

→ Servidores Web: Apache e IIS

Máster



Daniel Rodríguez
Departamento de Ciencias de la Computación
Universidad de Alcalá

→ Conceptos básicos

→ Contenidos

- Conceptos básicos
- Apache
 - Instalación
 - Servidores virtuales
 - Autenticación de usuarios
 - Monitorización y optimización del servidor Apache
 - Apache y la generación de contenido dinámico
 - Seguridad
- IIS (Internet Information Server)
 - Instalación
 - Servidores virtuales
 - Seguridad

Agradecimientos:

- Transparencias de Apache basadas en Alberto Abián, UAH.



Servidores Web
Daniel Rodríguez

2

→ Protocolo HTTP

- El **protocolo de transferencia de hipertexto** (HTTP, HyperText Transfer Protocol) es el protocolo usado en cada transacción de la Web (WWW).
 - Desarrollado por el consorcio W3C y la IETF, colaboración que culminó en 1999 con la publicación de una serie de RFCs, siendo el más importante de ellos el RFC 2616, que especifica la versión 1.1.
- HTTP define la sintaxis y la semántica que utilizan los elementos software de la arquitectura Web (clientes, servidores, proxies) para comunicarse.
 - Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición (un navegador o un spider) se lo conoce como "user agent" (agente del usuario). A la información transmitida se la llama recurso y se la identifica mediante un URL.
 - Los recursos pueden ser archivos, el resultado de la ejecución de un programa, una consulta a una base de datos, la traducción automática de un documento, etc.



Servidores Web
Daniel Rodríguez

4

→ HTTP – Protocolo sin Estado

- HTTP es un protocolo sin estado, es decir, en general no guarda ninguna información sobre conexiones anteriores.



- El desarrollo de aplicaciones web necesita frecuentemente mantener estado. Para esto se usan las **cookies**, que es información que un servidor puede almacenar en el sistema cliente.
- Esto le permite a las aplicaciones web instituir la noción de "sesión", y también permite rastrear usuarios ya que las cookies pueden guardarse en el cliente por tiempo indeterminado.



→ HTTP – Métodos principales

- Principalmente, se dan los siguientes métodos:
- GET
 - Devolver un fichero
- POST
 - Enviar datos al servidor

Ejemplos:

```
http://www.uah.es/          GET / HTTP/1.0
http://www.uah.es/index.html GET /index.html HTTP/1.0
http://www.uah.es/prog/appl.html GET /prog/appl.html HTTP/1.0
```



→ HTTP

- La conversación que se lleva a cabo entre un cliente y un servidor se realiza mediante texto normal. Sobre Internet el protocolo está sobre TCP y generalmente sobre el puerto 80. El cliente envía al servidor la siguiente información (de manera absolutamente transparente):

1. El método de la petición (GET o POST).
2. El nombre del documento que desea.
3. La versión de protocolo HTTP que se empleará en la comunicación.
4. Una lista de los tipos de datos que está dispuesto a aceptar (por ejemplo, si tenemos deshabilitada la opción de cargar gráficos, no incluirá el tipo gráfico en la lista).
5. Su propio nombre y versión.
6. Más cosas como el cliente que es, etc. y una **línea en blanco** para indicar el final.



→ Protocolo HTTP

- Para obtener un recurso con el URL `http://www.example.com/index.html`
 1. Se abre un con el host `www.tuhost.example`, puerto 80 que es el puerto por defecto para HTTP.
 2. Se envía un mensaje en el estilo siguiente:

```
GET /index.html HTTP/1.1
Host: www.example.com
User-Agent: HTTPTool/1.0
Connection: close
[Línea en blanco]
```

- La respuesta del servidor está formada por encabezados seguidos del recurso solicitado, en el caso de una página Web:

```
HTTP/1.1 200 OK
Date: Fri, 31 Dec 2003 23:59:59 GMT
Content-Type: text/html
Content-Length: 1221

<html><body>
<h1>Página principal de tuHost</h1>
...
</body></html>
```



→ HTTP – Códigos de retorno

- 200 OK
- 201 Created
- 202 Accepted
- 204 No Content
- 301 Moved Permanently
- 302 Moved Temporarily
- 304 Not Modified
- 400 Bad Request
- 401 Unauthorized
- 403 Forbidden
- 404 Not Found
- 500 Internal Server Error
- 501 Not Implemented
- 502 Bad Gateway
- 503 Service Unavailable



→ HTTP y las capas inferiores

TCP/IP



Modelo OSI



→ MIME Types

- MIME "Multipurpose Internet Mail Extensions" (MIME definido en RFC 1521) se definió como estándar para enviar por correo electrónico con datos binarios.
 - Los usos típicos de MIME incluyen el envío de imágenes, audio, documentos de procesadores de texto o incluso ficheros de texto cuando es importante que sistema de correo no modifique ninguna parte del fichero.
 - MIME permite además etiquetar partes de un mensaje para que el receptor (o programa de correo) puede determinar que hacer con él.
- El algoritmo “Base64” se utiliza para convertir ficheros binarios a texto y viceversa.
- Este mismo concepto se aplicó en la Web para el envío sobre el protocolo HTTP de las diferentes partes de las que se compone un documento en HTML



→ Ejemplo Fichero mime.types

```
# This file controls what Internet media types are sent to the client for
# given file extension(s). Sending the correct media type to the client
# is important so they know how to handle the content of the file.
# Extra types can either be added here or by using an AddType directive
# in your config files. For more information about Internet media types,
# please read RFC 2045, 2046, 2047, 2048, and 2077. The Internet type
# registry is at <http://www.iana.org/assignments/media-types/>.

# MIME type                                         Extensions
application/activemessage                         ez
application/andrew-inset                           atom
application/applefile                            jpeg jpg jpe
application/atom+xml                             html htm
...
image/jpeg
text/html
text/parityfec
text/plain
txt text conf def list log in
```



→ Ejecución de aplicaciones con CGI

- Para poder ejecutar aplicaciones externas, se puede utilizar un protocolo llamado CGI (Common Gateway Interface),
 - CGI define una serie de parámetros conocidos como variables de entorno, que describen las peticiones del cliente.
 - Esto define una interfaz independiente entre los programas o scripts y el servidor HTTP



→ Apache



The Apache Software Foundation
<http://www.apache.org/>

- Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server (un servidor "parcheado").
- El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.
- Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.
- Modular, Open source, Multi-plataforma, Extensible, Popular (fácil conseguir ayuda y soporte) y gratuito .



→ Apache Módulos

- La arquitectura del servidor Apache es modular. El servidor consta de un núcleo (core) y mucha de la funcionalidad que podría considerarse básica para un servidor Web es provista por módulos.
- Por ejemplo, módulos de la funcionalidad básica incluyen:
 - mod_ssl - Comunicaciones Seguras vía TLS.
 - mod_rewrite - reescritura de direcciones servidas (generalmente utilizado para transformar páginas dinámicas como php en páginas estáticas html para así engañar a los navegantes o a los motores de búsqueda en cuanto a como fueron desarrolladas estas páginas).
 - mod_dav - Soporte del protocolo WebDAV (RFC 2518).
 - mod_deflate - Compresión transparente con el algoritmo deflate del contenido enviado al cliente.
 - mod_auth_ldap - Permite autenticar usuarios contra un servidor LDAP.
 - mod_proxy_ajp - Conector para enlazar con el servidor Jakarta Tomcat de páginas dinámicas en Java (servlets y JSP).

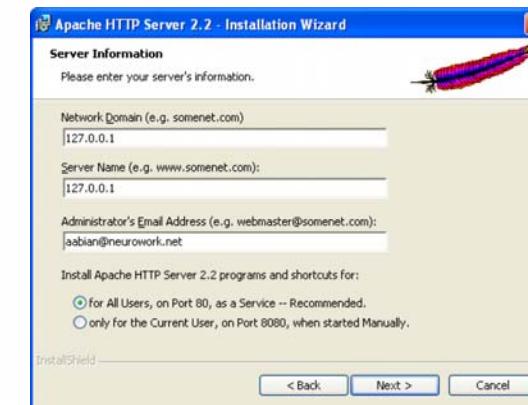


→ Modulos no básicos (externos)

- El servidor de base puede ser extendido con la inclusión de módulos externos entre los cuales se encuentran:
 - mod_perl - Páginas dinámicas en Perl.
 - mod_php - Páginas dinámicas en PHP.
 - mod_python - Páginas dinámicas en Python.
 - mod_rexx - Páginas dinámicas en REXX y Object REXX.
 - mod_ruby - Páginas dinámicas en Ruby.
 - mod_mono - Páginas dinámicas en Mono
 - mod_security - Filtrado a nivel de aplicación, para seguridad.

