

Práctica 4

Apache

Darío Martínez Kostyuk

Índice

1 Explica qué hace cada una de las líneas del bash script de instalación de Apache. Puedes colocar los comentarios dentro del propio archivo.....	4
2 Recorre las carpetas del servidor observando qué se guarda en ellas. Necesitarás acceder al archivo httpd.conf para modificar la configuración de Apache.....	5
3 Si quiero que el servidor web Apache escuche en otro puerto, por ejemplo, el 8080, ¿que opción del archivo configuración debo modificar?	6
4 Nos bajamos un archivo index.html de Internet y lo colocamos en la raíz. Comprobar que ahora se muestra este archivo en lugar del index.html.....	7
5 Añadir como posibilidad index.htm y probar si funciona. ¿Qué prioridad se utiliza si tenemos más de un archivo índice?.....	8
6 ¿Qué opción de qué directiva es la encargada de permitir mostrar el contenido de un directorio aunque éste no contenga ningún archivo índice? (pistas: Directory y Options Indexes..).....	9
7 Cambia dicha carpeta (DocumentRoot), para que sea la carpeta creada antes nuevo_root.....	10
8 Encuentra la directiva que especifica el usuario y grupo para la ejecución de Apache. ¿Cuáles son y para qué sirven?.....	10
9 Haz un telnet y escribe "GET / HTTP/1.0" y prueba a escribir HOLA. Explica qué sucede y los códigos de error que devuelve el servidor.....	11
10 Modifica la directiva para mostrar una página HTML personalizada para el error 404. Busca en Internet algún diseño llamativo.....	13
11 Identifica las directivas relacionadas con los archivos de logs de Apache.....	14
12 Redirecciona la dirección localhost/uco a https://www.uco.es, de manera que al acceder a localhost/uco aparezca la web de la Universidad de Córdoba.....	16
13 Crea dos Hosts virtuales.....	16
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.....	18

1 Explica qué hace cada una de las líneas del bash script de instalación de Apache. Puedes colocar los comentarios dentro del propio archivo.

Este es el script de instalación con los comentarios añadidos:

```
#!/bin/bash
```

```
#Creamos directorio temporal de instalación en el home
```

```
mkdir $HOME/tmp1
```

```
cd $HOME/tmp1
```

```
# Descargamos y decomprimos Apache 2.4.63
```

```
wget -O httpd-2.4.63.tar.bz2
```

```
https://dlcdn.apache.org/httpd/httpd-2.4.63.tar.bz2
```

```
tar jxvf httpd-2.4.63.tar.bz2
```

```
# Descargamos y descomprimos las librerías APR
```

```
wget -O apr-1.6.5.tar.bz2 https://dlcdn.apache.org/apr/apr-1.6.5.tar.bz2
```

```
wget -O apr-util-1.6.3.tar.bz2 https://dlcdn.apache.org/apr/apr-util-1.6.3.tar.bz2
```

```
tar jxvf apr-1.6.5.tar.bz2
```

```
tar jxvf apr-util-1.6.3.tar.bz2
```

```
# Instalamos el compilador gcc y paquetes esenciales
```

```
sudo apt-get update
```

```
sudo apt-get install build-essential
```

Movemos las librerías APR a la carpeta de apache, para que éste las utilice

```
mv apr-1.6.5 ./httpd-2.4.63/src/lib/apr
```

```
mv apr-util-1.6.3 ./httpd-2.4.63/src/lib/apr-util
```

Descargamos la librería pcre

```
sudo apt-get install libpcre3 libpcre3-dev
```

Descargamos la librería expat

```
sudo apt-get install libexpat1-dev
```

Compilamos apache e instalamos apache

```
cd httpd-2.4.63
```

```
./configure --prefix=$HOME/httpd --with-included-apr
```

```
sudo make -j4
```

```
sudo make install
```

Borramos el directorio temporal

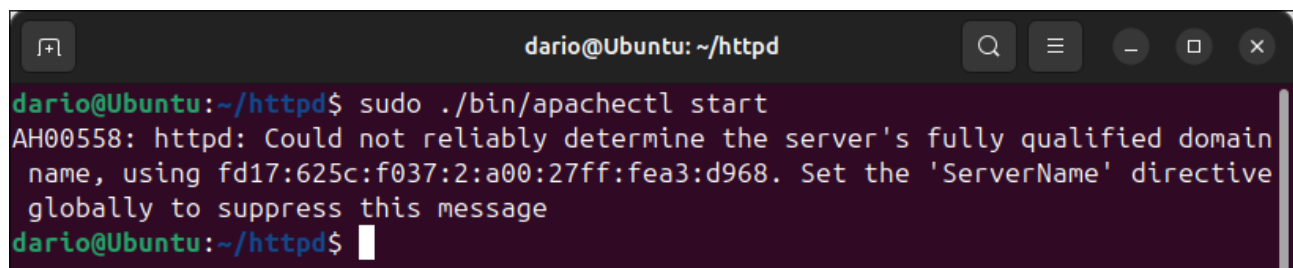
```
sudo rm -rf $HOME/tmp1
```

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

Efectivamente podemos comprobar que tras ejecutar el script de instalación se nos ha creado ~/httpd donde podemos encontrar todas las carpetas de apache y su configuración.

