



Ejercicio 1. Recorre las carpetas del servidor observando qué se guarda en ellas. Necesitarás acceder al archivo httpd.conf para modificar la configuración de Apache.

```
victor@victor-K56CB:~/httpd$ ls
bin build cgi-bin conf error htdocs icons include lib logs man manual modules
```

~/httpd/bin: ficheros ejecutables binarios.
~/httpd/build: ficheros de compilación para generar el servidor web.
~/httpd/cgi-bin: programas que se ejecutan en el servidor como pasarela.
~/httpd/conf: ficheros de configuración.
~/httpd/error: códigos de error en HTML.
~/httpd/htdocs: carpeta por defecto del servidor web.
~/httpd/icons: iconos.
~/httpd/include: dependencias.
~/httpd/lib: librerías compiladas.
~/httpd/logs: ficheros de registro.
~/httpd/man: manuales.
~/httpd/manual: manuales en HTML.
~/httpd/modules: módulos.

Ejercicio 2. Inicia y para el servidor Apache con apachectl. Recuerda que tendrías que llamar a [apachectl restart](#) cada vez que hagas un cambio en la configuración.

```
victor@victor-K56CB:~/httpd/bin$ sudo ./apachectl start
victor@victor-K56CB:~/httpd/bin$ sudo ./apachectl stop
```

IMPORTANTE: para evitar que Apache siga ejecutándose al salir de nuestra sesión, es obligatorio que añadáis a [\\$HOME/.bash_logout](#) la línea killall -s KILL httpd.

[Esto ya lo hace el script anterior.](#)

```
1 # ~/.bash_logout: executed by bash(1) when login shell exits.-
2 -
3 # when leaving the console clear the screen to increase privacy-
4 -
5 if [ "$SHLVL" = 1 ]; then-
6     [ -x /usr/bin/clear_console ] && /usr/bin/clear_console -q-
7 fi-
8 killall -s KILL httpd-
```



Ejercicio 3. Por defecto, Apache escucha por el puerto 80. Comprueba si esto produce algún error e investiga cuál es el motivo. Modifica el puerto de escucha para que sea el 8080. Comprueba que funciona con un navegador, accediendo a localhost: 8080.

```
victor@victor-K56CB:~/httpd/bin$ ./apachectl start
(13)Permission denied: AH00072: make_sock: could not bind to address [::]:80
(13)Permission denied: AH00072: make_sock: could not bind to address 0.0.0.0:80
    no listening sockets available, shutting down
AH00015: Unable to open logs
```

Al realizar la escucha por el puerto 80 se produce un error, esto se debe a que este puerto ya está ocupado por defecto y no se nos permite hacer la escucha a este mismo. Esto es lo que nos aparece al acceder al puerto por defecto (localhost:80):



Apache2 Ubuntu Default Page

ubuntu

It works!

Para cambiar el puerto de escucha modificamos el fichero httpd.conf del directorio ~/httpd/conf:

```
43  #
44  # Listen: Allows you to bind Apache to specific IP addresses and/or
45  # ports, instead of the default. See also the <VirtualHost>
46  # directive.
47  #
48  # Change this to Listen on specific IP addresses as shown below to
49  # prevent Apache from glomming onto all bound IP addresses.
50  #
51  #Listen 12.34.56.78:8080
52  Listen 8080
```

Tras modificar el puerto de escucha por el 8080 y reiniciar el servidor web, accedemos desde el navegador a localhost: 8080 y la escucha es satisfactoria:



It works!



Ejercicio 4. Los archivos que sirve Apache (los que puedo ver desde el cliente) están en una carpeta concreta. Cambia dicha carpeta (DocumentRoot), para que sea \$HOME/html-docs/. Crear la carpeta y copiar dentro un conjunto de archivos .html para tu servidor web, dándole como nombre a uno de ellos index.html e introduciendo en él tu nombre y apellidos. Si no sabes crear ficheros HTML, puedes obtener algunos en http://www.w3schools.com/html/html_examples.asp.

```
209  #  
210  # DocumentRoot: The directory out of which you will serve your  
211  # documents. By default, all requests are taken from this directory, but  
212  # symbolic links and aliases may be used to point to other locations.  
213  #  
214  DocumentRoot "/home/victor/html-docs"  
215  <Directory "/home/victor/html-docs">
```

Creamos el directorio “~/html-docs” y lo establecemos como DocumentRoot en el fichero de configuración ~/httpd/conf/httpd.conf.