→ Instalación Apache (en Windows)

- <http://httpd.apache.org/>



→



→ Apache – Arranque y parada

- Arranque y parada del servicio

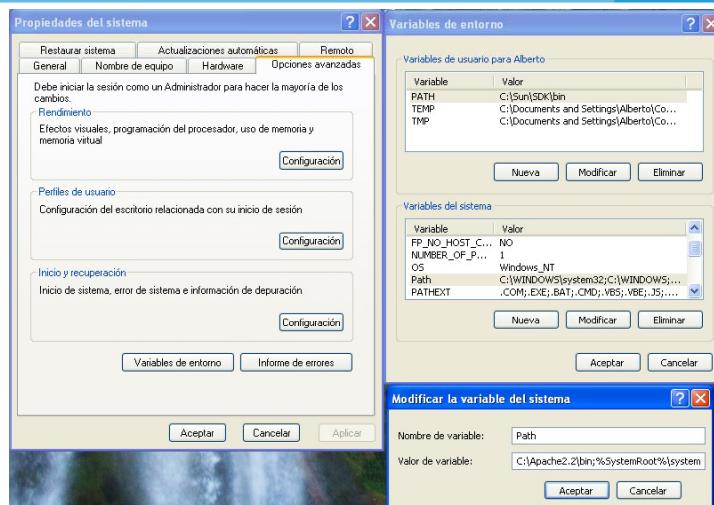


- ...y se comprueba con el navegador



It works!

→ Añadir el dir /bin al PATH



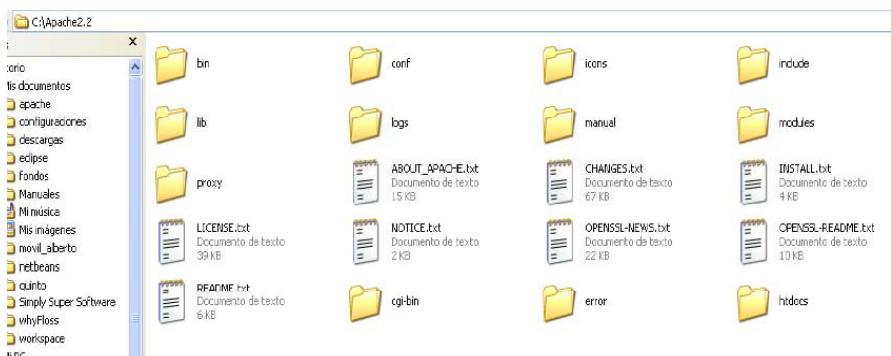
→ Herramientas de configuración

- La Web principal de la documentación del proyecto es:
- <http://httpd.apache.org/docs/2.2/>
- Otra de las páginas donde podemos obtener ayuda es:
- <http://www.apache-tools.com/>
- La herramienta más conocida para la configuración de Apache es **Comanche**, se trata de un programa gráfico multiplataforma que permite la modificación del archivo de configuración httpd.conf de forma sencilla. La página Web principal del proyecto es:
- <http://www.comanche.org/>



→ Directivas básicas de configuración

- Directorio de instalación:



→ Archivo httpd.conf

- El archivo de configuración principal de apache es

httpd.conf

- Comprobación de la sintaxis del archivo httpd.conf:

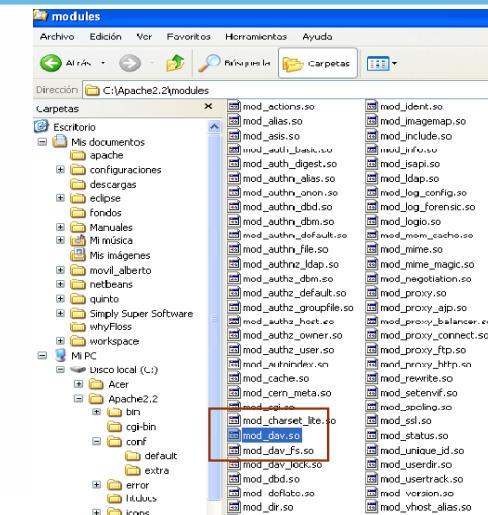
```
httpd -w -t -f c:\Apache2.2\conf\httpd.conf
```

```
httpd -w -t
```

→ Instalación de módulos dinámicos

- Apache es un servidor Web modular
 - Permite al administrador decidir cuáles son las funcionalidades que va a tener mediante la adición de módulos.
- Para entender cómo funcionan los módulos dinámicos se va a mostrar cómo instalar el módulo mod_dav:
 - WebDAV permite a usuarios remotos manipular de manera segura documentos concretos sin necesidad de FTP, permitiendo tareas como añadir, eliminar o actualizar archivos.
 - Lo primero que habría que hacer para instalar un módulo es bajarlo de internet y copiarlo en la carpeta "modules" que se encuentra dentro del directorio de instalación de Apache.
 - En el caso de WebDAV, se encuentra dentro de la distribución estándar de Apache.

→ Ejemplo Módulos dinámicos: WebDAV



→ Ejemplo Módulos dinámicos: WebDAV

- El siguiente paso que tenemos que dar es cargar el módulo de forma dinámica.
 - Para ello descomentamos la siguiente línea (o se crea si es que no existía):

```
#LoadModule dav_module modules/mod_dav.so
```

- Además, es conveniente activar los siguientes módulos:

```
LoadModule dav_fs_module modules/mod_dav_fs.so
LoadModule deflate_module modules/mod_deflate.so
```

→ Ejemplo Módulos dinámicos: WebDAV

- Posteriormente, incluir al final del archivo httpd.conf:

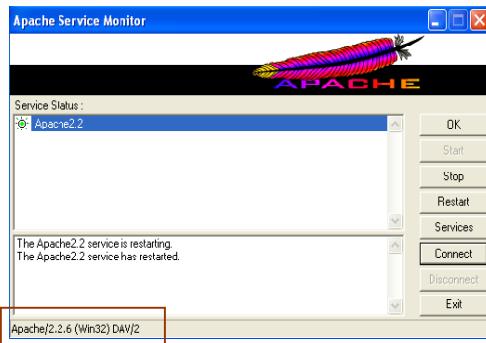
```
#Inclusión de mod_dav
Include conf/extra/httpd-dav.conf
```

- En el archivo conf/extra/httpd-dav.conf incluiremos las directivas propias del módulo WebDAV:

```
#CUIDADO: El directorio var debe existir
DavLockDB "C:/Apache2.2/var/DavLock"
DAVMinTimeout 600
Alias /test-dav "C:/Apache2.2/htdocs/test-dav"
<Directory "C:/Apache2.2/htdocs/test-dav">
    Dav On
    Order Allow,Deny
    Allow from all
</Directory>
```

→ Ejemplo Módulos dinámicos: WebDAV

- En la barra de estado del monitor de Apache ahora se muestra que está habilitado DAV versión 2.



