Práctica 4.2

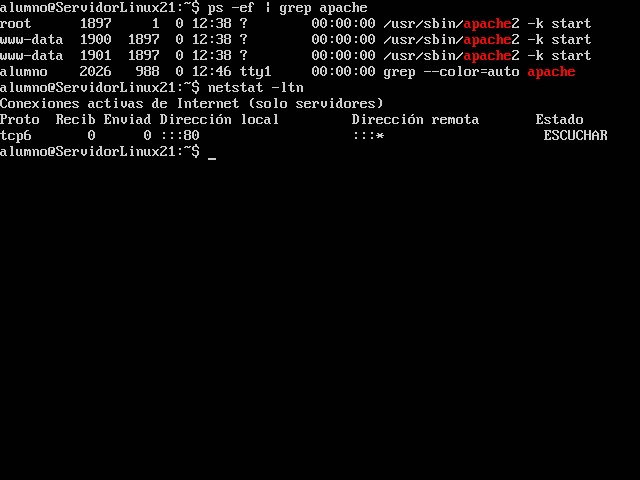
Administración de Apache parte I

Alexis Coves Berna DAW 2ºW Grupo 2

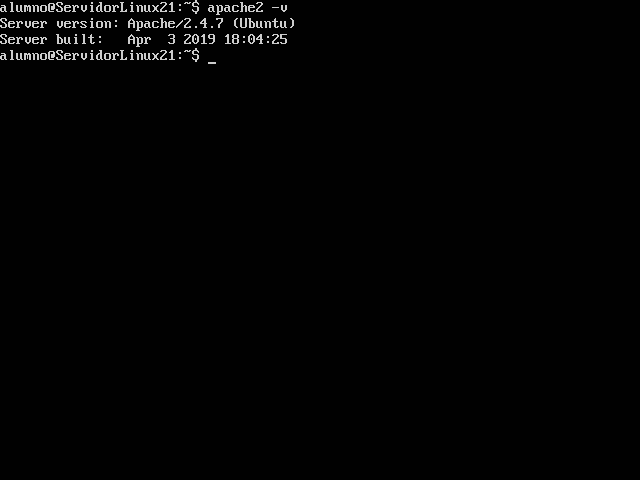
**Índice:**

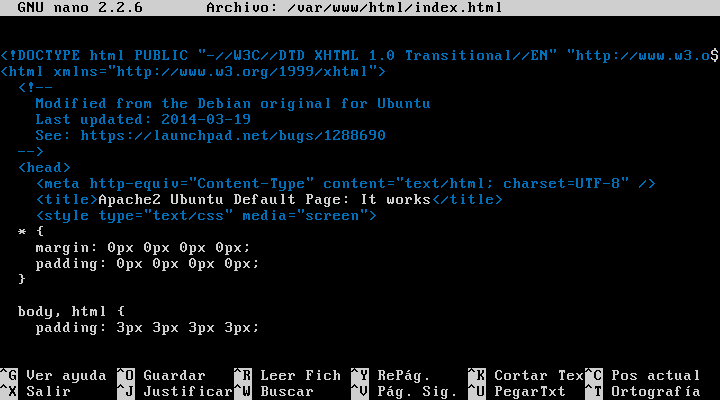
1. [Instalación del servidor web Apache 2.4 en Linux](#InstApacheLinux) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
2. [Instalación del servidor web Apache 2.2 en Windows](#InstApacheWindows) . . . . . . . . . . . . . . . . . . . . . . . . . . 4
3. [Ficheros de configuración y directivas en Linux](#ficherosConfig) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
4. [Ficheros de configuración y directivas en Windows](#ficherosConfigWindows)  . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
5. [Configuración básica en Linux](#confBasicaLinux) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
6. [Configuración básica en Windows](#confBasicaWindows) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16
7. [Módulos en Linux](#modulosLinux) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18
8. [Módulos en Windows](#modulosWindows) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22
9. [Control de acceso por IP y nombre de dominio](#controlAccesoIP) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
10. [Autenticación y autorización *Basic* y *Digest*](#BasicDigest). . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24
11. [Ficheros .htaccess](#htAccess) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
12. [Ficheros de registros (*logs*)](#registros) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
13. [Módulos *mod\_status* y *mod\_info*](#statusInfo) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
14. **Instalación del servidor web Apache 2.4 en Linux:**

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

Comprobamos que el servidor está iniciado y escuchando en el puerto 80/TCP mediante los comandos “ps -ef | grep apache” y “netstat -ltn”.

Comprobamos que se ha creado el directorio “/var/www/html” y que su propietario y grupo es root.

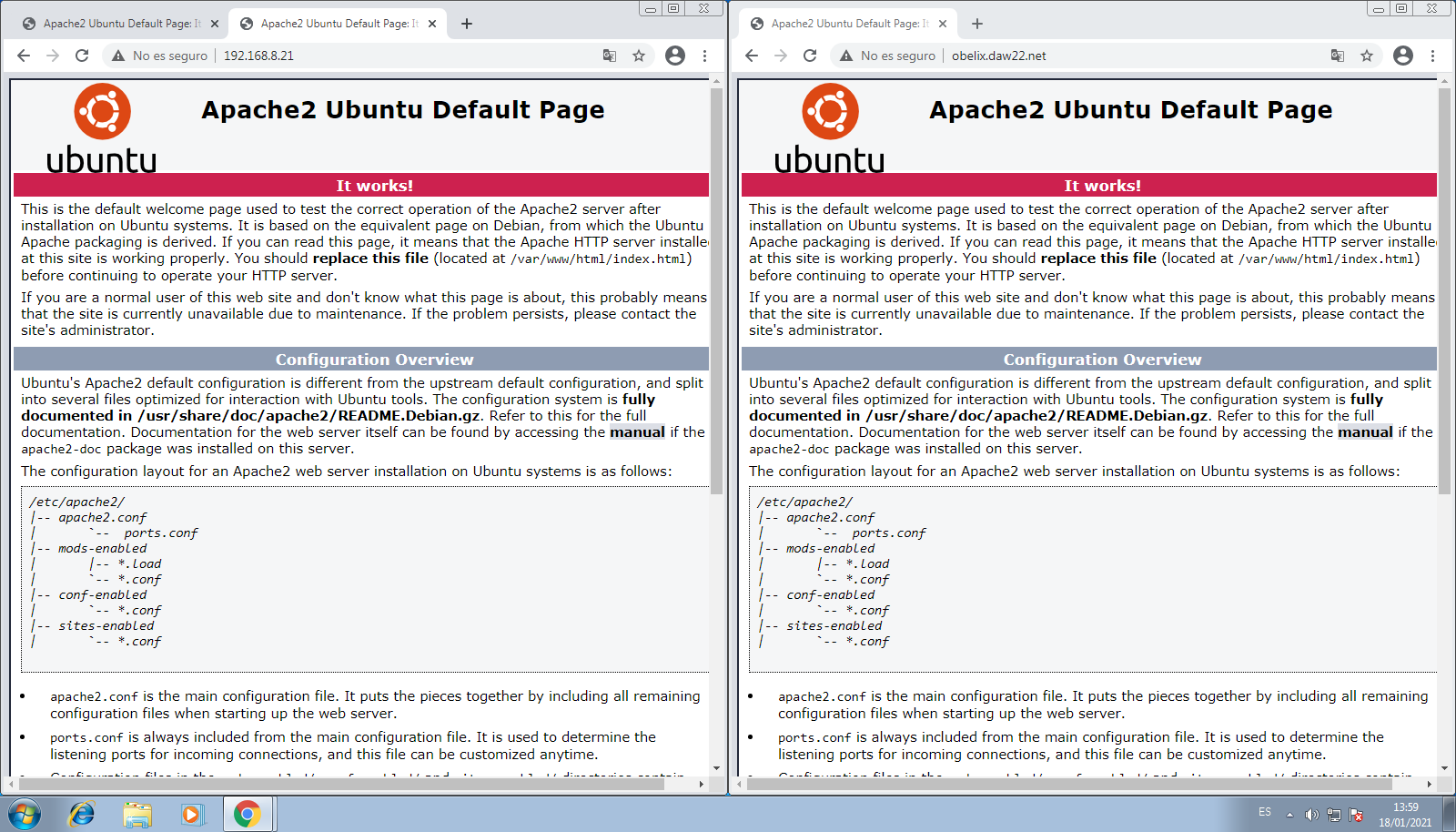
Comprobamos que la versión de Apache que se ha instalado es la 2.4.7.

Comprobamos el contenido del fichero “index.html”.

Este fichero es el que se cargará por defecto cuando accedamos al servidor por la web.

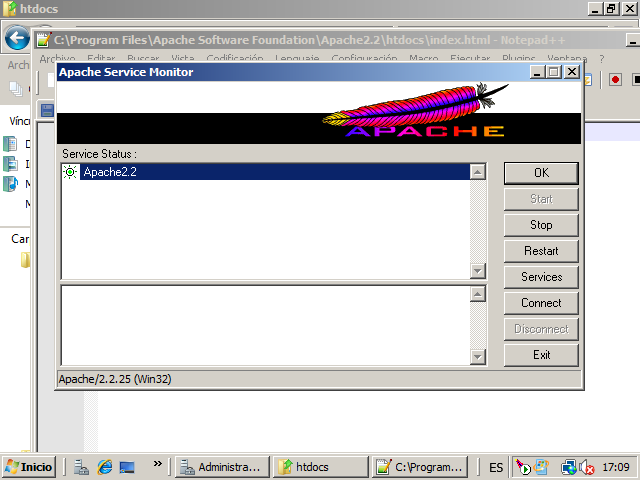
Comprobamos que el servidor Apache funciona correctamente y nos muestra la página por defecto de Apache en Linux (Ubuntu).

