

PRÁCTICA 4.2

Administración de Apache parte I

Alexis Coves
Berna DAW
2ºW Grupo 2

Índice:

1. Instalación del servidor web Apache 2.4 en Linux	2
2. Instalación del servidor web Apache 2.2 en Windows	4
3. Ficheros de configuración y directivas en Linux	7
4. Ficheros de configuración y directivas en Windows	9
5. Configuración básica en Linux	12
6. Configuración básica en Windows	16
7. Módulos en Linux	18
8. Módulos en Windows	22
9. Control de acceso por IP y nombre de dominio	23
10. Autenticación y autorización Basic y Digest	24
11. Ficheros .htaccess	27
12. Ficheros de registros (logs)	29
13. Módulos <i>mod_status</i> y <i>mod_info</i>	30

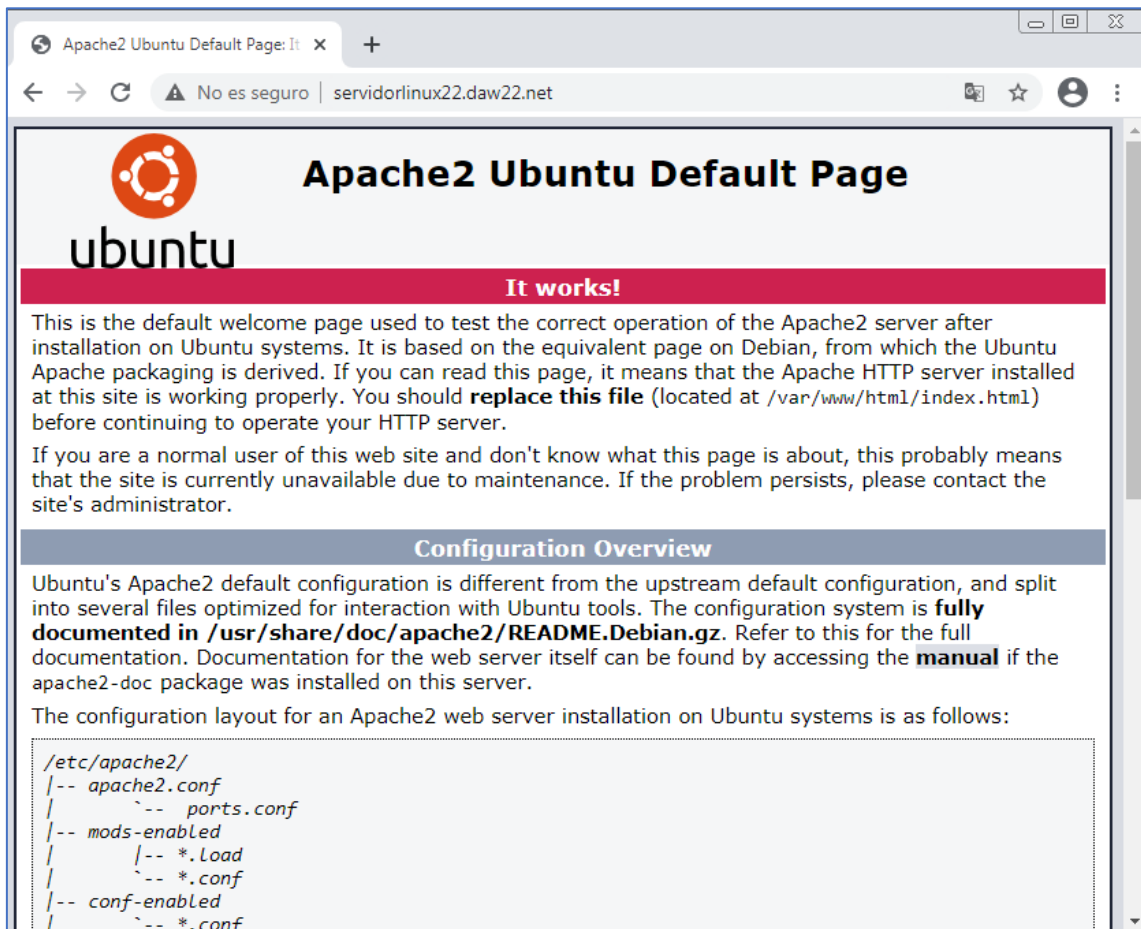
1. Instalación del servidor web Apache 2.4 en Linux:

Instalamos el servidor Apache mediante el comando `sudo apt-get install apache2`.

```

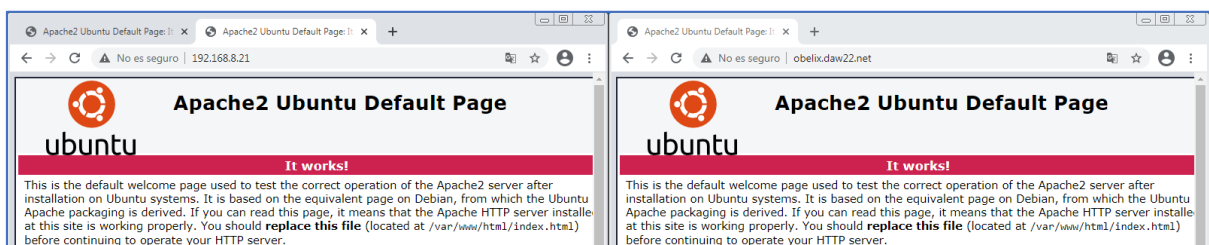
alumno@ServidorLinux21:~$ ps -ef | grep apache
root      1897      1  0 12:38 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  1900    1897  0 12:38 ?        00:00:00 /usr/sbin/apache2 -k start
www-data  1901    1897  0 12:38 ?        00:00:00 /usr/sbin/apache2 -k start
alumno    2026    988  0 12:46 tty1    00:00:00 grep --color=auto apache
alumno@ServidorLinux21:~$ netstat -ltn
Conexiones activas de Internet (solo servidores)
Proto Recib Envia Dirección local Dirección remota Estado
tcp6      0      0 :::80 :::* ESCUCHAR

```



Comprobamos que el servidor Apache funciona correctamente y nos muestra la página por defecto de Apache en Linux (Ubuntu).

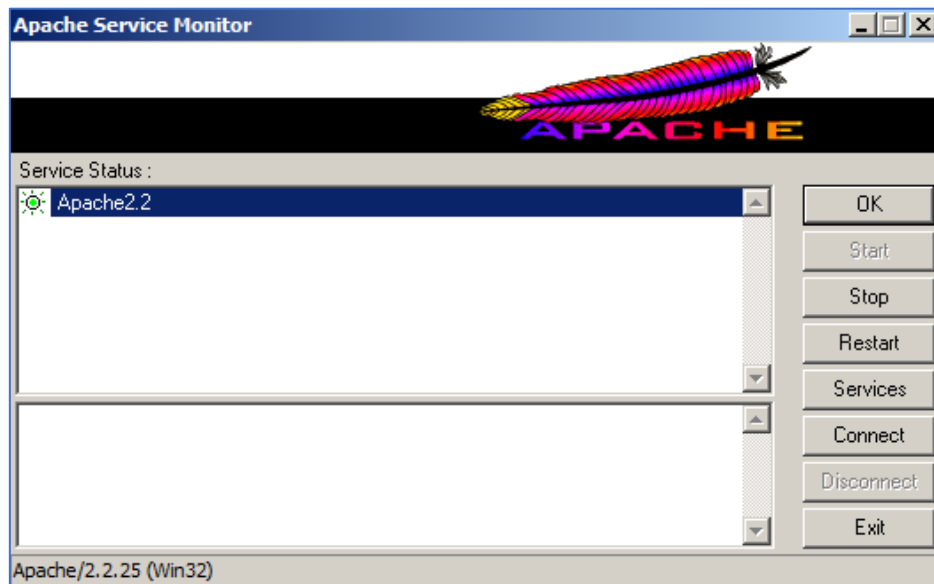
Observamos que podemos acceder desde los nombres DNS configurados en las prácticas anteriores "servidorlinux22.daw22.net", "obelix.daw22.net y a través de la dirección IP configurada en el servidor de Linux "192.168.8.21".



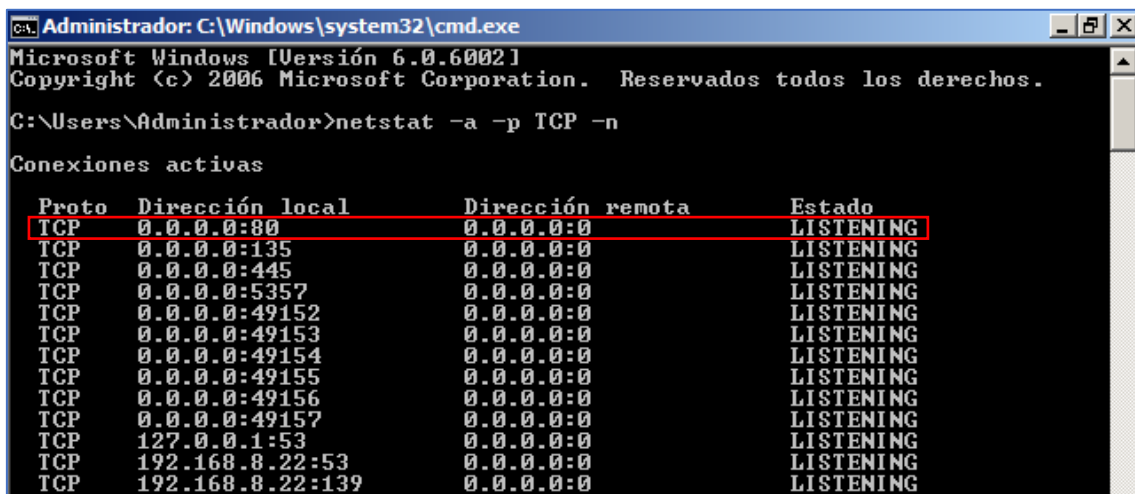
Acceso a apache desde la IP del servidor de Linux y desde el nombre de dominio "obelix.daw22.net".