29



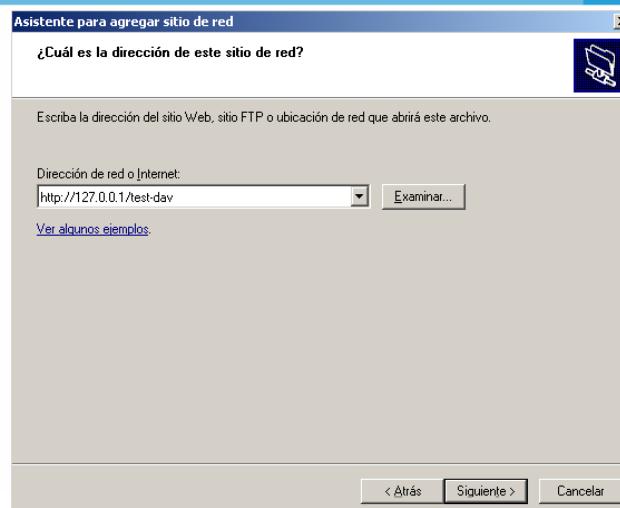
→ Ejemplo Módulos dinámicos: WebDAV

- Comprobación de mod_dav mediante el explorador de Windows:

1. Hacer click en **Mis sitios de Red**.
2. En la ventana derecha del explorador de Windows haga click en la opción **Añadir sitios de Red**.
3. En la ubicación, escribir: `http://127.0.0.1/test-dav/`



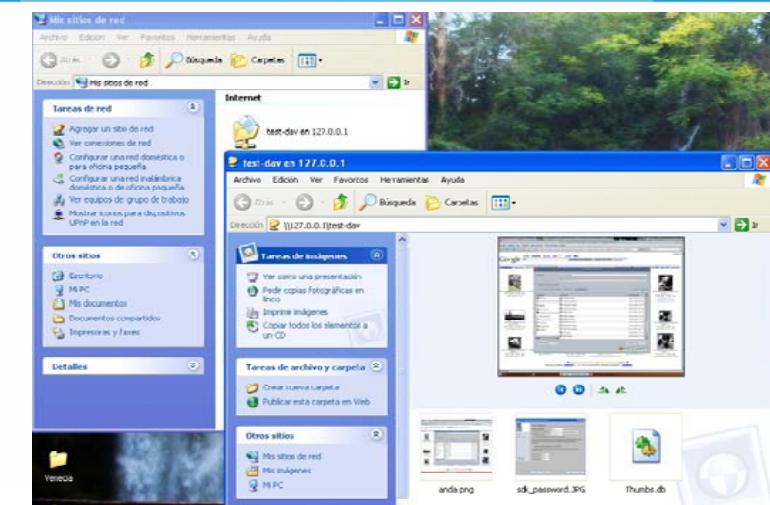
→ Ejemplo Módulos dinámicos: WebDAV



31



→ Ejemplo Módulos dinámicos: WebDAV



32



→ Servidores virtuales (*Virtual hosts*)

- El término Host Virtual se refiere a la práctica de ejecutar más de un sitio Web en la misma máquina:
 - Servidores virtuales por nombre
 - Servidores virtuales por IP
 - Servidores virtuales por PUERTO



→ Ejemplo. Servidores Virtuales

- Creación de dos *Hosts Virtuales*,
 - uno para la IP 127.0.0.1
 - y el otro para la IP de la red local. En Windows la IP local se puede ver desde la línea de comandos con:

 >ipconfig



→ Ejemplo. Servidores Virtuales

- Modificaciones en conf/extr/htpd-vhosts.conf:

```
<VirtualHost 127.0.0.1>
    ServerAdmin admin@127.0.0.1
    DocumentRoot "C:/Apache2.2/htdocs/vhost1/"
    ErrorLog "logs/127.0.0.1-error_log"
</VirtualHost>

<VirtualHost 192.168.1.55>
    ServerAdmin admin@192.168.1.55
    DocumentRoot "C:/Apache2.2/htdocs/vhost2/"
    ErrorLog "logs/192.168.1.55-error_log"
</VirtualHost>
```



→ Ejemplo. Servidores Virtuales

- Comprobar el archivo de configuración:

```
c:\>cd\Windows\system32\cmd.exe
UserDirModule <shared>
Syntax OK

C:\>apache2.2\bin\httpd.exe -v -t -f "C:/Apache2.2/conf/httpd.conf" -M -S
Server configuration:
  vhostbilibin (C:/Apache2.2/conf/extras/httpd-vhosts.conf):35
192.168.1.55:443

192.168.1.55:443
Loaded Modules:
core_module <static>
win32_module <shared>
mpm_win32_module <static>
http_module <static>
ssl_module <shared>
php7_module <shared>
actions_module <shared>
alias_module <shared>
rewrite_module <shared>
authn_basic_module <shared>
authn_default_module <shared>
authz_core_module <shared>
authz_host_module <shared>
authz_user_module <shared>
autoindex_module <shared>
curl_module <shared>
 dav_module <shared>
 dav_fs_module <shared>
 deflate_module <shared>
 dir_module <shared>
 env_module <shared>
 expires_module <shared>
 include_module <shared>
 isapi_module <shared>
 log_config_module <shared>
 mime_module <shared>
 negotiation_module <shared>
 setenvif_module <shared>
 userdir_module <shared>
 Syntax OK

C:\>apache2.2\bin\
```



→ Ejemplo 2. Servidores Virtuales

- Creación de cuatro Hosts Virtuales, uno para la ip 127.0.0.1 y el otro para la ip de la red local, además utilizar el puerto 80 y 8080 para distinguirlos.

```
Modificaciones en httpd.conf:  
Listen 127.0.0.1:80  
Listen 127.0.0.1:8080  
Listen 192.168.1.55:80  
Listen 192.168.1.55:8080  
  
# Virtual hosts  
Include conf/extra/httpd-vhosts.conf
```



→ Ejemplo 2. Servidores Virtuales

- Modificaciones en conf/extra/httpd-vhosts.conf

```
<VirtualHost 127.0.0.1:80>  
ServerAdmin admin@127.0.0.1  
DocumentRoot "htdocs/vhost1/"  
# ErrorLog "logs/127.0.0.1-error_log"  
# CustomLog "logs/127.0.0.1-access_log common"  
</VirtualHost>  
<VirtualHost 127.0.0.1:8080>  
ServerAdmin admin@127.0.0.1  
DocumentRoot "C:/Apache2.2/htdocs/vhost2/"  
# ErrorLog "logs/127.0.0.1-error_log"  
# CustomLog "logs/127.0.0.1-access_log common"  
</VirtualHost>  
<VirtualHost 192.168.1.55:80>  
ServerAdmin admin@192.168.1.55  
DocumentRoot "C:/Apache2.2/htdocs/vhost3/"  
ErrorLog "logs/192.168.1.55-error_log"  
# CustomLog "logs/192.168.1.55_log common"  
</VirtualHost>  
<VirtualHost 192.168.1.55:8080>  
ServerAdmin admin@192.168.1.55  
DocumentRoot "C:/Apache2.2/htdocs/vhost4/"  
# ErrorLog "logs/192.168.1.55-error_log"  
# CustomLog "logs/192.168.1.55_log common"  
</VirtualHost>
```



→ Ejemplo 2. Servidores Virtuales

```
C:\>apache2.2\bin>httpd.exe -v -t -f "C:/Apache2.2/conf/httpd.conf" -M -S  
VirtualHost configuration:  
192.168.1.55:80 winohlivion (C:/Apache2.2/conf/extra/httpd-vhosts.conf:42)  
192.168.1.55:8080 winohlivion (C:/Apache2.2/conf/extra/httpd-vhosts.conf:49)  
127.0.0.1:80 localhost (C:/Apache2.2/conf/extra/httpd-vhosts.conf:28)  
127.0.0.1:8080 localhost (C:/Apache2.2/conf/extra/httpd-vhosts.conf:35)  
Loaded Modules:  
core_module (static)  
crtsh_module (static)  
mpm_winnt_module (static)  
mpm_modular (static)  
http_module (static)  
so_module (static)  
dav_module (shared)  
dav_fs_module (shared)  
alias_module (shared)  
asis_module (shared)  
auth_basic_module (shared)  
authn_default_module (shared)  
authz_default_module (shared)  
authz_groupfile_module (shared)  
authz_host_module (shared)  
authz_user_module (shared)  
autoindex_module (shared)  
cgi_module (shared)  
 dav_module (shared)  
 dav_fs_module (shared)  
 deflate_module (shared)  
 dir_module (shared)  
 env_module (shared)  
 imagemap_module (shared)  
 include_module (shared)  
 isapi_module (shared)  
 langfilter_module (shared)  
 mime_module (shared)  
 negotiation_module (shared)  
 setenvif_module (shared)  
 userdir_module (shared)  
 Syntax OK  
C:\>apache2.2\bin>_
```



→ Ejemplo 2. Servidores Virtuales

- Comprobar el correcto funcionamiento



Virtual host 1



Virtual host 3



Virtual host 2



Virtual host 4



→ Autenticación de usuarios (Basic)