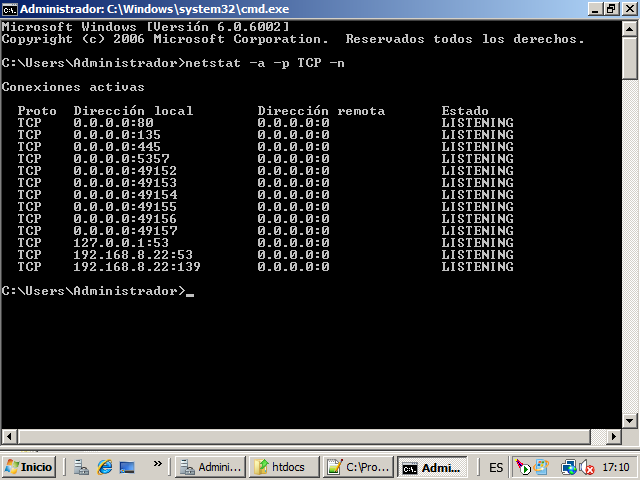
Observamos que podemos acceder desde los nombres DNS configurados en las prácticas anteriores “servidorlinux22.daw22.net”, “obelix.daw22.net y a través de la dirección IP configurada en el servidor de Linux “192.168.8.21".

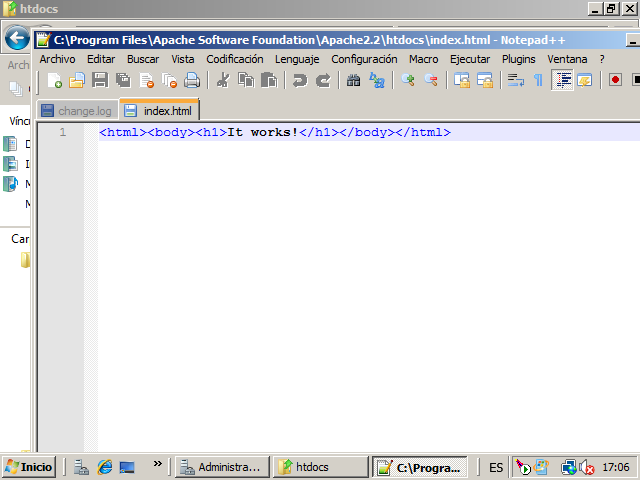
Acceso a apache desde la IP del servidor de Linux y desde el nombre de dominio “obelix.daw22.net”.

1. **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.

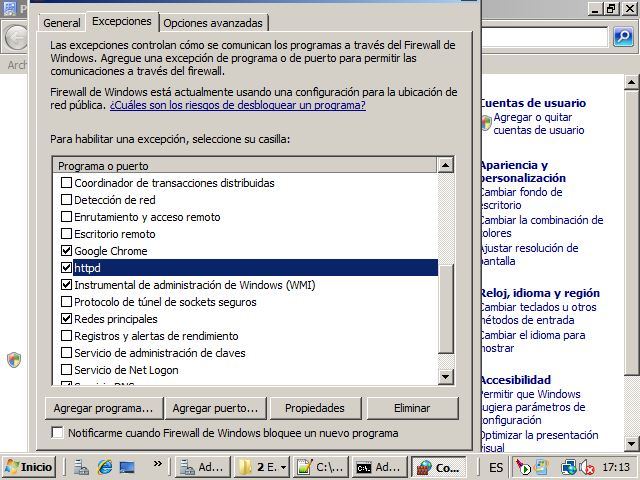
Comprobamos que el servidor está a la escucha en el puerto 80 (http).

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

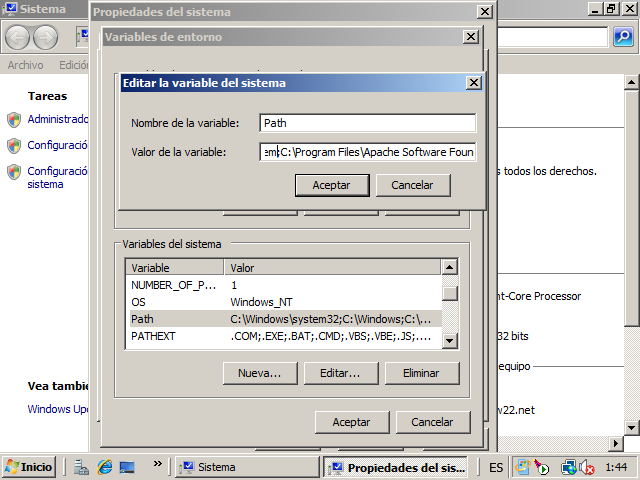
Como podemos observar, tanto las rutas como el contenido del fichero de la página por defecto de Apache, son distintas en Windows y Linux.

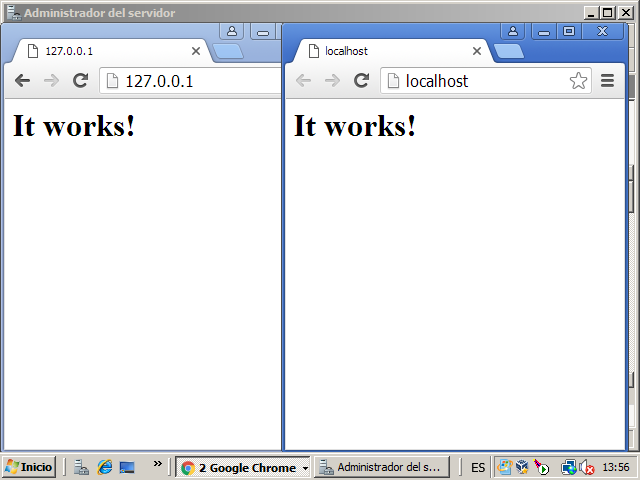
A continuación, configuraremos el *firewall* de Windows para que permita conexiones a Apache.

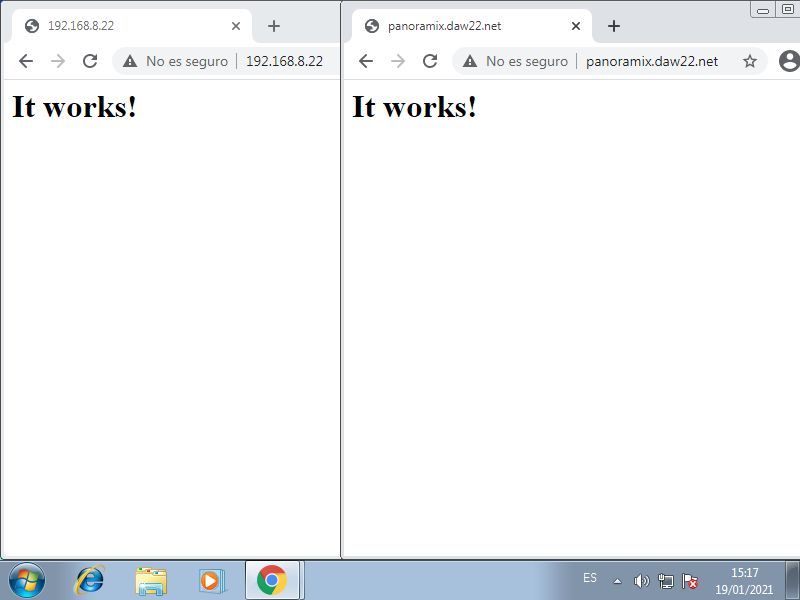
Para ello iremos a seguridad en el panel de control y en el *firewall* de Windows seleccionamos la opción “Deja pasar un programa a través de *firewall* de Windows”, después pincharemos en agregar programa y seleccionaremos el ejecutable httpd.exe en la ruta “C:\Program Files\Apache Software Foundation\Apache2.2\bin\

 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.

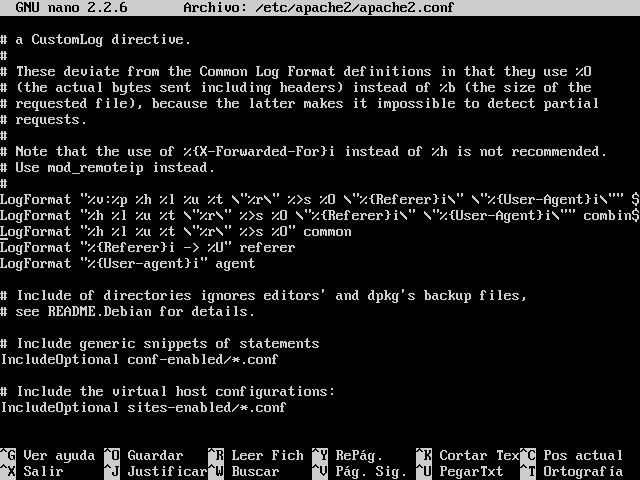
Con esta última configuración finalizamos la instalación de Apache en los servidores de Linux y Windows.

