco como separador y, después el cuerpo, que contendrá el recurso solicitado (una página web HTML en el ejemplo, pero puede ser cualquier tipo de texto plano, lenguaje de etiquetas o de programación).

Existe la posibilidad de que si las conexiones son muy lentas se visualice la página web parcialmente o que falten imágenes u otros objetos.

3.2. Códigos de estado

Los **códigos de estado** (*status code*) son números de tres cifras que indican la respuesta del servidor a una determinada petición. Los principales son:

- 1XX Mensajes informativos:
 - 100 Continuar (continuar con la petición).
- 2XX Operaciones exitosas:
 - 200 OK (todo correcto).
 - 206 Contenido Parcial (*Partial Content*).
- 3XX Redirección a otra URL:
 - 301 Mudado Permanentemente (*Moved Permanently*).
 - 307 Redirección Temporal (*Temporary Redirect*).
- 4XX Error por parte del cliente:
 - 401 No autorizado (*Unauthorized*).
 - 403 Prohibido (*Forbidden*).
 - 404 No Encontrado (*Not Found* o *File Not Found*, puede ser que no se haya escrito bien la URL, o que se haya puesto un espacio en blanco o tildes. Este error es el más común).
 - 408 Tiempo de Espera Agotado (*Request Timeout*).
- 5XX Error por parte del servidor:
 - 500 Error Interno (*Internal Server Error*).
 - 503 Servicio No Disponible (*Service Unavailable*).

caso práctico inicial

Álex debe conocer tanto los errores típicos del cliente como los del servidor antes de realizar sus pruebas.

3.3. Cookies

Las cookies o «galletas» son **archivos** que el navegador del cliente graba en el disco duro a petición del servidor. Estos archivos almacenan datos que normalmente utiliza el servidor en otras conexiones.

Se suelen utilizar para:

- Guardar los nombres de usuario y contraseñas (son útiles para «cestas de la compra», blogs y otras páginas que necesitan mantener datos entre sesiones).
- Recopilar información de virus, hábitos de navegación de los usuarios, spyware... y usos publicitarios.

saber más

Para informarte más a fondo sobre las especificaciones de las cookies, consulta el RFC 2109.

Las cookies solo se podían generar con CGI, pero ahora también se generan con JavaScript, etc. Según la legislación europea, es obligatorio pedir la instalación de plugins o tipos MIME (consultar la **Unidad 5. Gestión de servicios de correo electrónico**, en la que se trata la gestión del correo electrónico) para el navegador (como PDF, Flash, ActiveX y Silverlight).

Cuando el servidor envía una respuesta incluye el parámetro **Set-Cookie: name=value** y, a partir de ese momento, el navegador del cliente añade el parámetro **Cookie: name=value** a todas las peticiones de ese servidor (y de los servidores de elementos externos que contenga ese recurso).

En los navegadores las cookies se pueden activar, desactivar o preguntar cada vez que se vaya a enviar alguna.

4. Protocolo HTTPS (*HyperText Transfer Protocol Secure*)

saber más

Si quieras informarte sobre el TLS 1.0 o el SSL 3.0, consulta la web del IETF: <http://tools.ietf.org/html/draft-ietf-tls-ssl-version3-00>

El Protocolo Seguro de Transferencia de HiperTexto, **HTTPS** (*HyperText Transfer Protocol Secure*), es el mismo protocolo HTTP pero que ofrece más seguridad, bien sea con Protocolo de Capa de Conexión Segura, **SSL** (*Secure Sockets Layer*) o con Seguridad de la Capa de Transporte, **TLS** (*Transport Layer Security*, el servicio TLS 1.0 equivale al SSL 3.0).

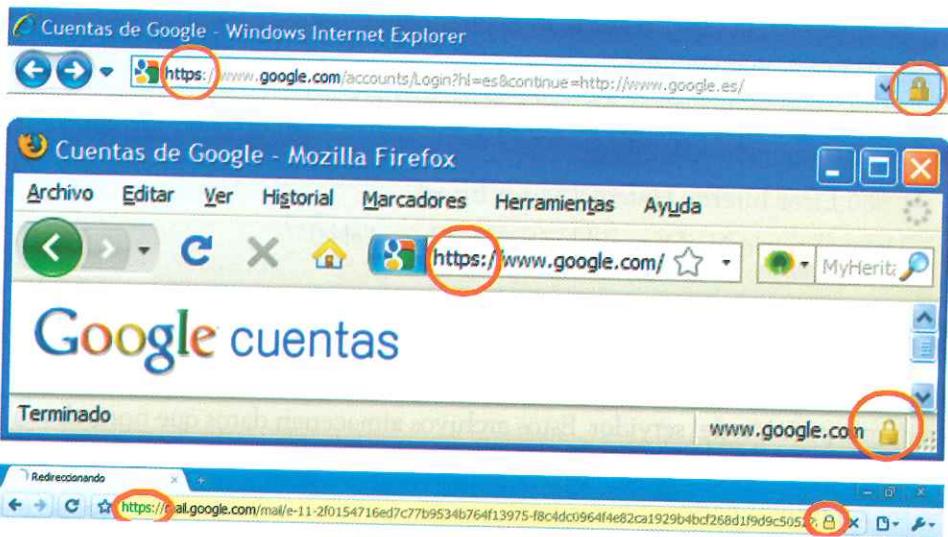
Utiliza el puerto 443 y se emplea para la transferencia de contraseñas, pagos con tarjeta, bancos, etc. Las URL de las páginas empiezan por `https://` y su especificación sobre TLS está en el RFC 2818. Este protocolo, además de los objetivos del SSL, evita el Eavesdropping (ataques de escucha).

Este servicio requiere la confianza de la Autoridad de Certificación que necesita instalar plugins en el navegador (VeriSign, MS Live, etc.). Una vez instalado, el navegador nos confirma que estamos en zona segura o insegura (Internet Explorer, Firefox y Google Chrome lo hacen con un candado abierto en la parte derecha de la dirección o en la parte inferior derecha de la barra de estado, y Netscape con una llave completa).

→ Observa en la imagen el protocolo HTTPS y la figura del candado cerrado en IE.

→ En Firefox el candado cerrado se encuentra en la parte inferior derecha de la ventana.

→ Google Chrome colorea de verde el protocolo HTTPS.



El servicio HTTP trabaja con una clave de sesión (que los navegadores piden como excepción) que se negocia en un *handshake* («apretón de manos») y que suele ser una contraseña de 128 bits (pero que desde hace unos años se puede romper en pocos minutos).

ACTIVIDADES

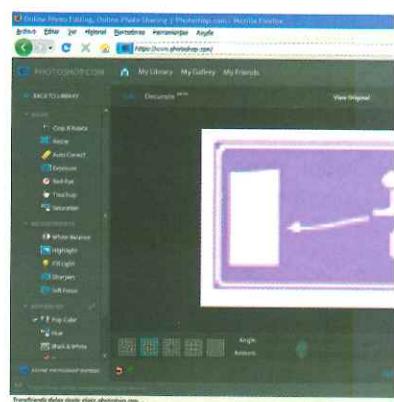
1. Busca en internet las URN: número de bastidor de un coche, IBAN, ISBN, código de barras, teléfono, número de pasaporte, etc.
2. Para visualizar las URL completas entra en: 20minutos.es y pulsa sobre la ciudad más cercana a tu municipio. Apunta las URL completas. ¿Puedes ver en qué directorio están guardadas las páginas web? En Google Vídeos, ¿puedes saber cuál es tu sistema operativo, tu idioma y el código de tu alfabeto?
3. Entra en Gmail o Hotmail. ¿Está activado el HTTPS? Localiza el candado, ¿cómo está? Entra en alguna tienda o banco hasta que te pida el número de tarjeta de crédito. ¿Sale HTTPS? ¿Identificas la AC?