- Se realiza mediante los ficheros .htaccess y asociados (.htpasswd, .htgroup y .htdigest)
- La creación de un fichero de usuarios y contraseñas se efectua con la orden:

```
htpasswd -c "fichero" user
```

- la misma orden pero sin la opción -c se van añadiendo usuarios al archivo de autenticación.
- Ej:

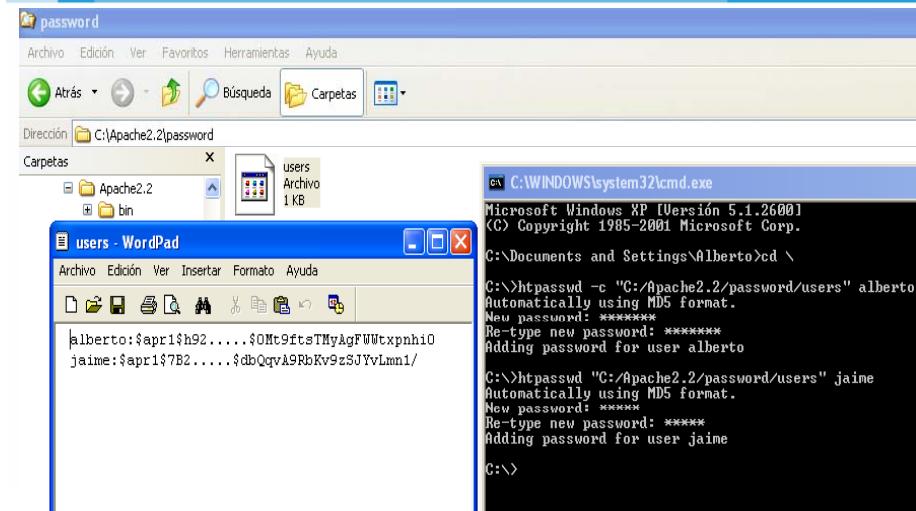
```
htpasswd -c users alberto
```

- El archivo de grupos, generalmente llamado .htgroups simplemente se edita con un editor de texto y el formato es el *nombre grupo* seguido por ":" y los *nombres de usuario* que pertenecen a cada grupo. Por ejemplo:

```
vendedores:alberto  
otros:alberto jaime
```



→ Autenticación de usuarios (Basic)



→ Autenticación de usuarios (Basic)

- Contenido del archivo .htaccess (a poner en cada directorio que queramos proteger) :

```
AuthType Basic  
AuthName "Contraseña requerida para entrar (.htaccess)"  
AuthUserFile "C:/Apache2.2/password/users"  
AuthGroupFile "C:/Apache2.2/password/groups"  
Require Group vendedores
```



→ Autenticación de usuarios (Basic)

- Protegiendo los archivos .htaccess y .htgroup en el fichero httpd.conf (suele estar activado por defecto) :

```
# The following lines prevent .htaccess and .htpasswd  
# files from being viewed by Web clients.  
#  
<Files ~ "^.ht">  
Order allow,deny  
Deny from all  
</Files>
```



→ Autenticación de usuarios (Basic)



Forbidden

You don't have permission to access /sensible/.htpasswd on this server.

45



→ Autenticación de usuarios (Basic)

- Utilizando los archivos .htaccess y .htgroups. Hay que copiar los archivos .ht* al directorio que queremos proteger con usuario y contraseña.
- Posteriormente hay que incluir la directiva AllowOverride.

```
<VirtualHost 127.0.0.1:80>
    ServerAdmin admin@127.0.0.1
    DocumentRoot "htdocs/vhost1/"
    # ErrorLog "logs/127.0.0.1-error_log"
    # CustomLog "logs/127.0.0.1-access_log common"
    # Para que funcione .htaccess
    <Directory "htdocs/vhost1/">
        AllowOverride All
    </Directory>
</VirtualHost>
```

46



→ Autenticación de usuarios (Basic)

- Las siguientes líneas son análogas a la utilización de .ht* pero mejoran la eficiencia.
- # prueba de autenticación severa alberto

```
<Directory "C:/Apache2.2/htdocs/vhost1/sensible">
    Satisfy All
    AuthType Basic
    AuthName "Entrar al directorio de documentación sensible"
    #user = alberto, pass = alberto
    AuthUserFile "C:/Apache2.2/password/users"
    # grupos existentes: vendedores, otros
    AuthGroupFile "C:/Apache2.2/passwrod/groups"
    Require group vendedores
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1
</Directory>
```

47



→ Autenticación de usuarios (Basic)

- Para comprobarlo hay que renombrar el archivo .htaccess y comprobar que en la pantalla en la que pide la contraseña ya no está escrito (.htaccess).

48



→ Autenticación de usuarios (digest)

- Para la autenticación utilizando el algoritmo *digest* (más seguro que Basic) es necesario cargar el módulo:

```
LoadModule auth_digest_module modules/mod_auth_digest.so
```

- Creando los archivos de usuario:

```
htdigest [ -c ] passwdfile realm username
```

-c - Crea el archivo de contraseñas

– Nota: si existe se **BORRA** y se crea vacío.

Passwdfile - Nombre del archivo en el que guardarán las contraseñas

Realm - Reino al que pertenece el nombre de usuario.

Username - Nombre de usuario que se desea crear, si no existe se crea una nueva entrada en el archivo, pero si existe tan solo se actualiza su password.



→ Autenticación de usuarios (digest)

```
C:\Apache2.2>htdigest -c "C:/Apache2.2/password/users.digest" "autenticacion-digest" alberto
Adding password for alberto in realm autenticacion-digest.
New password: *****
Re-type new password: *****

C:\Apache2.2>htdigest "C:/Apache2.2/password/users.digest" "autenticacion-digest"
" jaime
Adding user jaime in realm autenticacion-digest
New password: *****
Re-type new password: *****
C:\Apache2.2>
```



→ Autenticación de usuarios (digest)

```
# prueba de autenticación severa Alberto
<Directory "C:/Apache2.2/htdocs/vhost1/sensible_digest">
    Satisfy All
    AuthType Digest

    # Este es el reino al que debe pertenecer el usuario
    #debe estar escrito EXACTAMENTE igual que en la línea htdigest
    file REINO nombre
    AuthName "autenticacion-digest"
    AuthUserFile "C:/Apache2.2/password/users.digest"
    AuthGroupFile "C:/Apache2.2/password/groups"
    Require group vendedores
    Order deny,allow
    Deny from all
    Allow from 127.0.0.1
    Require valid-user

</Directory>
```



→ Gestión de Registros (*logs*)

- La configuración de los registros de acceso y los eventos que se producen en un servidor Web son de especial importancia pues pueden ayudar a:

- Monitorizar la utilización del sistema
- Detectar preventivamente un ataque al sistema, una debilidad que está siendo explotada
- Fallo en la configuración o en el sistema.



→ Tipos de registros

- Apache gestiona, por defecto, tres tipos de registros:
 - El registro de errores
 - Registro de acceso
 - Registro del identificador del proceso (PID) del demonio del servicio.
- Apache permite una selección flexible de lo que se quiere guardar por registro, dónde y en el formato en que se va a guardar esta información.



→ Registro de errores

- El registro de errores se guarda en un fichero que es marcado por la directiva `errorlog`.
 - Esta directiva también puede utilizarse para, en lugar de guardar la información en ficheros, utilizar el servicio syslog para registrar los eventos vía red y de forma centralizada.
- El uso del servicio syslog será de especial utilidad cuando tenemos sistemas de análisis de log centralizados o de correlación de eventos.

```
ErrorLog /ruta/ficheros/log/error_log
```

– ó

```
ErrorLog syslog:user
```



→ Registro de errores

- El nivel de información que se va a almacenar en los ficheros de registro se configura con la directiva `LogLevel` en la que se pueden ajustar diferentes valores:
 - `Emerg`: Sólo se almacenan los mensajes que dejan al sistema incapaz de ser utilizado.
 - `Alert`: Cuando se produce un error en el sistema que requiere la ejecución inmediata de una acción para corregirlo.
 - `Crit`: Fallos críticos del sistema. No requieren acción inmediata pero pueden dejar el sistema no disponible.
 - `Error`: Condiciones de error en el uso del sistema. No tiene porque afectar al uso del sistema.
 - `Warn`: Avisos. Se producen cuando algo no está realizándose correctamente. Puede producirse por un script o un cliente que no realiza la negociación correctamente con el servidor.
 - `Notice`: Información significativa del funcionamiento del sistema.
 - `Info`: Información general del sistema.
 - `debug`: Información de debugging del sistema. Cuando abre o cierra conexiones o ficheros, etc...



