

# **DESPLIEGUE DE APLICACIONES WEB. EJERCICIOS APACHE(I)**

Pablo Cidón  
Curso 2017-2018

## Sumario

1- Comando apache2ctl:.....	4
I. Comprobar si el paquete apache2 está instalado. En caso negativo, instalarlo.....	4
II. Comprobar el número de versión de apache.....	4
III. Infome completo del servidor apache.....	5
IV. Listar los módulos activos.....	6
V. Listar los sitios activos, ¿Muestra las variables de configuración, por ejemplo APACHE_LOG_DIR? Compararlo con el fichero /etc/apache2/envvars.....	6
VI. Chequear los ficheros de configuración.....	7
2. Comando a2query. Información que podemos visualizar a través de este comando.....	8
3. Listar los ficheros de configuración activos.....	8
4. Configuración del sitio por defecto:.....	8
I. Crea un directorio llamado ciclos en el directorio raíz del sitio por defecto.....	8
II. En dicho directorio, crea una página web, listado.html, donde muestra el listado con el nombre de los ciclos de grado superior de informática.....	9
III. Prohibir que el servidor realice un listado con el contenido de dicho directorio. Comprobación (http://IP/ciclos).....	9
IV. ¿Es posible acceder al archivo listado.html?.....	9
V. Personaliza el mensaje de error 403 (Acceso prohibido) y 404 (Recurso no encontrado en el servidor de xxx-us12.iniciales.local). Comprobación desde el cliente ( http://IP/ciclos/noexiste.html y también http://IP/ciclos). ErrorDocument 403 “no tienes permisos”.....	10
VI. Configura el servidor virtual por defecto para que cuando retorne el código de error 404 (página no encontrada) envíe la página 404.html almacenada en el directorio raíz del servidor. Comprobación. ErrorDocument 404 /404.html.....	10
VII. Crea un usuario “alumno” que en su home tenga una carpeta con sus apuntes. Modifica la configuración del sitio virtual por defecto para poder acceder a la carpeta apuntes (Directorio Virtuales).....	12
VIII. En el sitio por defecto crea una redirección. Utiliza la directiva Redirect para crear una redirección de /fp hacia http://www.todofp.es. Comprueba su funcionamiento en el navegador y también visualiza la cabeceras de la petición y de la respuesta ¿ código del estado?.....	12
IX. Si el recurso está disponible en el servidor, (crea el directorio fp a partir del directorio raíz del sitio por defecto) , ¿realiza la redirección? ¿Que código de estado devuelve?.....	13
X. Configura el servidor web para que los usuarios del sistema excepto el root, pueda acceder a sus directorios personales donde contiene un archivo personal.html (información personal del usuario) . Compruébalo con el usuario alumno.....	13
XI. En el sitio virtual por defecto, crea un directorio profesor que solo sea accedida desde el propio servidor (127.0.0.1) y también desde la IP de la máquina anfitriona del alumno. Comprobación:.....	15
1. En el propio servidor con el navegador lynx .....	16
2. En el ordenador del alumno con un navegador.....	16
XII. Usando autenticación básica se pide controlar que solo pueden acceder /var/www/html/profesor a través de HTTP los usuarios: profesor1 y profesor2 (usuarios de apache). Comprobación desde el cliente que pide la autenticación para acceder a la carpeta profesor. Observar el intercambio de tramas entre el cliente y el servidor, y ver el metadato Authorization.....	16
XIII. Crea el directorio /var/www/html/public_html/departamento con un archivo	

departamento.html . Usando autenticación digest crear un fichero con los usuarios (admin1 y admin2) y contraseñas asociadas al dominio (realm) informatica. Configurar las directivas asociadas al directorio departamento para el acceso a los usuarios admin1 y admin2. Intenta acceder a http://IP/departamento. ¿es posible?.....	18
XIV. Habilita en Apache el uso de ficheros de configuración personalizada de directorios (.htaccess). En el directorio /home/profesor/blog para que sea el propio usuario profesor el que pueda controlar como sirve Apache los contenidos. Para ello, se pide documentar las siguientes configuraciones.....	20
1. Desde el sitio por defecto:.....	20
2. Crea un usuario profesor.....	20
3. Desde la máquina anfitriona sin loggarte en el servidor (a través de FTP/ SFTP con Notepad++ o cliente FTP/SFTP) mediante el usuario profesor (es necesario crearlo) haz uso del fichero .htaccess y configura el directorio /home/profesor/blog.....	21
XV. Módulo status.....	22
1. En el servidor web debe estar habilitado el módulo mod_status (status) y configura para que se conecte desde cualquier equipo de la red del aula.....	22
2. Comprobación: http://IP/server-status.....	23
XVI. Módulo info: Visualizar la vista resumida de la configuración del servidor.....	23
1. En el servidor web debe estar habilitado el módulo mod_info (info) y restringir el acceso desde un solo equipo (anfitrión).....	23
2. Comprobación http://IP/server-info.....	24

## 1- Comando apache2ctl:

I. Comprobar si el paquete apache2 está instalado. En caso negativo, instalarlo.

```
miadmin@PCB-USED:~$ apache2ctl
Usage: /usr/sbin/apache2ctl start|stop|restart|graceful|graceful-stop|configtest
|status|fullstatus|help
      /usr/sbin/apache2ctl <apache2 args>
      /usr/sbin/apache2ctl -h                (for help on <apache2 args>)
```



II. Comprobar el número de versión de apache

```
miadmin@PCB-USED:~$ apache2ctl -v
Server version: Apache/2.4.18 (Ubuntu)
Server built:   2017-09-18T15:09:02
```