2. Instalación del servidor web Apache 2.2 en Windows

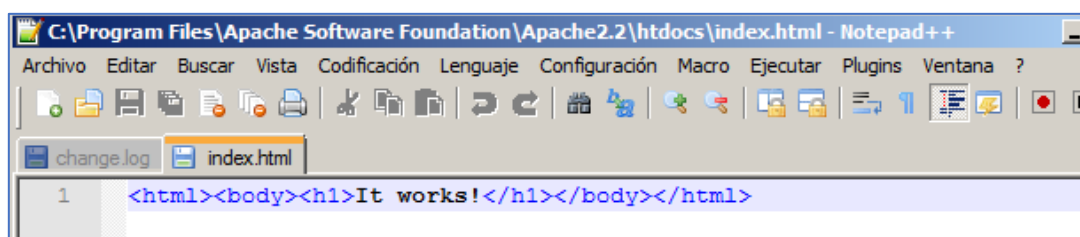
Instalamos la versión 2.2.25 de Apache en la máquina con Windows Server 2008.



Comprobamos que el servidor está iniciado.



Comprobamos que el servidor está a la escucha en el puerto 80 (http).

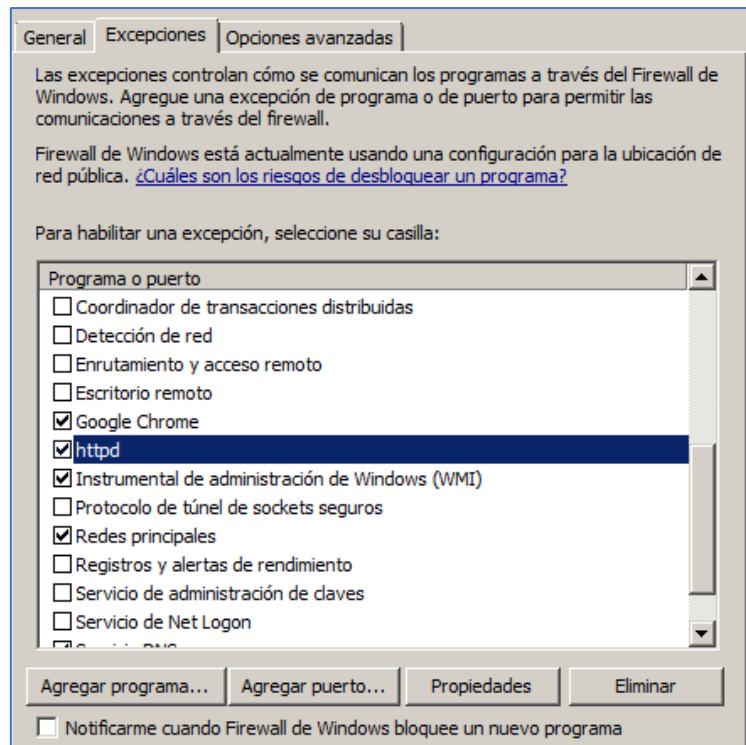


Comprobamos el contenido del fichero index.html en la ruta "C:\Program Files\Apache Software Foundation\Apache2.2\".

Como podemos observar, tanto las rutas como el contenido del fichero de la página por defecto de Apache, son distintas en Windows y Linux.

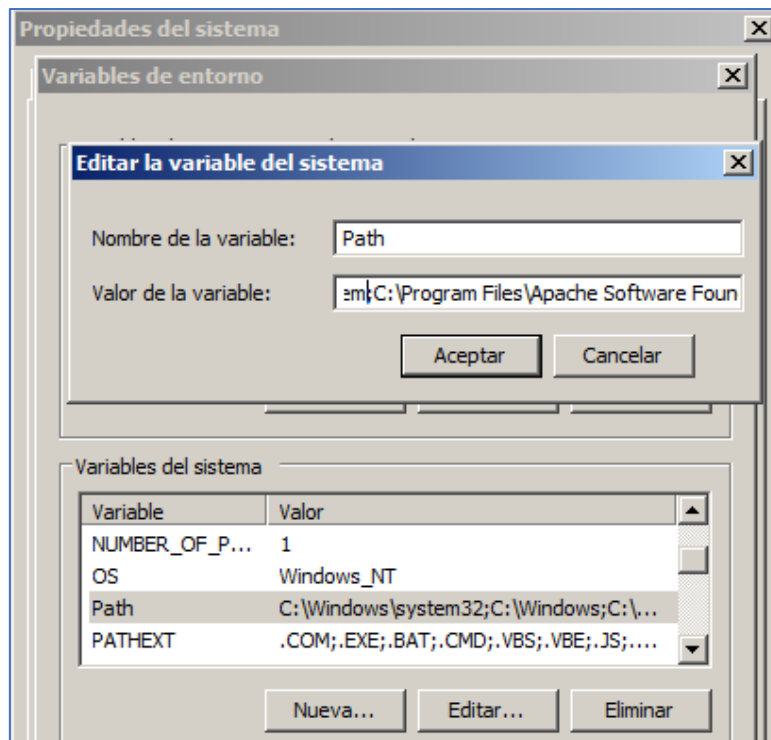
A continuación, configuraremos el *firewall* de Windows para que permita conexiones a Apache.

Para ello iremos a seguridad en el panel de control y en el *firewall* de Windows seleccionamos la opción "Deja pasar un programa a través de *firewall* de Windows", después pincharemos en agregar programa y seleccionaremos el ejecutable httpd.exe en la ruta "C:\Program Files\Apache Software Foundation\Apache2.2\bin\"

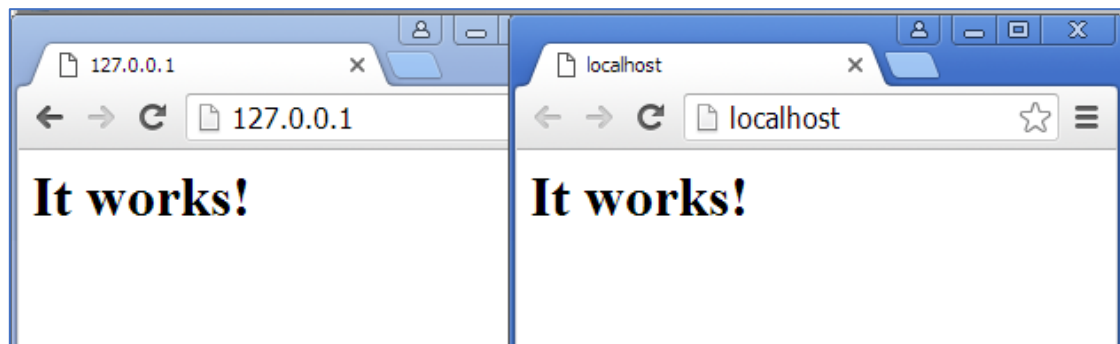


Comprobamos que se ha creado una excepción para el programa httpd.

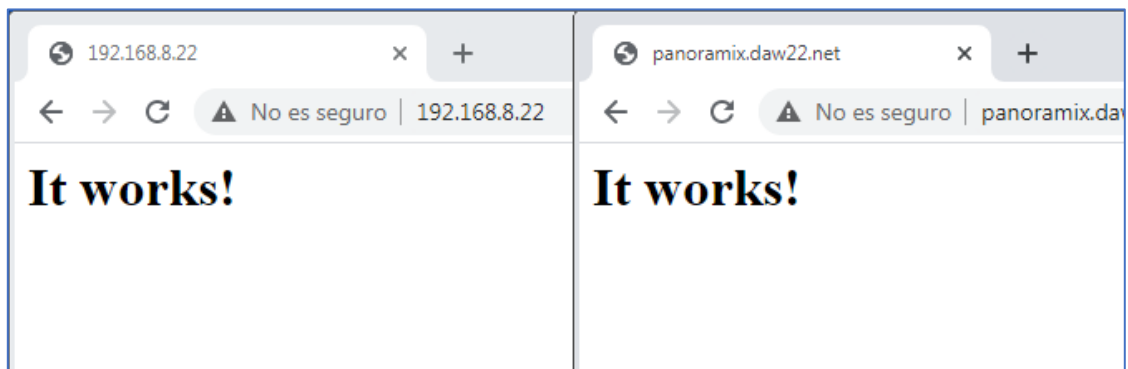
Para poder gestionar Apache desde la consola y desde cualquier directorio tendremos que añadirlo a la variable de entorno PATH. Para ello tendremos que ir al panel de control -> Sistema y mantenimiento -> Sistema o simplemente pulsando la tecla de Windows + Pausa. A continuación, pincharemos en configuración avanzada del sistema y seguidamente en variables de entorno, buscamos la variable de entorno Path y seleccionamos la opción "editar", al final añadimos la ruta "C:\Program Files\Apache Software Foundation\Apache2.2\bin".



Configuración de la variable de entorno "Path".



Comprobamos que tanto la dirección de loopback como localhost nos devuelve el index.html de Apache.



Comprobamos que funciona correctamente desde la máquina con Window 7.

Con esta última configuración finalizamos la instalación de Apache en los servidores de Linux y Windows.

3. Ficheros de configuración y directivas en Linux

Una vez instalado Apache nos centraremos en los principales ficheros de configuración y directivas del servidor web en el servidor de Linux, para ello primero analizaremos el contenido del fichero "apache2.conf" en la ruta "/etc/apache/". En este fichero es donde se configura la mayor parte del servidor Apache, aunque es recomendable separarlo en varios archivos más específicos para una mejor organización y simplicidad.