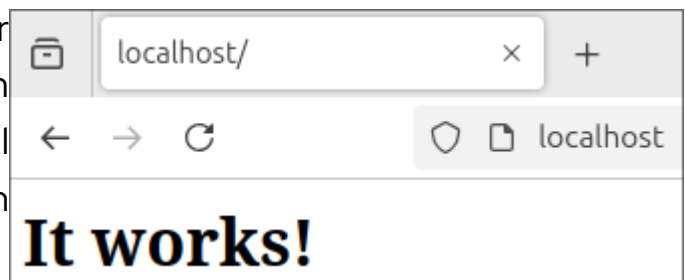
3 Si has instalado el servidor Apache en Ubuntu con el script de Moodle, inicia y para el servidor Apache utilizando el ejecutable `apachectl` (carpeta `/bin`) con las opciones `start` y `stop`. Recuerda que tendrás que ejecutar `apachectl restart` cada vez que hagas un cambio en la configuración. Abre el navegador con `http://localhost` y captura la pantalla. Si quiero que el servidor web Apache escuche en otro puerto, por ejemplo, el 8080, ¿que opción del archivo configuración debo modificar? Haga captura de pantalla de `http://localhost:8080`.

Tras ejecutar y el `apachectl` con `start` y entrar en `localhost` podemos ver lo siguiente:



```
dario@Ubuntu: ~/httpd
dario@Ubuntu:~/httpd$ sudo ./bin/apachectl start
AH00558: httpd: Could not reliably determine the server's fully qualified domain
name, using fd17:625c:f037:2:a00:27ff:fea3:d968. Set the 'ServerName' directive
globally to suppress this message
dario@Ubuntu:~/httpd$
```

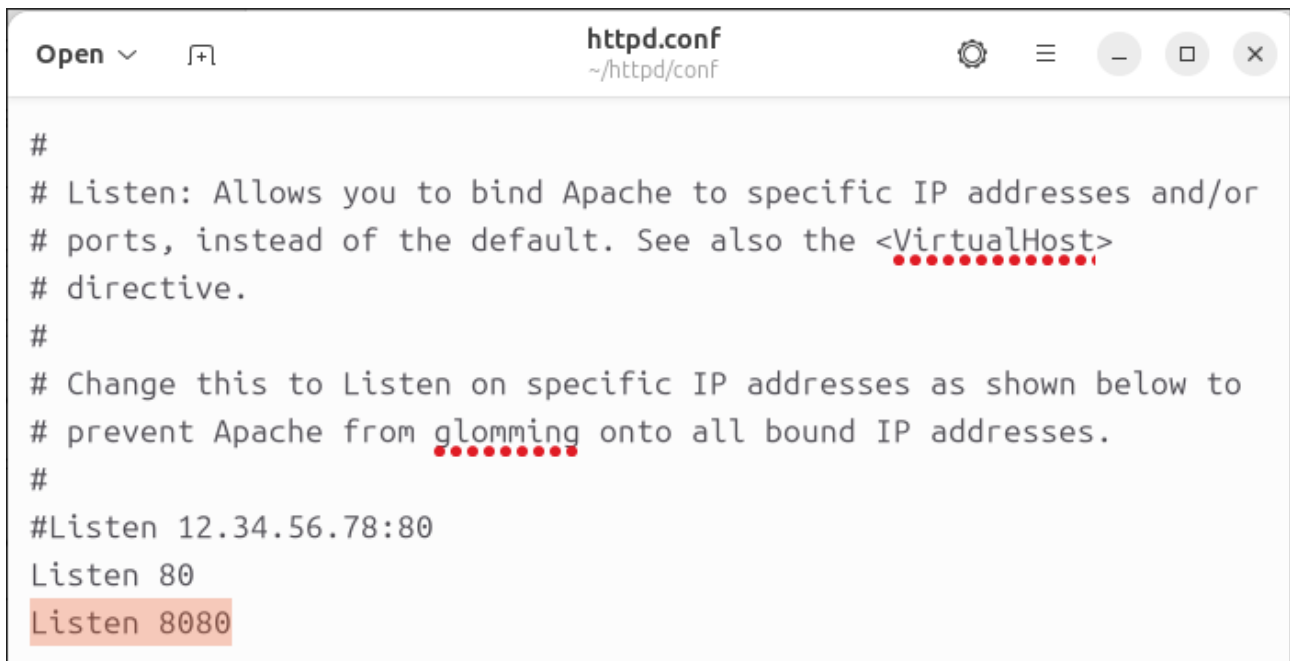
Como podemos comprobar esta funcionando correctamente, en `localhost` aparece el `index.html` generado por defecto en `~/httpd/htdocs`.



Sin embargo, si entramos en el puerto 8080 nos sale un error ya que tenemos que hacer que pueda escuchar el puerto:

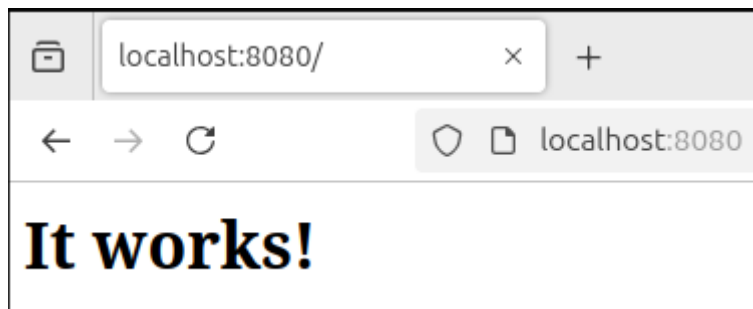


Para hacer esto tenemos que añadir al archivo de configuración el `Listen 8080`, como se ve en la siguiente imagen:



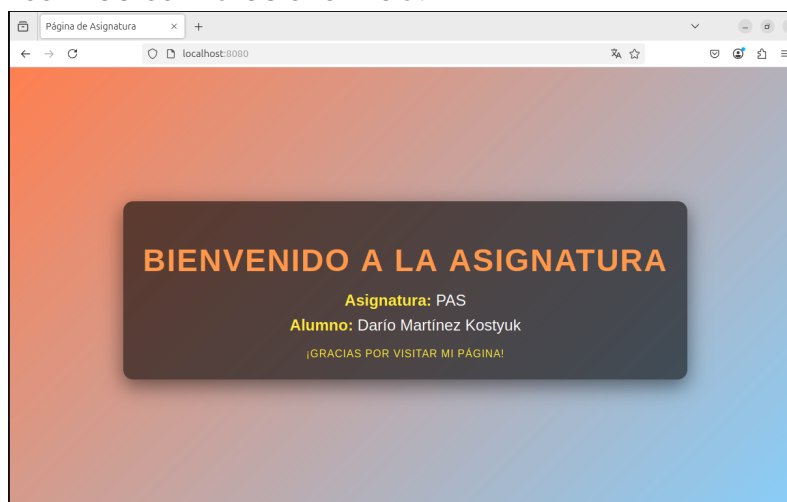
```
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 80
Listen 8080
```