5. Aplicaciones web

Las aplicaciones web son aquellas que los usuarios pueden utilizar accediendo a un servidor web. Suele tratarse de aplicaciones que se codifican en un lenguaje soportado por los navegadores (HTML, JavaScript, Java, PHP, ASP.NET [C#, VB.NET o VBScript], Perl, Ruby, Python, XML, ActionScript [Flash]).

Las más populares son:

- Webmail: correo electrónico web, como Hotmail o Gmail.
- Wiki: web cuyo contenido es colaborativo, cualquier usuario puede añadir o editar su contenido. Usa una tecnología donde el título de la página es la URL relativa de la página web. Los wikis más conocidos son Wikipedia, Mediawiki, etc.
- Weblogs: bitácoras o diarios, como Blogger, WordPress, Fotolog, Videolog, etc.
- Tiendas en línea: montadas por el usuario o, por ejemplo, por su caja o banco.
- Juegos: como Counter Strike, WOW, Aion, Conan, Warcraft, Lineage, Minijuegos, etc.
- Aplicaciones ofimáticas: Google Wave o Google Apps (suites), Google Spreadsheets (hoja de cálculo), Google Page Creator (editor de páginas web), Google Calc (calculadora), etc.
- SO: sistemas operativos, como EyeOS.
- Videochat: existen muchos de código abierto en Flash.
- Webmessenger: como el Windows Live Messenger Online.
- Otros: de diseño (Adobe Photoshop Express), de antivirus (como QuickScan de Bitdefender, Panda ActiveScan, Kaspersky Online, McAfee FreeScan, Eset NOD32), de compresores (Shrinkfile.net, Zip-online, Wobzip, Krunch), de vídeo (Youtube), de utilidades (File Destructor), etc.



5.1. Estructura y funcionamiento

Existen muchas posibilidades en la estructura de las aplicaciones web, pero la estructura más usual se basa en tres capas:

- Un navegador web.
- Un motor capaz de usar alguna tecnología web dinámica (con lenguaje de programación: Java, Flash Player, etc.).
- Una base de datos.

El funcionamiento es sencillo, el navegador web manda peticiones al motor, el cual ofrece servicios valiéndose de consultas y actualizaciones a la base de datos y, a su vez, proporciona una interfaz de usuario.



ACTIVIDADES

4. Prueba las aplicaciones web: Wikipedia, Blogger, Minijuegos, Web Messenger (ebuddy.com), Google Docs, Photoshop Express (<https://www.photoshop.com>), Panda Active Scan (<http://www.pandasecurity.com/spain/homeusers/solutions/activescan/>) y Youtube. ¿Qué opinas sobre poder hacer tantas cosas sin necesidad de instalar nada e independientemente de dónde estés?

6. Servidores web en sistemas operativos libres y propietarios

caso práctico inicial

Álex debe seleccionar el mejor servidor dependiendo del sistema operativo del que disponga.

Los servidores web son de los pocos donde la familia de sistemas operativos UNIX (Linux, Minix, Unix, etc.) no tiene casi la exclusividad. Tanto en Linux como en Windows se puede utilizar Apache (el más usado, es multiplataforma, gratuito y de código abierto), que tiene una cuota en los últimos años que ronda entre el 50 y el 70%; sin embargo crece el uso de Microsoft IIS (para Windows XP Pro, Vista Business, Windows 7 Ultimate, NT 3.51 y 4, Server 2003, 2008 y 2010) con una cuota del 25 al 35%. Los servidores web se instalan para el alojamiento de páginas web, bases de datos, aplicaciones web, etc., pero también para probar las aplicaciones web sin necesidad de subirlas a internet.

6.1. Instalación

saber más

Para instalar Xampp (en Linux, Windows 98, NT, 2000, XP, Vista y Windows 7, Mac OS X, Solaris...) se puede descargar en:

<http://www.apachefriends.org/es/xampp.html>

O en:

<http://sourceforge.net/projects/xampp>

Para Windows, instalaremos en modo gráfico, Wamp desde:

<http://www.wampserver.com>

O en:

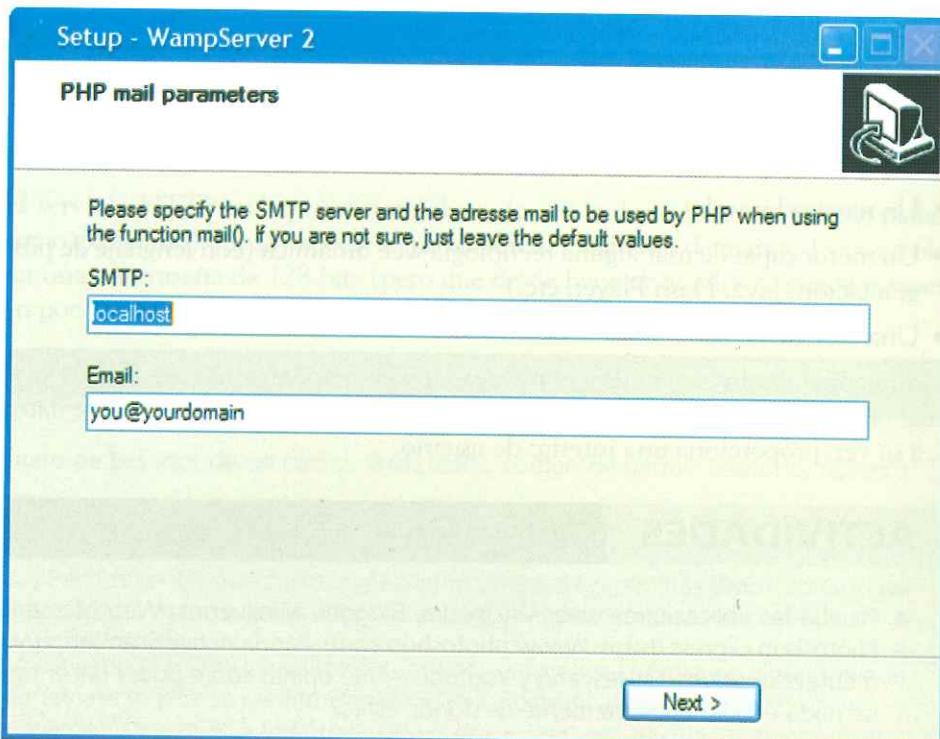
<http://sourceforge.net/projects/wampserver>

O en:

<http://www.apache.org>

Para instalar un servidor Apache necesitamos un procesador 486, con 4 MB de disco duro (15 libres para la instalación). Para establecerlo en el terminal instalamos el paquete apache2 y, opcionalmente, el paquete **apache2-mpm-prefork**. En el modo gráfico de Linux, instalaremos el grupo de aplicaciones Xampp, que es un programa que contiene Apache, MySQL, PHP y Perl. Existe una versión para Linux, llamada LAMP (soporta Python); otra para Windows, cuyo nombre es Wamp (es un archivo comprimido en RAR o ZIP a elegir); y otra llamada Mamp para Mac OS X, etc.