En este fichero podemos ver las configuraciones globales y los valores de las directivas:

- ServerRoot: "/etc/apache2".
- Timeout: 300 segundos.
- KeepAlive: On, con lo cual permite conexiones persistentes.
- User: \${APACHE_RUN_USER}.
- Group: \${APACHE_RUN_GROUP}.
- ErrorLog: \${APACHE_LOG_DIR}/error.log.

```
GNU nano 2.2.6      Archivo: /etc/apache2/apache2.conf

# a CustomLog directive.
#
# These deviate from the Common Log Format definitions in that they use %D
# (the actual bytes sent including headers) instead of %b (the size of the
# requested file), because the latter makes it impossible to detect partial
# requests.
#
# Note that the use of %{X-Forwarded-For}i instead of %h is not recommended.
# Use mod_remoteip instead.
#
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %D \"%{Referer}i\" \"%{User-Agent}i\" \"$"
LogFormat "%h %l %u %t \"%r\" %>s %D \"%{Referer}i\" \"%{User-Agent}i\" \"$" combin$
LogFormat "%h %l %u %t \"%r\" %>s %D" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

# Include of directories ignores editors' and dpkg's backup files,
# see README.Debian for details.

# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf

# Include the virtual host configurations:
IncludeOptional sites-enabled/*.conf
```

Observamos que permite incluir ficheros de configuración en los directorios "conf-enabled/" y "sites-enabled".


```
alumno@ServidorLinux21:/etc/apache2/sites-available$ ls -l
total 12
-rw-r--r-- 1 root root 1332 nov 26 2018 000-default.conf
-rw-r--r-- 1 root root 6437 nov 26 2018 default-ssl.conf
```

Comprobamos que existe el fichero "000-default.conf" en el directorio "sites-available".

```
alumno@ServidorLinux21:/etc/apache2/sites-enabled$ ls -l
total 0
lrwxrwxrwx 1 root root 35 ene 11 12:38 000-default.conf -> ../sites-available/000-default.conf
```

Comprobamos que existe el fichero "000-default.conf" en el directorio "sites-enabled".

```
GNU nano 2.2.6 Archivo: ports.conf

# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Comprobamos los puertos en los que escucha las peticiones en el fichero "ports.conf".

Observamos que escucha las peticiones en el puerto 80 (HTTP). Si habilitásemos los módulos SSL (Secure Sockets Layer) y TLS (Transport Layer Security), escucharía peticiones en el puerto 443 (HTTPS). Esto añadiría un nivel más de seguridad ya que el contenido estaría cifrado mediante OpenSSL.

```
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
```

Analizamos el fichero "000-default.conf" en el directorio "sites-available".

Observamos los valores de las directivas contenidas en VirtualHost:

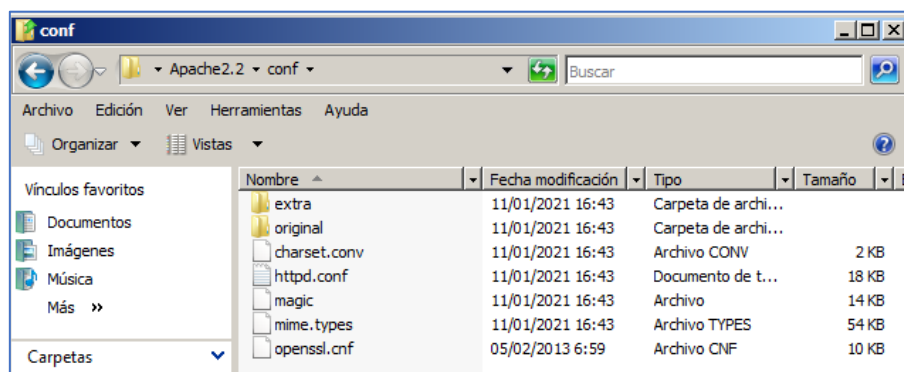
- ServerRoot: /var/www/html.
- ErrorLog: \${APACHE_LOG_DIR}/error.log.
- CustomLog: \${APACHE_LOG_DIR}/access.log.

```
<Directory /var/www/>
Options Indexes FollowSymLinks
AllowOverride None
Require all granted
</Directory>
```

Observamos la directiva <Directory>...</Directory> que determina cómo sirve el contenido del directorio "/var/www/".

4. Ficheros de configuración y directivas en Windows

Ficheros de configuración:



Contenido del directorio "Apache Software Foundation\Apache2.2\conf".

Analizamos el fichero "httpd.conf" y los valores de sus directivas:

- ServerRoot: "C:/Program Files/Apache Software Foundation/Apache2.2".
- Listen: puerto 80.
- DocumentRoot: "C:/Program Files/Apache Software Foundation/Apache2.2/htdocs".
- ErrorLog: "logs/error.log".

```

#
# This should be changed to whatever you set DocumentRoot to.
#
<Directory "C:/Program Files/Apache Software Foundation/Apache2.2/htdocs">
#
# Possible values for the options directive are "None", "All",
# or any combination of:
#   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI Multiviews
#
# Note that "Multiviews" must be named *explicitly* --- "options All"
# doesn't give it to you.
#
# The options directive is both complicated and important. Please see
# http://httpd.apache.org/docs/2.2/mod/core.html#options
# for more information.
#
Options Indexes FollowSymLinks

#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   Options FileInfo AuthConfig Limit
#
AllowOverride None

#
# Controls who can get stuff from this server.
#
Order allow,deny
Allow from all
</Directory>

#
# DirectoryIndex: sets the file that Apache will serve if a directory

```

Comprobamos el contenido de la directiva "Directory".

```

# Server-pool management (MPM specific)
#Include conf/extra/httpd-mpm.conf

# Multi-language error messages
#Include conf/extra/httpd-multilang-errordoc.conf

# Fancy directory listings
#Include conf/extra/httpd-autoindex.conf

# Language settings
#Include conf/extra/httpd-languages.conf

# User home directories
#Include conf/extra/httpd-userdir.conf

# Real-time info on requests and configuration
#Include conf/extra/httpd-info.conf

# Virtual hosts
#Include conf/extra/httpd-vhosts.conf

# Local access to the Apache HTTP Server Manual
#Include conf/extra/httpd-manual.conf

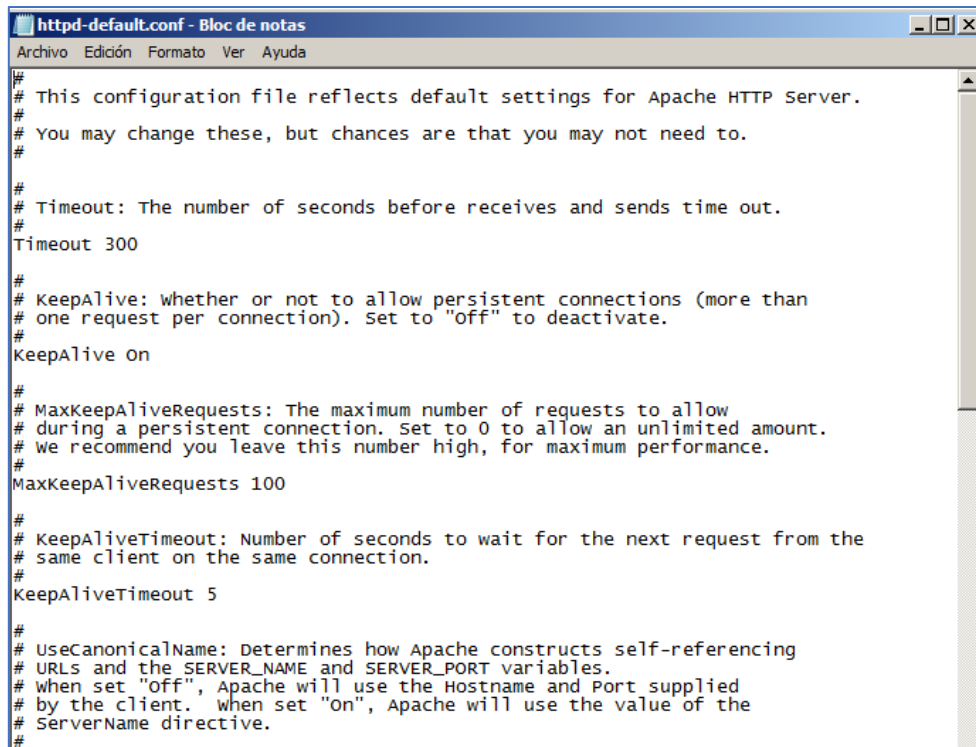
# Distributed authoring and versioning (WebDAV)
#Include conf/extra/httpd-dav.conf

# Various default settings
#Include conf/extra/httpd-default.conf

# Secure (SSL/TLS) connections
#Include conf/extra/httpd-ssl.conf
#
# Note: The following must be present to support
#       starting without SSL on platforms with no /dev/random equivalent
#       but a statically compiled-in mod_ssl.