### III. Infome completo del servidor apache

```
miadmin@PCB-USED:~$ apache2ctl fullstatus
      Apache Server Status for localhost (via ::1)

Server Version: Apache/2.4.18 (Ubuntu)
Server MPM: prefork
Server Built: 2017-09-18T15:09:02

-----

Current Time: Monday, 16-Apr-2018 10:06:21 CEST
Restart Time: Monday, 16-Apr-2018 08:35:52 CEST
Parent Server Config. Generation: 1
Parent Server MPM Generation: 0
Server uptime: 1 hour 30 minutes 29 seconds
Server load: 0.11 0.10 0.09
Total accesses: 4 - Total Traffic: 9 kB
CPU Usage: u0 s.02 cu0 cs0 - .000368% CPU load
.000737 requests/sec - 1 B/second - 2304 B/request
1 requests currently being processed, 5 idle workers

__W__ .....
.....
.....

Scoreboard Key:
"_" Waiting for Connection, "S" Starting up, "R" Reading Request,
"W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,
"C" Closing connection, "L" Logging, "G" Gracefully finishing,
"I" Idle cleanup of worker, "." Open slot with no current process

Srv PID Acc M CPU SS Req Conn Child Slot Client VHost Request
0-0 1181 0/3/3 _ 0.02 671 0 0.0 0.01 0.01 192.168.3.11 127.0.1.1:80 GET
/favicon.ico HTTP/1.1
2-0 1186 0/0/0 W 0.00 0 0 0.0 0.00 0.00 ::1 127.0.1.1:80 GET
/server-status HTTP/1.0
3-0 1189 0/1/1 _ 0.00 131 95 0.0 0.00 0.00 192.168.3.11 127.0.1.1:80
NULL

-----

Srv  Child Server number - generation
PID  OS process ID
Acc  Number of accesses this connection / this child / this slot
M    Mode of operation
CPU  CPU usage, number of seconds
SS   Seconds since beginning of most recent request
Req  Milliseconds required to process most recent request
Conn Kilobytes transferred this connection
Child Megabytes transferred this child
Slot  Total megabytes transferred this slot

-----

Apache/2.4.18 (Ubuntu) Server at localhost Port 80
```

## IV. Listar los módulos activos

```
miadmin@PCB-USED:~$ apache2ctl -t -D DUMP_MODULES
AH00558: apache2: Could not reliably determine the
in name, using 127.0.1.1. Set the 'ServerName' dir
is message
Loaded Modules:
  core_module (static)
  so_module (static)
  watchdog_module (static)
  http_module (static)
  log_config_module (static)
  logio_module (static)
  version_module (static)
  unixd_module (static)
  access_compat_module (shared)
  alias_module (shared)
  auth_basic_module (shared)
  authn_core_module (shared)
  authn_file_module (shared)
  authz_core_module (shared)
  authz_host_module (shared)
  authz_user_module (shared)
  autoindex_module (shared)
  deflate_module (shared)
  dir_module (shared)
  env_module (shared)
  filter_module (shared)
  mime_module (shared)
  mpm_prefork_module (shared)
  negotiation_module (shared)
  php7_module (shared)
  setenvif_module (shared)
  status_module (shared)
```

## V. Listar los sitios activos, ¿Muestra las variables de configuración, por ejemplo `APACHE_LOG_DIR`? Compararlo con el fichero `/etc/apache2/envvars`

```
miadmin@PCB-USED:~$ apache2ctl -S
AH00558: apache2: Could not reliably determine the server's fully qualified domain
in name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress th
is message
VirtualHost configuration:
*:80                  127.0.1.1 (/etc/apache2/sites-enabled/000-default.conf:1)
ServerRoot: "/etc/apache2"
Main DocumentRoot: "/var/www/html"
Main ErrorLog: "/var/log/apache2/error.log"
Mutex default: dir="/var/lock/apache2" mechanism=fcntl
Mutex mpm-accept: using_defaults
Mutex watchdog-callback: using_defaults
PidFile: "/var/run/apache2/apache2.pid"
Define: DUMP_VHOSTS
Define: DUMP_RUN_CFG
User: name="www-data" id=33 not_used
Group: name="www-data" id=33 not_used
```

## VI. Chequear los ficheros de configuración.

```
miadmin@PCB-USED:~$ apache2ctl configtest
Syntax OK
```

Si al chequear nos muestra un mensaje como este, no es nada malo, es un mensaje de advertencia, pero funcionará correctamente.

```
miadmin@PCB-USED:~$ apache2ctl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
Syntax OK
```

Para que no aparezca, solamente es necesario ir al fichero `/etc/apache2/apache2.conf`, y añadir la siguiente línea:

```
#ServerRoot "/etc/apache2"
ServerName localhost
```

## VII. Listado de puertos

```
miadmin@PCB-USED:~$ cat /etc/apache2/ports.conf
# If you just change the port or add more ports
# 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>
```



## 2. Comando a2query. Información que podemos visualizar a través de este comando

```
-a Returns the Apache 2 "Module Magic Version" (API version) number,
the server was compiled with. The returned version does not contain
any minor versions which are known to be compatible with the major
version returned.

-c [CONF]
Checks whether the configuration CONF is enabled. If no argument
was given, all enabled configuration files are being returned. CONF
is compared by string comparison by ignoring a leading "mod_"
prefix and possibly a '.conf' or '.load' suffix.

-h Displays a brief summary how the program can be called and exits.

-m [MODULE]
Checks whether the module MODULE is enabled, The argument is
interpreted in the same way, as for configuration files queried by
the -c switch.

-M Returns the currently enabled Apache 2 MPM (Multi Processing
Module).

-s [SITE]
Checks whether the module SITE is enabled, The argument is
interpreted in the same way, as for configuration files queried by
the -c switch.

-v returns the currently installed Apache 2 HTTP server version

-q suppress any output. This is useful to invoke a2query from another
script. This is useful if only the return code is of interest.

miadmin@PCB-USED:~$ a2query -a
20120211
```