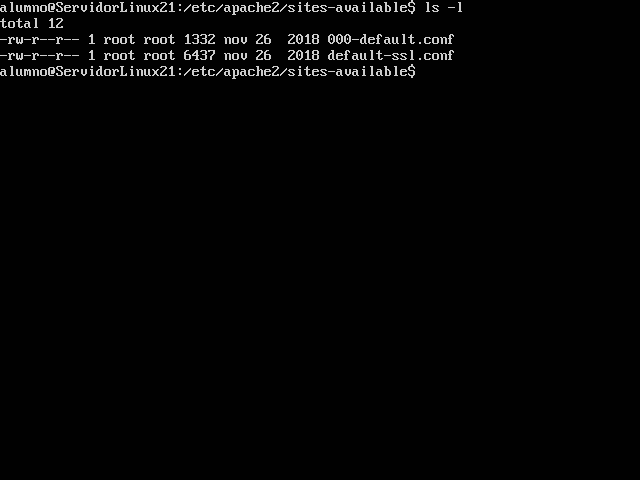
1. **Ficheros de configuración y directivas en Linux**

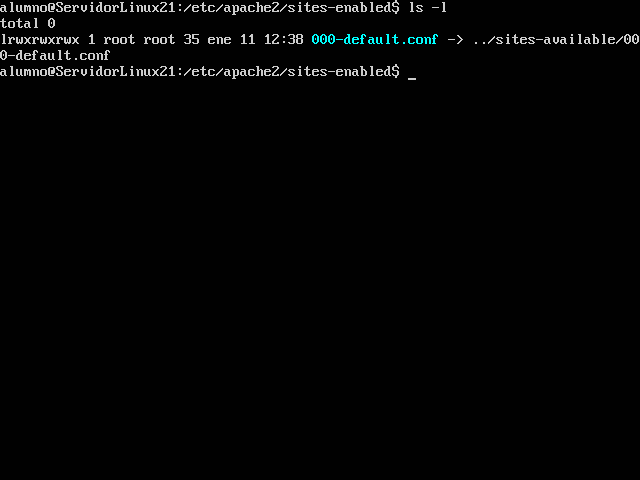
Una vez instalado Apache nos centraremos en los principales ficheros de configuración y directivas del servidor web en el servidor de Linux, para ello primero analizaremos el contenido del fichero “apache2.conf” en la ruta “/etc/apache/”. En este fichero es donde se configura la mayor parte del servidor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

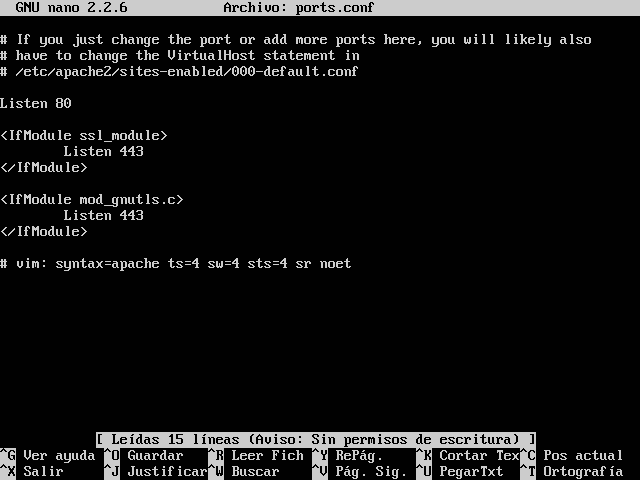
En este fichero podemos ver las configuraciones globales y los valores de las directivas:

* 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.

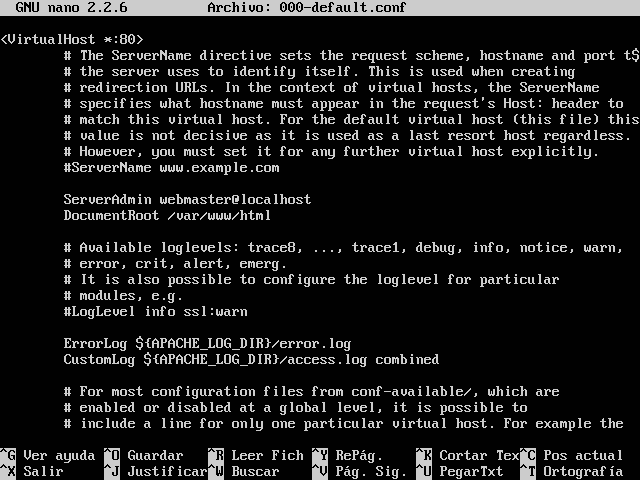
Observamos que permite incluir ficheros de configuración en los directorios “conf-enabled/” y “sites-enabled”.

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

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

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.

Analizamos el fichero “000-default.conf” en el directorio “sites-availabe”.

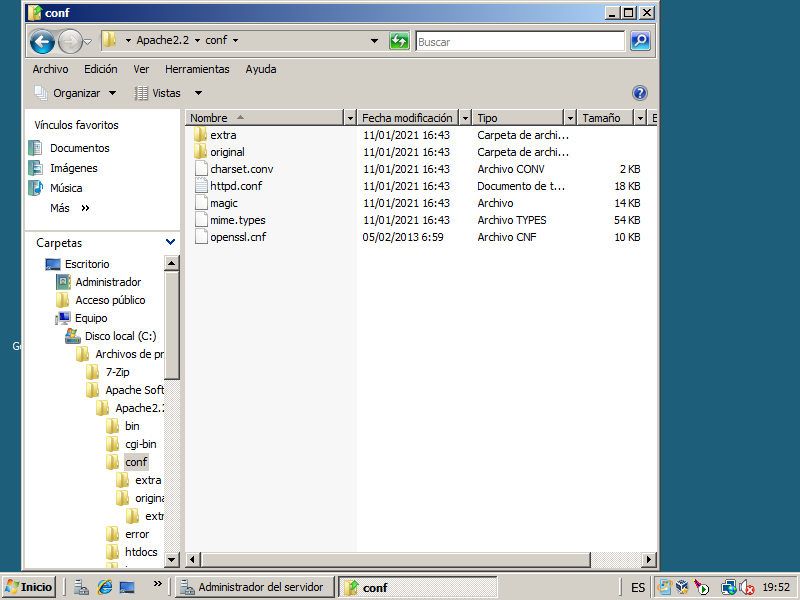
Observamos los valores de las directivas contenidas en VirtualHost:

* ServerRoot: /var/www/html.
* ErrorLog: ${APACHE\_LOG\_DIR}/error.log.
* CustomLog: ${APACHE\_LOG\_DIR}/access.log.

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

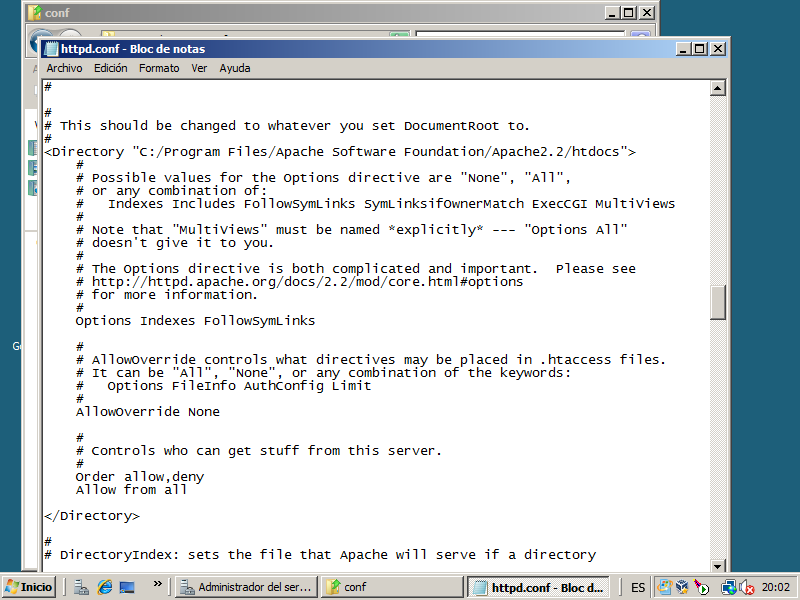
1. **Ficheros de configuración y directivas en Windows**