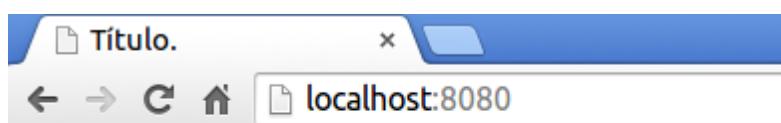
Ahora generamos un fichero index.html e introducimos el siguiente código en el fichero:

```
1  <html>  
2    <head>  
3      <meta charset="utf-8">  
4      <title>Título.</title>  
5    </head>  
6    <body>  
7      <h1>Victor Monserrat Villatoro.</h1>  
8    </body>  
9  </html>
```

Lo incluimos en la carpeta anterior (~html-docs) y reiniciamos el servidor:

```
victor@victor-K56CB:~/html-docs$ ls  
index.html
```

Y comprobamos su correcto funcionamiento accediendo a él desde el navegador:



Víctor Monserrat Villatoro.



Ejercicio 5. Cambia el nombre de los archivos que se utilizarán por Apache como archivos por defecto cuando solo se especifique el directorio (normalmente es index.html, añadir como posibilidad index.htm, probar si funciona, ¿qué prioridad se utiliza si tenemos más de un nombre de archivo?).

```
243  #  
244  # DirectoryIndex: sets the file that Apache will serve if a directory  
245  # is requested.  
246  #  
247  <IfModule dir_module>  
248  |   DirectoryIndex index.html index.htm  
249  </IfModule>
```

Modificamos el fichero de configuración de nuevo. La prioridad que utiliza si tenemos más de un nombre de archivo es respecto al orden en el que se encuentran en el fichero.

Ejercicio 6. ¿Qué opción de qué directiva es la encargada de permitir mostrar el contenido de un directorio aunque éste no contenga ninguno de los archivos índices explicados en la directiva anterior? Crea un directorio nuevo \$HOME/html-docs/prueba (es obligatorio especificar una nueva directiva [Directory](#)), incluye un archivo cualquiera y prueba a utilizar esta opción para prohibir o no el listado de archivos en dicha carpeta. Para probarlo tendrás que acceder a localhost:8080/prueba.

```
247  <Directory "/home/victor/html-docs/prueba/">  
248  |   Options Indexes FollowSymLinks  
249  |   AllowOverride None  
250  |   Require all granted  
251  </Directory>
```

Creamos el directorio prueba y añadimos dos ficheros (prueba1.html y prueba2.html) al directorio. Añadimos al fichero de configuración lo anterior. Al reiniciar el servidor, nos muestra la lista de directorios dentro del cual, como se observa en la siguiente imagen.



← → ⌂ ⌂ localhost:8080/prueba/

Index of /prueba

- [Parent Directory](#)
- [prueba1.htm](#)
- [prueba2.htm](#)

Para no permitir el listado de directorios debemos cambiar la opción +Indexes por -Indexes.

```
247 <Directory "/home/victor/html-docs/prueba/">-
248     Options -Indexes +FollowSymLinks-
249     AllowOverride None-
250     Require all granted-
251 </Directory>-
```

De esta manera, al intentar acceder al directorio prueba no se nos listan los directorios ya que no tenemos permisos como se observa abajo.

localhost:8080/prueba/

Forbidden

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

¿Cómo podríamos mejorar el aspecto visual del listado? (pistas, [Indexes](#) y [httpd-autoindex.conf](#)).

```
462 # Fancy directory listings-
463 Include conf/extrahtpd-autoindex.conf-
```

Para personalizar el listado de directorios basta con descomentar la línea que incluye httpd-autoindex.conf, en este caso la 463. Reiniciamos el servidor y tras volver a acceder al directorio prueba se nos lista de la siguiente forma:



Index of /prueba

	Name	Last modified	Size	Description
	Parent Directory		-	
	prueba1.htm	2015-05-04 17:39	150	
	prueba2.htm	2015-05-04 17:39	150	

Ejercicio 7. Encuentra la directiva que especifica el nombre del servidor e introduce su valor correcto.

```
183 #
184 # ServerName gives the name and port that the server uses to identify itself.-
185 # This can often be determined automatically, but we recommend you specify-
186 # it explicitly to prevent problems during startup.-
187 #
188 # If your host doesn't have a registered DNS name, enter its IP address here.-
189 #
190 ServerName 192.168.1.2:80
```

Especificamos el nombre del servidor en el fichero de configuración, para ello vamos a la directiva `ServerName` y escribimos el nombre del servidor. En este caso, no tenemos nombre de dominio asociado con lo que pondremos nuestra IP.