## 3. Listar los ficheros de configuración activos.

```
miadmin@PCB-USED:~$ ls /etc/apache2/conf-enabled/
charset.conf          other-vhosts-access-log.conf  serve-cgi-bin.conf
javascript-common.conf  phpmyadmin.conf
localized-error-pages.conf  security.conf
```

## 4. Configuración del sitio por defecto:

### I. Crea un directorio llamado ciclos en el directorio raíz del sitio por defecto.

```
miadmin@PCB-USED:~$ sudo mkdir /var/www/html/public_html/ciclos
miadmin@PCB-USED:~$ ls /var/www/html/public_html
ciclos  index.html  info.php
```



**II. En dicho directorio, crea una página web, listado.html, donde muestra el listado con el nombre de los ciclos de grado superior de informática.**

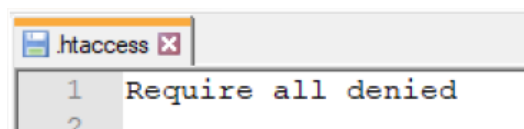
```
miadmin@PCB-USED:~$ sudo nano /var/www/html/public_html/ciclos/listado.html
miadmin@PCB-USED:~$ ls /var/www/html/public_html/ciclos
listado.html
```

**III. Prohibir que el servidor realice un listado con el contenido de dicho directorio. Comprobación (http://IP/ciclos).**

En primer lugar tendremos que configurar el sitio para que reconozca los archivos .htaccess, para ello iremos al fichero /etc/apache2/sites-enabled/000-default.conf, en donde añadiremos la siguiente directiva:

```
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/public_html
<Directory /var/www/html/public_html>
    AllowOverride All
</Directory>
```

Una vez hecho, chequeamos los ficheros y reiniciamos el servicio de apache. Seguidamente crearemos el fichero .htaccess, dentro del directorio /var/www/html/public\_html/ciclos, en el que añadiremos lo siguiente:



```
.htaccess
1 Require all denied
2
```

Finalmente vamos al navegador y comprobamos que funciona correctamente:



**IV. ¿Es posible acceder al archivo listado.html?**

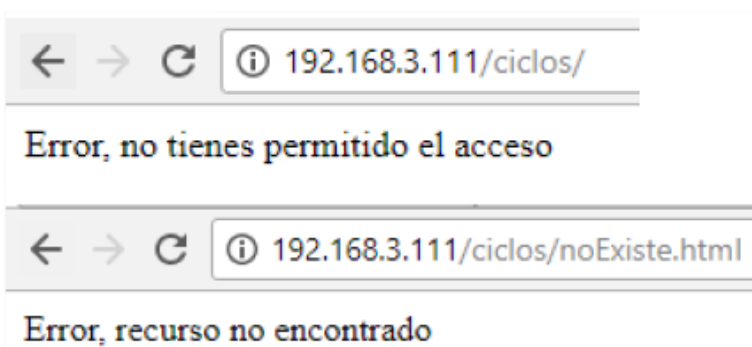
Sí, ya que la directiva del archivo .htaccess, impide que se haga un listado de los archivos, ya que no hay ningún archivo index que se encuentran dentro del directorio, pero si se accede directamente al archivo, podremos ver su contenido.



**V. Personaliza el mensaje de error 403 (Acceso prohibido) y 404 (Recurso no encontrado en el servidor de xxx-us12.iniciales.local). Comprobación desde el cliente ( <http://IP/ciclos/noexiste.html> y también <http://IP/ciclos>). ErrorDocument 403 “no tienes permisos”.**

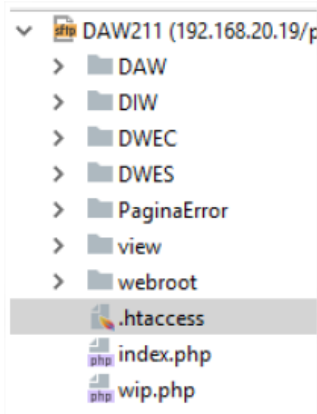
```
Require all Denied
ErrorDocument 404 "Error, recurso no encontrado"
ErrorDocument 403 "Error, no tienes permitido el acceso"
```

Añadimos los errores y lo que vamos a mostrar en cada uno de ellos. Para comprobar el error 403, tendremos que dejar el acceso denegado, por el contrario para comprobar el error 404, tendremos que permitir el acceso.



**VI. Configura el servidor virtual por defecto para que cuando retorne el código de error 404 (página no encontrada) envíe la página 404.html almacenada en el directorio raíz del servidor. Comprobación. ErrorDocument 404 /404.html**

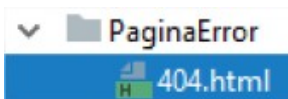
En primer lugar crearemos el archivo .htaccess en el raíz de nuestro directorio.



Una vez creado, añadimos la siguiente línea en el archivo:

```
1 ErrorDocument 404 /DAW211/public_html/PaginaError/404.html
```

Tendremos que crear el directorio para almacenar nuestras páginas de error. Para cargar esas páginas, será necesario introducir la ruta absoluta ya que para que muestre la página va a depender de donde se encuentre el error.