Y una vez reiniciado lo tenemos hecho:



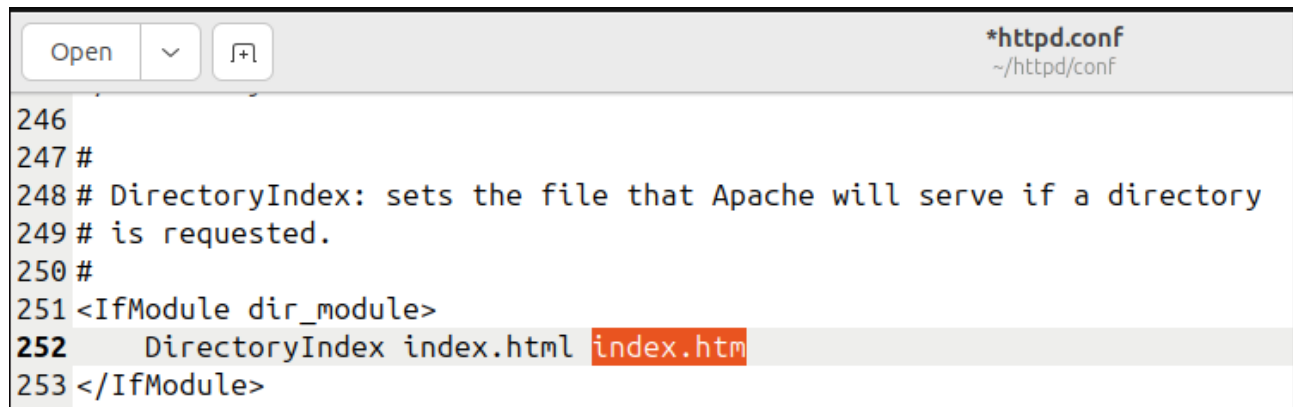
4 La carpeta raíz del servidor Apache es /htdocs. Esta es la carpeta donde primero busca Apache el archivo index.html. Nos bajamos un archivo index.html de Internet y lo colocamos en la raíz. Comprobar que ahora se muestra este archivo en lugar del index.html. Puedes buscar en Internet archivos básicos de HTML (Mirad lo que se puede hacer con HTML5: <https://www.tutorialrepublic.com/html-examples.php>).

Como podemos comprobar, con cambiar el index.html y recargar el localhost se aplican los cambios a la web:



5 Cambia el nombre de los archivos índice (archivos que servirá Apache cuando el cliente sólo especifique un directorio). Normalmente es index.html, añadir como posibilidad index.htm y probar si funciona. ¿Qué prioridad se utiliza si tenemos más de un archivo índice?

Para añadirlo tenemos que modificar el DirectoryIndex para añadir las opciones que queremos:



```
246
247 #
248 # DirectoryIndex: sets the file that Apache will serve if a directory
249 # is requested.
250 #
251 <IfModule dir_module>
252     DirectoryIndex index.html index.htm
253 </IfModule>
```

Una vez añadido reiniciamos el apache y cambiamos el nombre del index.html a index.htm y como podemos comprobar en el localhost sigue exactamente igual que antes:



Para la prioridad se recorren en orden los nombres que aparecen en la directiva DirectoryIndex y esa es la prioridad de uso.

6 ¿Qué opción de qué directiva es la encargada de permitir mostrar el contenido de un directorio aunque éste no contenga ningún archivo índice? Crea un nuevo directorio llamado `new_root` dentro de la carpeta `/htdocs`, incluye un archivo cualquiera (que no sea `index.html`) y una imagen y prueba a utilizar esta opción para prohibir o no el listado de archivos en dicha carpeta. Para probarlo tendrás que acceder a `localhost/new_root`. Agrega la Descripción "Cuadro de mando" al archivo de imagen que está en la carpeta (pistas: `Directory` y `Options Indexes`..)

Creamos dentro de `htdocs` la carpeta y añadimos un `.txt` y una imagen `.jpg`:

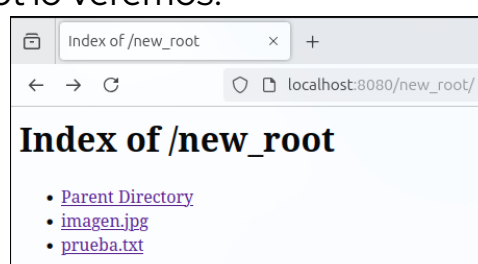
```
dario@Ubuntu: ~/httpd/htdocs/new_root
dario@Ubuntu:~/httpd/htdocs$ mkdir new_root
dario@Ubuntu:~/httpd/htdocs$ cd new_root/
dario@Ubuntu:~/httpd/htdocs/new_root$ echo "Prueba de archivo" >> prueba.txt
dario@Ubuntu:~/httpd/htdocs/new_root$ cp ../../
bin/      cgi-bin/  error/    icons/   lib/      man/      modules/
build/    conf/      htdocs/  include/ logs/     manual/
dario@Ubuntu:~/httpd/htdocs/new_root$ cp ../../../../Do
Documents/ Downloads/
dario@Ubuntu:~/httpd/htdocs/new_root$ cp ../../../../Downloads/servidores_rack.jpg
imagen.jpg
dario@Ubuntu:~/httpd/htdocs/new_root$ ls
imagen.jpg  prueba.txt
```