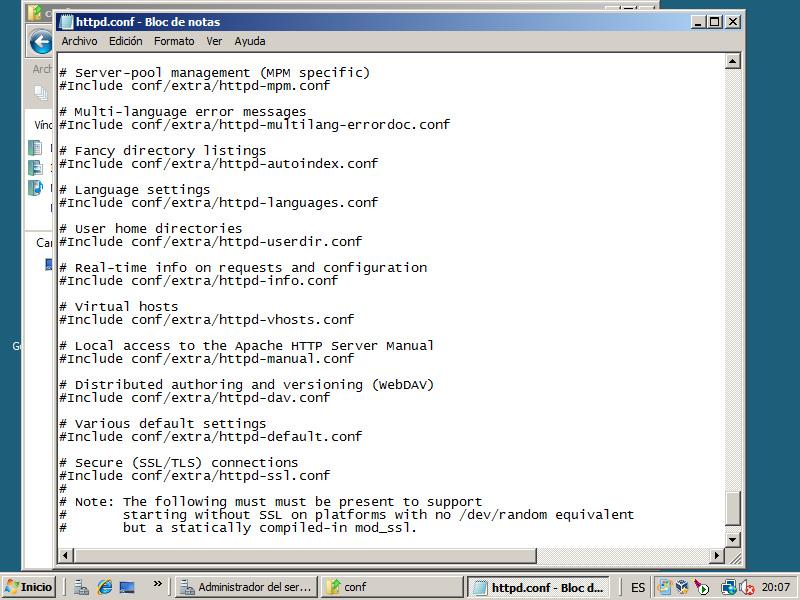
**Ficheros de configuración:**

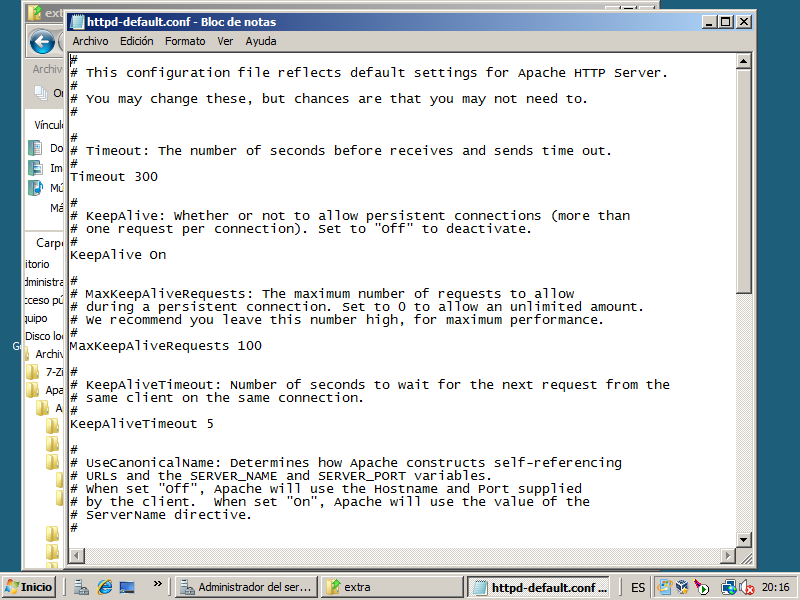
**** Contenido del directorio “Apache Software Foundation\Apache2.2\conf”.

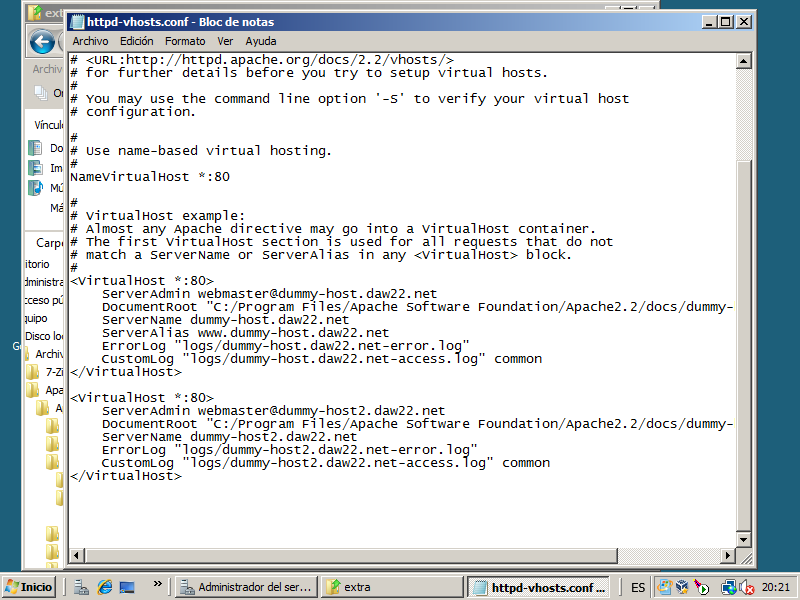
Analizamos el fichero “httpd.conf” y los valores de sus directivas:

* 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”.

Comprobamos el contenido de la directiva “Directory”.

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

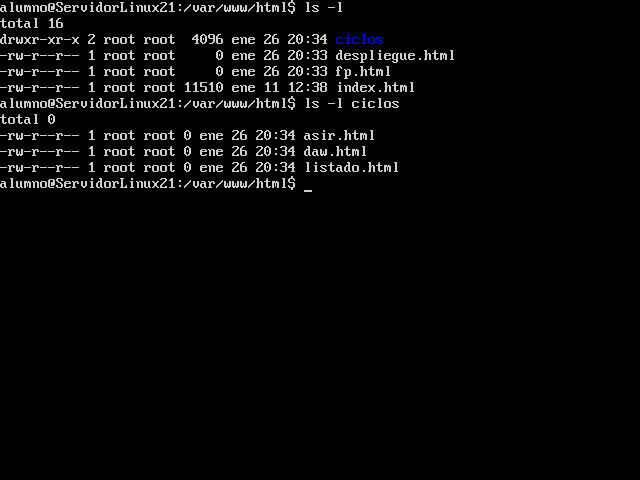
Observamos los valores por defecto en el fichero “httpd-default”.

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

1. **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:

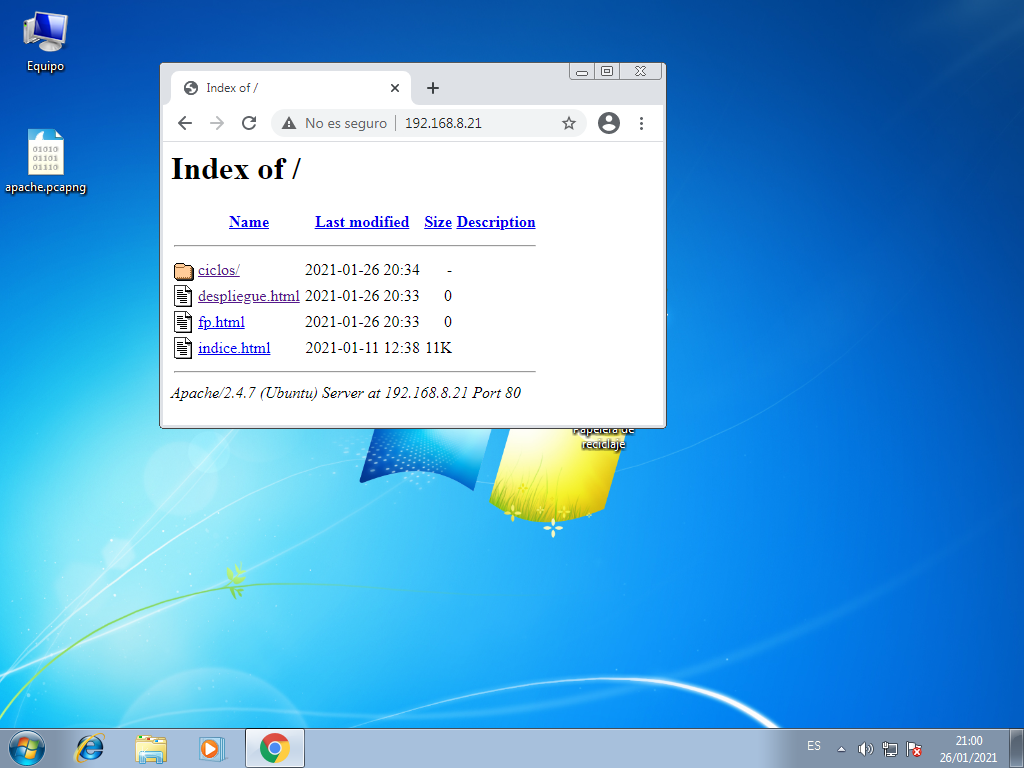
* /var/www/html/despliegue.html
* /var/www/html/fp.html
* /var/www/html/ciclos/listado.html
* /var/www/html/ciclos/asir.html
* /var/www/html/ciclos/daw.html
* /var/www/html/ciclos/dam.html

 estructura de directorios y ficheros creada en la ruta “/var/www/html/”.

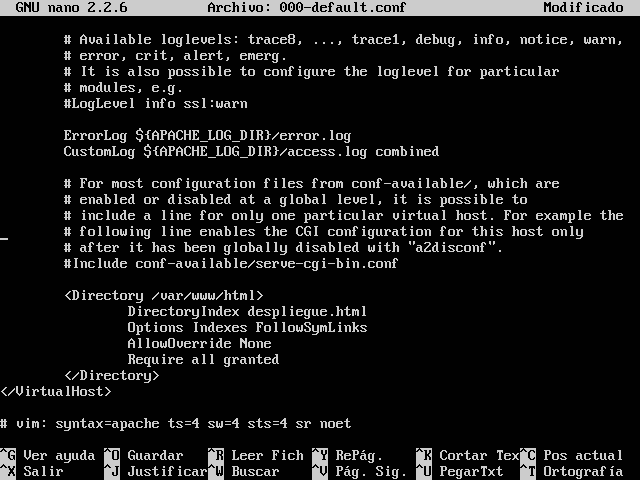
A continuación, buscamos en el navegador de la máquina con Windows 7, los archivos que acabamos de crear en el directorio raíz de Apache en Linux.

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”.