Vamos al navegador y comprobamos que funciona la ventana del error:



## VII. Crea un usuario “alumno” que en su home tenga una carpeta con sus apuntes. Modifica la configuración del sitio virtual por defecto para poder acceder a la carpeta apuntes (Directorio Virtuales).

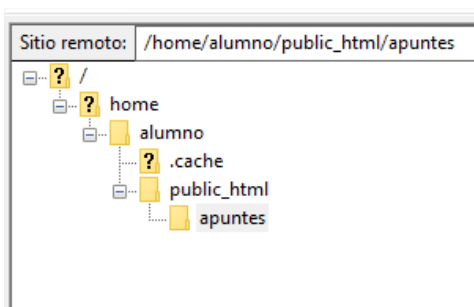
En primer lugar tendremos que activar el módulo Userdir, que es el que permite que los usuarios tengan sus propios directorios web:

```
sudo a2enmod userdir.load
```

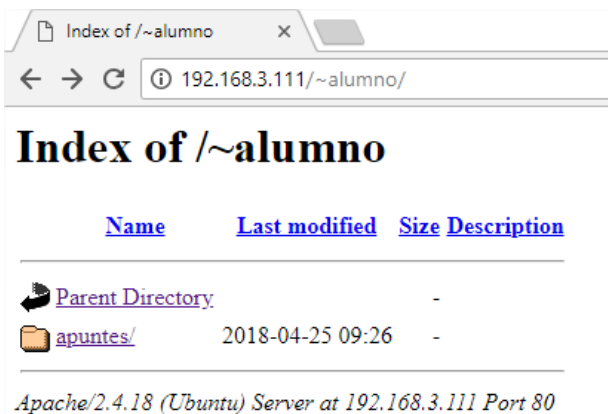
Reiniciamos el servicio, y creamos el usuario alumno:

```
sudo adduser alumno
```

Vamos al Filezilla y creamos los siguientes directorios:

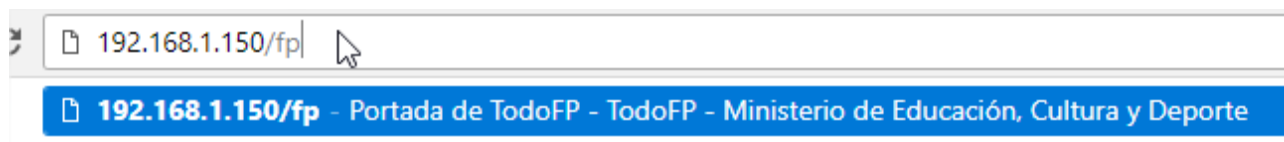


Mapeamos la URL en el navegador y comprobamos que funciona:

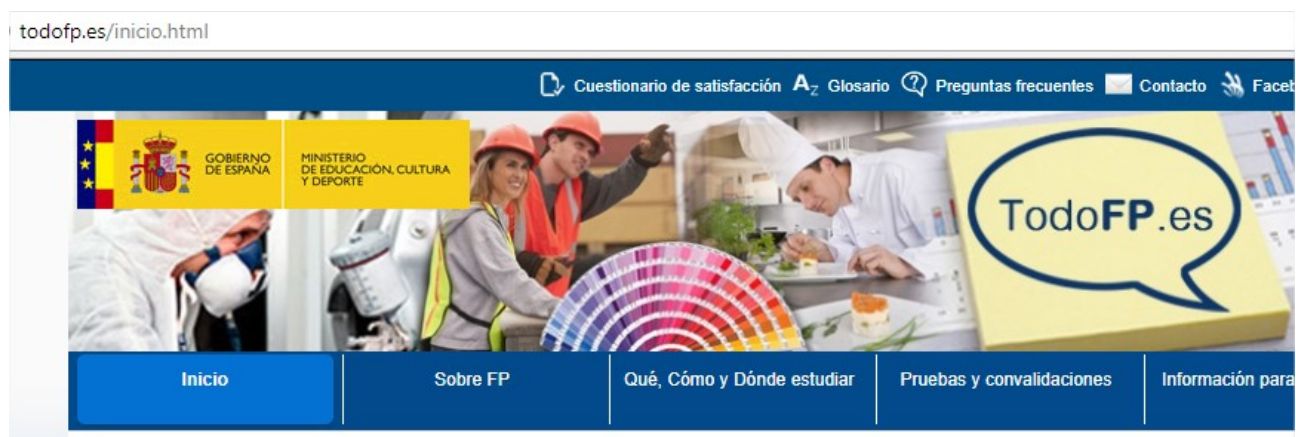


## VIII. En el sitio por defecto crea una redirección. Utiliza la directiva Redirect para crear una redirección de /fp hacia http://www.todofp.es. Comprueba su funcionamiento en el navegador y tambien visualiza la cabeceras de la petición y de la respuesta ¿ código del estado?

En el fichero .htaccess del raíz de nuestro sitio, añadiremos la siguiente directiva, de forma que cuando introduzcamos “host/fp” en el navegador nos envíe a la URL indicada.



Una vez hecho, vamos al navegador y comprobamos que funciona:

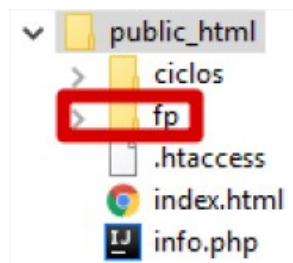


Inspeccionamos en la consola el network y observamos el código que nos devuelve:

<input type="checkbox"/> fp	302	text/html
-----------------------------	-----	-----------

Nos devolverá el código 302, que es el que tiene asignado el código de las redirecciones.