Ejercicio 8. Encuentra la directiva que especifica el usuario y grupo para el demonio `httpd` (en vuestro caso, ¿sirve para algo cambiarla?).

```
150 <IfModule unixd_module>
151 #
152 # If you wish httpd to run as a different user or group, you must run-
153 # httpd as root initially and it will switch.-
154 #
155 # User/Group: The name (or #number) of the user/group to run httpd as.-
156 # It is usually good practice to create a dedicated user and group for-
157 # running httpd, as with most system services.-
158 #
159 User daemon-
160 Group daemon-
```



Las directivas User y Group indican al servidor los usuarios y grupos a los que responderá peticiones. En nuestro caso no sirve de nada ya que estamos ejecutando apache desde un usuario no root.

Ejercicio 9. Prueba a hacer un telnet al puerto del servidor (telnet localhost 8080) y a mandarle mensajes HTML (escribe “GET / HTTP/1.0” y pulsa dos veces intro). Prueba a escribir HOLA y pulsar dos veces a Intro. Explica qué sucede y los códigos de error que devuelve el servidor.

```
victor@victor-K56CB:~$ telnet localhost 8080
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
```

Una vez conectados al servidor local, enviamos un mensaje HTML:

```
GET / HTTP/1.0

HTTP/1.1 200 OK
Date: Sat, 16 May 2015 11:51:19 GMT
Server: Apache/2.4.12 (Unix)
Last-Modified: Mon, 04 May 2015 15:39:05 GMT
ETag: "96-515435cb3fa17"
Accept-Ranges: bytes
Content-Length: 150
Connection: close
Content-Type: text/html

<html>
  <head>
    <meta charset="utf-8">
    <title>Título.</title>
  </head>
  <body>
    <h1>Victor Monserrat Villatoro.</h1>
  </body>
</html>
```

El servidor nos devuelve el código HTML del fichero index.html.

A continuación, enviamos el mensaje HOLA al servidor. Este nos devuelve el error 501, que significa que el método HOLA no está implementado.

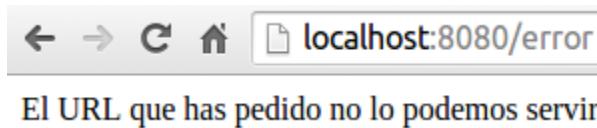


```
HOLA
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>501 Not Implemented</title>
</head><body>
<h1>Not Implemented</h1>
<p>HOLA to /index.htm not supported.<br />
</p>
</body></html>
```

Ejercicio 10. Encuentra la directiva que se utiliza para la visualización de páginas de error. Modifícalo para personalizar el mensaje de error 404 que mostrará el servidor y que sea una cadena de tipo “El URI que has pedido no lo podemos servir”. ¿Se podría especificar un fichero .html de error?. Modifica el error 501 para que muestre “Método no implementado” y comprueba que funciona.

```
408  #
409  # Customizable error responses come in three flavors:
410  # 1) plain text 2) local redirects 3) external redirects
411  #
412  # Some examples:
413  #ErrorDocument 500 "The server made a boo boo."
414  ErrorDocument 501 "Método no implementado"
415  ErrorDocument 404 "El URL que has pedido no lo podemos servir"
416  #ErrorDocument 404 "/cgi-bin/missing_handler.pl"
417  #ErrorDocument 402 http://www.example.com/subscription_info.html
418  #
```

La directiva que se utiliza para la visualización de páginas de error es ErrorDocument. A continuación indicamos los códigos de error 404 y el 501 personalizados y reiniciamos el servidor.