En Windows bajamos Wamp (Wamp es para el modo texto y Wamp para el modo gráfico). Inicialmente nos pedirá el dominio SMTP y el correo electrónico del administrador. Los requerimientos en Windows dependen del sistema operativo.



Para instalar IIS en Windows Server 2003, iremos a INICIO > HERRAMIENTAS ADMINISTRATIVAS > ADMINISTRE SU SERVIDOR, y después a AGREGAR O QUITAR FUNCIÓN. Pulsemos SIGUIENTE y seleccionaremos SERVIDOR DE APLICACIONES (IIS, ASP.NET). A continuación pincharemos en SIGUIENTE y seleccionaremos EXTENSIONES DE SERVIDOR DE FRONTPAGE y/o HABILITAR ASP.NET si fuese necesario (en el caso de querer programar en ASP.NET o usar FrontPage). De nuevo en SIGUIENTE (dos veces). Nos pedirá el CD de Windows. Otra opción es realizarlo desde INICIO > PANEL DE CONTROL > AGREGAR O QUITAR PROGRAMAS > SERVIDOR DE APLICACIONES.

En Windows XP Profesional (para pocos clientes a la vez) debemos instalar el servidor IIS desde INSTALAR COMPONENTES OPCIONALES DE WINDOWS (que aparece al introducir el CD de instalación de Windows). También se podría hacer desde la opción AGREGAR O QUITAR PROGRAMAS, del PANEL DE CONTROL, y luego pulsar AGREGAR O QUITAR COMPONENTES DE WINDOWS. En ambos casos, seleccionaríamos la opción SERVICIOS DE INTERNET INFORMATION SERVER (IIS).

En Windows Vista debemos ir a INICIO > PROGRAMAS > ACTIVAR O DESACTIVAR LAS CARACTERÍSTICAS DE WINDOWS. Desplegar INTERNET INFORMATION SERVICES y activar las opciones HERRAMIENTAS DE ADMINISTRACIÓN WEB y SERVICIOS WORLD WIDE WEB.

Y, en Windows 7, desde INICIO > PANEL DE CONTROL > PROGRAMAS Y COMPONENTES, en la esquina superior izquierda aparece AGREGAR O QUITAR COMPONENTES DE WINDOWS y, en el cuadro de diálogo, aparece INTERNET INFORMATION SERVER. Pulsar ACEPTAR.

6.2. Arranque y parada

En el caso de Apache, en el modo de texto de Linux, para arrancar el servidor ejecutaremos /etc/init.d/apache2 o /usr/local/apache2/bin/apachectl -f en versiones antiguas. Para pararlo apachectl -k stop.

Desde Windows, para arrancar Wamp (Apache) nos vamos a INICIO > PROGRAMAS > WAMP SERVER > START WAMP SERVER; para pararlo, con el botón secundario sobre el ícono del residente WAMP SERVER (como un cuentakilómetros), pulsaremos EXIT.

Una vez arrancado tendremos que pulsar sobre el ícono con el botón primario del ratón y pinchar en START ALL SERVICES (si deseamos arrancar todos los servicios).

Wampp, Xampp, Mampp, Lampp, etc., tienen archivos .bat o scripts en sus carpetas de instalación llamados start y stop (por ejemplo en Wampp, en la carpeta ..\wampp2\apache_start.bat y mysql_start.bat, etc.).

Por último, en Windows, para arrancar IIS (Microsoft) desde INICIO > PANEL DE CONTROL > HERRAMIENTAS ADMINISTRATIVAS > IIS y con el botón secundario del ratón, pulsaremos INICIAR.

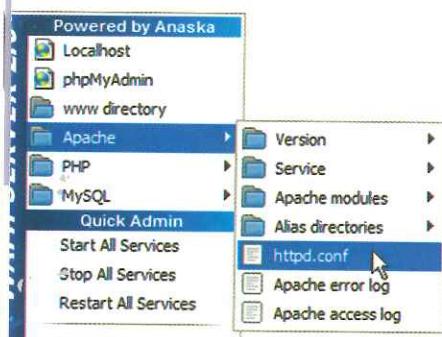
Para pararlo sería igual, pero pulsando la opción DETENER.



6.3. Ficheros y parámetros de configuración

Apache tiene el archivo de configuración httpd.conf, que se encuentra en la carpeta de Linux /etc/apache2. Si hemos instalado Xampp lo encontraremos en /etc/lampp/apache/conf o en ..\wampp2\apache\conf en Windows.

Si hemos instalado Wamp está ubicado en C:/wamp/bin/apache/Apache2.2.11/conf, pero accesible desde el ícono del residente APACHE > HTTPD.CONF.



↑ WampServer 2.0.

recuerda

En algunas versiones de Linux, el archivo de configuración se llama apache2.conf

caso práctico inicial

El archivo de configuración de Apache es crucial para que el servidor funcione correctamente.

EJEMPLO

```

Powered by Anaska
localhost
phpMyAdmin
www directory
Apache
PHP
MySQL
Quick Admin
Start All Services
Stop All Services
Restart All Services
Put Online

LoadModule php4_module libexec/libphp4.so //Carga módulo.
Include /etc/apache2/ssl-global.conf
Listen 80 //Escucha el puerto 80.
Listen 8080 //Escucha el puerto 8080.
<IfDefine SSL> Listen 443 </IfDefine> //Solo si el módulo
                                            SSL está instalado,
                                            lo escucha.
Port 80 //Define el puerto 80 por defecto.
ServerAdmin webmaster@here.com //mail del admin.
ServerName www.here.com //Nombre público DNS o IP.
DocumentRoot "/var/www/html" //Dir. raíz del site ppal.
<Directory "/var/www/html">
    Order allow,deny
    Allow from all
</Directory> //Orden página inicio por omisión.
DirectoryIndex index.html index.htm index.shtml default
                .htm index.php
TypesConfig /usr/local/apache/conf/mime.types //MIME Default.
    Type text/plain
AddLanguage es .es //Alfabeto para .es.
AddCharset ISO-8859-1 .iso8859-1 .latin1
ErrorDocument 404 /missing.html //Redirección error 404.

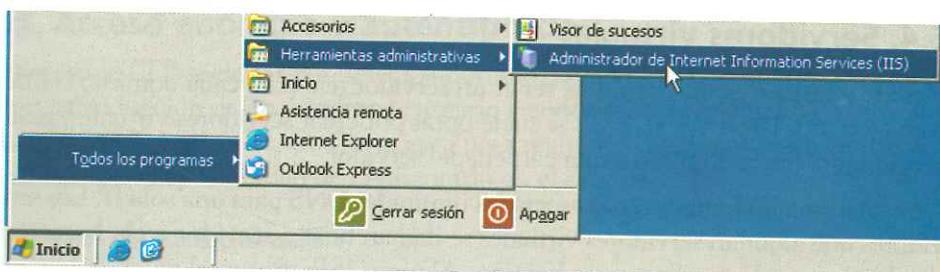
```

Este archivo contiene unos parámetros básicos que suelen ser:

- LoadModule: carga un módulo (applet, plugin...) en el ejemplo el de PHP.
- Include: añade otros archivos de configuración.
- Listen: puerto/s por los que escucha las peticiones HTTP.
- Port: puerto por defecto al que redirecciona si no se especifica otro.
- ServerAdmin: correo del administrador.
- ServerName: nombre público del servidor (DNS o IP).
- DocumentRoot: carpeta raíz por defecto del sitio web.
- DirectoryIndex: determina cuáles son los archivos por defecto (si existe omisión al especificarlo por parte del cliente) y en qué orden deben buscarse.
- TypesConfig: la especificación de los tipos MIME.
- AddLanguage: idiomas soportados.
- AddCharset: mapas de caracteres soportados.
- ErrorDocument: redirecciona un error a una página web local, a una URL o envía un mensaje de texto (se especifica entre dobles comillas anglosajonas [" "]).