**IX. Si el recurso está disponible en el servidor, (crea el directorio fp a partir del directorio raíz del sitio por defecto) , ¿realiza la redirección? ¿Que código de estado devuelve?**



Creando el directorio, realiza igualmente la redirección ya que el archivo .htaccess es leído antes de acceder al directorio. El código de error que devolverá es el mismo ya que también hace la redirección.

**X. Configura el servidor web para que los usuarios del sistema excepto el root, pueda acceder a sus directorios personales donde contiene un archivo personal.html (información personal del usuario) . Compruébalo con el usuario alumno.**

Comprobamos que la directiva alias se encuentra activa:

```
miadmin@PCB-USED:~$ ls /etc/apache2/mods-enabled/ | grep user
authz_user.load
userdir.conf
userdir.load
```

Creamos el directorio en el raíz de nuestro servidor:

```
miadmin@PCB-USED:~$ ls /home
alumno  miadmin
```

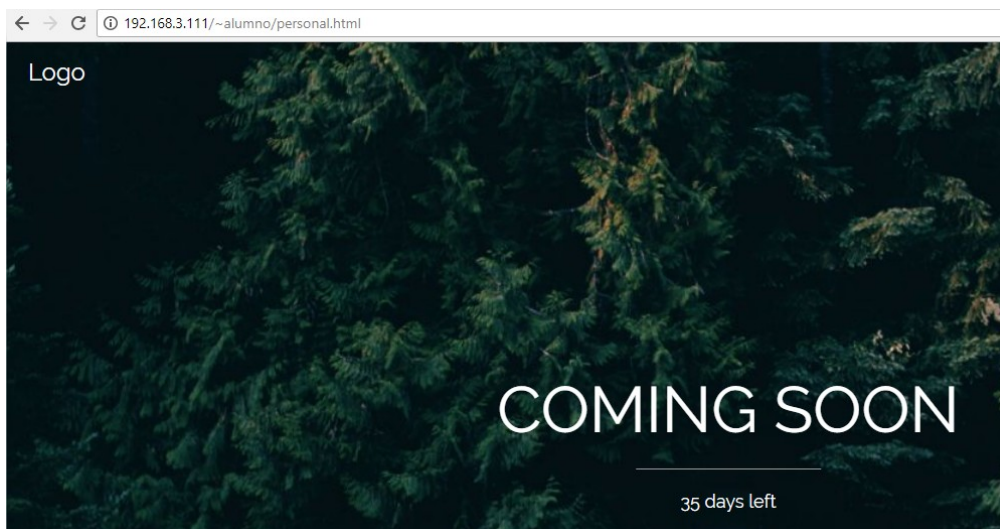
Vamos al fichero de configuración del módulo:

```
sudo nano /etc/apache2/mods-enabled/userdir.conf
```

Dentro del fichero haremos lo siguiente:

```
<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>
```

Guardamos los cambios, reiniciamos el servicio y comprobamos que funciona:



Para no tener que estar poniendo el carácter '~', podremos establecer un alias, para ello haremos lo siguiente:



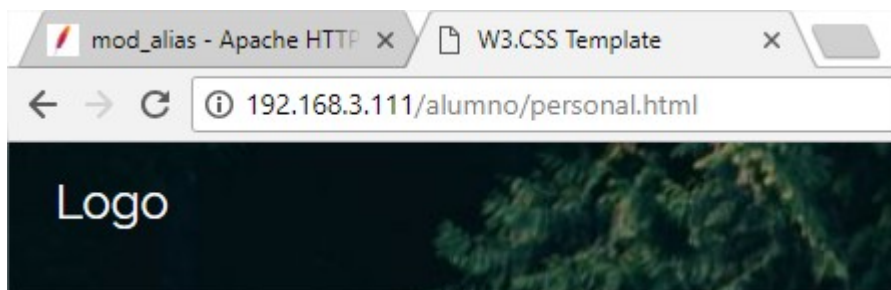
En primer lugar iremos al fichero de configuración del módulo, ya que es algo que solo afecta a este módulo:

```
<IfModule mod_userdir.c>
    UserDir public_html
    UserDir disabled root
    Alias /alumno "/home/alumno/public_html"
    <Directory /home/*/public_html>
        AllowOverride FileInfo AuthConfig Limit Index
        Options MultiViews Indexes SymLinkIfOwnerMatch
```

Guardamos los cambios y reiniciamos el servicio:

```
sudo service apache2 restart
```

Finalmente comprobamos que funciona:



**XI. En el sitio virtual por defecto, crea un directorio profesor que solo sea accedida desde el propio servidor (127.0.0.1) y también desde la IP de la máquina anfitriona del alumno. Comprobación:**

En primer lugar crearemos el directorio:

```
miadmin@PCB-USED:~$ sudo mkdir /var/www/html/public_html/profesor
[sudo] password for miadmin:
miadmin@PCB-USED:~$ ls /var/www/html/public_html
ciclos  fp  index.html  info.php  profesor
```

Ahora iremos al fichero de configuración del sitio:

```
miadmin@PCB-USED:~$ sudo nano /etc/apache2/sites-enabled/000-default.conf
```

Dentro del fichero introduciremos lo siguiente:

```
<Directory /var/www/html/public_html/profesor>
    Require host localhost
    Require ip 192.168.3.11
</Directory>
```

También tendremos que comprobar que se encuentra activado el módulo `mod_authz_host`, para ello haremos un listado de los módulos activos filtrando por la palabra `host`:

```
miadmin@PCB-USED:~$ ls /etc/apache2/mods-enabled/ | grep host
authz_host.load
```