Al intentar acceder a un directorio no existente, el servidor nos da error 404, es decir, que no ha podido encontrar lo que se le ha solicitado y nos muestra el mensaje de error que hemos especificado. Como podemos ver abajo, al enviarle al servidor el método METODO, el cual no existe, nos muestra el mensaje de error con código 501. Todos estos mensajes, pueden ser también HTML.



```
victor@victor-K56CB:~/httpd/bin$ telnet localhost 8080
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
METODO
Método no implementadoConnection closed by foreign host.
```

Ejercicio 11. Identifica las directivas relacionadas con los archivos de *logs* de Apache. Haz un acceso normal y acceso erróneo (por ejemplo, pidiendo un archivo o directorio que no existe). Comprueba los *logs* y muestra cómo se han modificado.

```
259  #
260  # ErrorLog: The location of the error log file.
261  # If you do not specify an ErrorLog directive within a <VirtualHost>
262  # container, error messages relating to that virtual host will be
263  # logged here. If you *do* define an error logfile for a <VirtualHost>
264  # container, that host's errors will be logged there and not here.
265  #
266  ErrorLog "logs/error_log"
```

La directiva para el log de errores es `ErrorLog` y para el log de accesos es `CustomLog`.

```
288  ....#
289  ....# The location and format of the access logfile (Common Logfile Format).
290  ....# If you do not define any access logfiles within a <VirtualHost>
291  ....# container, they will be logged here. Contrariwise, if you *do*
292  ....# define per-<VirtualHost> access logfiles, transactions will be
293  ....# logged therein and *not* in this file.
294  ....#
295  ....CustomLog "logs/access_log" common
```

A continuación hacemos un acceso normal y vemos que modificación ha tenido el fichero `access_log`.

```
127.0.0.1 - - [16/May/2015:15:33:17 +0200] "GET / HTTP/1.0" 200 150
```

Ahora comprobamos con un acceso erróneo el fichero `error_log`.

```
[Sat May 16 15:34:27.495497 2015] [core:error] [pid 3227:tid 140541686470400] [client 127.0.0.1:45560] AH00135: Invalid method in request ERROR
```



Ejercicio 12. Redirecciona la dirección /uco a www.uco.es, de manera que al acceder a localhost:8080/uco aparezca la web de la Universidad de Córdoba.

```
304 <IfModule alias_module>
305   ...
306   # Redirect: Allows you to tell clients about documents that used to
307   # exist in your server's namespace, but do not anymore. The client
308   # will make a new request for the document at its new location.
309   # Example:
310   Redirect permanent /uco http://www.uco.es/
```

La directiva para la redirección de rutas es `Redirect`, a la que le indicamos la ruta que queremos dirigir y el destino de redirección. De esta forma, al acceder al directorio `/uco`, se nos redirigirá automáticamente a la web de la Universidad de Córdoba como podemos ver.



Ejercicio 13. Crea un *Host* virtual de manera que cuando un cliente se conecte al servidor usando `localhost:8080` el servidor muestre la carpeta raíz original, y cuando se conecte usando `IPMAQUINA:8080` muestre la subcarpeta `/prueba`.
`IPMAQUINA` es la IP de la máquina en la que estás, puedes averiguarla con `ifconfig`. Haz que tengan ficheros de *log* independientes (`local-access.log`, `local-error.log`, `ip-access.log` y `ip-error.log`).

```
47 <VirtualHost 192.168.1.2>
48   DocumentRoot /home/victor/html-docs/prueba/
49   ServerName 192.168.1.2
50   CustomLog logs/ip-access.log combined
51   ErrorLog logs/ip-error.log
52 </VirtualHost>
```



Creamos un VirtualHost con la IP de la máquina y le indicamos que muestre la carpeta prueba y los logs de acceso y error.

Name	Last modified	Size	Description
prueba1.htm	2015-05-04 17:39	150	
prueba2.htm	2015-05-04 17:39	150	

Como se puede observar, al reiniciar el servidor, accediendo con la IP se listan los ficheros del directorio prueba.

```
276 ErrorLog "logs/local-error.log"
```

Y por último, modificamos los ficheros logs de accesos y error.

```
305 ... CustomLog "logs/logs/local-access.log" common
```