→ Registros de acceso

- Los registros de acceso al servidor se almacenan aparte de los errores del servidor.
 - Estos registros guardan información relativa a todos los accesos a documentos o intentos de acceso a los documentos.
- Para almacenar esta información Apache se apoya en el uso de dos módulos, que son:
 - `mod_log_config`: se utiliza para configurar el lugar y el formato de los ficheros de registro
 - `mod_setenvif`: para utilizar variables de entorno que permitan generar ficheros de registro condicionales, que serán muy útiles a la hora de detectar sucesos en entornos con mucho tráfico.



→ Registros de acceso

- Para configurar los registros de acceso utilizamos las directivas LogFormat y CustomLog.
 - LogFormat determina que información se va a escribir y de qué forma, en el fichero de registro.
- La forma común es tiene esta estructura:

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```
- Parámetros:
 - %h: IP del cliente.
 - %I: Identificación del cliente utilizando el servicio indentd.
 - %u: Usuario cliente.
 - %t: Fecha y hora.
 - %r: Petición realizada
 - %s: Código de status del servidor.
 - %b: Tamaño de la respuesta.



- Con la directiva LogFormat se puede crear el formato de fichero log utilizando estas variables o añadiendo información de cabeceras del cliente como:

```
LogFormat "%h %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" miformato
```
- Aquí hemos añadido %{Referer} y %{User-Agent} al formato de registro y hemos quitado %I y %u.
- Una vez definido el formato que a utilizar hemos de marcar el lugar de registro con:

```
CustomLog /ruta/logs/access_log miformato
```
- Podemos crear tantos ficheros de log como deseemos, así por ejemplo, en uno podemos almacenar la dirección IP y la hora y en otro la dirección IP, la petición y el referer, etc



→ Control de Logs

- Los ficheros de log suelen tener un rápido crecimiento por lo que deben ser controlados y tener un mantenimiento muy ajustado.
 - Para ello se pueden utilizar los sistemas de log rotacionales, que permiten, utilizando el programa externo rotatelogs crear archivos automáticos cada cierto tamaño o tiempo.
 - Para ello definimos en ErrorLog el formato de los ficheros:

```
ErrorLog "| bin/rotatelogs /var/logs/errorlog.%Y-%m-%d-%H_%M_%S 50M"
```



- Es importante el uso del pipe "|" para la dirección de los registros de logs.
 - En este ejemplo se utilizan las variables de fecha %Y (year), %m (month), %d (day),... para crear un nuevo archivo de log cada 50 MB de tamaño.
- O directamente con CustomLog, que creará logapache.nnnn (nnnn el tiempo de creación):

```
CustomLog "|bin/rotatelogs /ratalogs/logapache 50M" miformato
```



→ Generación de contenido dinámico

- Consiste en la ejecución en el servidor de programas.
- Como ejemplo, se muestra como instalar el PHP para poder crear páginas dinámicas
- PHP funciona con el módulo mod_php

<http://www.php.net/manual/es/install.windows.manual.php>



→ PHP: Descargar y descomprimir PHP

- El primer paso consiste en descargar la última versión de PHP. Podremos hacerlo desde la página oficial de PHP, en la sección de descargas.
 - <http://www.php.net/downloads.php>
- Debemos elegir la versión "zip package" que contiene todas las funcionalidades de PHP y el módulo necesario para instalarlo en Apache. Una vez descargado el paquete comprimido en .zip de PHP necesitamos descomprimirllo en nuestro disco duro.
 - Podemos utilizar el directorio raíz del disco duro para descomprimir los archivos. En ese caso, se creará un directorio llamado algo como "php-5.2.5-Win32" que colgará de nuestro directorio raíz. Se recomienda cambiar el nombre del directorio creado a algo como "c:\php". En todo caso, nos advierten en la página de PHP sobre no colocar ningún nombre de directorio que contenga espacio, pues algún servidor web puede dar problemas. Por ejemplo, cuidado con instalar PHP en un directorio como este "c:\archivos de programa\php", pues en la ruta tenemos directorios con espacios.



→ PHP: Copia de las DLL

- A continuación nos informan sobre la necesidad de copiar en nuestro directorio de sistema una serie de bibliotecas (.dll): php5apache2*.dll
- En Windows XP, el directorio de sistema donde debemos copiar las dll, es "C:\WINDOWS\system32".
 - Nota: no se deben mezclar las DLL de diversas versiones de PHP, porque de lo contrario, podría causarnos problemas.
 - El mencionado directorio de sistema puede variar de unas versiones a otras de Windows.



→ PHP: Definir un archivo php.ini

- Otro archivo que debemos copiar, esta vez en nuestro directorio Windows, es el php.ini, que guarda las opciones de configuración definidas para PHP.
- En la distribución de PHP se incluyen dos archivos php.ini que podemos utilizar directamente en nuestro sistema. Estos dos archivos se llaman "php.ini-dist" y "php.ini-recommended" y contienen unas opciones típicas de configuración de PHP.
 - Se recomienda utilizar "php.ini-recommended", porque viene optimizado para obtener los mejores niveles de seguridad. En cualquier caso, podemos editar en cualquier momento el contenido del archivo para modificar la configuración de PHP a nuestro gusto o necesidades.
 - Para definir el php.ini debemos hacer una copia del archivo de configuración escogido ("php.ini-dist" o "php.ini-recommended") y renombrarlo como el "php.ini". Posteriormente debemos copiarlo en nuestra carpeta Windows, que en Windows XP es "c:\windows"



→ PHP: Editar httpd.conf

- Posteriormente deberemos editar el archivo de configuración de Apache añadiendo un par de líneas de configuración del módulo de Apache.
- ```
LoadModule php5_module C:/php/php5apache2_2.dll
AddType application/x-httdp-php .php
Action application/x-httdp-php "c:/php/php.exe"
```
- El lugar adecuado para añadir esas líneas es en el bloque de carga de módulos, que podemos encontrar si buscamos por el texto LoadModule. Podemos añadir las líneas de carga del módulo PHP después de la carga de los otros módulos que vienen ya configurados en archivo httpd.conf de Apache.
  - Si no instalamos PHP en el directorio c:\php, debemos editar las líneas a colocar en el httpd.conf para colocar la ruta correcta al directorio donde está php5apache2.dll.



## → PHP: Comprobación

- Antes de acabar y probar si PHP se ha instalado correctamente, necesitamos copiar una última dll en el directorio sapi. Concretamente, la dll php5ts.dll, que podemos encontrar en nuestro directorio de instalación de PHP "c:\php\".
- Para terminar, podemos crear una página de prueba de PHP, que colocaremos en nuestro directorio de publicación de Apache. Podemos crear un archivo llamado, por ejemplo, "prueba.php", en el que colocaremos dentro el siguiente código:  
<?phpinfo();?>
- Esta función simplemente creará una página de muestra de las configuraciones definidas para PHP en ese servidor.
- Para acceder al archivo creado desde nuestro explorador, escribiremos en la barra de direcciones esta URL:  
<http://localhost/prueba.php>



## → PHP: Comprobación

PHP Version 5.0.4



| System                              | Linux genet 2.6.8-24.14-default #1 Tue Mar 29 09:27:43 UTC 2005 i686                                                  |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| Build Date                          | Apr 24 2005 20:39:33                                                                                                  |
| Configure Command                   | './configure' '--prefix=/opt/local/php' '--with-apxs=/opt/local/apache/bin/apxs' '--with-libdb2=/home/db2inst1/sqlib' |
| Server API                          | Apache                                                                                                                |
| Virtual Directory Support           | disabled                                                                                                              |
| Configuration File (php.ini) Path   | /opt/local/php/lib                                                                                                    |
| PHP API                             | 20031224                                                                                                              |
| PHP Extension                       | 20041030                                                                                                              |
| Zend Extension                      | 220040412                                                                                                             |
| Debug Build                         | no                                                                                                                    |
| Thread Safety                       | disabled                                                                                                              |
| IPv6 Support                        | enabled                                                                                                               |
| Registered PHP Streams              | php, file, http, ftp                                                                                                  |
| Registered Stream Socket Transports | tcp, udp, unix, udg                                                                                                   |