Ahora vamos a configurar Apache para permitir el listado y añadir una descripción a la imagen:

```
*httpd.conf
~/httpd/conf

244     Require all granted
245 </Directory>
246
247 <Directory "/home/dario/httpd/htdocs/new_root">
248     Options +Indexes
249     AddDescription "Cuadro de mando" imagen.jpg
250 </Directory>
```

Una vez hecho esto podemos reiniciar apache y al entrar a `localhost:8080/new_root` lo veremos:



7 Los archivos que sirve Apache (los que puedo ver desde el cliente) están en una carpeta concreta. Cambia dicha carpeta (DocumentRoot), para que sea la carpeta creada antes nuevo_root. 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 curso y tu asignatura preferida. Puedes buscar en Internet archivos básicos html.

Creamos un index.html en new_root y cambiamos el DocumentRoot en la configuración al new_root que acabamos de crear

Reiniciamos apache y al entrar al localhost podremos ver el index.html del new_root:



8 Encuentra la directiva que especifica el usuario y grupo para la ejecución de Apache. ¿Cuáles son y para qué sirven?

Se encuentra dentro de esta directiva y se especifica mediante la línea User y Group:



```
153
154 <IfModule unixd_module>
155 #
156 # If you wish httpd to run as a different user or group, you must run
157 # httpd as root initially and it will switch.
158 #
159 # User/Group: The name (or #number) of the user/group to run httpd as.
160 # It is usually good practice to create a dedicated user and group for
161 # running httpd, as with most system services.
162 #
163 User daemon
164 Group daemon
165
166 </IfModule>
```

Indican el usuario y grupo del sistema bajo los cuales corren los procesos hijos de Apache. Esto permite que, tras arrancar con root, Apache pase a un usuario menos privilegiado y controle el acceso a archivos y directorios según los permisos Unix.

9 Prueba a hacer un telnet al puerto del servidor (telnet localhost 80) y a mandarle mensajes HTML (escribe "GET / HTTP/1.0" y pulsa dos veces intro). Prueba a escribir HOLA y pulsar dos veces intro. Explica qué sucede y los códigos de error que devuelve el servidor.

Al hacer el telnet y poner el GET / HTTP/1.0 podemos ver que pone un HTTP/1.1 OK significando que efectivamente todo salió bien, así que busco el / es decir el index.html y lo devolvió con éxito:

```
dario@Ubuntu: ~/httpd/htdocs/new_root
dario@Ubuntu:~/httpd/htdocs/new_root$ telnet localhost 80
Trying ::1...
Connected to localhost.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.1 200 OK
Date: Fri, 23 May 2025 17:32:54 GMT
Server: Apache/2.4.63 (Unix)
Last-Modified: Fri, 23 May 2025 16:05:50 GMT
ETag: "1ac-635cfc5a100c0"
Accept-Ranges: bytes
Content-Length: 428
Connection: close
Content-Type: text/html

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Mi Curso Favorito</title>
</head>
<body>
  <h1>Bienvenidos a la página de mi curso</h1>
  <p>Mi asignatura preferida es <strong>Programación y administración de Sistemas</strong>.</p>
  <p>Aquí aprenderemos a programar y administrar sistemas xd</p>
</body>
</html>
Connection closed by foreign host.
```

Sin embargo cuando ponemos algo no reconocido como la palabra "HOLA" se nos devuelve un error 400 Bad Request

```
dario@Ubuntu: ~/httpd
dario@Ubuntu:~/httpd$ telnet localhost 80
Trying ::1...
Connected to localhost.
Escape character is '^]'.
HOLA

HTTP/1.1 400 Bad Request
Date: Sat, 24 May 2025 10:04:01 GMT
Server: Apache/2.4.63 (Unix)
Content-Length: 226
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
</body></html>
Connection closed by foreign host.
```

Con esto podemos comprobar que los codigos de success (Todo salió bien) son los 200 mientras que los 400 son de error bad request.

10 Encuentra la directiva que se utiliza para la visualización de páginas de error y describe cada uno de los tipos de errores posibles. Modifica la directiva para mostrar una página HTML personalizada para el error 404. Busca en Internet algún diseño llamativo.

Estos son los tipos de errores posibles:

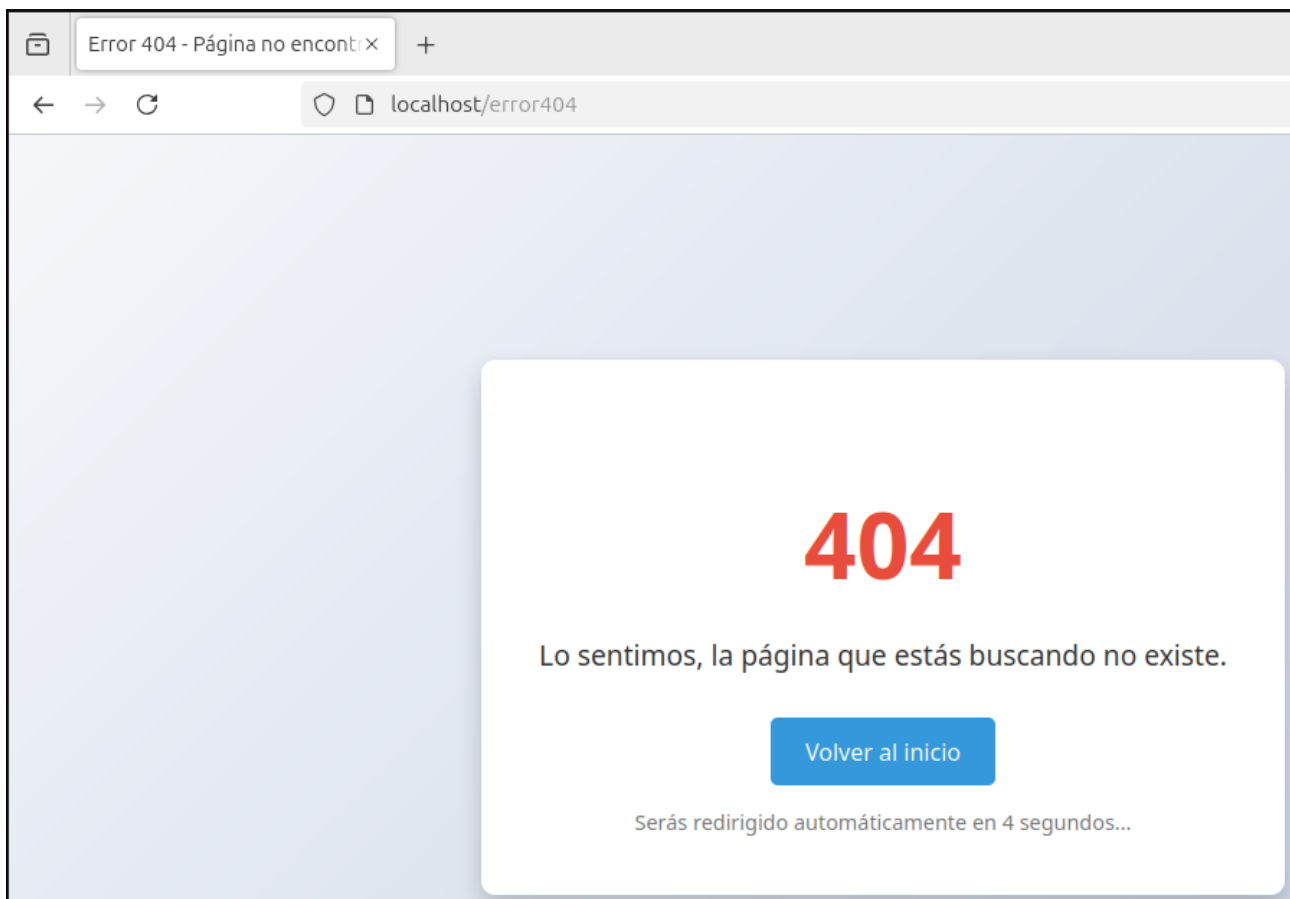
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
500	Internal Server Error
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout

Para poner una página HTML personalizada para el error 404 tenemos que añadir a la configuración la directiva `ErrorDocument`, como en la imagen:



```
427 # Customizable error responses come in three flavors:
428 # 1) plain text 2) local redirects 3) external redirects
429 #
430 # Some examples:
431 #ErrorDocument 500 "The server made a boo boo."
432 #ErrorDocument 404 /missing.html
433 #ErrorDocument 404 "/cgi-bin/missing_handler.pl"
434 #ErrorDocument 402 http://www.example.com/subscription_info.html
435 #
436 ErrorDocument 404 /missing.html
```

Y creamos el `missing.html` dentro de `htdocs/new_root` para que lo pueda leer correctamente y al entrar a un enlace no definido para que nos salga un error 404 podemos ver que efectivamente funciona:

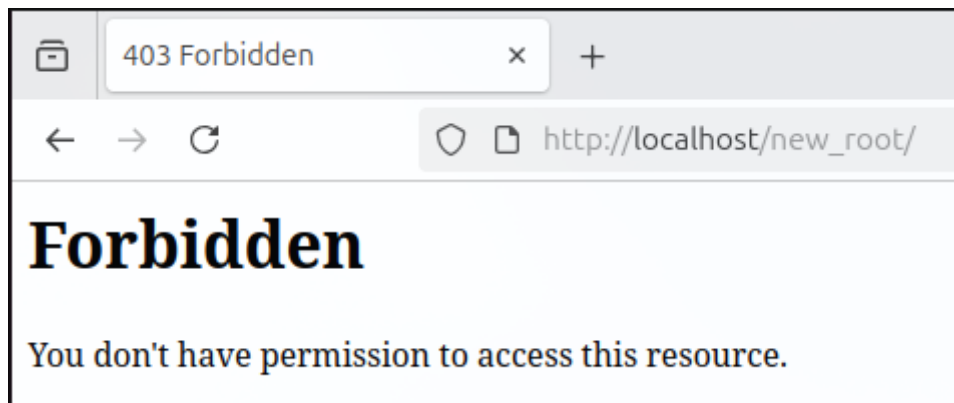


11 Identifica las directivas relacionadas con los archivos de logs de Apache. Para ello desactiva las Options en un directorio del ejercicio 6, para que muestre el error 403 de Acceso Prohibido. Comprueba los logs y muestra las nuevas líneas que se han agregado. Reinicia apache y comprueba de nuevo si se ha incluido alguna línea y en qué archivos de log. Confeccione una tabla con los principales errores que se pueden dar en un servidor HTTP como Apache.