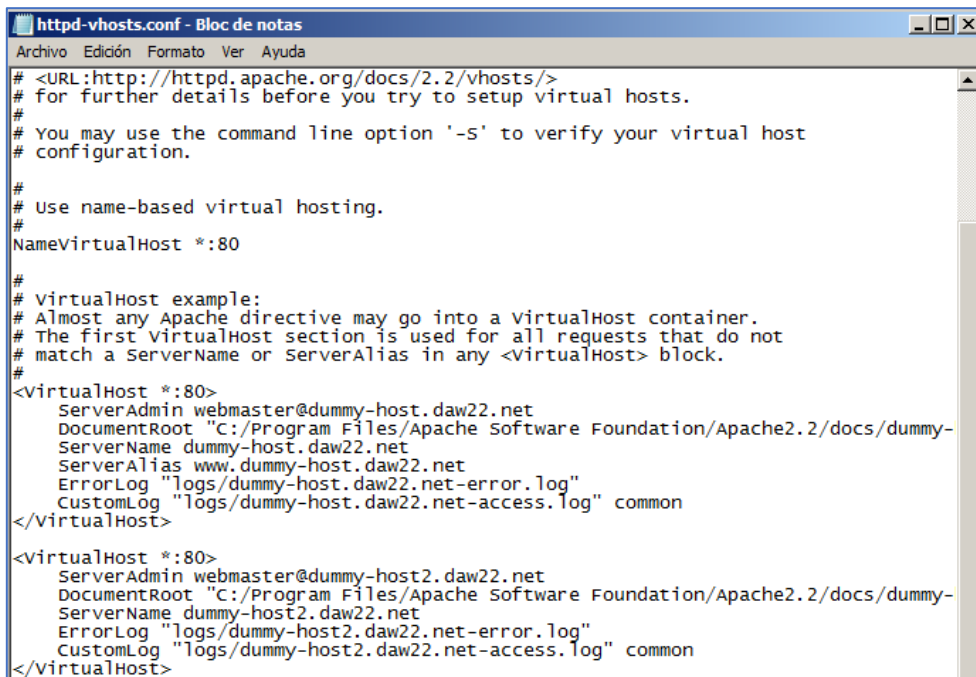
```

Observamos que existen varias directivas "Include" comentadas, estas añadirán nuevas funcionalidades.



```
# httpd-default.conf - Bloc de notas
# This configuration file reflects default settings for Apache HTTP Server.
# You may change these, but chances are that you may not need to.
#
# Timeout: The number of seconds before receives and sends time out.
#
Timeout 300
#
# KeepAlive: whether or not to allow persistent connections (more than
# one request per connection). Set to "off" to deactivate.
#
KeepAlive on
#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
#
MaxKeepAliveRequests 100
#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout 5
#
# UseCanonicalName: Determines how Apache constructs self-referencing
# URLs and the SERVER_NAME and SERVER_PORT variables.
# When set "off", Apache will use the Hostname and Port supplied
# by the client. When set "on", Apache will use the value of the
# ServerName directive.
#
```

Observamos los valores por defecto en el fichero "httpd-default".



```
# httpd-vhosts.conf - Bloc de notas
# <URL:http://httpd.apache.org/docs/2.2/vhosts/>
# for further details before you try to setup virtual hosts.
#
# You may use the command line option '-s' to verify your virtual host
# configuration.
#
# Use name-based virtual hosting.
#
NameVirtualHost *:80
#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
# The first VirtualHost section is used for all requests that do not
# match a ServerName or ServerAlias in any <VirtualHost> block.
#
<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host.daw22.net
    DocumentRoot "C:/Program Files/Apache Software Foundation/Apache2.2/docs/dummy-"
    ServerName dummy-host.daw22.net
    ServerAlias www.dummy-host.daw22.net
    ErrorLog "logs/dummy-host.daw22.net-error.log"
    CustomLog "logs/dummy-host.daw22.net-access.log" common
</VirtualHost>
<VirtualHost *:80>
    ServerAdmin webmaster@dummy-host2.daw22.net
    DocumentRoot "C:/Program Files/Apache Software Foundation/Apache2.2/docs/dummy-"
    ServerName dummy-host2.daw22.net
    ErrorLog "logs/dummy-host2.daw22.net-error.log"
    CustomLog "logs/dummy-host2.daw22.net-access.log" common
</VirtualHost>
```

Observamos el fichero "httpd-vhosts" desde el que podremos añadir servidores virtuales.

5. Configuración básica en Linux

Una vez comprobados los ficheros de configuración vamos a realizar algunas pruebas, para ello empezaremos creando los siguientes ficheros y directorios:

- `/var/www/html/despliegue.html`
- `/var/www/html/fp.html`
- `/var/www/html/ciclos/listado.html`
- `/var/www/html/ciclos/asir.html`
- `/var/www/html/ciclos/daw.html`
- `/var/www/html/ciclos/dam.html`

```
alumno@ServidorLinux21:/var/www/html$ ls -l
total 16
drwxr-xr-x 2 root root 4096 ene 26 20:34 ciclos
-rw-r--r-- 1 root root 0 ene 26 20:33 despliegue.html
-rw-r--r-- 1 root root 0 ene 26 20:33 fp.html
-rw-r--r-- 1 root root 11510 ene 11 12:38 index.html
alumno@ServidorLinux21:/var/www/html$ ls -l ciclos
total 0
-rw-r--r-- 1 root root 0 ene 26 20:34 asir.html
-rw-r--r-- 1 root root 0 ene 26 20:34 daw.html
-rw-r--r-- 1 root root 0 ene 26 20:34 listado.html
```

estructura de directorios y ficheros creada en la ruta `"/var/www/html/"`.

A continuación, buscamos en el navegador de la máquina con Windows 7, los archivos que acabamos de crear en el directorio raíz de Apache en Linux.

Comprobamos que funciona correctamente y nos abre una página en blanco ya que no tienen ningún contenido y que si no especificamos ningún recurso nos devuelve el `index.html` especificado en la directiva `"DirectoryIndex"`.

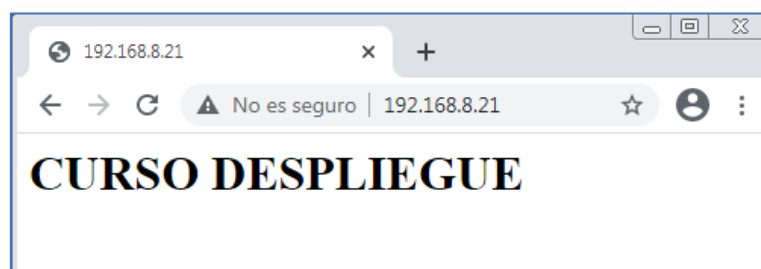
Seguidamente, cambiamos el nombre del fichero `"index.html"` por `"indice.html"` y volvemos y recargamos la página del navegador sin especificar el recurso y observamos que al no encontrar el fichero `index.html`, nos muestra el directorio raíz `"/var/www/html"`.



Ahora vamos a añadir la sección "<Directory /var/www/html>...</Directory>" en el fichero "000-default.conf" con la siguiente directiva:

```
<Directory /var/www/html>
    DirectoryIndex despliegue.html
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

De este modo cuando no pidamos ningún recurso en el navegador el servidor nos redirigirá a la página despliegue.html.



Comprobamos que la directiva funciona correctamente.

A continuación, probamos a abrir en el navegador el directorio "/var/www/html/ciclos"



Comprobamos que nos devuelve su contenido.

Esto es porque ha heredado la configuración de "/var/www/html" y no encuentra el fichero despliegue.html con lo cual nos muestra su contenido.

Posteriormente creamos una nueva directiva para "/var/www/html/ciclos".

```
<Directory /var/www/html/ciclos>
    Options FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

Puesto que no hemos añadido la opción "Indexes" y tampoco la opción "DirectoryIndex", el navegador nos devuelve un mensaje de prohibición.



Comprobamos que no tenemos acceso al directorio "/var/www/html/ciclos".



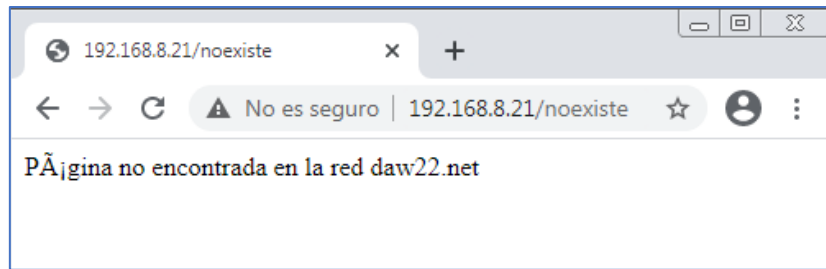
Comprobamos que tenemos acceso a ficheros dentro del directorio "/var/www/html/ciclos".

Códigos del error (ErrorDocument):

Ahora cambiaremos el mensaje de error 404 por defecto. Para ello editaremos el fichero "/etc/apache2/sites-available/000-default.conf".

```
<Directory /var/www/html/ciclos>
    Options FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

ErrorDocument 404 "Página no encontrada en la red daw22.net"
</VirtualHost>
```



Comprobamos que el navegador nos devuelve el mensaje de error que hemos especificado.

Directorios Virtuales (Alias):

A continuación, vamos a crear el directorio "home/alumno/apuntes" y dentro creamos el fichero "apuntes.html". Seguidamente editamos el fichero "/etc/apache2/sites-available/000-default.conf" y añadimos la directiva Alias para crear un directorio virtual que hará referencia al directorio "home/alumno/apuntes" seguido de la directiva directory con la configuración del directorio quedando de la siguiente manera:

```
Alias /apuntes /home/alumno/apuntes
<Directory /home/alumno/apuntes>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

ErrorDocument 404 /404.html
```

Redirrecciones (Redirect):

Vamos a editar de nuevo el fichero "etc/apache2/sites-available/000-default.conf" y vamos a añadir la directiva Redirect, de modo que nos redirija a la dirección "http://www.todofp.es".