## → Otra forma más fácil en Windows...

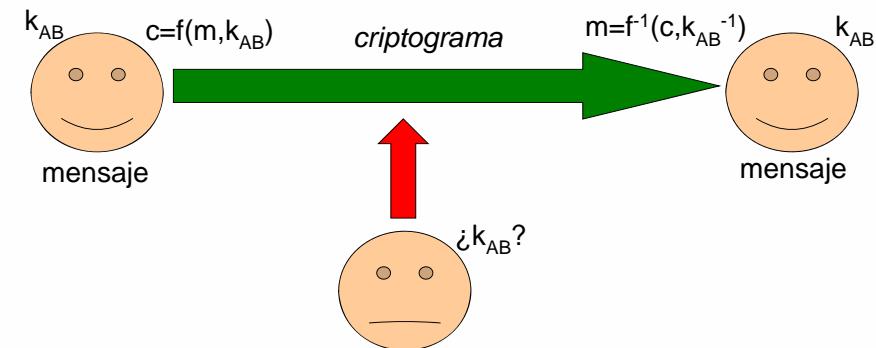
- Existen distribuciones que empaquetan todo junto en un instalable para Windows. Por ejemplo:
  - EasyPHP
    - <http://www.easypht.org/>
  - WampServer
    - <http://www.wampserver.com/>
  - Ambos instalan y configuran:
    - Apache + PHP + mySQL + mySQLAdmin



## → Seguridad en Servidores Web

- Antes de ver el protocolo de seguridad en Web, la terminología necesaria incluye:
  - **Codificación**... expresar en un código diferente
  - **Criptología**
    - Criptografía... hacer ilegible la información
    - Criptoanálisis... violar un sistema criptográfico
  - **Esteganografía**... ocultar información
  - Cifrar:
    - 1. tr. Transcribir en guarismos, letras o símbolos, de acuerdo con una clave, un mensaje cuyo contenido se quiere ocultar.
- En la Web, el protocolo de comunicaciones SSL se utiliza para cifrar la comunicación entre dos equipos y autenticar a los participantes de la misma.
  - SSL es complejo, y a continuación se describe superficialmente sus bases.

## → Criptografía simétrica (I)

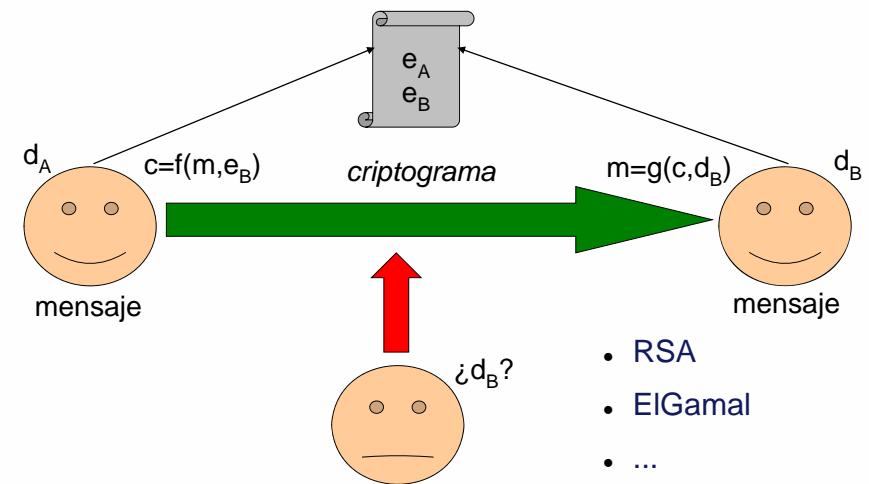


- DES, 3DES, AES, Twofish, IDEA...

## → Criptografía simétrica

- Algoritmos conocidos
- Una única para ambos extremos
- La clave debe ser intercambiada
- Ventajas
  - Proceso muy rápido
  - Requiere pocos recursos
- Inconvenientes
  - Gestión y seguridad de las claves

## → Criptografía asimétrica (I)



- RSA
- ElGamal
- ...

## → Criptografía asimétrica

- Algoritmos conocidos
- Dos claves complementarias por extremo
  - Clave pública conocida y disponible
  - Clave privada secreta
- Ventajas
  - Gestión de claves más sencilla
  - Gran seguridad
- Inconvenientes
  - Gran consumo de recursos



## → Sistemas híbridos - SSL

- Sistema SSL:
  - Generación de claves de sesión
  - Intercambio asimétrico de claves
  - Cifrado simétrico de sesión



## → SSL

- Siempre que se utilice SSL estamos cifrando las comunicaciones extremo a extremo, sin embargo el proceso de autenticación requiere de una configuración con más cuidado.
- Si se desea autenticar al servidor, es decir, que los clientes tengan la certeza de que se están comunicando con el servidor que ellos desean, es necesario utilizar un certificado emitido por una **Entidad Emisora de Certificados** contrastable por los usuarios de nuestro sistema, o lo que es lo mismo, una Entidad en la que los clientes confíen y tengan la clave pública de esta instalada en su máquina.
- Si esto no se produce, el uso de SSL ayuda a cifrar las comunicaciones pero no ayudará a detectar un certificado "falso" emitido por un atacante en medio.
  - Es por ello, aquí como muestra didáctica únicamente, se utiliza un certificado emitido por nosotros mismos, se recomienda utilizar un certificado de servidor emitido por una CA de confianza para nuestros usuarios.
- Con SSL, además de autenticar el servidor, se pueden autenticar a los clientes mediante certificados digitales a la hora de iniciar la conexión con SSL, aunque esta no es una práctica muy extendida debido a la complejidad en el despliegue y mantenimiento de los certificados de los clientes.



## →

- Creación de las claves y los certificados:

```
openssl genrsa -out hostname.key 1024
```

```
openssl req -new -key "hostname.key" -out "hostname.csr" -config "c:\Archivos de programa\Apache Software Foundation\Apache2.2\conf\openssl.cnf"
```

```
openssl x509 -req -days 365 -in "hostname.csr" -signkey "hostname.key" -out "hostname.crt"
```





```
C:\OpenSSL\bin>openssl.exe genrsa -out hostname.key 1024
Loading 'screen' into random state - done
Generating RSA private key. 1024 bit long modulus
.....+++++
e is 65537 <0x10001>
C:\OpenSSL\bin>openssl.exe req -new -key "hostname.key" -out "hostname.csr"
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
Country Name (2 letter code) [AU]:es
State or Province Name (full name) [Some-State]:Madrid
Locality Name (eg, city) []:Madrid
Organization Name (eg, company) [Internet Widgits Pty Ltd]:mi company
Organizational Unit Name (eg, section) []:informatica
Common Name (eg, YOUR name) []:Alberto
Email Address []:abian@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:alberto
An optional company name []:alberto

C:\OpenSSL\bin>openssl x509 -req -days 365 -in "hostname.csr" -signkey "hostname
.key" -out "hostname.crt"
Loading 'screen' into random state - done
Signature ok
subject=/C=es/ST=Madrid/L=Madrid/O=mi company/OU=informatica/CN=Alberto/emailAdd
ress=abian@gmail.com
Getting Private key
```



- Con el certificado digital emitido ya tenemos lo necesario para configurar el soporte SSL en nuestro servidor web. Lo primero que hemos de preparar es que el servidor Apache cargue el módulo SSL.
- Una vez cargado el módulo SSL en el servidor Apache, ahora deberemos configurar una serie de parámetros para dar soporte a SSL a nivel de servidor o de Virtual Host. Para ello se deben configurar las opciones en el archivo
  - SSL viene acompañado de muchas opciones y es recomendable, para ajustes especiales, consultar la documentación de mod\_ssl que está disponible en la siguiente URL:  
[http://httpd.apache.org/docs/2.2/en/mod/mod\\_ssl.html](http://httpd.apache.org/docs/2.2/en/mod/mod_ssl.html)



## → Opciones de SSL

- La siguiente opciones muestran las configuraciones que se deben realizar para que nuestro servidor esté funcionando correctamente:
  - **SSLEngine on/off:** Este parámetro activa el uso de SSL en nuestro servidor. Si la configuración del servicio fuera errónea el soporte no se activaría y puede llegar a no levantar los demonios de Apache, luego es importante tener correctamente configurado el servicio antes de ponerlo activo.
  - **SSLProtocol:** Este parámetro se utiliza para determinar cuáles van a ser los protocolos de cifrado que se van a utilizar en nuestro servidor. Hay que tener en cuenta que cuando se produce el “handshake” o saludo inicial entre el cliente y el servidor, estos negocian el protocolo a utilizar. Si no deseamos que se utilice un protocolo antiguo o inseguro debemos deshabilitar el uso de todos a excepción de los protocolos seguros. Esta acción puede producir problemas de acceso en clientes antiguos. La lista de protocolos que vienen con SSL son: SSLv2, SSLv3, TLSv1.