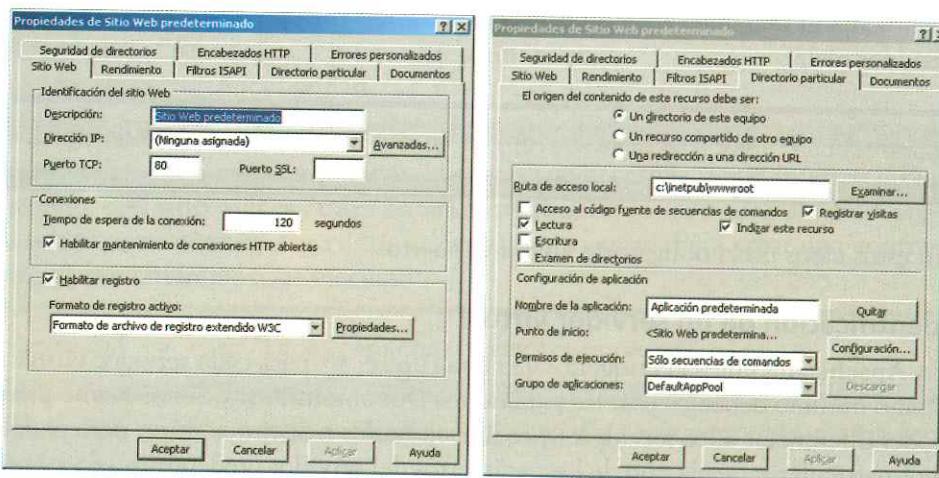
En IIS la configuración es gráfica y podemos modificar estas configuraciones entrando en INICIO > PANEL DE CONTROL > HERRAMIENTAS ADMINISTRATIVAS, donde haremos doble clic sobre el ícono de ADMINISTRADOR DE INTERNET INFORMATION SERVICES (IIS) (ver imagen en la página siguiente). Desplegaremos nuestro equipo y, con el botón secundario del ratón, pulsaremos en SITIO WEB PREDETERMINADO y seleccionaremos PROPIEDADES.

Gestión de servidores web

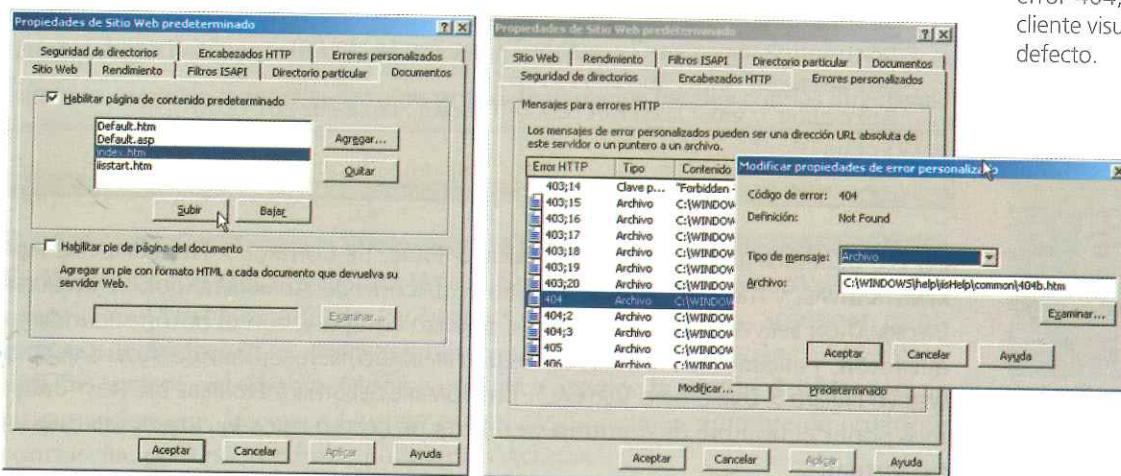


Las opciones básicas que podemos modificar son, por solapas:

- SITIO WEB:
 - DIRECCIÓN IP: usaremos 127.0.0.1 o la IP del servidor.
 - PUERTO TCP: dejaremos el 80 o pondremos el que nos interese.



- DIRECTORIO PARTICULAR: en RUTA DE ACCESO LOCAL escribiremos la ruta del directorio del sitio web. Deberemos comprobar que solo está activada la de LECTURA.
- DOCUMENTOS: en la opción de HABILITAR PÁGINA DE CONTENIDO PREDETERMINADO podremos añadir archivos con la opción AGREGAR, por ejemplo index.html, y ordenarlos (según preferencia) con las flechas.
- ERRORES PERSONALIZADOS: seleccionaremos el error (según su código), iremos a PROPIEDADES, en TIPO DE MENSAJE seleccionaremos ARCHIVO, URL o MENSAJE y, en el siguiente cuadro, seleccionaremos o escribiremos el destino (por ejemplo en ARCHIVO, pulsaremos en EXAMINAR y buscaremos el archivo a visualizar).



recuerda

Los documentos predeterminados pueden ser varios (por ejemplo, index.html, default.htm y home.php). Si el usuario omite el nombre del archivo en la URL se visualizará el primero de la lista, si este no existe será el segundo y así sucesivamente. Si no encuentra ninguno buscará la redirección del error 404, sino el navegador del cliente visualizará el que tenga por defecto.

6.4. Servidores virtuales

caso práctico inicial

Alex debe tener claro si desea un servidor propio, alquilado o uno virtual de alquiler. Los parámetros para elegirlo suelen ser económicos.

Debido a que no es muy rentable tener un servidor real para cada dominio o sitio web (servidor RAID, SAI, etc.) se suele optar por crear servidores virtuales, redireccionando cada dominio a una carpeta del servidor.

De esta forma podríamos tener asociadas ilimitadas DNS para una sola IP. Los servidores que alquilan servidores virtuales se llaman **host** o **servidor web**, la acción de alquilar es a la que llamamos **hosting**. Algunos ISP ofrecen alquiler de DNS y hosting desde 20 euros. Si deseamos un servidor propio, se debe alquilar un **servidor dedicado**, aunque muchos ya lo alquilan sobre **máquinas virtuales**.

Nombre de encabezado de host

Para crear servidores virtuales necesitamos activar esa opción. En Apache solo debemos añadir una línea al archivo de configuración (`httpd.conf`), con la IP del servidor real:

```
NameVirtualHost 192.168.0.100:80
```

//Site virtual.