Para hacer un error 403 vamos a cambiar en el directory el options a none y vamos a renombrar el index.html a otro nombre para que no lo detecte

```
Open  v  [+]  
httpd.conf  
~/httpd/conf  
246  
247 <Directory "/home/dario/httpd/htdocs/new_root">  
248     Options None  
249     AddDescription "Cuadro de mando" imagen.jpg  
250 </Directory>  
251
```

Ahora en el momento en el que intentamos acceder al localhost/new_root nos aparece:



Los logs gracias al script de instalación se nos guardan en httpd/logs y hay 3 archivos (access_log, error_log y httpd.pid) si ahora hacemos un tail de las ultimas lineas de access_log y error_log podremos observar este 403:

```
dario@Ubuntu: ~/httpd
dario@Ubuntu:~/httpd$ tail -n 10 logs/error_log
[Sat May 24 11:55:52.018704 2025] [mpm_event:notice] [pid 4964:tid 4964] AH00489:
Apache/2.4.63 (Unix) configured -- resuming normal operations
[Sat May 24 11:55:52.018882 2025] [core:notice] [pid 4964:tid 4964] AH00094: Comm
and line: '/home/dario/httpd/bin/httpd'
[Sat May 24 11:56:30.777474 2025] [autoindex:error] [pid 4967:tid 4977] [client 1
27.0.0.1:36674] AH01276: Cannot serve directory /home/dario/httpd/htdocs/new_root
/: No matching DirectoryIndex (index.html,index.htm) found, and server-generated
directory index forbidden by Options directive
[Sat May 24 11:56:31.590741 2025] [autoindex:error] [pid 4967:tid 4981] [client 1
27.0.0.1:36674] AH01276: Cannot serve directory /home/dario/httpd/htdocs/new_root
/: No matching DirectoryIndex (index.html,index.htm) found, and server-generated
directory index forbidden by Options directive
[Sat May 24 12:04:15.686425 2025] [mpm_event:notice] [pid 4964:tid 4964] AH00494:
SIGHUP received. Attempting to restart
AH00558: httpd: Could not reliably determine the server's fully qualified domain
name, using fd17:625c:f037:2:62f:6a61:81b2:3530. Set the 'ServerName' directive g
lobally to suppress this message
[Sat May 24 12:04:15.738155 2025] [mpm_event:notice] [pid 4964:tid 4964] AH00489:
Apache/2.4.63 (Unix) configured -- resuming normal operations
[Sat May 24 12:04:15.738185 2025] [core:notice] [pid 4964:tid 4964] AH00094: Comm
and line: '/home/dario/httpd/bin/httpd'
[Sat May 24 12:04:27.053257 2025] [autoindex:error] [pid 5317:tid 5335] [client 1
27.0.0.1:49046] AH01276: Cannot serve directory /home/dario/httpd/htdocs/new_root
/: No matching DirectoryIndex (index.html,index.htm) found, and server-generated
directory index forbidden by Options directive
[Sat May 24 12:04:28.355778 2025] [autoindex:error] [pid 5317:tid 5337] [client 1
27.0.0.1:49046] AH01276: Cannot serve directory /home/dario/httpd/htdocs/new_root
/: No matching DirectoryIndex (index.html,index.htm) found, and server-generated
directory index forbidden by Options directive
```

```
dario@Ubuntu: ~/httpd
dario@Ubuntu:~/httpd$ tail -n 10 logs/access_log
127.0.0.1 - - [24/May/2025:10:51:28 +0000] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [24/May/2025:10:51:29 +0000] "GET / HTTP/1.1" 200 1969
127.0.0.1 - - [24/May/2025:10:51:29 +0000] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [24/May/2025:11:55:16 +0000] "GET / HTTP/1.1" 304 -
127.0.0.1 - - [24/May/2025:11:55:33 +0000] "GET /new_root HTTP/1.1" 301 234
127.0.0.1 - - [24/May/2025:11:55:33 +0000] "GET /new_root/ HTTP/1.1" 200 428
127.0.0.1 - - [24/May/2025:11:56:30 +0000] "GET /new_root/ HTTP/1.1" 403 199
127.0.0.1 - - [24/May/2025:11:56:31 +0000] "GET /new_root/ HTTP/1.1" 403 199
127.0.0.1 - - [24/May/2025:12:04:27 +0000] "GET /new_root/ HTTP/1.1" 403 199
127.0.0.1 - - [24/May/2025:12:04:28 +0000] "GET /new_root/ HTTP/1.1" 403 199
```


Los principales errores que pueden salir los he hecho en el ejercicio anterior en una tabla

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

Para hacer esto tenemos que añadir la opción redirect como he realizado en la siguiente imagen:



The screenshot shows a text editor window titled '*httpd.conf' with the path '~/.httpd/conf'. The editor contains the following lines of code:

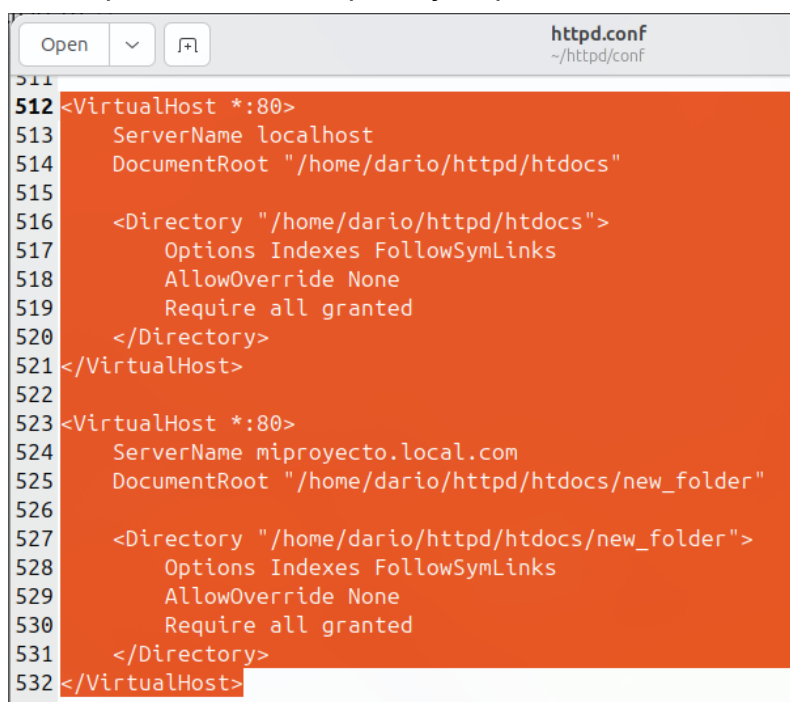
```
312
313 <IfModule alias_module>
314     #
315     # Redirect: Allows you to tell clients about documents that used to
316     # exist in your server's namespace, but do not anymore. The client
317     # will make a new request for the document at its new location.
318     # Example:
319     # Redirect permanent /foo http://www.example.com/bar
320     Redirect "/uco" "https://www.uco.es"
```

The line 320 is highlighted in orange.