## → Opciones de SSL

- ...
- **SSLCipherSuite:** Una vez elegido el protocolo SSL a utilizar, en mod\_ssl podremos configurar las opciones de cifrado, para ello podemos elegir los algoritmos de generación de clave, decantándonos por el uso de RSA o de Diffie-Hellman con claves RSA o Diffie-Hellman con claves DSA, etc... Así mismo podremos elegir los algoritmos de firma, de codificación, y las longitudes de cifrado a usar. Es decir, podemos realizar un ajuste fino de la criptografía que nos va a permitir secupear las comunicaciones hasta nuestro deseo. Hay que tener en cuenta que el deshabilitar ciertas opciones de cifrado puede generar conflictos con clientes que no tengan una suite criptográfica amplia y moderna.
- **SSLOptions:** Este parámetro se va a utilizar para configurar diferentes comportamientos en diferentes situaciones. La opción +StrictRequire se va a utilizar para deshabilitar el acceso por medio http a aquellas rutas en las que se exija SSL



## → Opciones de SSL

- ...
- **SSLCertificateFile:** Ruta al archivo del certificado del sitio
  - **SSLCertificateKeyFile:** Ruta al archivo key del certificado.
  - **SSLCACertificateFile:** El certificado digital de la Entidad Certificadora.
  - **SSLCARevocationFile:** Archivo dónde se encuentra la CRL (Lista de Certificados Revocados).
  - **SSLRequire:** Este parámetro se utiliza para exigir un cumplimiento de opciones SSL a la hora de acceder a una determinada ruta del servidor. Se utilizan expresiones regulares para poder afinar las restricciones.
  - **SSLRequireSSL:** Parámetro para forzar el uso de http-s en un determinado directorio. Si está configurada la opción +StrictRequire se prohibirá el uso de http.



## → Ejemplo: Inclusión SSL en un servidor

- SSL necesita de servidores virtuales
  - (Se recomienda terminar la práctica de servidores virtuales antes de realizar esta otra)
- Crear un directorio en `seguro` dentro de `htdocs` para ponerlo en un servidor virtual
  - `...htdocs/vhost1/seguro/`



## → SSL: modificaciones `httpd.conf`

- Crear modificaciones en `httpd.conf`:

```
Listen 127.0.0.1:80
Listen 127.0.0.1:443
Listen 127.0.0.1:8080
Listen 192.168.1.55:80
Listen 192.168.1.55:8080

LoadModule ssl_module modules/mod_ssl.so

#Para comprobar el correcto funcionamiento
#de esta directiva hay que #poner
#http://127.0.0.1/seguro/index.html (NO FUNCIONA)
#https://127.0.0.1/seguro/index.html (SÍ FUNCIONA)
<Directory "htdocs/vhost1/seguro/">
SSLRequireSSL
</Directory>
```

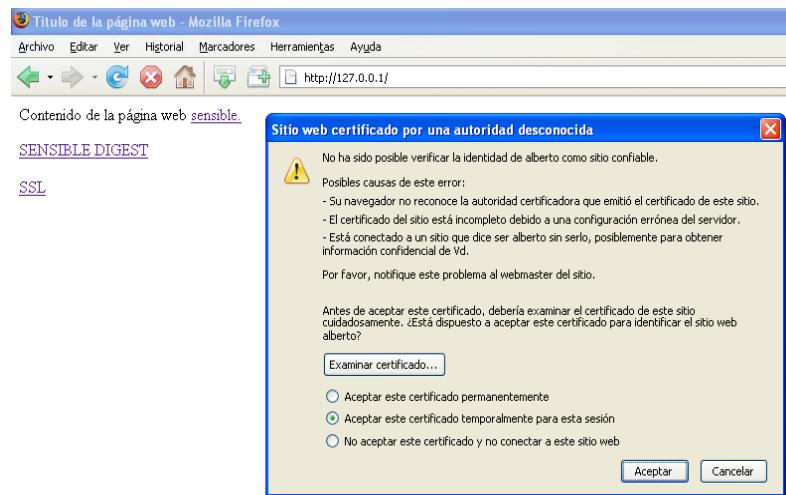


## → SSL: Modificaciones `httpd-vhosts.conf`

```
<VirtualHost 127.0.0.1:80>
ServerAdmin admin@127.0.0.1
DocumentRoot "htdocs/vhost1/"
ErrorLog "logs/127.0.0.1-error_log"
CustomLog "logs/127.0.0.1-access_log common"
Para que funcione .htaccess
<Directory "htdocs/vhost1/">
AllowOverride All
</Directory>
#Regla de reescritura para poder acceder a las páginas con SSL
Redirect /seguro/ https://127.0.0.1/seguro/
</VirtualHost>
<VirtualHost 127.0.0.1:443>
SSLEngine On
SSLCertificateFile "openssl/usuario.crt"
SSLCertificateKeyFile "openssl/usuario.key"
ServerName secure.example.org
DocumentRoot "htdocs/vhost1/"
</VirtualHost>
```

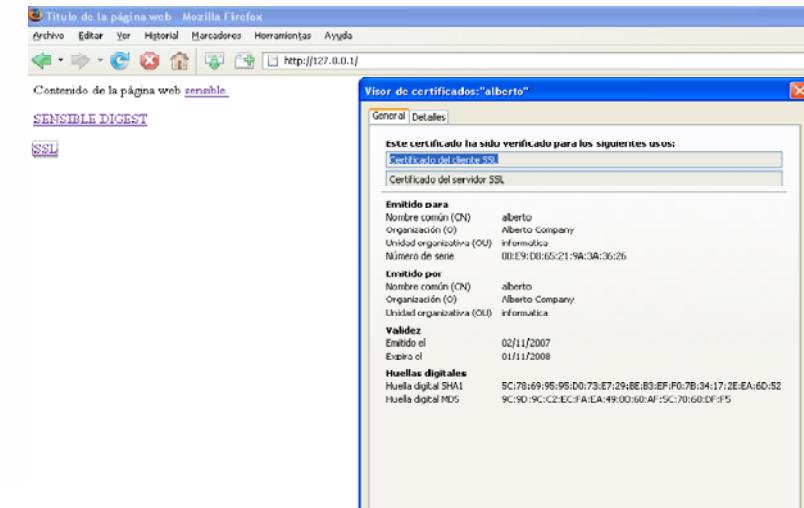


## → Ejemplo: Usando SSL en un servidor



85

## → Ejemplo: Usando SSL en un servidor



86

## → Internet Information Server

## → MS IIS (Internet Information Server)

- Todo lo visto se puede configurar para Microsoft IIS
- Para servidores profesionales, lo normal es utilizar Windows Server, que contiene
  - Windows 2003 Server
    - Servidor de archivos e impresión.
    - Servidor web y aplicaciones Web.
    - Servidor de correo.
    - Terminal Server
    - Servidor de acceso remoto/red privada virtual (VPN).
    - Servidor de directorio, Sistema de dominio (DNS), y servidor DHCP.
    - Servidor de transmisión de multimedia en tiempo real (Streaming).
    - Servidor de infraestructura para aplicaciones de negocios on-line



## → Familia Windows 2003 Server

- **Windows Server 2003 Standard Edition**
  - Soporta hasta 4 procesadores y la compartición de archivos e impresoras. Ofrece conectividad segura en Internet. Permite la implementación de aplicaciones centralizadas de escritorio.
- **Windows Server 2003 Enterprise Edition**
  - Es un sistema operativo completo de servidor que soporta hasta ocho procesadores. Ofrece funciones de tipo clase-empresarial tales como clustering de cuadro nodos y soporta hasta 32 GB de memoria. Disponible para ordenadores con Intel Itanium.
- **Windows Server 2003 Datacenter Edition**
  - Es el sistema operativo de servidor más potente y funcional que Microsoft haya ofrecido. Soporta hasta 32 vías SMP y 64 GB de RAM. Ofrece clustering de ocho nodos y servicios de balanceo de carga como funciones estándar. Plataformas de 64 bit capaz de soportar 32 procesadores y 128 GB de RAM.
- **Windows Server 2003 Web Edition**
  - Está diseñado para crear y alojar aplicaciones Web, páginas Web y servicios Web XML. Está diseñado para ser usado principalmente como un servidor Web IIS 6.0. Ofrece una plataforma rápida de desarrollo e implementación de servicios y aplicaciones Web XML que usan la tecnología ASP.NET, como parte principal del sistema .NET Framework.