En estos casos no es obligatorio añadir el puerto.

Identificación de un servidor virtual

caso práctico inicial

Alex debe saber configurar servidores virtuales (distintos sites) con Apache, pero también debe tener claro si desea las configuraciones particulares.

En Apache añadimos la etiqueta `<VirtualHost *>` para cada servidor virtual. Como mínimo debemos usar los parámetros `DocumentRoot` y `ServerName` para poder diferenciar a los servidores y redirigirlos a distintos sitios, pero podemos particularizar cada uno de los parámetros del servidor predeterminado (si no los especificamos, heredan los del servidor real).

```
<VirtualHost 192.168.0.100> //Configuración particular del virtual.
  ServerAdmin webmaster@3colegas.com
  DocumentRoot /var/www/html/3colegas.com
  ServerName www.3colegas.com
  ErrorLog /var/log/httpd/3colegas.com-error_log
</VirtualHost>

<VirtualHost 192.168.0.100>
  ServerAdmin webmaster@3colegas.info
  DocumentRoot /var/www/html/3colegas.info
  ServerName www.3colegas.info
  ErrorLog /var/log/httpd/3colegas.info-error_log
</VirtualHost>
```

En IIS debemos añadir sites desde **INICIO > PANEL DE CONTROL > HERRAMIENTAS ADMINISTRATIVAS**, y hacemos doble clic sobre el ícono de **ADMINISTRADOR DE INTERNET INFORMATION SERVER (IIS)**. Desplegamos nuestro equipo y, con el botón secundario del ratón, pulsamos en **SITIO WEB PREDETERMINADO**, seleccionando **AGREGAR SITIO WEB** (o **NUEVO > DIRECTORIO VIRTUAL**). En **NOMBRE DEL SITIO** y **NOMBRE DEL HOST** debemos poner el nombre de dominio y en **RUTA DE ACCESO FÍSICA** la carpeta de este sitio virtual.

6.5. Acceso anónimo y autenticado

El acceso HTTP suele ser por defecto anónimo, pero existe la posibilidad de forzar la introducción de un nombre de usuario y contraseña. Hay mucho código escrito en PHP, ASP, CGI, C#, Java, etc., para realizarlo mediante bases de datos, pero tenemos la posibilidad de configurarlo en el servidor.

Métodos de autenticación

En Apache tenemos dos métodos para la autenticación de usuarios locales para el acceso a un directorio: dentro del archivo principal de configuración con la sección `<Directory>` o creando un archivo `.htaccess` en cada directorio que requiera la autenticación. En cualquier caso, se necesita la directiva o el parámetro de configuración `AllowOverride` y los módulos `mod_access` y `mod_auth`:

```
LoadModule mod_access libexec/mod_access.so
LoadModule mod_auth libexec/mod_auth.so
AllowOverride AuthConfig
```

Primero debemos crear un archivo de claves (no accesible desde internet). Por ejemplo, si el sitio web es `/usr/local/apache/htdocs/inetpub`, guardaremos las contraseñas en `/usr/local/apache/passwd`. Para crear una contraseña nueva tendremos que ejecutar:

```
htpasswd -c /usr/local/apache/passwd/passwords alex
```

Esta opción reclamará introducir una contraseña dos veces para cada usuario (en el ejemplo `alex`).

Imaginemos que deseamos proteger la carpeta `/usr/local/apache/htdocs/secreto`, entonces introduciremos la siguiente sección en el archivo de configuración:

```
<Directory /usr/local/apache/htdocs/secreto >
  AuthType Basic
  AuthName "Acceso denegado"
  AuthUserFile /usr/local/apache/passwd/passwords
  Require user alex
</Directory>
```

O bien crearemos un archivo `.htaccess` en la carpeta e introduciremos las anteriores directivas pero sin meterlas en una sección `<Directory>`.

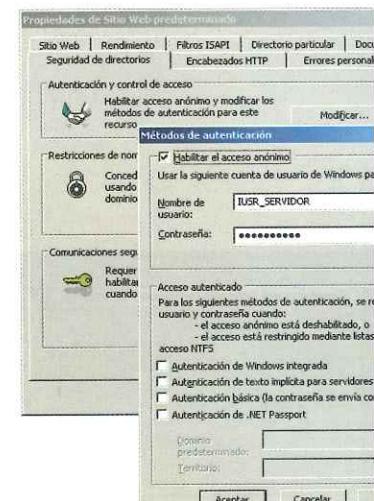
También se puede dejar acceder a un grupo de usuarios locales con las directivas:

```
AuthGroupFile /usr/local/apache/passwd/groups
Require group GroupName
```

En Windows con IIS, desde SITIOS WEB (o SITIO WEB PREDETERMINADO) seleccionaremos con el botón secundario PROPIEDADES e iremos a la solapa SEGURIDAD DE DIRECTORIOS (o SEGURIDAD DE ARCHIVOS). Una vez hecho esto, en AUTENTICACIÓN Y CONTROL DE ACCESO, pulsaremos MODIFICAR y activaremos la casilla del MÉTODO DE AUTENTICACIÓN que deseemos (AUTENTICACIÓN BÁSICA requiere nombre de usuario y contraseña, pero no lo cifra) y pulsaremos ACEPTAR.

caso práctico inicial

Alex necesita la configuración de autenticación.



Restricciones de acceso a recursos

Para filtros genéricos de Apache, en donde en vez del nombre de usuario prime la DNS o IP de la que provienen los usuarios, usaremos `Allow` (para permitir) y `Deny` (para denegar). Se puede utilizar el comodín `all` (para todos), partes de URL o TLD (`.net.es`) e inicio de redes (`192.168.0`) o rangos de IP (`192`). Con `Order` damos preferencia a denegar o a permitir.

EJEMPLO

```
Order deny,allow
Deny from all
Deny from hackers.org //O solo .ru.
Deny from 192.101.2.111 //O solo 192.168.
Allow from 3colegas.com
```

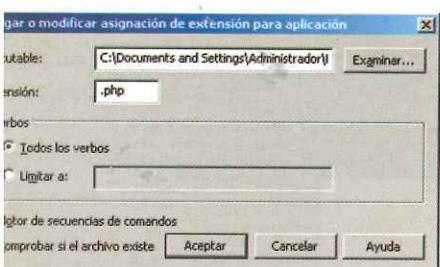
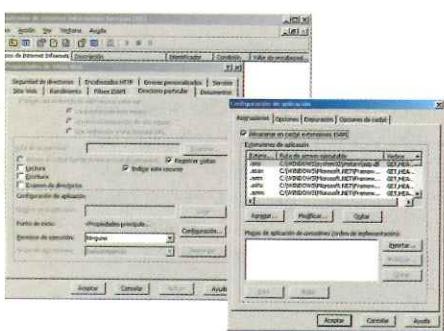
Si lo que queremos es filtrar direcciones IP y DNS en Windows con IIS, debemos pulsar **RESTRICCIONES DE NOMBRE DE DOMINIO Y DIRECCIÓN IP**. Pulsaremos sobre **CONCEDERÁ EL ACCESO** y después en **AGREGAR**; en el cuadro de diálogo **DENEGAR ACCESO A** escribiremos la opción para posteriormente pulsar **ACEPTAR**.