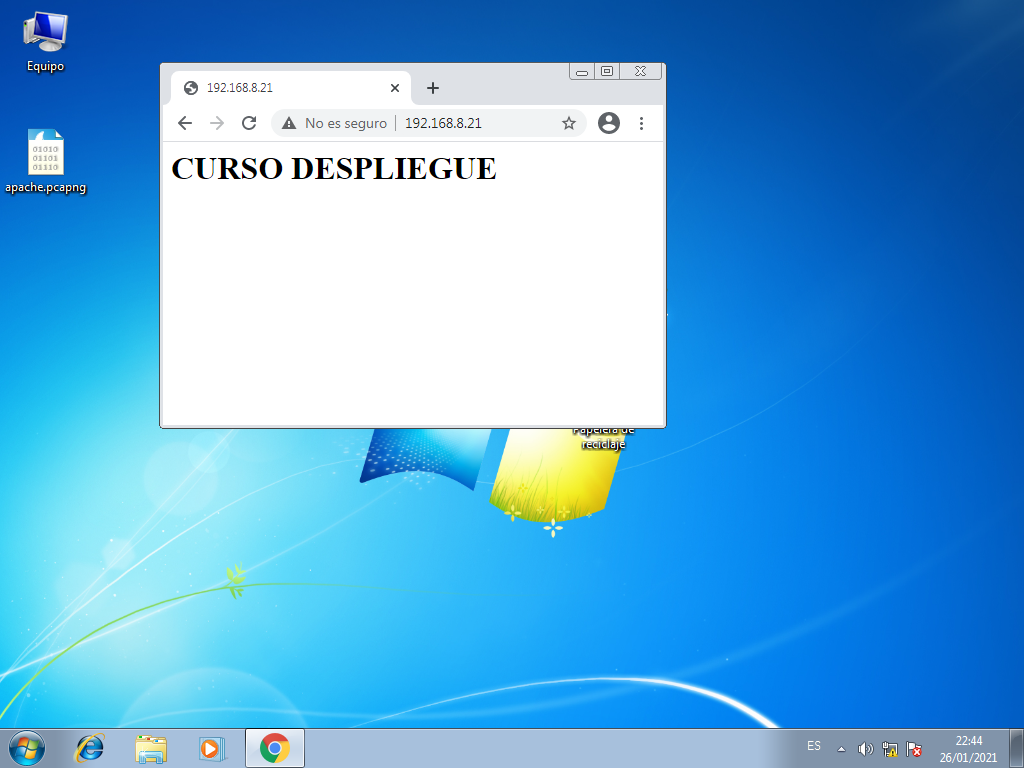
Seguidamente, cambiamos el nombre del fichero “index.html” por “indice.html” y volvemos y recargamos la página del navegador sin especificar el recurso y observamos que al no encontrar el fichero index.html, nos muestra el directorio raíz “/var/www/html”.



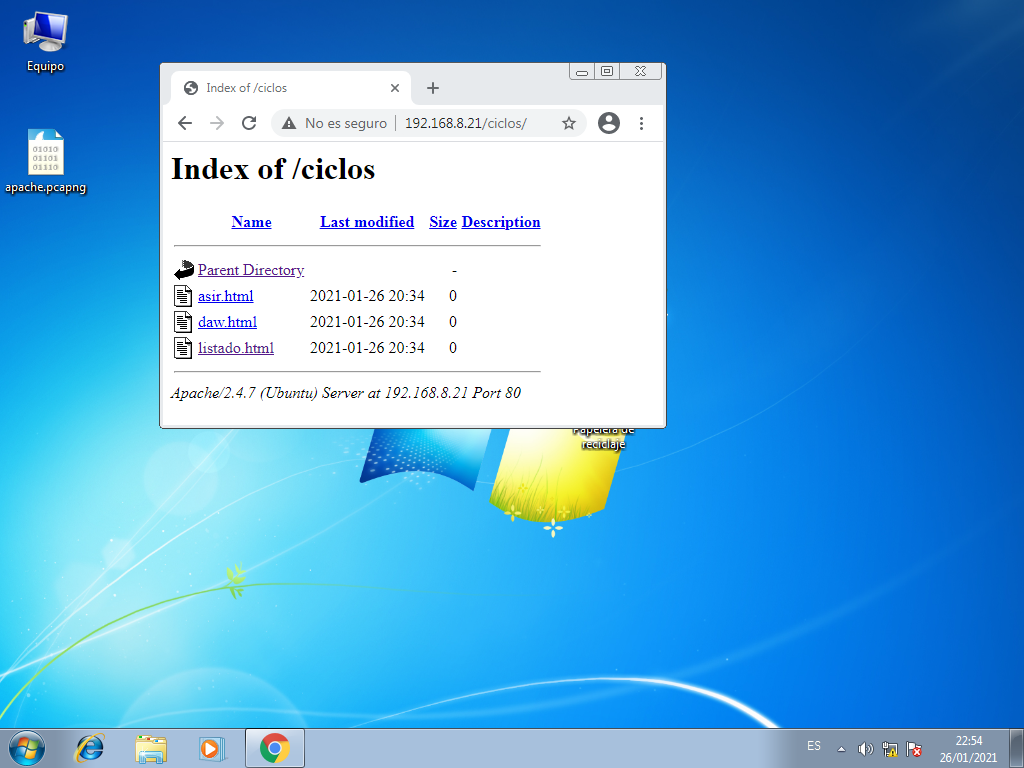
Ahora vamos a añadir la sección “<Directory /var/www/html>…</Directory>” en el fichero ”000-default.conf” con la siguiente directiva:



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”

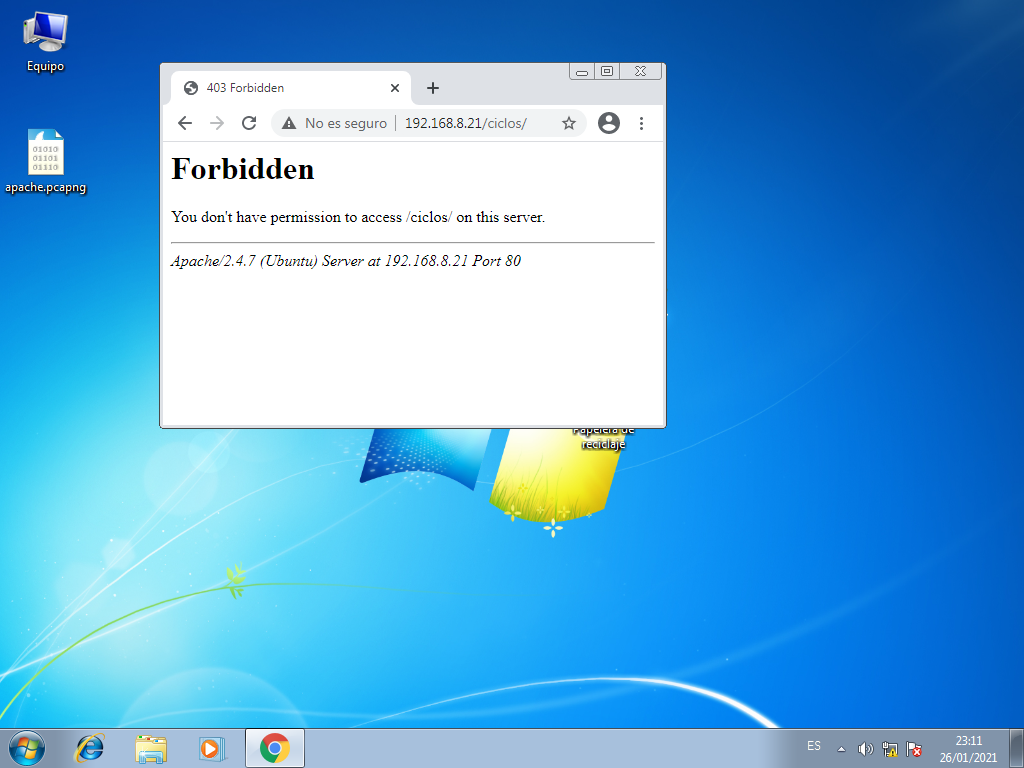
 Comprobamos que nos devuelve su contenido.

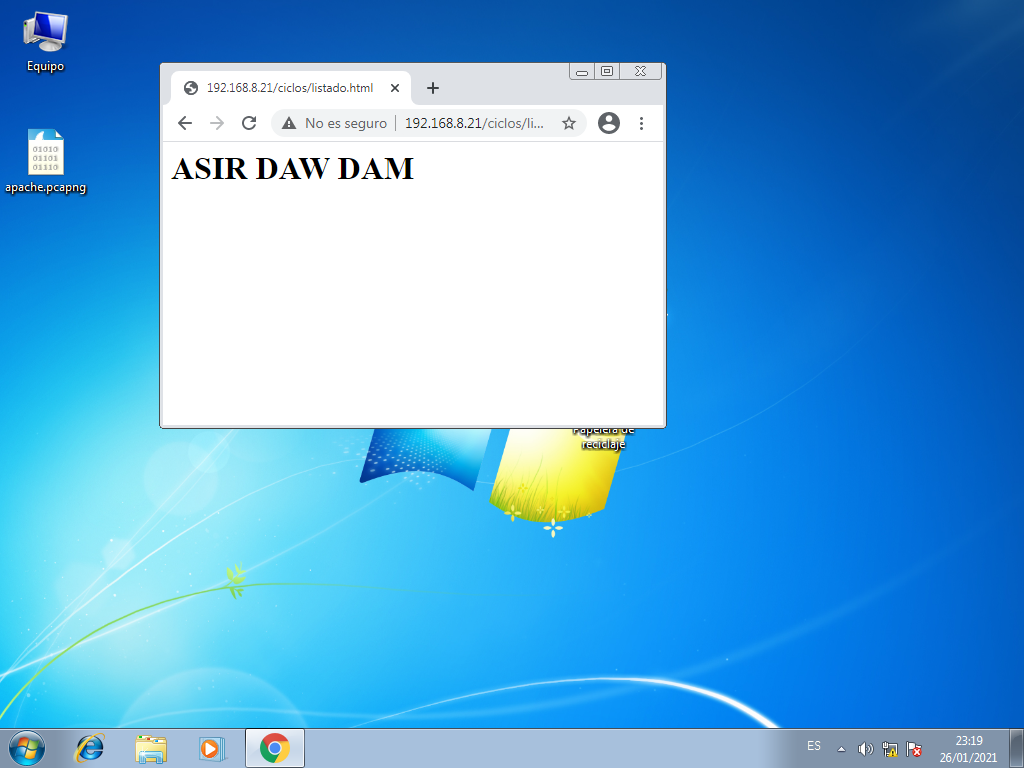
Esto es porque ha heredado la configuración de “/var/www/html” y no encuentra el fichero despliegue.html con lo cual nos muestra su contenido.