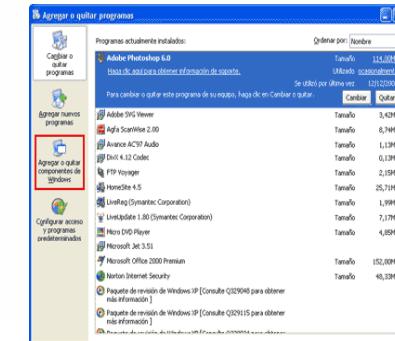
## → Instalación IIS

- En componentes:



## → Instalación IIS

- En el Panel de control, seleccionar la opción de "Agregar o quitar programas" y después "Seleccionar o quitar componentes de Windows".



## → Instalación IIS

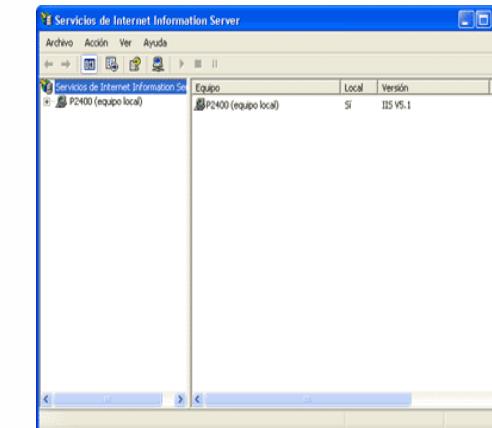
- Al igual que en Apache, se accede como:

– <http://localhost> ó <http://127.0.0.1/>



## → Administración de IIS

- Para administrar IIS en Windows XP, disponemos de un panel de control llamado "Servicios de Internet Information Server" bien:
  - Pulsando con el botón derecho en MI PC y seleccionando la opción que pone "Administrar". Esto nos abre "Microsoft Management Console" y desde ahí accedemos a "Servicios y aplicaciones", entre los que encontraremos: "Servicios de Internet Information Server"
  - Desde el panel de control.
  - Ejecutando en consola "inetmgr.exe".

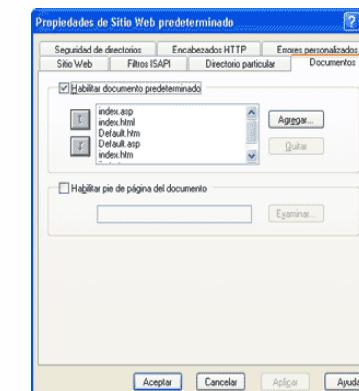


## → Documentos por defecto

- En IIS viene definido en un principio en los archivos default.asp, default.htm o index.htm
- Se puede cambiar desde las propiedades:

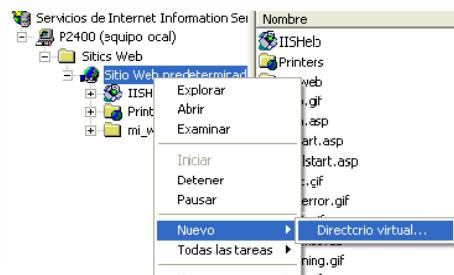


- Y desde propiedades:



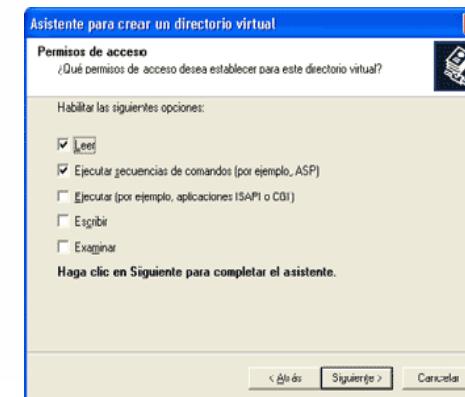
## → Directorios virtuales en IIS

- Para definir un directorio virtual se pulsar con el botón derecho del ratón sobre el sitio web en el que queremos definirlo y seleccionar "Nuevo > Directorio Virtual...".
  - Entonces aparece un asistente que nos guiará paso a paso en el proceso



## → Directorios virtuales en IIS

- Asistente:



## → Seguridad y certificados en IIS

- En el Administrador de IIS, expanda el equipo local y, después, expanda la carpeta Sitios Web.
- Haga clic con el botón secundario en el sitio Web para el que desea obtener un certificado de servidor comodín y, a continuación, haga clic en Propiedades.
- En la ficha Seguridad de directorios, en Comunicaciones seguras, haga clic en Certificado de servidor.
- En el Asistente para certificados de servidor Web, haga clic en Crear un certificado nuevo.
- Siga el Asistente para certificados de servidor Web, el cual le guiará a lo largo del proceso de petición de un nuevo certificado de servidor. En la página Nombre común de su sitio, escriba un nombre en el cuadro de diálogo Nombre común, con el siguiente formato:  
\*.<nombreDeSitio>, por ejemplo, \*.contoso.com.  
De forma predeterminada, el archivo de petición de certificado se guarda como C:\Certreq.txt, pero el asistente permite especificar una ubicación diferente.
- Haga clic en Finalizar para completar el asistente.

## → Seguridad y certificados en IIS

- En <http://support.microsoft.com/kb/299875/es> contiene la información sobre seguridad en IIS.
- Para instalación de un certificado:
  - Abra el Administrador de servicios Internet y expanda el nombre de servidor para poder ver los sitios Web.
  - Haga clic con el botón secundario en el sitio Web para el que creó la solicitud de certificado y haga clic en Propiedades.
  - Haga clic en la ficha Seguridad de directorios. En Comunicaciones seguras, haga clic en Certificado de servidor. Esto inicia el Asistente para la instalación de certificados. Haga clic en Siguiente para continuar en Siguiente.
  - Seleccione Procesar la petición pendiente e instalar el certificado y haga clic en Siguiente.
  - Tipo en Siguiente, sección entonces hace clic la ubicación del certificado de descargar en "el problema y descarga un certificado". El Asistente muestra el Resumen del certificado. Compruebe que la información es correcta entonces haga clic en Siguiente para continuar en Siguiente.
  - Haga clic en Finalizar

## → Seguridad y certificados en IIS

- Configure y pruebe el certificado

1. En la ficha Seguridad de directorios en Comunicaciones seguras, anote que hay ahora tres opciones disponibles. Para establecer el sitio Web para requerir conexión segura, haga clic en Modificar. El cuadro de diálogo Comunicaciones seguras aparece.
2. Seleccione Requerir canal seguro y haga clic en Aceptar.
3. Haga clic en Aplicar y a continuación en Aceptar para cerrar la hoja de propiedades.
4. Examine al sitio y compruebe que funciona. Para ello:
  - Tenga acceso al sitio a través de HTTP escribiendo `http://localhost/Postinfo.html` En el explorador. Aparecerá un mensaje de error similar al siguiente: HTTP 403.4 - Forbidden: SSL required.
  - Intente explorar a la misma página Web que utiliza unas conexiones seguras (HTTPS) escribiendo `https://localhost/postinfo.html` En el explorador. Puede recibir un alerta de seguridad que indica que el certificado no es que se sigue produciendo a la página Web desde una raíz de confianza Sí Click de CA. Si la página aparece, ha instalado su certificado correctamente.