6.6. Ejecución de código

Para configurar la ejecución de código en el servidor debemos activar los lenguajes de programación o las tecnologías soportadas.

En el caso de Linux hemos instalado Xamp, que ya lleva una configuración por defecto de PHP y MySQL. Aun así, debemos cerciorarnos de que estos servidores estén configurados y de que se pueda acceder a ellos desde el servidor web. En el archivo de configuración deben aparecer estas líneas:

```
LoadModule php5_module "c:/wamp/bin/php/php5.3.0/
php5apache2_2.dll"
LoadModule cgi_module modules/mod_cgi.so
AddType application/x-httpd-php .php
AddType application/x-httpd-php .php3
<Directory "cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>
```



En Windows, al instalar el servidor nos pide si queremos instalar **EXTENSIONES DE SERVIDOR DE FRONTPAGE** y/o **HABILITAR ASP.NET** para soportar ASP, CGI, etc. Una vez instalado podemos añadir esta compatibilidad desde las **PROPIEDADES DE SITIOS WEB**, en la solapa **DIRECTORIO PARTICULAR**, en **CONFIGURAR APlicACIÓN**, pulsaremos **CONFIGURACIÓN** y en la ventana **ASIGNACIONES PARA LA APlicACIÓN** en **AGREGAR**. En la ventana **AGREGAR O MODIFICAR ASIGNACIÓN DE EXTENSIÓN PARA APlicACIÓN** pulsaremos **EXAMINAR** y buscaremos el ejecutable (`php.exe` por ejemplo) y lo asociaremos a una extensión escribiendo en **EXTENSIÓN:** `.php` por ejemplo. Para finalizar pincharemos en **ACEPTAR**. Este método requiere la instalación de los servidores PHP y MySQL.

Scripts de servidor y de cliente

Los **scripts** (guiones) son un archivo de órdenes o de procesamiento por lotes. Suelen ser muy simples y se almacenan en texto plano. Habitualmente se utilizan para combinar componentes o interactuar con el SO o el usuario.

Existen dos tipos de scripts de servicios HTTP:

- **Scripts de servidor:** se ejecutan en el servidor. No tienen problemas de accesibilidad, pueden modificar cabeceras HTTP u obtener acceso a bases de datos y a otros archivos del servidor (incluso en directorios no públicos). Estos suelen programarse en JSP, ASP, PHP, CGI (C, C++, C#, Perl), .NET, Java y Python.
- **Scripts de cliente:** se ejecutan en el cliente. Tienen restricciones de acceso al servidor. Se suelen programar en JavaScript, JavaScript con XML (AJAX), ActionScript de Adobe Flash y en VBScript para Internet Explorer. Pueden modificar archivos HTML, XML, CSS, etc.

Todos ellos suelen tener una parte de HTML en el <HEAD> y están entre etiquetas <script>. Se encuentran en el <body> entre comentarios <--! o tienen su propia etiqueta <asp:> o son llamados con una etiqueta <object>.

En la web existen millones de scripts de servidor y cliente, aunque también tenemos las API de Google, Microsoft, Adobe, SUN, etc., donde se nos ofrece la posibilidad de conocer las cabeceras, datos de entrada y salida, y los prototipos de funciones de sus librerías de programas y aplicaciones.

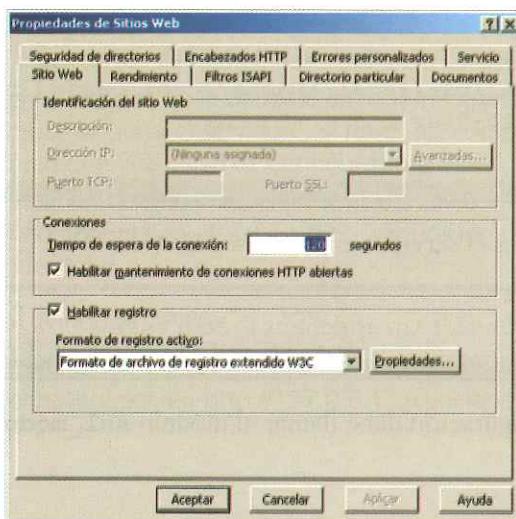
6.7. Monitorización y logs

En Apache podemos activar la monitorización del servidor añadiendo en el archivo de configuración la línea:

```
ErrorLog "c:/wamp/logs/apache_error.log"
```

Desde Wamp tenemos archivos de error log en los servidores Apache, PHP y MySQL accesibles pulsando con el botón principal del ratón sobre el ícono del programa residente. Apache también incluye un archivo llamado access.log que controla los accesos.

En IIS lo podemos configurar desde PROPIEDADES DE SITIOS WEB, en la solapa SITIO WEB, pulsando en el cuadro de opción HABILITAR REGISTRO, e incluso pulsando en PROPIEDADES podemos modificar el formato de archivo de registro.



recuerda

Autoridad de Certificación (AC), en inglés *Certificate Authority* o *Certification Authority (CA)*, será el término que emplearemos para la configuración de los servidores.

6.8. Establecimiento de conexiones seguras (HTTPS)

A la hora de establecer una conexión segura, tenemos que tener en cuenta que el servicio HTTP no es un protocolo seguro, ya que permite que las claves viajen por la Red a merced de programas sniffer. Para evitarlo podemos configurar accesos **HTTPS** con el servicio SSL o TLS. Para ello debemos crear nuestra propia Autoridad de Certificación (AC) o comprar una de las que comercializan VeriSign o utilizar las AC estatales. Para crear nuestra AC podemos hacerlo en Linux con el paquete openssl:

```
/usr/lib/ssl/misc/CA.sh -newca
```

saber más

OpenSSL se utiliza para implementar SSL, la web del proyecto es:

<http://www.openssl.org/>

Nos preguntará el Nombre de la CA (si existe, si no pulsaremos INTRO para crearlo), la Pass Phrase que sería la frase para acceder a la clave privada, así como otro tipo de datos (país, provincia...). Nos creará un archivo llamado `cacer.pem` para firmar nuestro certificado en `/private/carey.pem` donde estará la clave de la AC. Ahora creamos una clave CSR, clave para nuestro servidor triple-DES:

```
openssl genrsa -des3 -out server.key 1024
openssl req -new -key server.key -out server.csr
```

Cuando nos pregunte por nuestro Common Name, tendremos que poner nuestra DNS (`www.3colegas.com` en nuestro ejemplo). Ahora crearemos el CRT, firmando el CSR como AC.

```
ln -s server.csr newreq.pem
CA.sh -signreq
mv newcert.pem server.crt
```

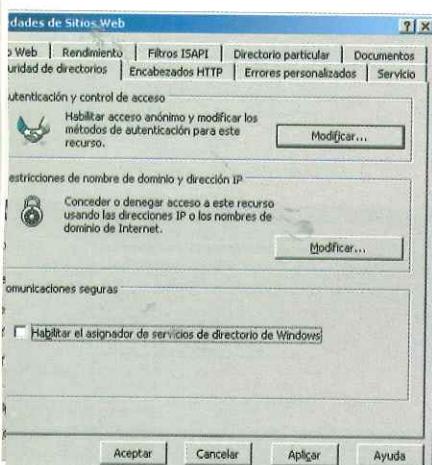
A lo largo de este proceso nos preguntará por la **pass phrase** de nuestro CA. Cuando nos pregunte si queremos firmarla contestaremos que **Sí**, así como también a la pregunta de hacer **comit**. Moveremos los archivos `Server.key` a la carpeta `ssl.key` de Apache y el archivo `Server.crt` a `ssl.crt`. En el archivo de configuración añadiremos:

```
<Directory "/usr/local/apache2/htdocs/secreto">
    SSLRequireSSL //fuerza acceso https
    AuthName "privatefiles"
    AuthType Basic
    AuthUserFile /usr/local/apache2/conf/passwd_basic
    Require valid-user
</Directory>
```

Ahora arrancaremos el servidor en modo seguro (HTTPS):

```
/usr/local/apache2/bin/apachectl startssl
```

El archivo de configuración debe llamar al módulo **ssl module**.