```
Options Indexes FollowSymLinks
AllowOverride None
Require all granted
</Directory>

Redirect /fp http://www.todofp.es

ErrorDocument 404 /404.html
```




Comprobamos que la directiva Redirect funciona correctamente al acceder a
 "http://192.168.8.21/fp".

6. Configuración básica en Windows

Ficheros y directorios de prueba

Creamos la misma estructura de directorios y ficheros que en la configuración de Linux.

- ...\\Apache2.2\\htdocs\\despliegue.html
- ...\\Apache2.2\\htdocs\\ciclos\\listado.html
- ...\\Apache2.2\\htdocs\\ciclos\\asir.html
- ...\\Apache2.2\\htdocs\\ciclos\\daw.html
- ...\\Apache2.2\\htdocs\\ciclos\\dam.html
- C:\\Usuarios\\Administrador\\apuntes.html

A continuación, vamos a editar el fichero "C:\\Program Files\\Apache Software Foundation\\Apache2.2\\conf\\httpd.conf" y creamos las directivas necesarias.

```
<Directory "C:/Program Files/Apache Software Foundation/Apache2.2/htdocs/ciclos">
    Options FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

Alias /apuntes C:/Users/Administrador/apuntes
<Directory "C:/Users/Administrador/apuntes">
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

Redirect /fp http://www.todofp.es
```

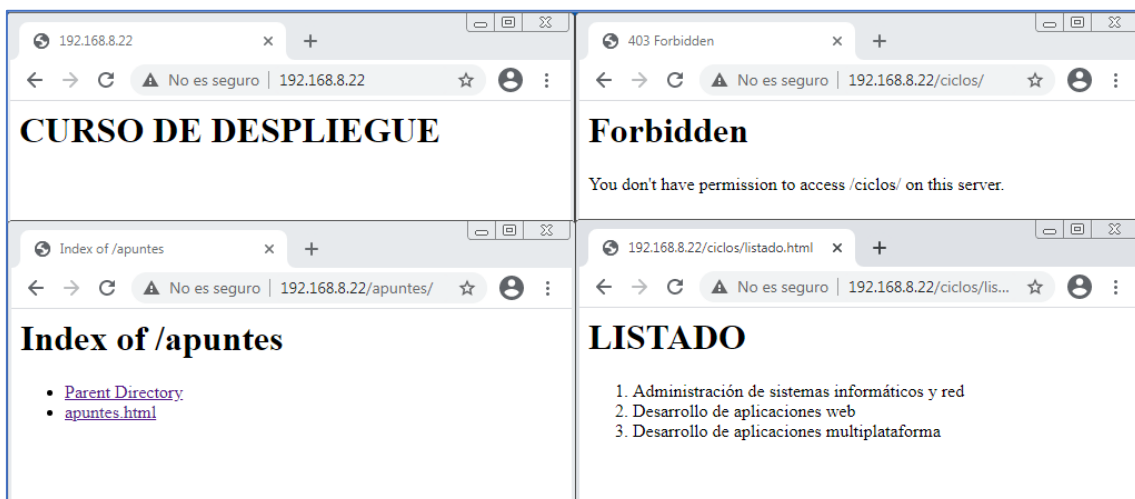
```
<Directory "C:/Program Files/Apache Software Foundation/Apache2.2/htdocs">
#
# Possible values for the options directive are "None", "All",
# or any combination of:
#   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
# The options directive is both complicated and important. Please see
# http://httpd.apache.org/docs/2.2/mod/core.html#options
# for more information.
#
DirectoryIndex despliegue.html
Options Indexes FollowSymLinks

#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   Options FileInfo AuthConfig Limit
#
AllowOverride None

#
# Controls who can get stuff from this server.
#
Order allow,deny
Allow from all

</Directory>
```

Añadimos también la directiva "DirectoryIndex" en la sección "<Directory \"C:/Program Files/Apache Software Foundation/Apache2.2/htdocs\">" con el fichero "despliegue.html".



Comprobamos que las directivas funcionan correctamente tras reiniciar el servidor Apache.

7. Módulos en Linux

Módulos:

```
Compiled in modules:
core.c
mod_so.c
mod_watchdog.c
http_core.c
mod_log_config.c
mod_logio.c
mod_version.c
mod_unixd.c
```

Comprobamos los módulos estáticos cargados al compilar el servidor mediante el comando "sudo apache2ctl -l".

```
alumno@ServidorLinux21:/etc/apache2/mods-enabled$ ls
access_compat.load  authz_host.load  dir.load          negotiation.conf
alias.conf          authz_user.load  env.load          negotiation.load
alias.load          autoindex.conf  filter.load       setenvif.conf
auth_basic.load     autoindex.load  mime.conf         setenvif.load
authn_core.load     deflate.conf     mime.load         status.conf
authn_file.load     deflate.load     mpm_event.conf    status.load
authn_core.load     dir.conf         mpm_event.load
```

Comprobamos los módulos que se han cargado dinámicamente al arrancar el servidor, en el directorio "/etc/apache2/mods-enabled/".

Estos módulos hacen referencia a los módulos contenidos en el directorio "/etc/apache2/mods-available" donde estarán todos los módulos disponibles en el servidor Apache.

```
authn_file.load      info.conf           reqtimeout.load
authn_socache.load  info.load           request.load
authnz_ldap.load    lbmethod_bybusyness.load rewrite.load
authz_core.load     lbmethod_byrequests.load sed.load
authz_dbd.load      lbmethod_bytraffic.load session_cookie.load
authz_dbm.load      lbmethod_heartbeat.load session_crypto.load
authz_groupfile.load ldap.conf           session_dbd.load
authz_host.load     ldap.load           session.load
authz_owner.load    log_debug.load      setenvif.conf
authz_user.load     log_forensic.load   setenvif.load
autoindex.conf      lua.load            slotmem_plain.load
autoindex.load      macro.load          slotmem_shm.load
buffer.load         mime.conf           socache_dbm.load
cache_disk.conf     mime.load           socache_memcache.load
cache_disk.load     mime_magic.conf     socache_shmcb.load
cache.load          mpm_event.conf      spelling.load
cache_socache.load  mpm_event.load      ssl.conf
cgid.conf           mpm_prefork.conf    ssl.load
cgid.load           mpm_prefork.load    status.conf
cgi.load            mpm_worker.conf     status.load
charset_lite.load   mpm_worker.load     substitute.load
data.load           negotiation.conf     suexec.load
dav_fs.conf         negotiation.load    unique_id.load
dav_fs.load         proxy_a.jp.load      userdir.conf
dav.load            proxy_balancer.conf  userdir.load
dav_lock.load       proxy_balancer.load  usertrack.load
dbd.load            proxy.conf           vhost_alias.load
deflate.conf        proxy_connect.load   xml2enc.load
deflate.load
```

Módulos disponibles en el directorio "/etc/apache2/mods-available".

```
GNU nano 2.2.6 Archivo: dir.load
LoadModule dir_module /usr/lib/apache2/modules/mod_dir.so
```

Analizamos el fichero “/etc/apache2/mods-available/dir.load”.

Comprobamos que carga el módulo desde la ruta “/usr/lib/apache2/modules/” en la cual está contenido el código del módulo en el fichero “mod_dir.so”.

```
GNU nano 2.2.6 Archivo: dir.conf
<IfModule mod_dir.c>
    DirectoryIndex index.html index.cgi index.pl index.php index.xhtml inde$
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Analizamos el fichero “/etc/apache2/mods-available/dir.conf”.

Observamos cómo se añaden directivas dentro de una declaración “<IfModule nombreModulo>...</IfModule>”. Estos módulos se ejecutarán sólo si se carga el módulo en la ejecución del servidor.

```
mod_auth_basic.so      mod_env.so            mod_proxy_wstunnel.so
mod_auth_digest.so     mod_expires.so        mod_ratelimit.so
mod_auth_form.so       mod_ext_filter.so     mod_reflector.so
mod_authn_anon.so      mod_file_cache.so     mod_remoteip.so
mod_authn_core.so      mod_filter.so         mod_reqtimeout.so
mod_authn_dbd.so       mod_headers.so        mod_request.so
mod_authn_dbm.so       mod_heartbeat.so      mod_rewrite.so
mod_authn_file.so      mod_heartmonitor.so   mod_sed.so
mod_authn_socache.so   mod_include.so        mod_session_cookie.so
mod_authnz_ldap.so     mod_info.so           mod_session_crypto.so
mod_authz_core.so      mod_lbmethod_bybusyness.so
mod_authz_dbd.so       mod_lbmethod_byrequests.so
mod_authz_dbm.so       mod_lbmethod_bytraffic.so
mod_authz_groupfile.so mod_lbmethod_heartbeat.so
mod_authz_host.so      mod_ldap.so           mod_session_dbd.so
mod_authz_owner.so     mod_log_debug.so      mod_session.so
mod_authz_user.so      mod_log_forensic.so   mod_setenvif.so
mod_autoindex.so       mod_macro.so          mod_slotmem_plain.so
mod_buffer.so          mod_mime_magic.so     mod_slotmem_shm.so
mod_cache_disk.so      mod_mime.so           mod_socache_dbm.so
mod_cache.so           mod_mpm_event.so      mod_socache_memcache.so
mod_cache_socache.so   mod_mpm_prefork.so    mod_socache_shmcb.so
mod_cgid.so            mod_mpm_worker.so     mod_speling.so
mod CGI.so             mod_negotiation.so    mod_ssl.so
mod_charset_lite.so    mod_proxy_ajp.so       mod_status.so
mod_data.so            mod_proxy_balancer.so  mod_substitute.so
mod_dav_fs.so          mod_proxy_connect.so   mod_suexec.so
mod_dav_lock.so        mod_proxy_express.so   mod_unique_id.so
mod_dav.so             mod_proxy_fcgi.so      mod_userdir.so
                      mod_proxy_http.so      mod_usertrack.so
                      mod_proxy_ftp.so       mod_vhost_alias.so
                      mod_proxy_html.so      mod_xml2enc.so
```