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



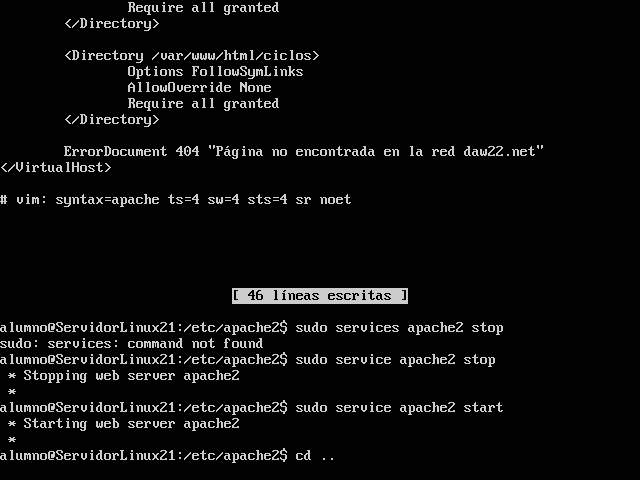
Puesto que no hemos añadido la opción “Indexes” y tampoco la opción “DirectoryIndex”, el navegador nos devuelve un mensaje de prohibición.

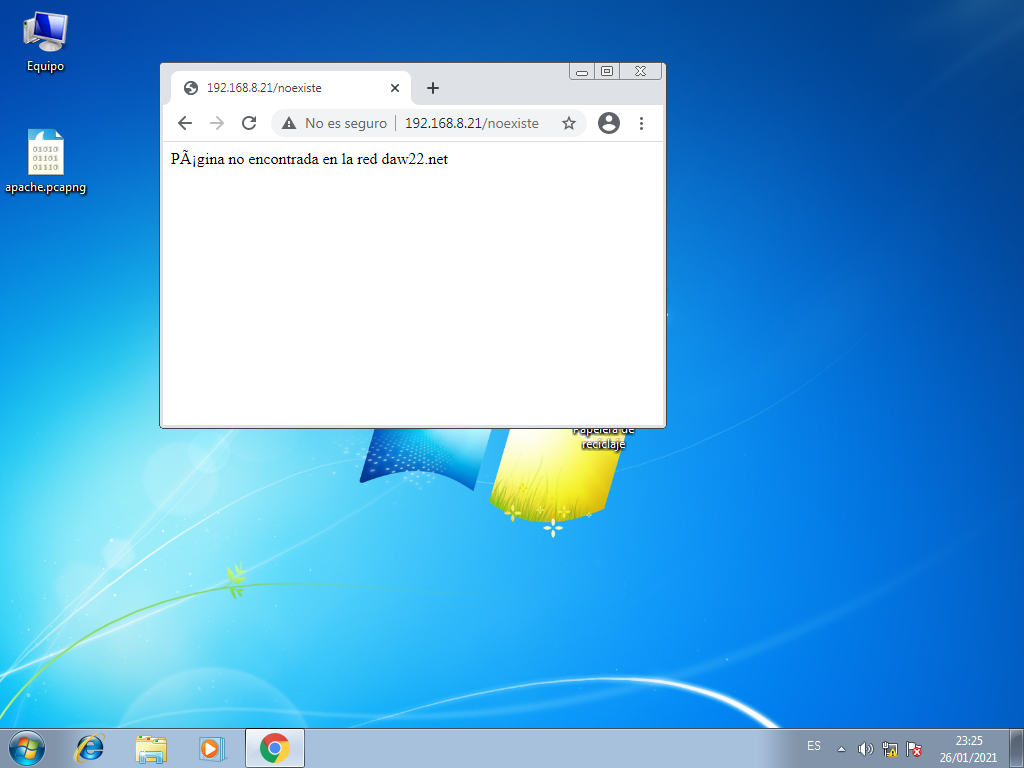
 Comprobamos que no tenemos acceso al directorio “/var/www/html/ciclos”.

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

**Códigos del error (ErrorDocument):**

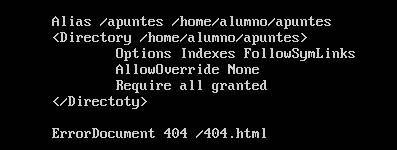
Ahora cambiaremos el mensaje de error 404 por defecto. Para ello editaremos el fichero “/etc/apache2/sites-available/000-default.conf”.



 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:



**Redirreciones (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”.



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

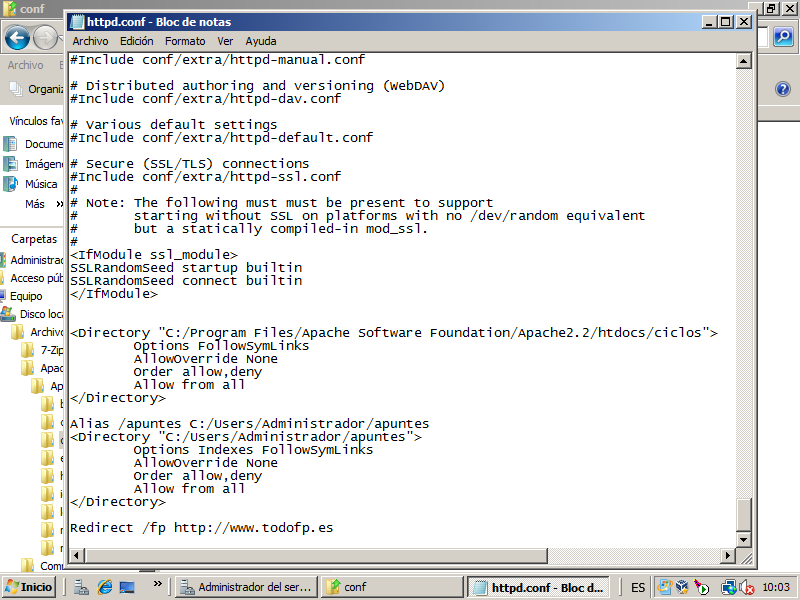
1. **Configuración básica en Windows**

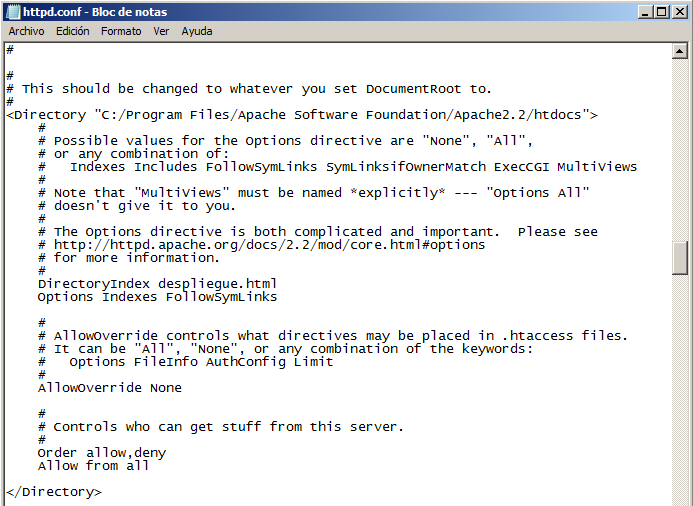
**Ficheros y directorios de prueba**

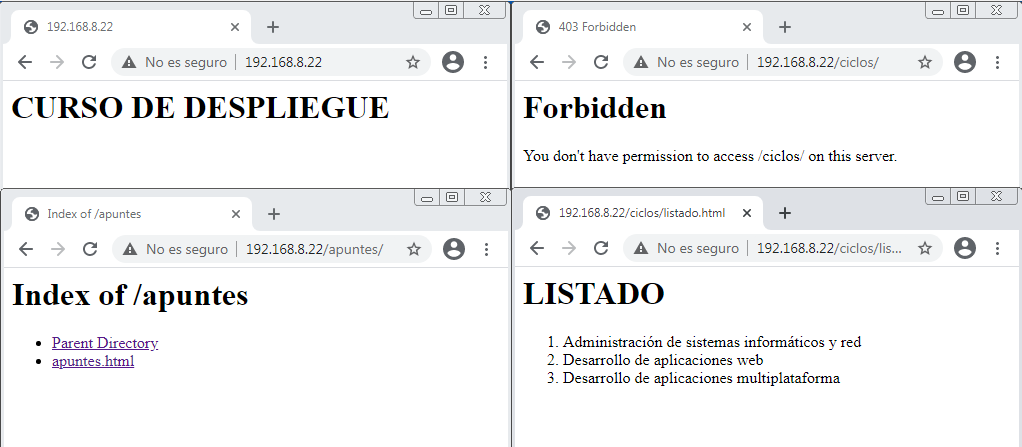
Creamos la misma estructura de directorios y ficheros que en la configuración de Linux.

* …\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.