Una vez reiniciado apache funcionará correctamente

13 Crea dos Hosts virtuales de manera que cuando un cliente se conecte al servidor usando localhost el servidor muestre la carpeta raíz original, y cuando se conecte usando miproyecto.local.com muestre el index.html de la subcarpeta /new_folder.

Para hacer esto hay varios pasos, primero de todo tenemos que añadir los virtual hosts al httpd.conf, como por ejemplo:



The screenshot shows a text editor window titled 'httpd.conf' with the path '~/.httpd/conf'. The editor contains the following lines of code:

```
511
512 <VirtualHost *:80>
513     ServerName localhost
514     DocumentRoot "/home/dario/httpd/htdocs"
515
516     <Directory "/home/dario/httpd/htdocs">
517         Options Indexes FollowSymLinks
518         AllowOverride None
519         Require all granted
520     </Directory>
521 </VirtualHost>
522
523 <VirtualHost *:80>
524     ServerName miproyecto.local.com
525     DocumentRoot "/home/dario/httpd/htdocs/new_folder"
526
527     <Directory "/home/dario/httpd/htdocs/new_folder">
528         Options Indexes FollowSymLinks
529         AllowOverride None
530         Require all granted
531     </Directory>
532 </VirtualHost>
```

The entire code block is highlighted in orange.

Una vez hecho esto tenemos que reiniciar apache y modificar /etc/hosts y añadir miproyecto.local.com

```
dario@Ubuntu: ~/httpd
GNU nano 7.2 /etc/hosts
127.0.0.1    localhost.localdomain localhost
::1         localhost6.localdomain6 localhost6
127.0.0.1    miproyecto.local.com
```

Ademas, tenemos que crear la carpeta new_folder y poner un index.html dentro

```
dario@Ubuntu: ~/httpd
dario@Ubuntu:~/httpd$ mkdir htdocs/new_folder
dario@Ubuntu:~/httpd$ cp htdocs/new_root/
imagen.jpg  index.ht  missing.html  prueba.txt
dario@Ubuntu:~/httpd$ cp htdocs/new_root/index.ht htdocs/new_folder/index.html
dario@Ubuntu:~/httpd$ nano htdocs/new_folder/index.html
```

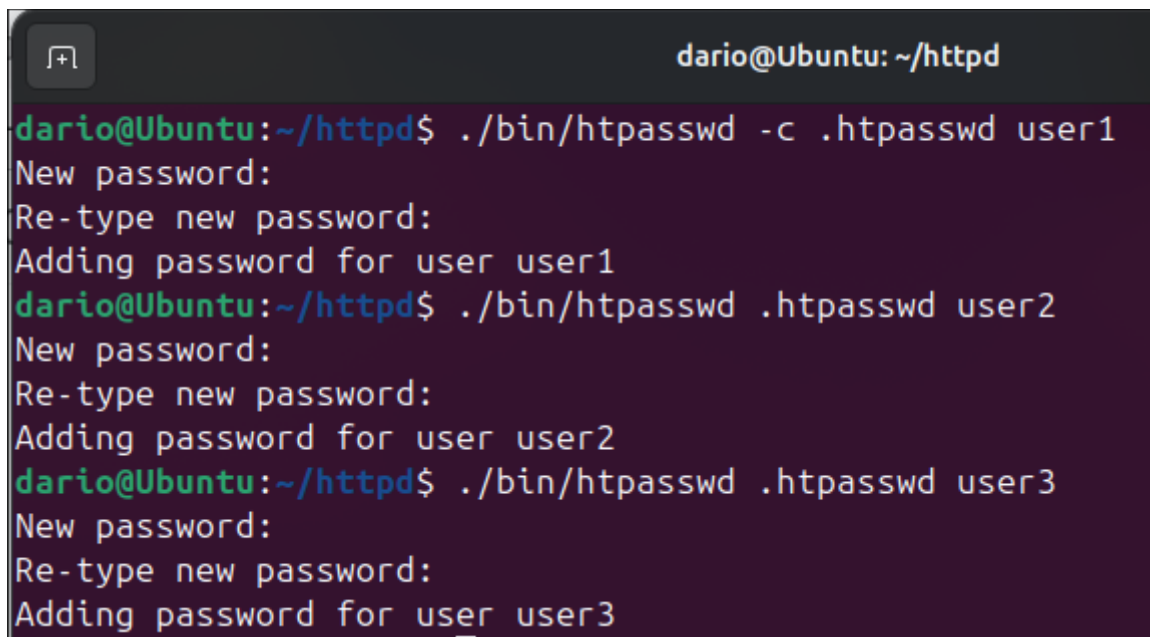
Y como podemos ver, ahora depende de la URL que coloquemos en el buscador, ya sea localhost o miproyecto.local.com saldrá un index.html u otro:



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:

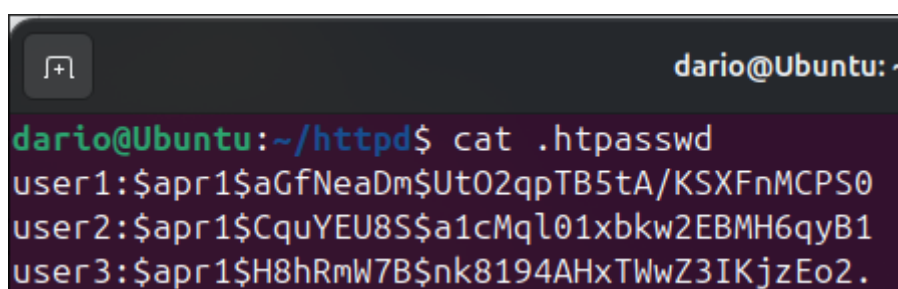
- crear los usuarios y contraseñas correspondientes dentro de un archivo llamado `.htpasswd` (para esto, tendrás que utilizar el comando `./htpasswd -c .htpasswd usuario`, incluido en la carpeta `bin` de Apache, una vez por cada usuario a añadir y luego copiar el archivo resultante a un sitio protegido; ojo, el `-c` indica que el fichero se cree nuevo, por lo que sólo debe utilizarse para el primer usuario, los demás usuarios con `-b`)
- Aplicar las restricciones de acceso anteriores a la carpeta `new_folder` del host virtual `miproyecto.local.com`. Los usuarios que tendrán acceso a ese directorio serán: `user1`, `user2` y `user3`; y la contraseña, para todos será: `entrar`. Explica los pasos realizados para conseguirlo, y el contenido de los archivos creados.

Vamos a crear los usuarios con `htpasswd`:



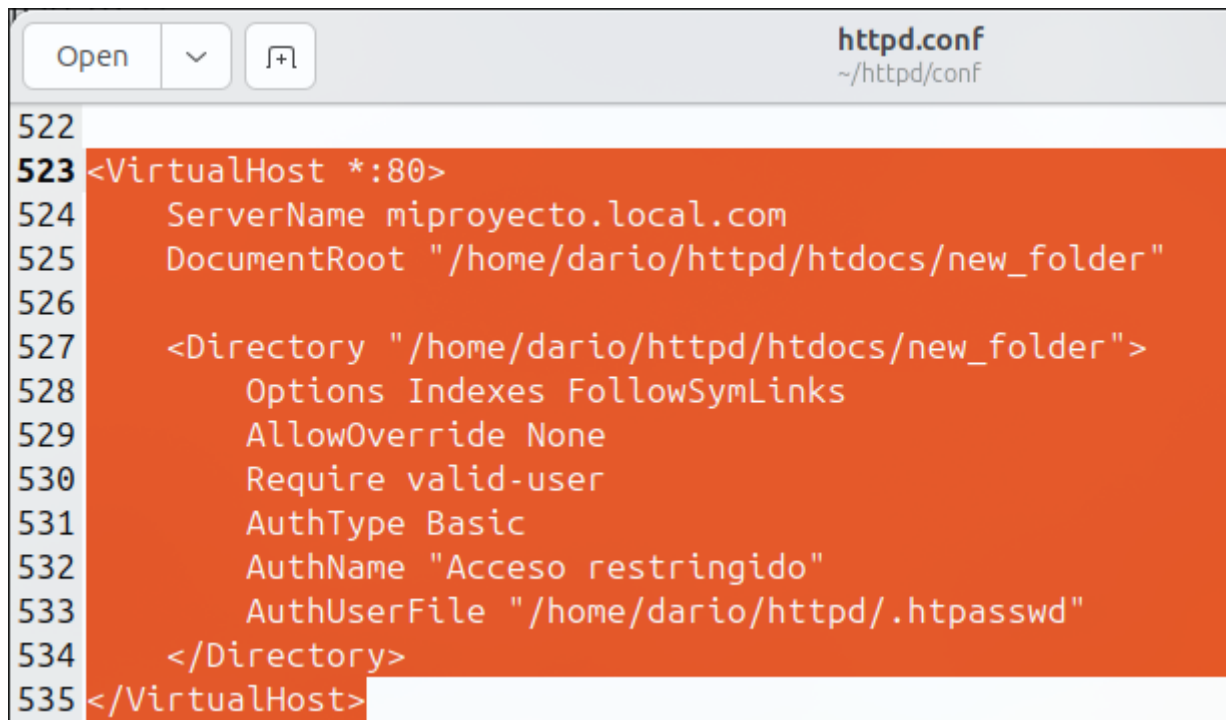
```
dario@Ubuntu: ~/httpd
dario@Ubuntu:~/httpd$ ./bin/htpasswd -c .htpasswd user1
New password:
Re-type new password:
Adding password for user user1
dario@Ubuntu:~/httpd$ ./bin/htpasswd .htpasswd user2
New password:
Re-type new password:
Adding password for user user2
dario@Ubuntu:~/httpd$ ./bin/htpasswd .htpasswd user3
New password:
Re-type new password:
Adding password for user user3
```

Con esto ya tenemos creados los 3 usuarios como podemos comprobar si miramos en `.htpasswd`:



```
dario@Ubuntu: ~/httpd$ cat .htpasswd
user1:$apr1$aGfNeaDm$Ut02qpTB5tA/KSXFmMCPS0
user2:$apr1$CquYEU8S$a1cMql01xbkw2EBMH6qyB1
user3:$apr1$H8hRmW7B$nk8194AHxTWwZ3IKjzEo2.
```

Ahora tenemos que modificar la configuración de apache para que cuando entremos en miproyecto.local.com sea necesaria autenticación



```
522
523 <VirtualHost *:80>
524     ServerName miproyecto.local.com
525     DocumentRoot "/home/dario/httpd/htdocs/new_folder"
526
527     <Directory "/home/dario/httpd/htdocs/new_folder">
528         Options Indexes FollowSymLinks
529         AllowOverride None
530         Require valid-user
531         AuthType Basic
532         AuthName "Acceso restringido"
533         AuthUserFile "/home/dario/httpd/.htpasswd"
534     </Directory>
535 </VirtualHost>
```

Tras reiniciar apache si queremos entrar en miproyecto.local.com nos saltara un popup pidiendonos que inciemos sesión y en el momento que iniciemos sesión con un perfil de los que hemos creado antes nos dejará acceder sin problema.