Consultamos el directorio “/usr/lib/apache2/modules/” y vemos todos los módulos disponibles para cargar.

```

libapache2-mod-xsendfile - Serve large static files efficiently from web applica
tions
libapache2-modsecurity - Dummy transitional package
libocamlnet-ocaml-dev - OCaml application-level Internet libraries - core develo
pment libraries
mapcache-cgi - tile caching server - CGI binary
mono-apache-server2 - ASP.NET 2.0 backend for mod_mono2 Apache module
mono-apache-server4 - ASP.NET 1.1 backend for mod_mono Apache module
mono-fastcgi-server2 - ASP.NET 2.0 backend for FastCGI web servers
mono-fastcgi-server4 - ASP.NET 4.0 backend for FastCGI web servers
php5-fpm - server-side, HTML-embedded scripting language (FPM-CGI binary)
libapache-mod-jk-doc - Documentación del paquete libapache2-mod-jk
libapache2-mod-apreq2 - generic Apache request library - Apache module
libapache2-mod-axis2c - Apache web services engine - apache module
libapache2-mod-fcgid-dbg - símbolos de depuración para mod_fcgid
libapache2-mod-log-sql - Use SQL to store/write your Apache queries logs - Base
libapache2-mod-log-sql-dbi - Use SQL to store/write your Apache queries logs - D
BI interface
libapache2-mod-log-sql-mysql - Use SQL to store/write your Apache queries logs -
MySQL interface
libapache2-mod-neko - Apache module for running server-side Neko programs
libapache2-mod-removeip - Module to remove IP from apache2's logs
libapache2-mod-scgi - Apache module implementing the SCGI protocol
libapache2-mod-webauth - Apache module for WebAuth authentication
libapache2-mod-webauthldap - Apache module for WebAuth LDAP lookup and authoriza
tion
libapache2-mod-webkdc - Apache modules for a WebAuth authentication KDC
libapache2-mod-wsgi-py3 - Python 3 WSGI adapter module for Apache
mod-musicindex-common - Archivos comunes para mod-musicindex

```

Consultamos los paquetes disponibles en los repositorios de Ubuntu que permiten instalar módulos adicionales en Apache mediante el comando "sudo apt-cache search libapache2-mod".

Módulo userdir:

```

alumno@ServidorLinux21:/etc/apache2/mods-enabled$ ls
access_compat.load  authz_host.load  dir.load          negotiation.conf
alias.conf          authz_user.load  env.load          negotiation.load
alias.load          autoindex.conf  filter.load       setenvif.conf
auth_basic.load     autoindex.load  mime.conf         setenvif.load
auth_core.load      deflate.conf     mime.load         status.conf
auth_file.load      deflate.load     mpm_event.conf   status.load
authz_core.load     dir.conf        mpm_event.load

```

Comprobamos que no tenemos habilitado el módulo "userdir".

```

alumno@ServidorLinux21:/etc/apache2/mods-enabled$ sudo a2enmod userdir
Enabling module userdir.
To activate the new configuration, you need to run:
service apache2 restart

```

Habilitamos el módulo "userdir" mediante el comando "sudo a2enmod userdir".

```

alumno@ServidorLinux21:/etc/apache2/mods-enabled$ ls
access_compat.load  authz_user.load  filter.load      setenvif.load
alias.conf          autoindex.conf  mime.conf        status.conf
alias.load          autoindex.load  mime.load        status.load
auth_basic.load     deflate.conf    mpm_event.conf  userdir.conf
auth_core.load      deflate.load     mpm_event.load  userdir.load
auth_file.load      dir.conf        negotiation.conf
auth_core.load      dir.load        negotiation.load
auth_host.load      env.load        setenvif.conf

```

Verificamos que se han creado los enlaces "userdir.conf" y "userdir.load" que referencian al módulo en el directorio "/etc/apache2/mods-available/".

```

GNU nano 2.2.6      Archivo: userdir.conf
<IfModule mod_userdir.c>
    UserDir public_html
    UserDir disabled root

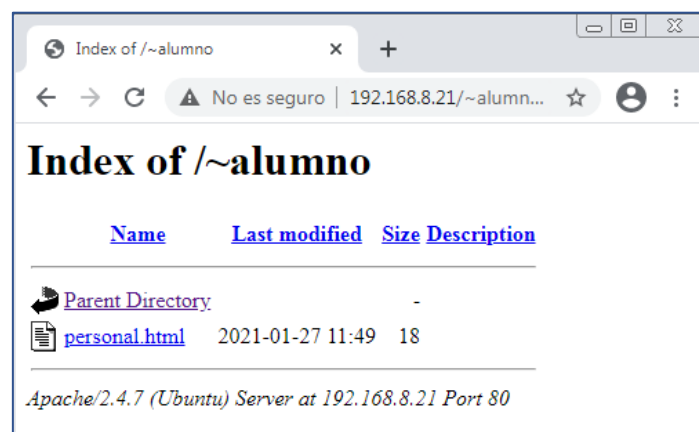
    <Directory /home/*/public_html>
        AllowOverride FileInfo AuthConfig Limit Indexes
        Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
        <Limit GET POST OPTIONS>
            Require all granted
        </Limit>
        <LimitExcept GET POST OPTIONS>
            Require all denied
        </LimitExcept>
    </Directory>
</IfModule>

```

Consultamos el fichero "/etc/apache2/mods-enabled/userdir.conf".

Observamos que está habilitado el uso de directorios personales para todos los usuarios excepto para el usuario "root" y que "public_html" es el nombre del subdirectorio en el que los usuarios podrán publicar sus páginas personales en su directorio "home".

A continuación, creamos el directorio "/home/alumno/public_html" y dentro, el fichero "personal.html"



Comprobamos que el módulo "userdir" funciona correctamente.

8. Módulos en Windows

Módulos:

```
C:\Users\Administrador>httpd -l
Compiled in modules:
  core.c
  mod_win32.c
  mpm_winnt.c
  http_core.c
  mod_so.c

C:\Users\Administrador>_
```

Comprobamos los módulos estáticos cargados con el arranque del servidor Apache mediante el comando "httpd -l".

```
LoadModule actions_module modules/mod_actions.so
LoadModule alias_module modules/mod_alias.so
LoadModule asis_module modules/mod_asis.so
LoadModule auth_basic_module modules/mod_auth_basic.so
#LoadModule auth_digest_module modules/mod_auth_digest.so
#LoadModule authn_alias_module modules/mod_authn_alias.so
#LoadModule authn_anon_module modules/mod_authn_anon.so
#LoadModule authn_dbd_module modules/mod_authn_dbd.so
#LoadModule authn_dbm_module modules/mod_authn_dbm.so
LoadModule authn_default_module modules/mod_authn_default.so
LoadModule authn_file_module modules/mod_authn_file.so
#LoadModule authnz_ldap_module modules/mod_authnz_ldap.so
#LoadModule authz_dbm_module modules/mod_authz_dbm.so
LoadModule authz_default_module modules/mod_authz_default.so
LoadModule authz_groupfile_module modules/mod_authz_groupfile.so
LoadModule authz_host_module modules/mod_authz_host.so
```

Consultamos los módulos cargados de forma dinámica al arrancar el servidor en el fichero "C:\Program Files\Apache Software Foundation\Apache2.2\conf\httpd.conf".

Módulo (userdir):

Eliminamos el comentario del módulo "userdir_module" y lo habilitamos.

```
#LoadModule status_module modules/mod_status.so
#LoadModule substitute_module modules/mod_substitute.so
#LoadModule unique_id_module modules/mod_unique_id.so
LoadModule userdir_module modules/mod_userdir.so
#LoadModule usertrack_module modules/mod_usertrack.so
#LoadModule version_module modules/mod_version.so
```

```
# User home directories
#Include conf/extra/httpd-userdir.conf
Include conf/extra/httpd-userdir.conf
```

Añadimos la directiva "Include" para que habilite el módulo.

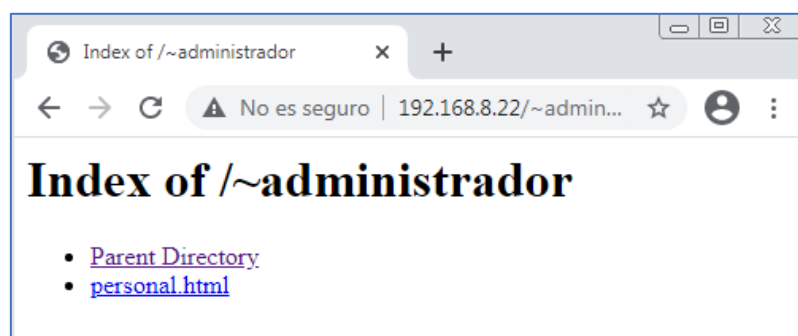

```
# Settings for user home directories
#
# Required module: mod_userdir
#
# UserDir: The name of the directory that is appended onto a user's home
# directory if a ~user request is received. Note that you must also set
# the default access control for these directories, as in the example below.
#
UserDir "Documents/Website"
#
# Control access to UserDir directories. The following is an example
# for a site where these directories are restricted to read-only.
#
<Directory "C:/Users/*/Documents/Website">
    AllowOverride FileInfo AuthConfig Limit Indexes
    Options Multiviews Indexes SymLinksIfOwnerMatch IncludesNoExec
    <Limit GET POST OPTIONS>
        order allow,deny
        Allow from all
    </Limit>
    <LimitExcept GET POST OPTIONS>
        order deny,allow
        Deny from all
    </LimitExcept>
</Directory>
```

Consultamos el fichero "C:\Program Files\Apache Software Foundation\Apache2.2\conf\extra\httpd-userdir.conf".