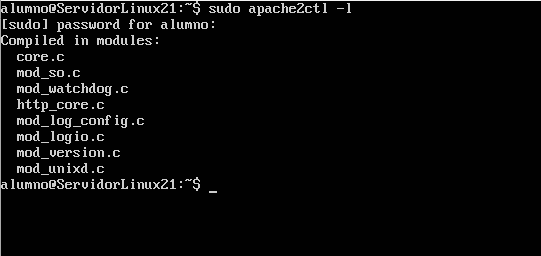


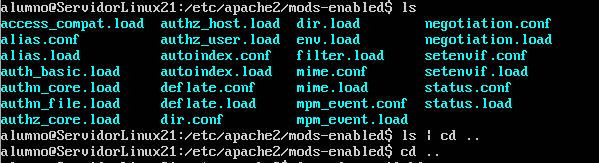
Añadimos también la directiva “DirectoryIndex” en la sección “<Directory "C:/Program Files/Apache Software Foundation/Apache2.2/htdocs">” con el fichero “despliegue.html”.

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

1. **Módulos en Linux**

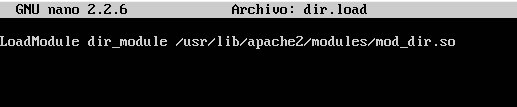
**Módulos:**

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

  
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.

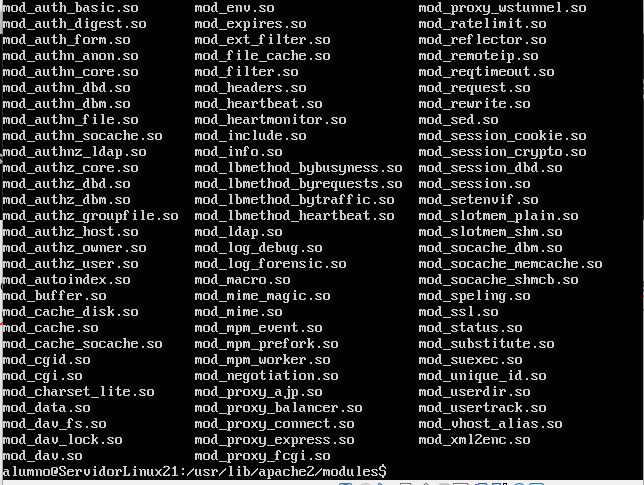
  
Módulos disponibles en el directorio “/etc/apache2/mods-available”.

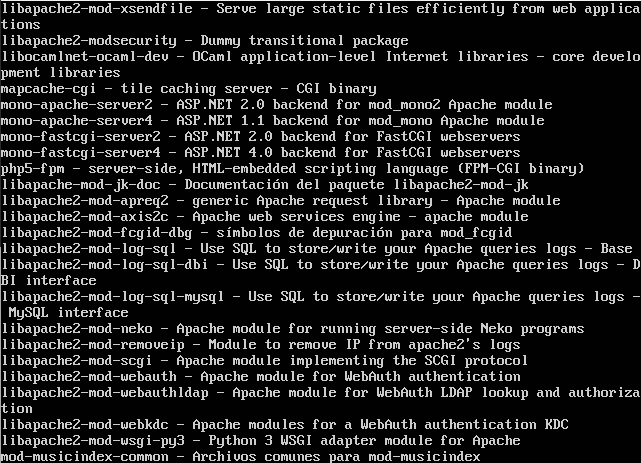
  
Analizamos el fichero “/etc/apache2/mods-available/dir.load”.

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”.

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

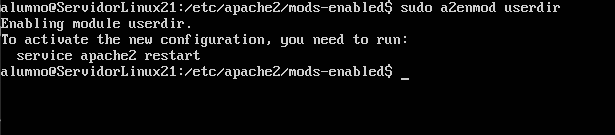
Observamos cómo se añaden directivas dentro de una declaración “<IfModule nombreModulo>…<IfModule>”. Estos módulos se ejecutarán sólo si se carga el módulo en la ejecución del servidor.

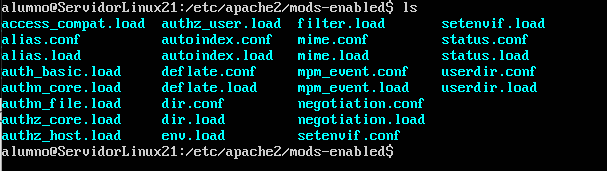
  
Consultamos el directorio “/usr/lib/apache2/modules/” y vemos todos los módulos disponibles para cargar.

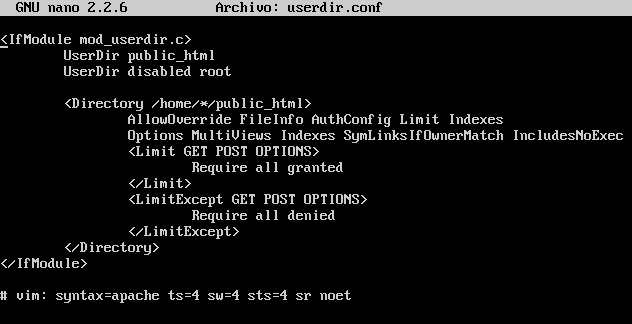
  
Consultamos los paquetes disponibles en los repositorios de Ubuntu que permiten instalar módulos adicionales en Apache mediante el comando “sudo apt-cache search libapache2-mod”.

**Módulo userdir:**

  
Comprobamos que no tenemos habilitado el módulo “userdir”.

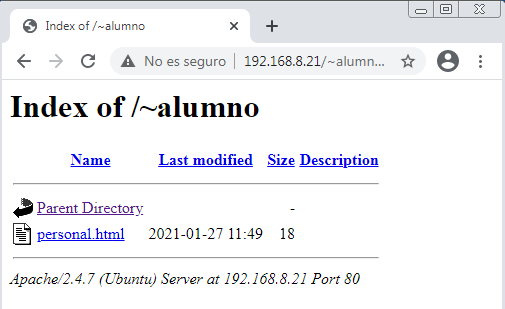
  
Habilitamos el módulo “userdir”mediante el comando“sudo a2enmod userdir”.

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

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

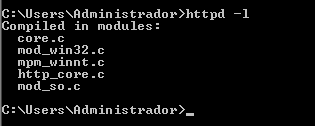
Observamos que está habilitado el uso de directorios personales para todos los usuarios excepto para el usuario “root” y que “public\_html” es el nombre del subdirectorio en el que los usuarios podrán publicar sus páginas personales en su directorio “home”.

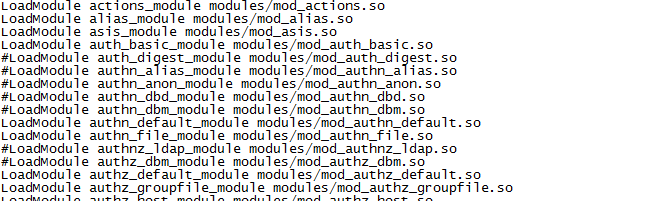
A continuación, creamos el directorio “/home/alumno/public\_html” y dentro, el fichero “personal.html”

  
Comprobamos que el módulo “userdir” funciona correctamente.

1. **Módulos en Windows**

**Módulos:**

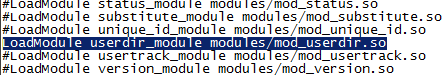
  
Comprobamos los módulos estáticos cargados con el arranque del servidor Apache mediante el comando “httpd -l”.

  
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”.

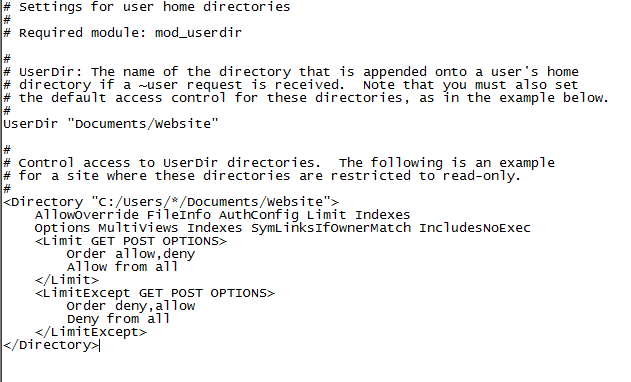
.

**Módulo (userdir):**

Eliminamos el comentario del módulo “userdir\_module” y lo habilitamos.



  
Añadimos la directiva “Include” para que habilite el módulo.

  
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.

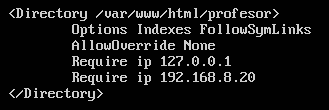
A continuación, creamos el fichero “C:\Usuarios\Administrador\Websites\personal.html”.

  
Comprobamos que el módulo “userdir” se ha habilitado correctamente.

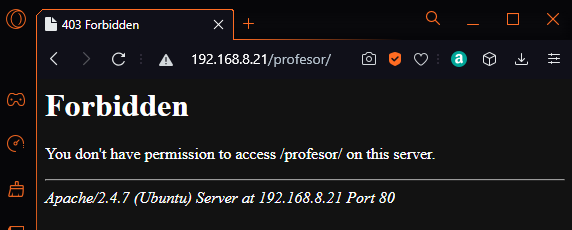
1. **Control de acceso por IP y nombre de dominio**

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”:

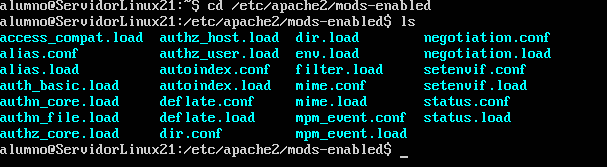
  
Utilizamos la directiva “Require ip” para que sólo se pueda acceder de el equipo local y el equipo con la dirección “192.168.8.20” asociada a Windows 7.