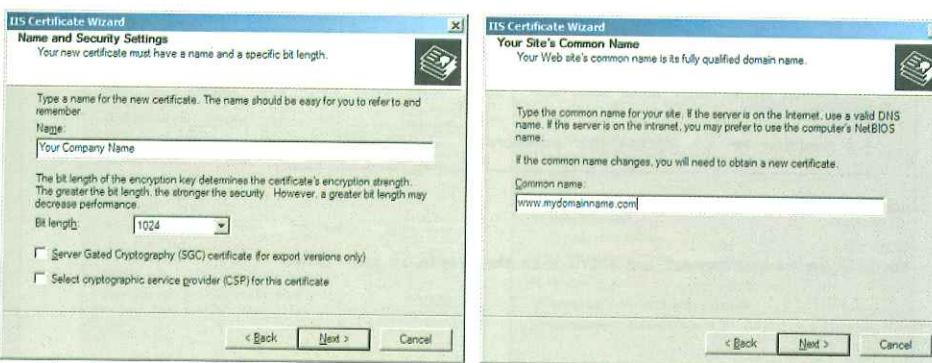
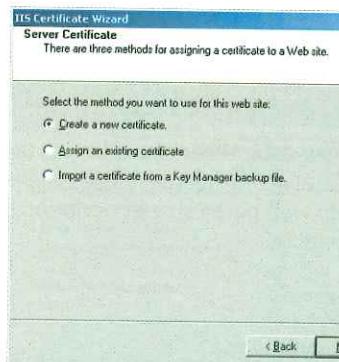


Gestión de servidores web

En Windows con IIS debemos tener instalado el servicio SSL (ver la Unidad de Trabajo correspondiente) y luego ir a INICIO > CONFIGURACIÓN > PANEL DE CONTROL > HERRAMIENTAS ADMINISTRATIVAS > ADMINISTRADOR DE SERVICIOS DE INTERNET y seleccionar el sitio web, pulsar con el botón secundario del ratón en PROPIEDADES, ir a la solapa SEGURIDAD DE DIRECTORIOS > COMUNICACIONES SEGURAS > MODIFICAR > REQUERIR CANAL SEGURO (SSL) > CIFRADO REQUIERE 128 BITS y pulsar ACEPTAR.

En la opción de CERTIFICADO DE SERVIDOR podemos CREAR UN NUEVO CERTIFICADO. Para ello pulsaremos en SIGUIENTE y nos crearemos nuestro certificado seleccionando el NOMBRE y la LONGITUD.

Pulsaremos varias veces en SIGUIENTE. Cuando nos pida el COMMON NAME escribiremos el nombre de la DNS. Para que funcione totalmente tendremos que crear-nos una AC con openssl.



ACTIVIDADES

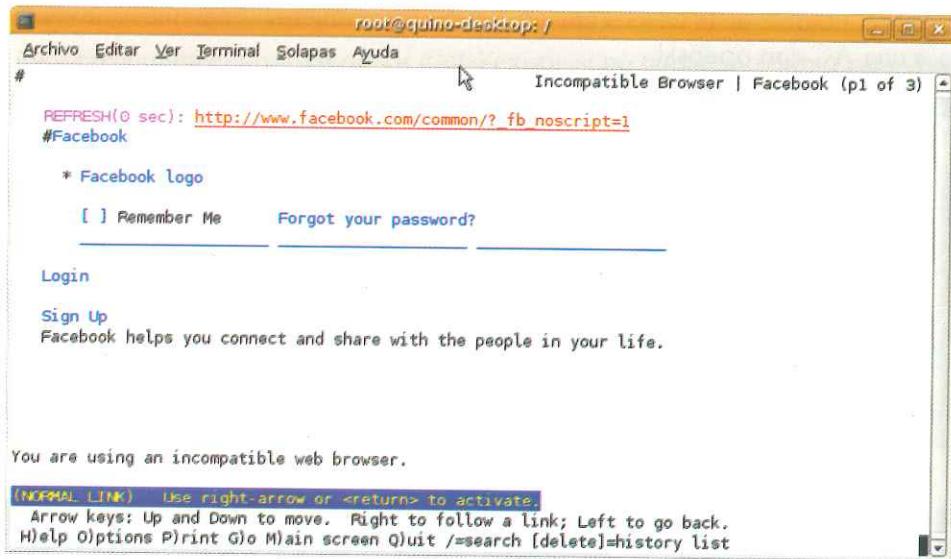
5. Instala el servidor Wamp. Arráncalo. Consulta los archivos log para comprobar que todo ha ido correctamente.
6. Visualiza el archivo de configuración de Apache y escribe cuáles son las configuraciones básicas por defecto.
7. En Windows, con Apache o Wamp arrancados:
 - Crea una página web llamada index.html en la carpeta c:/wamp/www/.
 - Entra en el navegador, en la barra de direcciones escribe http://127.0.0.1:80/index.html o http://localhost:80/index.html. ¿Funciona el servidor?, si no es así comprueba si está arrancado o si has cambiado algo sin querer en el archivo de configuración.
 - Prueba con http://127.0.0.1/index.html, http://127.0.0.1:80 y http://127.0.0.1. ¿Funciona con todos? ¿Por qué? Anota en el archivo de configuración dónde direcciona el puerto 80 y el documento predeterminado index.html.
8. En Linux, con Apache arrancado:
 - Crea una página web llamada index.html en la carpeta /var/www.
 - Entra en el navegador, en la barra de direcciones escribe http://127.0.0.1:80/index.html o http://localhost:80/index.html. ¿Funciona el servidor?, si no es así comprueba si el servidor está arrancado o si has cambiado algo sin querer en el archivo de configuración.
 - Prueba con http://127.0.0.1/index.html, http://127.0.0.1:80 y http://127.0.0.1:80/index.html. ¿Funciona con todos? ¿Por qué? Anota en el archivo de configuración dónde direcciona el puerto 80 y el documento predeterminado index.html.
9. Instala IIS en Windows XP Pro:
 - Crea una página web llamada default.asp en la carpeta C:.
 - Entra en el navegador, en http://127.0.0.1. ¿Funciona el servidor?

7. Navegadores web

caso práctico inicial

Alex debe tener claras las ventajas e inconvenientes de los clientes web para así seleccionar uno o varios.