Editamos el nombre del directorio personal en el que los usuarios podrán publicar sus páginas personales por el nombre "Documents/Website" y reiniciamos el servidor para aplicar los cambios.

A continuación, creamos el fichero

"C:\Usuarios\Administrador\Websites\personal.html".



Comprobamos que el módulo "userdir" se ha habilitado correctamente.

9. Control de acceso por IP y nombre de dominio

En este apartado vamos a controlar el acceso al servidor mediante direcciones IP.

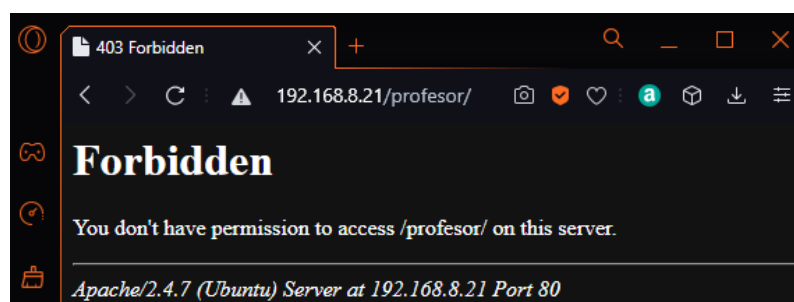
Para ello creamos el directorio y fichero "/var/www/html/profesor/profesor.html" y añadimos la directiva correspondiente en el fichero "/etc/apache2/sites-available/000-default.conf":

```
<Directory /var/www/html/profesor>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require ip 127.0.0.1
    Require ip 192.168.8.20
</Directory>
```

Utilizamos la directiva "Require ip" para que sólo se pueda acceder de el equipo local y el equipo con la dirección "192.168.8.20" asociada a Windows 7.



Comprobamos que la máquina de Windows 7 tiene acceso al directorio “profesor”.



Comprobamos que desde la máquina real no tenemos acceso al directorio “profesor”.

10. Autenticación y autorización Basic y Digest

Autenticación HTTP Basic:

```
alumno@ServidorLinux21:/etc/apache2/mods-enabled$ ls
access_compat.load  authz_host.load  dir.load          negotiation.conf
alias.conf          authz_user.load  env.load          negotiation.load
alias.load          autoindex.conf  filter.load       setenvif.conf
auth_basic.load     autoindex.load  mime.conf         setenvif.load
auth_core.load      deflate.conf     mime.load         status.conf
auth_file.load      deflate.load     mpm_event.conf   status.load
authz_core.load     dir.conf        mpm_event.load
```

Comprobamos que está habilitado el módulo “auth_basic.load”.

A continuación, crearemos un fichero en el que se guardarán los usuarios y sus contraseñas. Para ello instalaremos el paquete “apache2-utils”.

```
alumno@ServidorLinux21:/etc/apache2/mods-enabled$ sudo htpasswd -c /etc/apache2/
passwd profesor1
New password:
Re-type new password:
Adding password for user profesor1
```

Creamos el fichero y el usuario “profesor1” mediante el comando “sudo htpasswd -c /etc/apache2/passwd profesor1”.

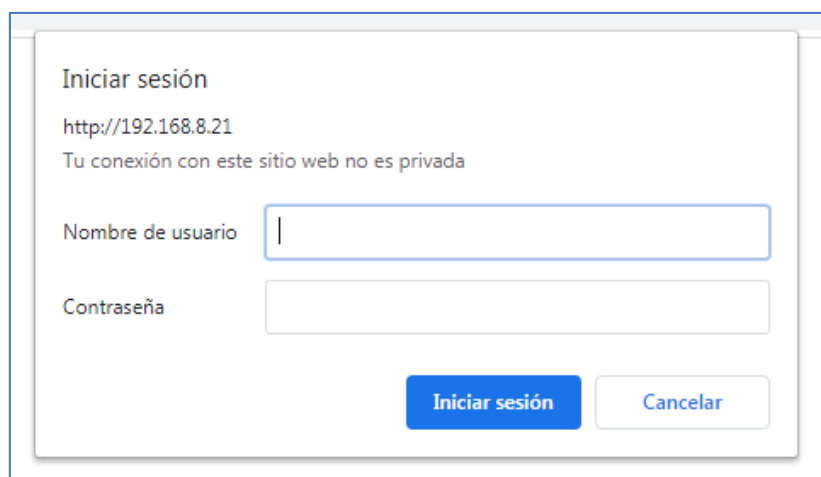
```
alumno@ServidorLinux21:/etc/apache2/mods-enabled$ sudo htpasswd /etc/apache2/passwd profesor2
New password:
Re-type new password:
Adding password for user profesor2
```

Creamos el usuario "profesor2" mediante el comando "sudo htpasswd /etc/apache2/passwd profesor2".

```
<Directory /var/www/html/profesor>
    Options Indexes FollowSymLinks
    AllowOverride None
    AuthType Basic
    AuthName "Acceso restringido"
    AuthUserFile /etc/apache2/passwd
    <RequireALL>
        Require user profesor1 profesor2
    <RequireAny>
        Require ip 127.0.0.1
        Require ip 192.168.8.20
    </RequireAny>
</RequireALL>
</Directory>
```

Editamos el fichero "/etc/apache2/sites-available/000-default.conf" con las directivas necesarias.

De este modo, a los usuarios "profesor1" y "profesor2" se les requerirá su usuario y contraseña para poder acceder al directorio "/var/html/profesor/".



Una vez reiniciado el servidor comprobamos que desde la máquina de Windows 7 nos pide usuario y contraseña para acceder al directorio.



Comprobamos que tenemos acceso tanto con el usuario "profesor1" como "profesor2".

Autenticación HTTP Digest:

```
alumno@ServidorLinux21:/etc/apache2$ sudo a2enmod auth_digest
Considering dependency authn_core for auth_digest:
Module authn_core already enabled
Enabling module auth_digest.
To activate the new configuration, you need to run:
service apache2 restart
```

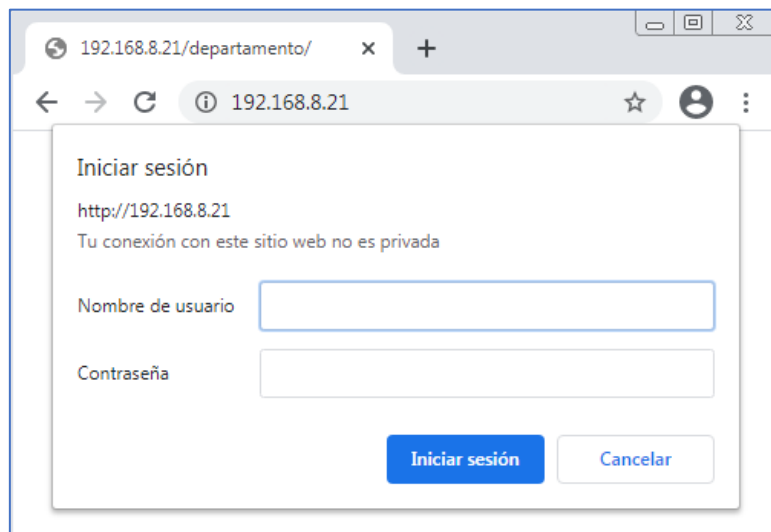
Habilitamos el módulo "auth_digest" mediante el comando "sudo a2enmod auth_digest".

```
alumno@ServidorLinux21:/etc/apache2$ sudo htdigest -c /etc/apache2/digest informatica admin1
Adding password for admin1 in realm informatica.
New password:
Re-type new password:
alumno@ServidorLinux21:/etc/apache2$ sudo htdigest /etc/apache2/digest informatica admin2
Adding user admin2 in realm informatica
New password:
Re-type new password:
```

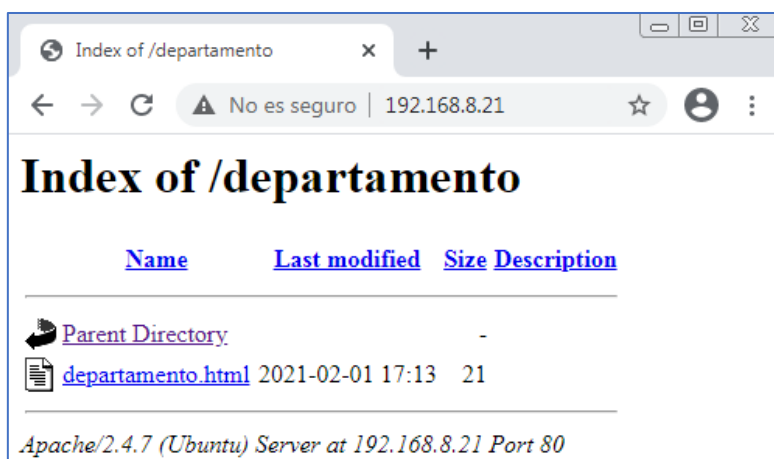
Creamos el directorio y los usuarios "admin1" y "admin2" mediante el comando "htdigest".

```
<Directory /var/www/html/departamento>
    Options Indexes FollowSymLinks
    AllowOverride None
    AuthType Digest
    AuthName "informatica"
    AuthDigestProvider file
    AuthUserFile /etc/apache2/digest
    Require user admin1 admin2
</Directory>
```

Editamos el fichero "/etc/apache2/sites-available/000-default.conf" con las directivas necesarias.



Comprobamos que nos pide usuario y contraseña para el directorio "departamento".



Comprobamos que tenemos acceso tanto con el usuario "admin1" como "admin2".