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

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

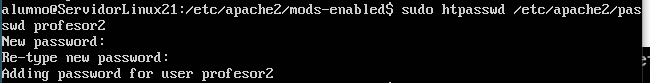
1. **Autenticación y autorización *Basic* y *Digest***

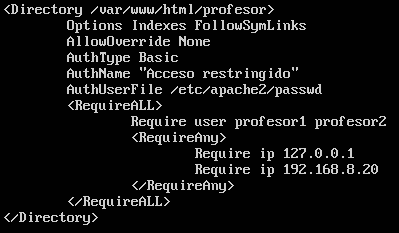
**Autenticación HTTP *Basic*:**

  
Comprobamos que está habilitado el módulo “auth\_basic.load”.

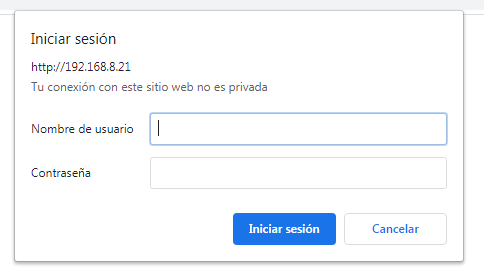
A continuación, crearemos un fichero en el que se guardarán los usuarios y sus contraseñas. Para ello instalaremos el paquete “apache2-utils”.

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

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

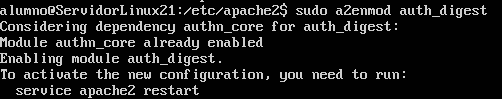
  
Editamos el fichero “/etc/apache2/sites-available/000-default.conf” con las directivas necesarias.

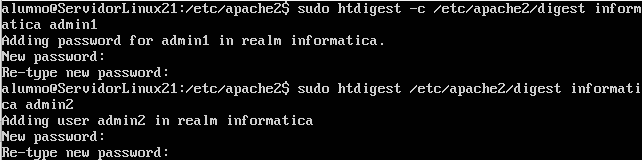
De este modo, a los usuarios “profesor1” y “profesor2” se les requerirá su usuario y contraseña para poder acceder al directorio “/var/html/profesor/”.

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

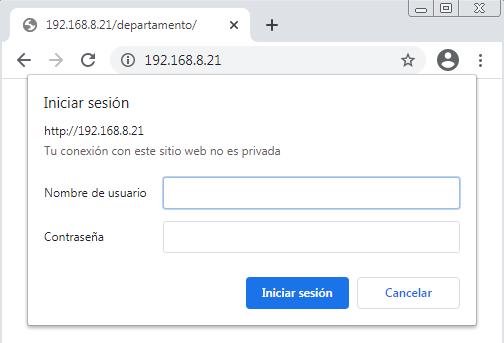
  
Comprobamos que tenemos acceso tanto con el usuario “profesor1” como “profesor2”.

**Autenticación HTTP *Digest*:**

  
Habilitamos el módulo “auth\_digest” mediante el comando “sudo a2enmod auth\_digest”.

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

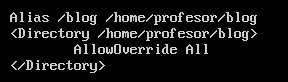
  
Editamos el fichero “/etc/apache2/sites-available/000-default.conf” con las directivas necesarias.

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

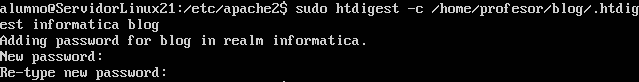
  
Comprobamos que tenemos acceso tanto con el usuario “admin1” como “admin2”.

1. **Ficheros .htaccess**

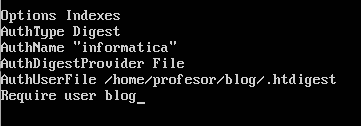
A continuación vamos a habilitar el uso de ficheros de configuración personalizada de directorios (.htacces). Para ello, en primer lugar crearemos el usuario profesor en la Máquina de Linux y tras cambiar la sesión al usuario profesor, creamos el directorio y fichero “/home/profesor/blog/blog.html”.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

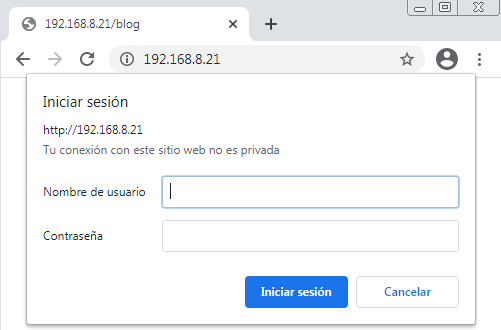


Una vez reiniciado el servidor y aplicado los cambios, creamos el fichero “/home/profesor/blog/.htdigest” y añadimos al usuario blog.

  
Creamos el fichero “/home/profesor/blog/.htdigest” y añadimos el usuario “blog” mediante el comando “htdigest /home/profesor/blog/.htdigest informatica blog”.

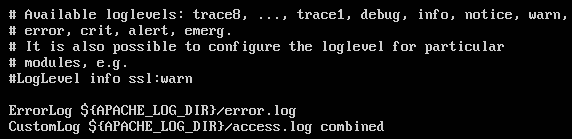
A continuación, creamos el fichero “/home/profesor/blog/.htaccess” y añadimos las directivas necesarias.

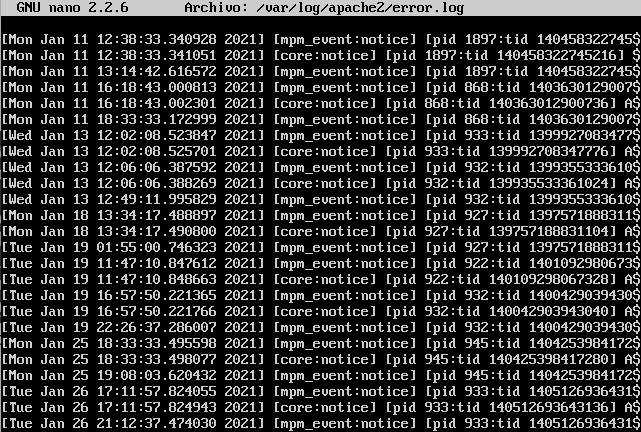


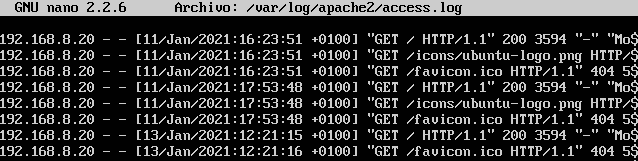
  
Comprobamos que nos pide usuario y contraseña para el directorio “blog”.

  
Comprobamos que tenemos acceso tanto con el usuario “blog”.

1. **Ficheros de registros (logs)**

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

  
Consultamos el log de errores.

  
Consultamos el log de accesos.

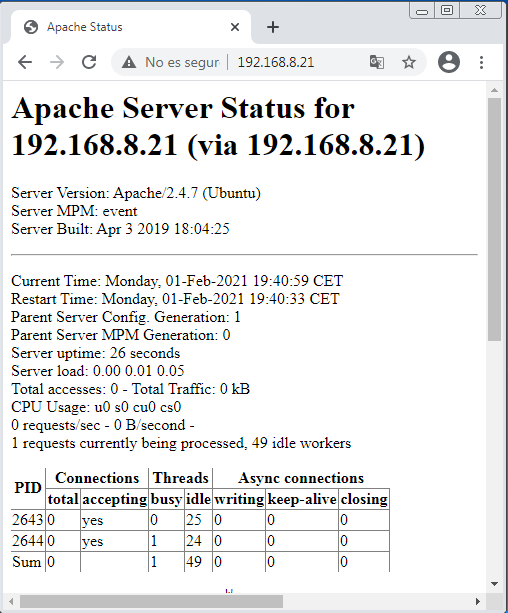
1. **Módulos *mod\_status* y *mod\_info***

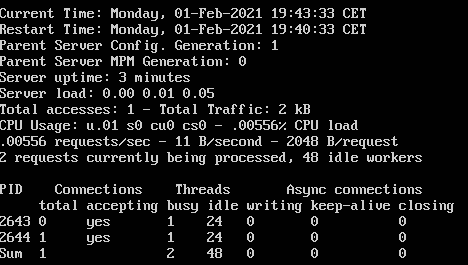
En este apartado vamos a probar los módulos ”mod\_status” y “mod\_info”.

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

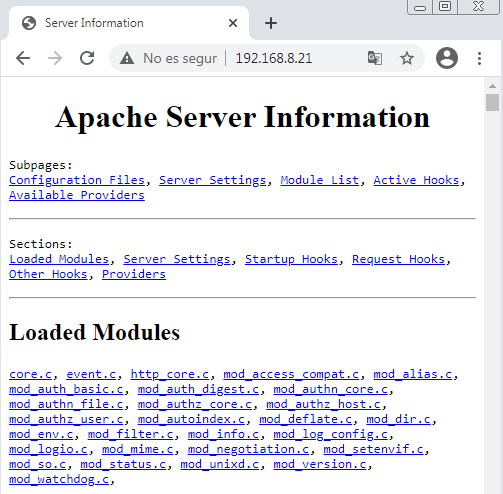
A continuación, editamos el fichero de configuración del módulo con las directivas necesarias para tener acceso desde la máquina de Windows 7 y reiniciamos el servidor.



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

  
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.