Un navegador web (*browser*) es un programa que permite visualizar páginas web. Es un intérprete de código (normalmente HTML, XML, CSS...). Existen muchos navegadores gráficos. En 1993 nació Mosaic, que dio lugar a Netscape en 1994, a Internet Explorer (IE) en 1995 y a Mozilla en 1998. En 1994 nació Opera, otra nueva familia de navegadores; en el 2000 nació Konqueror, que dio lugar a Safari en 2003 y a Google Chrome (GC) en 2008. En el modo texto tenemos un gran número de navegadores, el más utilizado en Linux es Lynx (ya no viene integrado en las distribuciones actuales por lo que tendremos que instalar el paquete lynx).

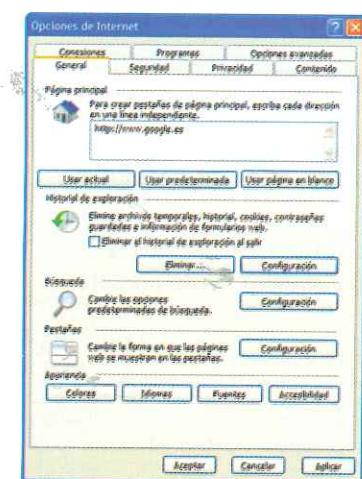


→ Aspecto de Facebook.com en Lynx.

En 2009 se registró un uso de las versiones de Internet Explorer (IE) que había bajado del 90 al 60%, de Mozilla Firefox (MF) superior al 30% y de otros el 8% (Google Chrome sigue creciendo en el mercado).

7.1. Parámetros de configuración

Para configurar las opciones básicas de Internet Explorer, desde HERRAMIENTAS, podemos seleccionar ELIMINAR EL HISTORIAL DE EXPLORACIÓN y escoger ARCHIVOS TEMPORALES DE INTERNET, COOKIES, HISTORIAL... y todo lo que deseemos. Desde HERRAMIENTAS > OPCIONES DE INTERNET tenemos las siguientes solapas:

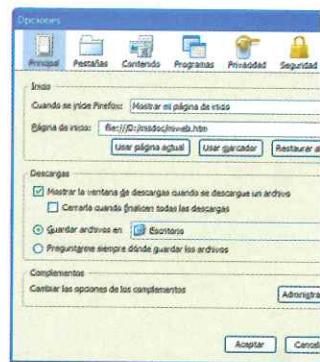
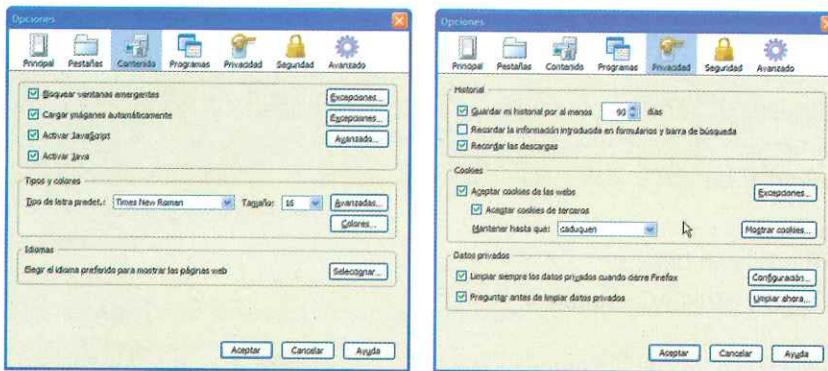


- **GENERAL:** donde podemos seleccionar la PÁGINA PRINCIPAL o página inicial (la que visualizaremos al entrar en el navegador o al pulsar al ícono HOME, que se corresponde con la casita). Desde el botón ELIMINAR podemos borrar los archivos temporales o con el de CONFIGURACIÓN DE HISTORIAL DE EXPLORACIÓN podemos configurar que lo elimine cuando llegue a unas cantidades concretas de espacio de disco duro.
- **SEGURIDAD:** en esta solapa se nos da la opción de crear nuestra lista de sitios web a los que no deseamos tener acceso. Pulsaremos en SITIOS RESTRINGIDOS y después en SITIOS, para por último pulsar AGREGAR y añadir las direcciones que queramos.
- **PRIVACIDAD:** también se nos ofrece la opción de configurar con privacidad MEDIA, para denegar las cookies de terceros, y ACTIVAR EL BLOQUEADOR DE VENTANAS EMERGENTES.
- **CONEXIONES:** desde el botón CONFIGURACIÓN DE LAN, podemos configurar un proxy.

Gestión de servidores web

Para configurar básicamente Firefox tenemos que ir a HERRAMIENTAS > OPCIONES y encontramos las solapas:

- PRINCIPAL: donde podemos escribir la PÁGINA DE INICIO o especificar dónde queremos que se graben los archivos de descarga.
- CONTENIDO: solapa en la que podremos BLOQUEAR VENTANAS EMERGENTES o aumentar el tamaño de letra en TIPOS Y COLORES, entre otras posibilidades.
- PRIVACIDAD: en esta solapa podemos especificar el número de días para: GUARDAR EL HISTORIAL POR AL MENOS... También se nos ofrece la posibilidad de desactivar ACEPTAR COOKIES DE TERCEROS y activar LIMPIAR SIEMPRE LOS DATOS PRIVADOS CUANDO SE CIERRE FIREFOX o pulsar LIMPIAR AHORA.
- AVANZADO: en la subsolapa RED, con el botón CONFIGURACIÓN, podremos activar el proxy.



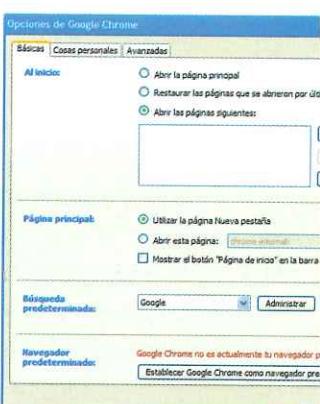
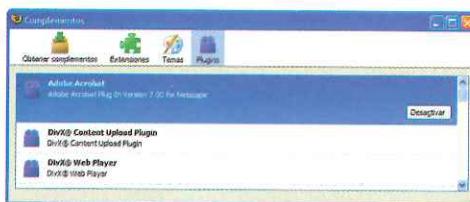
Para Google Chrome, se accede a las opciones de configuración desde el ícono LLAVE INGLESA > OPCIONES, en donde aparece una ventana con las siguientes solapas:

- BÁSICAS: para especificar la PÁGINA PRINCIPAL.
- AVANZADAS: para CAMBIAR LA CONFIGURACIÓN DEL PROXY, política de CONFIGURACIÓN DE COOKIES, directorio de DESCARGAS, etc.



7.2. Complementos

Todos los navegadores aceptan complementos (plugins, gadgets) que nos permiten añadir al navegador funciones de búsqueda, traducción, lectura de webmail (MS Live de Hotmail y Messenger o Gmail), ejecución de Adobe Reader y Flash, DivX o Java, extensiones o temas (para cambiar el aspecto), Windows Media Player, Real Player, Quicktime y Silverlight, etc.



En Internet Explorer se gestionan desde HERRAMIENTAS > OPCIONES DE INTERNET > PROGRAMAS y el botón ADMINISTRAR COMPLEMENTOS.

En Mozilla Firefox podemos acceder desde HERRAMIENTAS > COMPLEMENTOS.

Google Chrome ya soporta los programas antes citados. Los complementos de aspecto y otros scripts de usuario se ejecutan desde las páginas de Google.