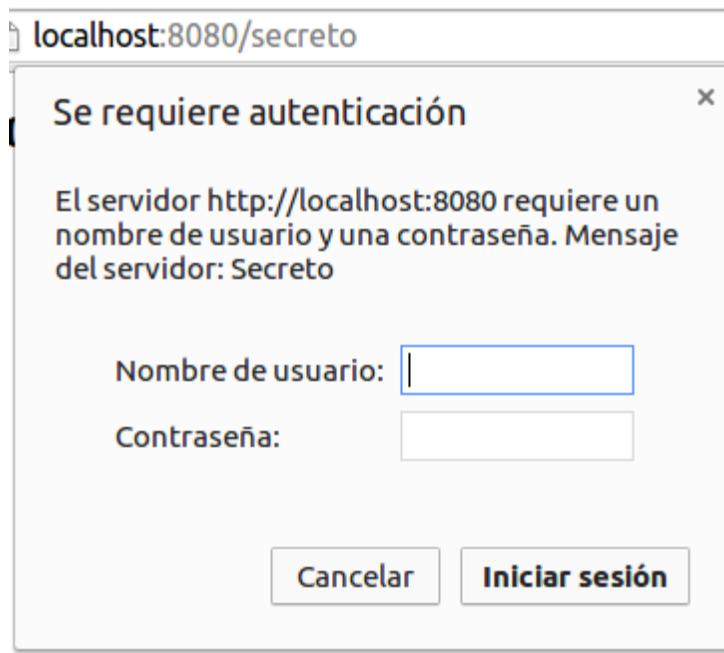
Ejercicio 14. Apache permite el acceso a recursos restringidos mediante la creación de usuarios y grupos que deberán autenticarse antes de acceder a dicho recurso protegido. Se debe:

1. crear los usuarios y contraseñas correspondientes dentro de un archivo llamado .htpasswd (para esto, tendrás que utilizar el comando `./htpasswd -c .htpasswd usuario`, incluido en Apache, una vez por cada usuario a añadir y luego copiar el archivo resultante a la carpeta que quieras proteger; ojo, el -c indica que el fichero se cree nuevo, por lo que sólo debe utilizarse para el primer usuario)



```
victor@victor-K56CB:~/httpd/bin$ sudo ./htpasswd -c .htpasswd profesor
New password:
Re-type new password:
Adding password for user profesor
victor@victor-K56CB:~/httpd/bin$ sudo ./htpasswd .htpasswd alumno1
New password:
Re-type new password:
Adding password for user alumno1
victor@victor-K56CB:~/httpd/bin$ sudo ./htpasswd .htpasswd alumno2
New password:
Re-type new password:
Adding password for user alumno2
```

2. e incluir un archivo .htaccess (por defecto, aunque estos nombres de archivo pueden modificarse en httpd.conf) con los usuarios o grupos de usuarios que tendrán acceso. La sintaxis de ese archivo es la misma que el resto de configuraciones de directorios de Apache.



Debes crear un directorio, que se llamará secreto, de forma que, para acceder a él, habrá que autenticarse. Los usuarios que tendrán acceso a ese directorio serán: profesor, alumno1 y alumno2; y la contraseña, para todos será: quieroentrar. Explica los pasos realizados para conseguirlo, y el contenido de los archivos creados.

NOTAS: para que el archivo .htaccess se interprete por Apache es necesario que, previamente, hayamos activado la directiva AllowOverride All sobre el directorio correspondiente (por defecto está a None) en httpd.conf. Consultar el ejemplo de



autenticación en

<http://www.cristalab.com/tutoriales/proteger-carpetas-con-.htaccess-y-.htpasswd-c213l/>.

Ejercicio 15. Existe también la posibilidad de permitir o denegar el acceso a diferentes directorios o archivos dependiendo de la dirección IP del cliente. Para ello, disponemos de las directivas Allow from, Deny from y Order, que pueden utilizarse en el archivo de configuración httpd.conf o en cada uno de los directorios mediante el archivo .htaccess. Prueba esta opción de seguridad para permitir el acceso a nuestro servidor web, únicamente para direcciones IP de la subred de los equipos de sobremesa del laboratorio. Prueba también a denegar el acceso a todas las direcciones IP.

NOTA: para especificar una subred se debe utilizar la siguiente notación **172.16.215.0/24**, donde **24** es el número de bits de la subred (tres primeras cifras) y la subred sería **172.16.215**. Averigua cuál es la subred del laboratorio con el comando ifconfig.

```
247 <Directory "/home/victor/html-docs/prueba/">
248     Options Indexes FollowSymLinks
249     Deny from 192.168.1.0/24
250     AllowOverride None
251     Require all granted
252 </Directory>
```

Añadimos al directorio prueba la directiva Deny from con la IP de la máquina y el servidor no nos permite el acceso a través de la IP como se puede observar a continuación.

← → ⌂ ⌂ 192.168.1.2:8080/prueba

Forbidden

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