Dado que nos aparece como activo, solamente nos queda reiniciar el servicio de apache



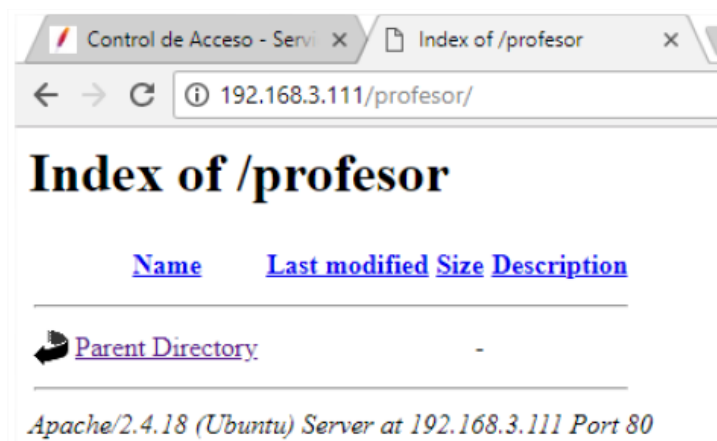
para que se apliquen los cambios del fichero del sitio:

```
miadmin@PCB-USED:~$ sudo service apache2 restart
```

### 1. En el propio servidor con el navegador lynx .

```
Index of /profesor
[ICO] [Name] [Last modified] [Size] [Description]
[PARENTDIR] Parent Directory -
Apache/2.4.18 (Ubuntu) Server at localhost Port 80
```

### 2. En el ordenador del alumno con un navegador.



(\*)Añadir el cambio de propietario para ese directorio.

**XII. Usando autenticación básica se pide controlar que solo pueden acceder /var/www/html/profesor a través de HTTP los usuarios: profesor1 y profesor2 (usuarios de apache). Comprobación desde el cliente que pide la autenticación para acceder a la carpeta profesor. Observar el intercambio de tramas entre el cliente y el servidor, y ver el metadato Authorization.**

En primer lugar vamos a comprobar que se encuentra activado el módulo mod\_authz\_basic, para ello haremos un listado de los módulos activos, filtrando por authz.

```
miadmin@PCB-USED:~$ ls /etc/apache2/mods-enabled/ | grep authz
authz_core.load
authz_host.load
authz_user.load
```

A continuación crearemos los usuarios, para ello haremos lo siguiente:

```
miadmin@PCB-USED:~$ sudo htpasswd -c /etc/apache2/passwd profesor1
New password:
Re-type new password:
Adding password for user profesor1

miadmin@PCB-USED:~$ sudo htpasswd /etc/apache2/passwd profesor2
New password:
Re-type new password:
Adding password for user profesor2
```

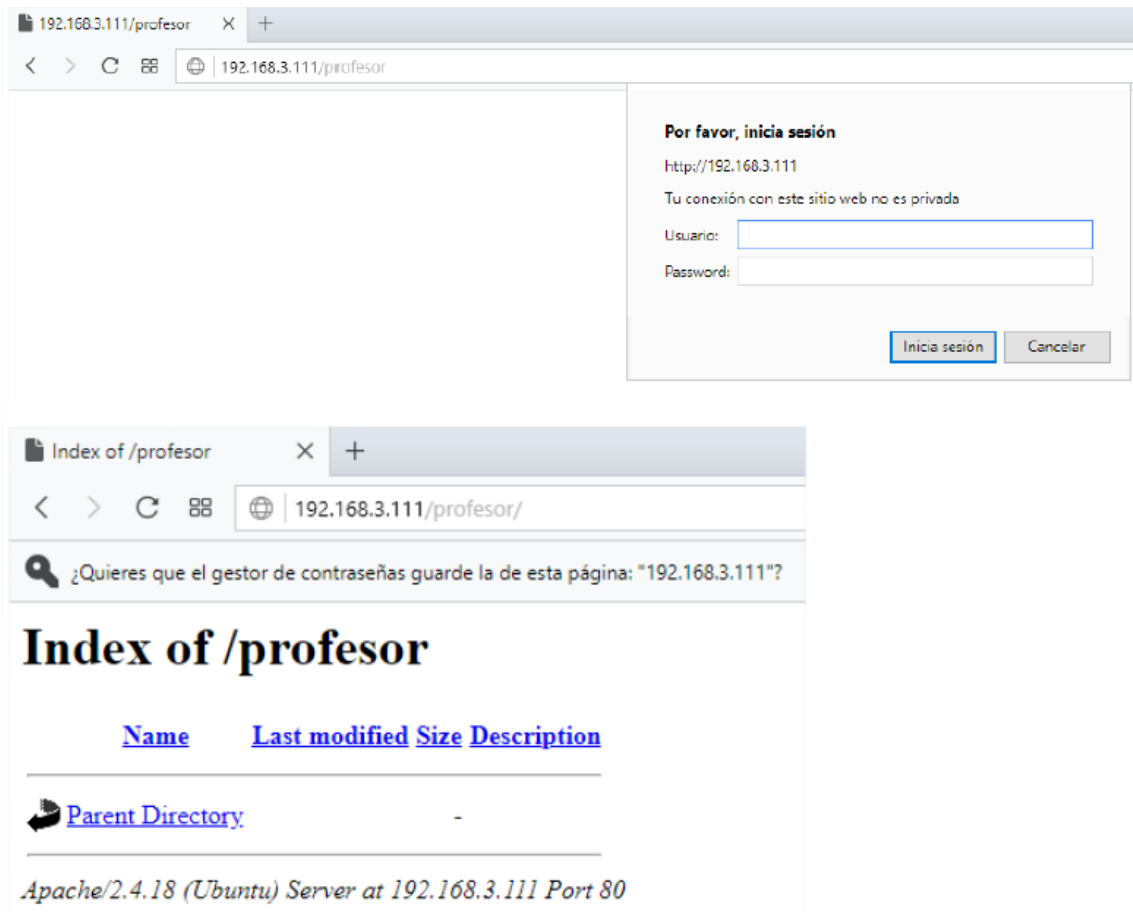
Una vez hecho comprobaremos el fichero:

```
miadmin@PCB-USED:~$ cat /etc/apache2/passwd
profesor1:$apr1$z9f.q/3.$ncWSLBW6av59KLFEN873B.
profesor2:$apr1$60MpKelN$Qa7B4EBRpaxt6l.WmMahR/
```