11. Ficheros .htaccess

A continuación vamos a habilitar el uso de ficheros de configuración personalizada de directorios (.htaccess). Para ello, en primer lugar crearemos el usuario profesor en la Máquina de Linux y tras cambiar la sesión al usuario profesor, creamos el directorio y fichero `"/home/profesor/blog/blog.html"`. seguidamente editamos el fichero `"/etc/apache2/sites-available/000.default.conf"` y añadimos la siguiente directiva:

```
Alias /blog /home/profesor/blog
<Directory /home/profesor/blog>
    AllowOverride All
</Directory>
```

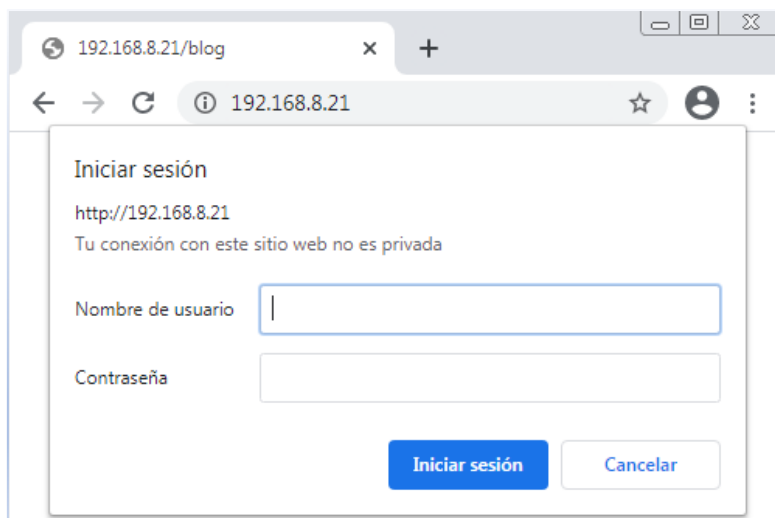
Una vez reiniciado el servidor y aplicado los cambios, creamos el fichero `"/home/profesor/blog/.htdigest"` y añadimos al usuario `blog`.

```
alumno@ServidorLinux21:/etc/apache2$ sudo htdigest -c /home/profesor/blog/.htdigest informatica blog
Adding password for blog in realm informatica.
New password:
Re-type new password:
```

Creamos el fichero `"/home/profesor/blog/.htdigest"` y añadimos el usuario `"blog"` mediante el comando `"htdigest /home/profesor/blog/.htdigest informatica blog"`.

A continuación, creamos el fichero `"/home/profesor/blog/.htaccess"` y añadimos las directivas necesarias.

```
Options Indexes
AuthType Digest
AuthName "informatica"
AuthDigestProvider File
AuthUserFile /home/profesor/blog/.htdigest
Require user blog_
```



Comprobamos que nos pide usuario y contraseña para el directorio `"blog"`.



Comprobamos que tenemos acceso tanto con el usuario "blog".

12. Ficheros de registros (logs)

```
# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn
```

```
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
```

Comprobamos cuales son los ficheros de logs de errores y accesos en el fichero
"/etc/apache2/sites-available/000-default.conf".

```
GNU nano 2.2.6 Archivo: /var/log/apache2/error.log
[Mon Jan 11 12:38:33.340928 2021] [mpm_event:notice] [pid 1897:tid 140458322745$
[Mon Jan 11 12:38:33.341051 2021] [core:notice] [pid 1897:tid 140458322745216] $
[Mon Jan 11 13:14:42.616572 2021] [mpm_event:notice] [pid 1897:tid 140458322745$
[Mon Jan 11 16:18:43.000813 2021] [mpm_event:notice] [pid 868:tid 1403630129007$
[Mon Jan 11 16:18:43.002301 2021] [core:notice] [pid 868:tid 140363012900736] A$
[Mon Jan 11 18:33:33.172999 2021] [mpm_event:notice] [pid 868:tid 1403630129007$
[Wed Jan 13 12:02:08.523847 2021] [mpm_event:notice] [pid 933:tid 1399927083477$
```

Consultamos el log de errores.

```
GNU nano 2.2.6 Archivo: /var/log/apache2/access.log
192.168.8.20 - - [11/Jan/2021:16:23:51 +0100] "GET / HTTP/1.1" 200 3594 "-" "Mo$
192.168.8.20 - - [11/Jan/2021:16:23:51 +0100] "GET /icons/ubuntu-logo.png HTTP/$
192.168.8.20 - - [11/Jan/2021:16:23:51 +0100] "GET /favicon.ico HTTP/1.1" 404 5$
192.168.8.20 - - [11/Jan/2021:17:53:48 +0100] "GET / HTTP/1.1" 200 3594 "-" "Mo$
192.168.8.20 - - [11/Jan/2021:17:53:48 +0100] "GET /icons/ubuntu-logo.png HTTP/$
192.168.8.20 - - [11/Jan/2021:17:53:48 +0100] "GET /favicon.ico HTTP/1.1" 404 5$
192.168.8.20 - - [13/Jan/2021:12:21:15 +0100] "GET / HTTP/1.1" 200 3594 "-" "Mo$
192.168.8.20 - - [13/Jan/2021:12:21:16 +0100] "GET /favicon.ico HTTP/1.1" 404 5$
```

Consultamos el log de accesos.

13. Módulos *mod_status* y *mod_info*

En este apartado vamos a probar los módulos "mod_status" y "mod_info".

Para ello, en primer lugar habilitaremos el módulo "status" mediante el comando "sudo a2enmod status".

A continuación, editamos el fichero de configuración del módulo con las directivas necesarias para tener acceso desde la máquina de Windows 7 y reiniciamos el servidor.

```
<Location /server-status>
    SetHandler server-status
    Require local
    Require ip 192.168.8.20
    #Require ip 192.0.2.0/24
</Location>
```

Apache Server Status for 192.168.8.21 (via 192.168.8.21)

Server Version: Apache/2.4.7 (Ubuntu)
 Server MPM: event
 Server Built: Apr 3 2019 18:04:25

Current Time: Monday, 01-Feb-2021 19:40:59 CET
 Restart Time: Monday, 01-Feb-2021 19:40:33 CET
 Parent Server Config. Generation: 1
 Parent Server MPM Generation: 0
 Server uptime: 26 seconds
 Server load: 0.00 0.01 0.05
 Total accesses: 0 - Total Traffic: 0 kB
 CPU Usage: u0 s0 cu0 cs0
 0 requests/sec - 0 B/second -
 1 requests currently being processed, 49 idle workers

PID	Connections		Threads		Async connections		
	total	accepting	busy	idle	writing	keep-alive	closing
2643	0	yes	0	25	0	0	0
2644	0	yes	1	24	0	0	0
Sum	0		1	49	0	0	0

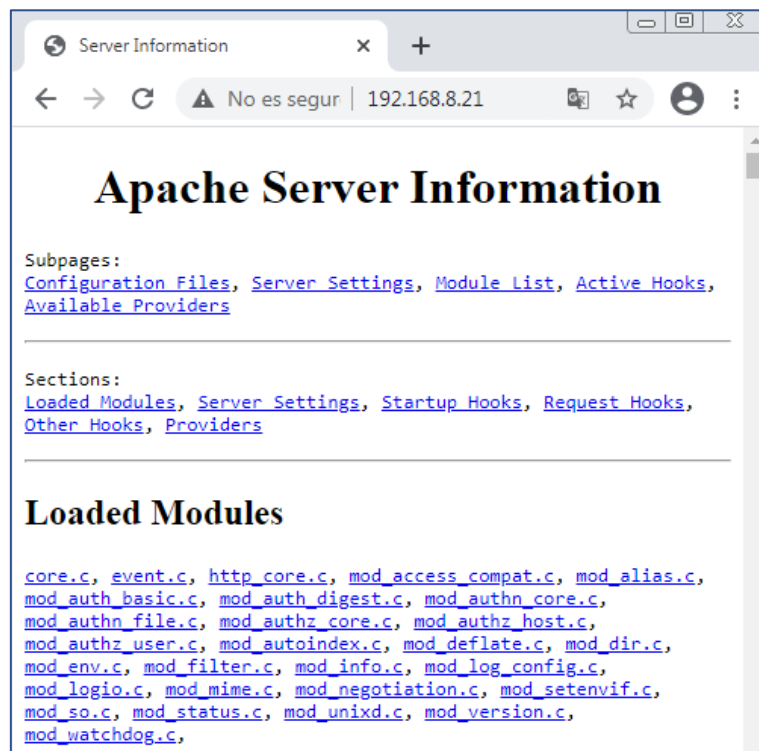
Comprobamos que tenemos acceso al módulo y podemos consultar el estado del servidor.

```
Current Time: Monday, 01-Feb-2021 19:43:33 CET
Restart Time: Monday, 01-Feb-2021 19:40:33 CET
Parent Server Config. Generation: 1
Parent Server MPM Generation: 0
Server uptime: 3 minutes
Server load: 0.00 0.01 0.05
Total accesses: 1 - Total Traffic: 2 kB
CPU Usage: u.01 s0 cu0 cs0 - .00556% CPU load
.00556 requests/sec - 11 B/second - 2048 B/request
2 requests currently being processed, 48 idle workers
```

PID	Connections		Threads		Async connections		
	total	accepting	busy	idle	writing	keep-alive	closing
2643	0	yes	1	24	0	0	0
2644	1	yes	1	24	0	0	0
Sum	1		2	48	0	0	0

Comprobamos que también podemos consultar el estado mediante el comando "sudo apache2ctl status".

Repetimos los mismos pasos para el módulo "info".



Comprobamos que tenemos acceso desde la máquina de Windows 7.