A continuación iremos al fichero `/etc/apache2/sites-enabled/000-default.conf`, donde crearemos las restricciones para ese directorio:

```
<Directory /var/www/html/public_html/profesor>
    Options FollowSymLinks Indexes
    AllowOverride None
    AuthType Basic
    AuthName "Acceso restringido"
    AuthUserFile /etc/apache2/passwd
    Require user profesor1 profesor2
    #Require host localhost
    #Require ip 192.168.3.11
</Directory>
```

Reiniciamos el servicio y comprobamos que funciona:



**XIII. Crea el directorio `/var/www/html/public_html/departamento` con un archivo `departamento.html`. Usando autenticación digest crear un fichero con los usuarios (`admin1` y `admin2`) y contraseñas asociadas al dominio (realm) `informatica`. Configurar las directivas asociadas al directorio `departamento` para el acceso a los usuarios `admin1` y `admin2`. Intenta acceder a `http://IP/departamento`. ¿es posible?**

En primer lugar vamos a comprobar que se encuentra activo el módulo `mod_authz_digest`, para ello haremos un listado filtrando por `digest`, dado que no aparece como activo lo buscamos en los disponibles:

```
miadmin@PCB-USED:~$ ls /etc/apache2/mods-available/ | grep digest
auth_digest.load
```

Activamos el módulo y reinicamos el servicio:

```
miadmin@PCB-USED:~$ sudo a2enmod auth_digest.load
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
miadmin@PCB-USED:~$ sudo service apache2 restart
```

Ahora crearemos los usuarios:

```
miadmin@PCB-USED:~$ sudo htdigest -c /var/www/html/public_html/.htdigest Admin admin1
Adding password for admin1 in realm Admin.
New password:
Re-type new password:
miadmin@PCB-USED:~$ sudo htdigest /var/www/html/public_html/.htdigest Admin admin2
Adding user admin2 in realm Admin
New password:
Re-type new password:
```

Una vez creados, iremos al fichero 000-default.conf, donde realizaremos la configuración para el directorio, el authname debe coincidir con el nombre que se da al dominio, indicado delante del nombre del usuario al ser creado:

```
<Directory /var/www/html/public_html/departamentos>
    Options FollowSymLinks Indexes
    AuthType Digest
    AuthName "Admin"
    AuthUserFile /var/www/html/public_html/.htdigest
    Require user admin1 admin2
</Directory>
```

Reiniciamos el servicio y comprobamos que funciona:

**Por favor, inicia sesión**

http://192.168.3.111

Tu conexión con este sitio web no es privada

Usuario:



Password:

Index of /departamentos

< > ↻ 🌐 192.168.3.111/departamentos/

🔍 ¿Quieres que el gestor de contraseñas guarde la de esta página: "192.168.3.111"?

## Index of /departamentos

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
 <a href="#">Parent Directory</a>		-	
 <a href="#">departamentos.html</a>	2018-04-20 09:52	1.1K	

Apache/2.4.18 (Ubuntu) Server at 192.168.3.111 Port 80

**XIV. Habilita en Apache el uso de ficheros de configuración personalizada de directorios (.htaccess). En el directorio /home/profesor/blog para que sea el propio usuario profesor el que pueda controlar como sirve Apache los contenidos. Para ello, se pide documentar las siguientes configuraciones.**

**1. Desde el sitio por defecto:**

a) Configura un alias /blog que permite acceder al directorio /home/profesor/blog

```
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/public_html
Alias /blog "/home/profesor/blog"
<Directory /home/profesor/blog>
    Options FollowSymLinks Indexes
    Require all granted
</Directory>
```

b) Habilita el uso del fichero .htaccess

Para habilitar el uso de ficheros .htaccess tendremos que ir al fichero de configuración del sitio por defecto, en donde añadiremos lo siguiente dentro de la directiva <Directory>.

```
<Directory /var/www/html/public_html>
    AllowOverride All
</Directory>
```

Finalmente guardamos cambios y reiniciamos el servicio de apache:

```
miadmin@PCB-USED:~$ sudo service apache2 restart
```

**2. Crea un usuario profesor.**

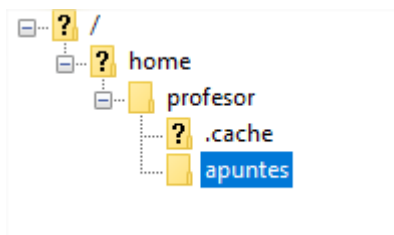
Creamos el usuario:

```
sudo adduser profesor
```

Comprobamos que se ha creado su home:

```
miadmin@PCB-USED:~$ ls /home
alumno  miadmin  profesor
miadmin@PCB-USED:~$
```

Vamos al filezilla y creamos el directorio:



Finalmente vamos al navegador y comprobamos que funciona:

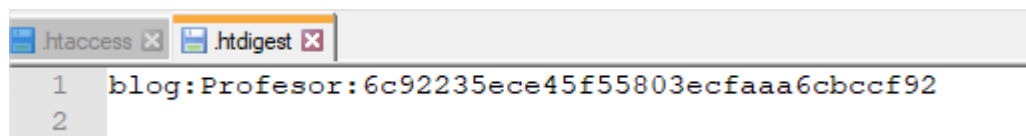
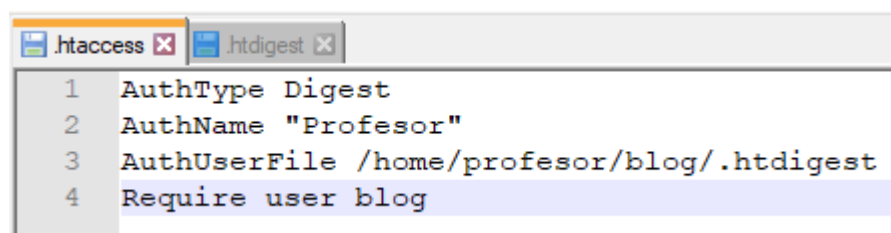


**3. Desde la máquina anfitriona sin loggarte en el servidor (a través de FTP/SFTP con Notepad++ o cliente FTP/SFTP) mediante el usuario profesor (es necesario crearlo) haz uso del fichero .htaccess y configura el directorio /home/profesor/blog**

Vamos al fichero de configuración del sitio por defecto, y para ese directorio añadiremos la siguiente directiva:

```
<Directory /home/profesor/blog>
    AllowOverride All
    Options FollowSymLinks Indexes
    Require all granted
</Directory>
```

Vamos al Filezilla, y crearemos los ficheros .htaccess y .htdigest, para el último tendremos que acceder a un generador de htdigest para rellenar el fichero:



Vamos al navegador y comprobamos:

Iniciar sesión

http://192.168.3.111

Tu conexión con este sitio web no es privada

Nombre de usuario

blog

Contraseña

....

Iniciar sesión

Cancelar

## Index of /blog

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
 <a href="#">Parent Directory</a>		-	

Apache/2.4.18 (Ubuntu) Server at 192.168.3.111 Port 80

## XV. Módulo status

**1. En el servidor web debe estar habilitado el módulo `mod_status` (`status`) y configura para que se conecte desde cualquier equipo de la red del aula.**

Realizamos un listado del directorio de los módulos activos buscando “status”, para comprobar que el módulo ya se encuentra activo:

```
miadmin@PCB-USED:~$ ls /etc/apache2/mods-enabled/ | grep status
status.conf
status.load
```

Dado que no nos permite el acceso desde el cliente, tendremos que realizar la configuración del mismo para que nos permita el acceso, para ello vamos al fichero de configuración del archivo que se encuentra en `/etc/apache2/mods-enabled/status.conf`. Dentro de este fichero comentaremos la siguiente línea:

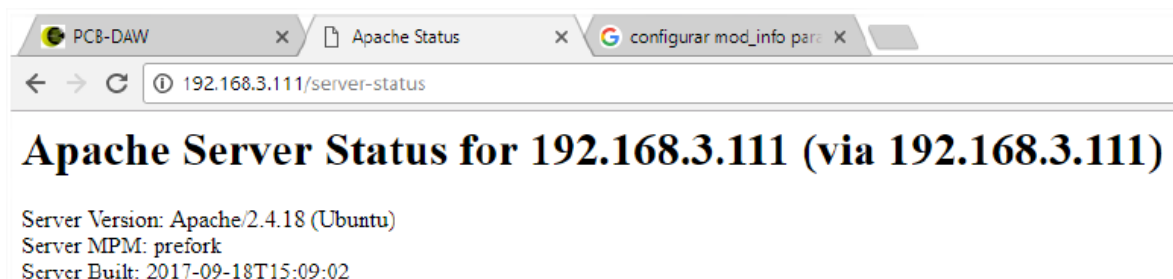
```
<Location /server-status>
    SetHandler server-status
    #Require local
    #Require ip 192.0.2.0/24
</Location>
```

Una vez editado, guardamos los cambios y reiniciamos el servicio:

```
miadmin@PCB-USED:~$ sudo service apache2 restart
```



## 2. Comprobación: <http://IP/server-status>.



### XVI. Módulo info: Visualizar la vista resumida de la configuración del servidor.

#### 1. En el servidor web debe estar habilitado el módulo `mod_info` (`info`) y restringir el acceso desde un solo equipo (anfitrión).

En primer lugar vamos a ver los módulos que están disponibles, buscando aquellos que contengan 'info':

```
miadmin@PCB-USED:~$ ls /etc/apache2/mods-available/ | grep info
info.conf
info.load
```

Una vez encontrado, tendremos que activarlo, para ello haremos lo siguiente:

```
miadmin@PCB-USED:~$ sudo a2enmod info.load
[sudo] password for miadmin:
Enabling module info.
To activate the new configuration, you need to run:
    service apache2 restart
```

Por último reiniciamos el servicio de apache:

```
miadmin@PCB-USED:~$ sudo service apache2 restart
```

A continuación vamos al fichero de configuración del módulo y comentamos la siguiente línea, de modo que permita el acceso desde cualquier equipo. Una vez hecho reiniciamos el servicio de apache:

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

## 2. Comprobación <http://IP/server-info>

Vamos al navegador y comprobamos que podemos acceder:

