

Práctica 1: Apache

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

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

El archivo *httpd.conf* se encuentra en el directorio */home/httpd/conf/httpd.conf*.

2. Inicia y para el servidor Apache con *apachectl*. Recuerda que tendrás que llamar a *apachectl restart* cada vez que hagas un cambio en la configuración.

```
alex@ThinkPAlex:~/httpd/bin$ ls
ab                apu-1-config      dbmmanage         fcgid            htdigest          httxt2dbm
apachectl         apxs              envvars           htcacheclean    httpd             logresolve
apr-1-config      checkgid          envvars-std       htdbm            httpd             rotatelogs
alex@ThinkPAlex:~/httpd/bin$ sudo ./apachectl start
httpd (pid 10643) already running
```

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*.

En el puerto 80 no funciona porque ese puerto es el asignado al protocolo http. Cambiándolo al 8080:

```
51 #Listen 12.34.56.78:80
52 Listen 8080
```

Al acceder con un navegador:

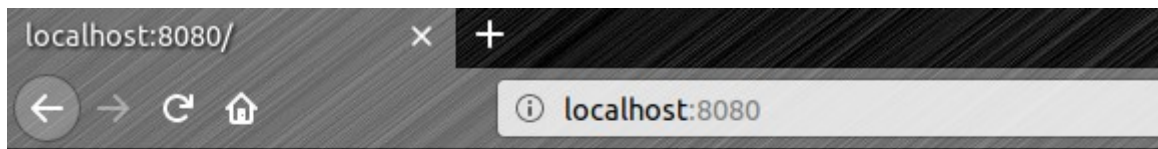


It works!

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 */*. 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.

```
DocumentRoot "/home/alex/httpd-docs"
<Directory "/home/alex/httpd-docs">
```

Al acceder con un navegador:



Manuel Alejandro Luque León

Mi asignatura favorita sin lugar a dudas es PAS.

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 index2.html y probar si funciona. ¿Qué prioridad se utiliza si tenemos más de un archivo índice?

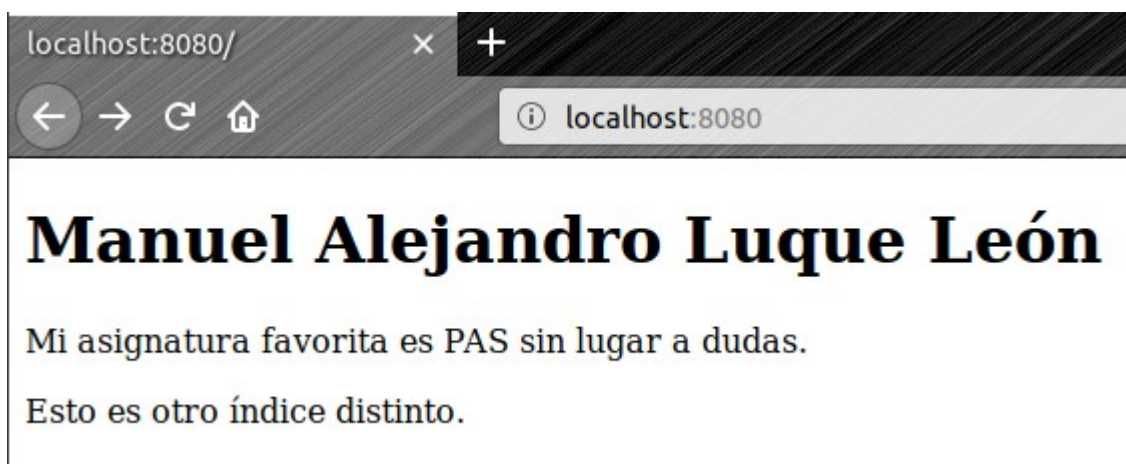
Al añadir más posibles índices:

```
alex@ThinkPAlex:~/httpd-docs$ ls
index2.html  index.html
```

Y añadir estos al httpd.conf:

```
<IfModule dir_module>
#DirectoryIndex index.html
DirectoryIndex index2.html
</IfModule>
```

Se dará más prioridad al que esté antes en el archivo httpd.conf. En este caso se ha comentado el primero para poder observar que el nuevo índice funciona correctamente:



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 directorio nuevo `$HOME/httpd-docs/newFolder` (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/newFolder`. ¿Cómo podríamos mejorar el aspecto visual del listado? (pistas, *Indexes* y *httpd-autoindex.conf*).

Esta directiva nos permite listar el contenido de un directorio:

```
<Directory "/home/alex/httpd-docs/newFolder/">  
    Options +Indexes  
</Directory>
```

Al entrar con el navegador podemos comprobarlo:



Al modificar la directiva anterior:

```
<Directory "/home/alex/httpd-docs/newFolder/">  
    Options -Indexes  
</Directory>
```

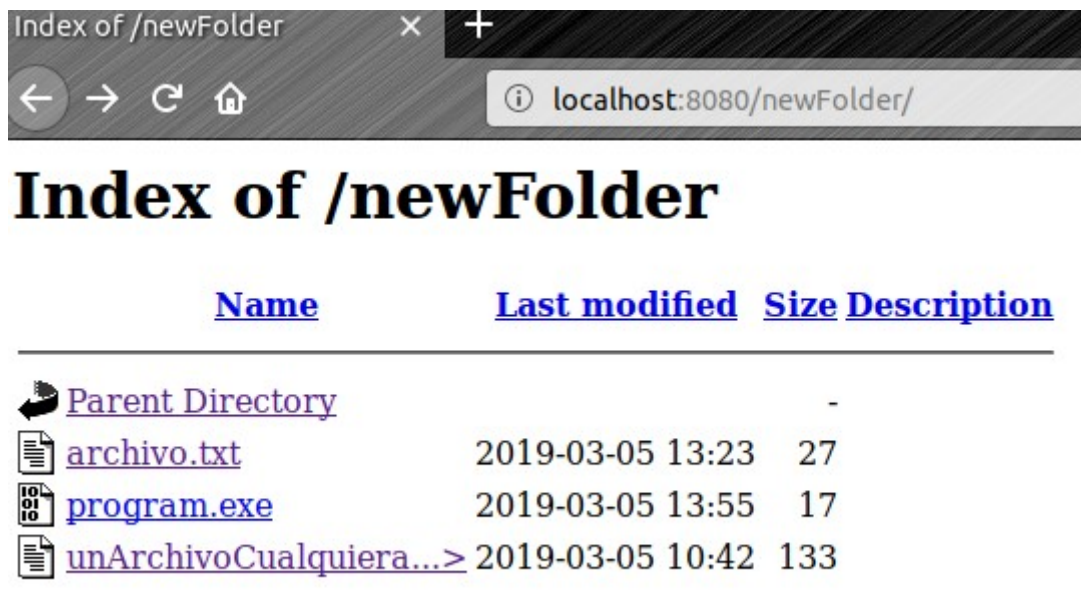
Comprobando su funcionamiento con el navegador:



Si se habilita la opción:

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

Y lo comprobarlo con el navegador podemos observar como el más atractivo el listado de los archivos:



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

```
ServerName localhost
```

8. Encuentra la directiva que especifica el usuario y grupo para el demonio *httpd*. ¿Sirve para algo cambiarla?

```
User daemon
Group daemon
```

Esto permite elegir los permisos que tiene el servidor. En este caso se está ejecutando como un demonio. Si se asignaran otros, se le asignarían los permisos asignados a ese usuario y grupo.

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 intro. Explica qué sucede y los códigos de error que devuelve el servidor.

```
alex@ThinkPAlex:~/httpd-docs/newFolder$ telnet localhost 8080
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.1 200 OK
Date: Tue, 05 Mar 2019 14:42:40 GMT
Server: Apache/2.4.38 (Unix)
Last-Modified: Mon, 04 Mar 2019 18:05:06 GMT
ETag: "c4-5834898b4137e"
Accept-Ranges: bytes
Content-Length: 196
Connection: close
Content-Type: text/html

<!DOCTYPE html>
<html>
<body>

<h1>Manuel Alejandro Luque Le&ocirc;ne;n</h1>

<p>Mi asignatura favorita es PAS sin lugar a dudas.</p>

<p>Esto es otro &iacute;ndice distinto.</p>

</body>

</html>

Connection closed by foreign host.
```

Se puede observar que devuelve “200 OK” esto es la prueba de que la petición ha sido recibida correctamente.

En el caso de enviar “*HOLA*”:


```
alex@ThinkPAlex:~/httpd-docs/newFolder$ telnet localhost 8080
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
HOLA
HTTP/1.1 400 Bad Request
Date: Tue, 05 Mar 2019 14:46:31 GMT
Server: Apache/2.4.38 (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.
```

El servidor devuelve “400 Bad Request”, esto se debe a que no se ha especificado a donde se manda esta petición. Si se incluye escribiendo “HOLA / HTTP/1.0”:

```
alex@ThinkPAlex:~/httpd/bin$ telnet localhost 8080
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
HOLA / HTTP/1.0

HTTP/1.1 501 Not Implemented
Date: Wed, 06 Mar 2019 09:22:33 GMT
Server: Apache/2.4.38 (Unix)
Allow: POST,OPTIONS,HEAD,GET,TRACE
Content-Length: 203
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!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 /index2.html not supported.<br />
</p>
</body></html>
Connection closed by foreign host.
```

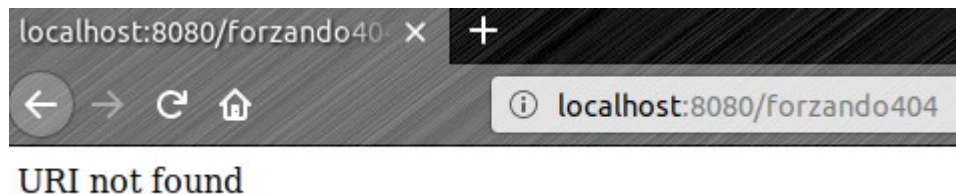
El error que devuelve (“501 Not Implemented”) que dicha petición no está implementada.

10. Encuentra la directiva que se utiliza para la visualización de páginas de error. Modifícala para personalizar el mensaje de error 404 que mostrará el servidor y que sea una cadena del tipo *"URI not found"*. ¿Se podría especificar un fichero .html de error? Modifica el error 501 para que muestre *"Method not implemented"* y comprueba que funciona.

Al incluir la siguiente línea en httpd.conf:

```
ErrorDocument 404 "URI not found"
```

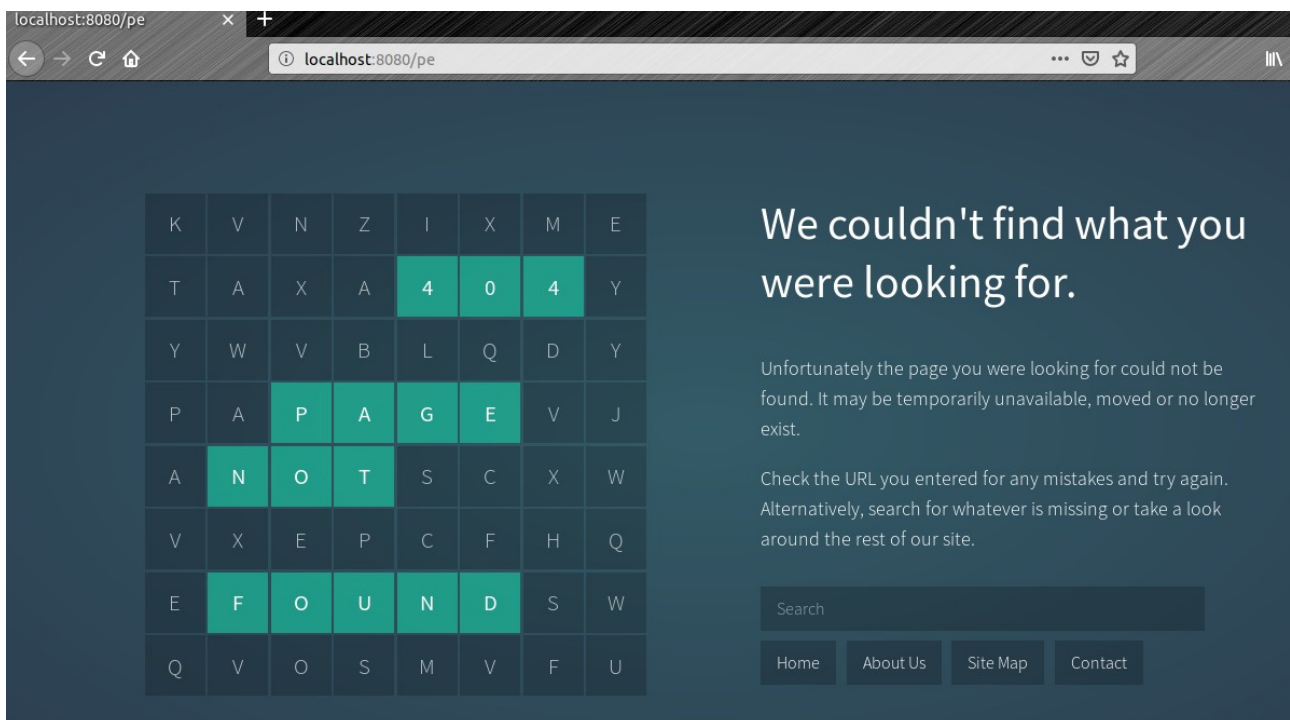
E intentar acceder desde navegador:



De la siguiente manera se puede incluir un html (con css y js en este caso):

```
ErrorDocument 404 /404.html
```

Al acceder con navegador:



Si se incluye:

```
ErrorDocument 501 "Method not implemented."
```

Y al preguntar por un método no existente se puede comprobar que funciona:

```
alex@ThinkPAlex:~/httpd/bin$ telnet localhost 8080
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
HOLA / HTTP/1.0

HTTP/1.1 501 Not Implemented
Date: Wed, 06 Mar 2019 17:50:38 GMT
Server: Apache/2.4.38 (Unix)
Allow: GET,POST,OPTIONS,HEAD,TRACE
Content-Length: 23
Connection: close
Content-Type: text/html; charset=iso-8859-1

Method not implemented.Connection closed by foreign host.
```

11. Identifica las directivas relacionadas con los archivos de logs de Apache. Haz un acceso normal y acceso erróneo (por ejemplo, un error de tipo 501). Comprueba los logs y muestra cómo se han modificado.

En el caso de realizar un log erróneo, la siguiente directiva indica donde debe registrarse este:

```
ErrorLog "logs/error_log"
```

En caso de que el log fuera exitoso:

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

#
# If you prefer a logfile with access, agent, and referer information
# (Combined Logfile Format) you can use the following directive.
#
#CustomLog "logs/access_log" combined
```

Tras realizar un log exitoso a lo carpeta newFolder y se comprueba el archivo que coincide con la ruta antes mencionada:


```
127.0.0.1 - - [06/Mar/2019:20:19:40 +0100] "GET /
newFolder/ HTTP/1.1" 200 1544
```

Si se realiza uno erróneo con un error 501 y se comprueba el archivo correspondiente:

```
[Wed Mar 06 20:21:28.044127 2019] [core:error] [pid
10284:tid 139711504471808] [client 127.0.0.1:36494]
AH00135: Invalid method in request Hola / HTTP/1.0
```

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

Añadiendo esto al `httpd.conf`:

```
Redirect permanent /uco http://www.uco.es/
```

Se puede hacer que al acceder a la dirección *localhost:8080/uco* se abra la página de la *uco*.

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 */newFolder*. *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*).

Al añadir al `httpd.conf` las siguientes líneas habilitaremos el *virtual host*:

```
<VirtualHost 192.168.1.105:8080>
    DocumentRoot "/home/alex/httpd-docs/newFolder"
    ServerName 192.168.1.105:8080

    ErrorLog "/home/alex/httpd-docs/newFolder/logs/ip-error.log"
    CustomLog "/home/alex/httpd-docs/newFolder/logs/ip-access.log" common

    # Other directives here
</VirtualHost>
```

Se puede observar que también se han generado los archivos de log independientes y que han registrado correctamente la entrada al servidor:

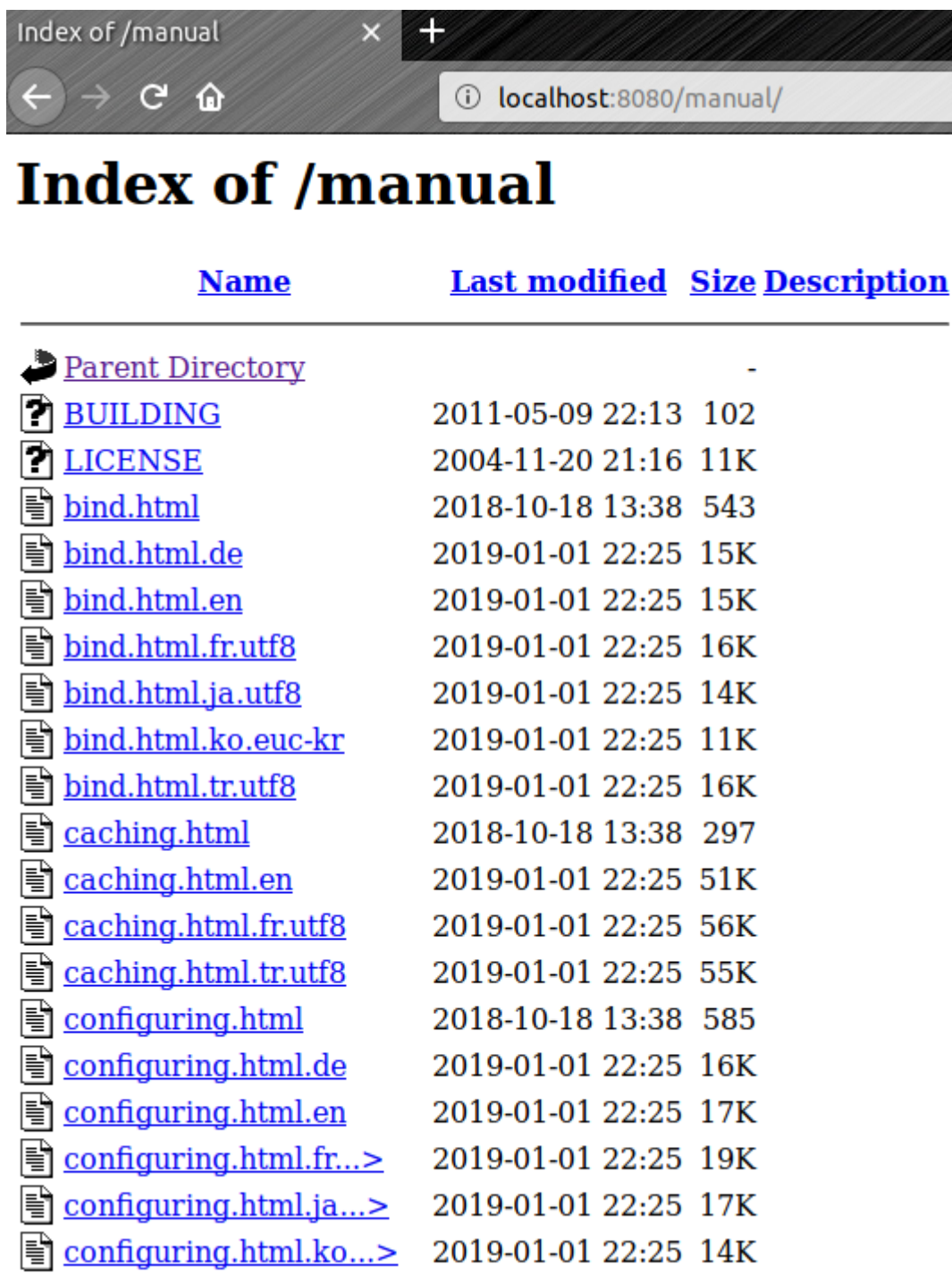
```
alex@ThinkPAlex:~/httpd-docs/newFolder/logs$ ls
ip-access.log ip-error.log
alex@ThinkPAlex:~/httpd-docs/newFolder/logs$ cat ip-access.log
192.168.1.105 - - [07/Mar/2019:19:22:58 +0100] "GET / HTTP/1.1" 403 209
```

14. Haz que el servidor web sirva toda la documentación de Apache. Para ello, habilita la configuración del manual de Apache (*extra/httpd-manual.conf*) y los módulos que necesite. Explica las distintas líneas que aparecen en dicha configuración, incluyendo el significado de las expresiones regulares.

Al habilitar las siguientes líneas del servidor:

```
LoadModule negotiation_module modules/mod_negotiation.so
# Local access to the Apache HTTP Server Manual
Include conf/extra/httpd-manual.conf
```

Se podrá acceder al manual:



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/manual/'. The page title is 'Index of /manual'. The main content is a table listing files and directories in the manual. The table has four columns: 'Name', 'Last modified', 'Size', and 'Description'. The first row is a directory entry for 'Parent Directory'. Subsequent rows are file entries for 'BUILDING', 'LICENSE', and various localized versions of 'bind.html', 'caching.html', and 'configuring.html' in different languages and encodings (e.g., utf8, ko.euc-kr). Each file entry includes its last modified date and time, and its size in bytes or kilobytes.

Name	Last modified	Size	Description
Parent Directory	-	-	-
BUILDING	2011-05-09 22:13	102	
LICENSE	2004-11-20 21:16	11K	
bind.html	2018-10-18 13:38	543	
bind.html.de	2019-01-01 22:25	15K	
bind.html.en	2019-01-01 22:25	15K	
bind.html.fr.utf8	2019-01-01 22:25	16K	
bind.html.ja.utf8	2019-01-01 22:25	14K	
bind.html.ko.euc-kr	2019-01-01 22:25	11K	
bind.html.tr.utf8	2019-01-01 22:25	16K	
caching.html	2018-10-18 13:38	297	
caching.html.en	2019-01-01 22:25	51K	
caching.html.fr.utf8	2019-01-01 22:25	56K	
caching.html.tr.utf8	2019-01-01 22:25	55K	
configuring.html	2018-10-18 13:38	585	
configuring.html.de	2019-01-01 22:25	16K	
configuring.html.en	2019-01-01 22:25	17K	
configuring.html.fr...>	2019-01-01 22:25	19K	
configuring.html.ja...>	2019-01-01 22:25	17K	
configuring.html.ko...>	2019-01-01 22:25	14K	

En cuanto al contenido del fichero anteriormente habilitado, se puede observar que este activa un alias a */manual* de manera que este sea redireccionado a la carpeta *httpd/manual* donde se encuentran los html con la información correspondiente al manual. Además se puede observar que se aplica *SetHandler* y *type-map* a todos los archivos *.html*.

15. 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:

- a) 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).

Se ejecutará varias veces como prueba, una probando a creando el archivo nuevo y otra añadiendo un nuevo usuario al ya existente:

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

Una vez hecho esto se listará el contenido del archivo *.htpasswd* y así comprobar que la creación de usuarios ha sido exitosa:

```
alex@ThinkPAlex:~/httpd/bin$ cat .htpasswd
user1:$apr1$KF1AK74q$1ZmsMPDBSyh38pM8rlmu70
user2:$apr1$vMlbmKQM$aP0g2Nphtl4zFdd1l9qUU/
user3:$apr1$yZSluWzr$mhEw5zi1DoNQsrS/zMoIa1
```

- b) 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.

Es necesario activar la siguiente directiva en el *httpd.conf*:

```
AllowOverride All
```

Se creará el archivo *.htaccess* con el siguiente contenido:

```
alex@ThinkPAlex:~/httpd/htdocs/secretFolder$ cat .htaccess
AuthUserFile /home/alex/httpd/bin/.htpasswd
AuthName secretFolder
AuthType Basic

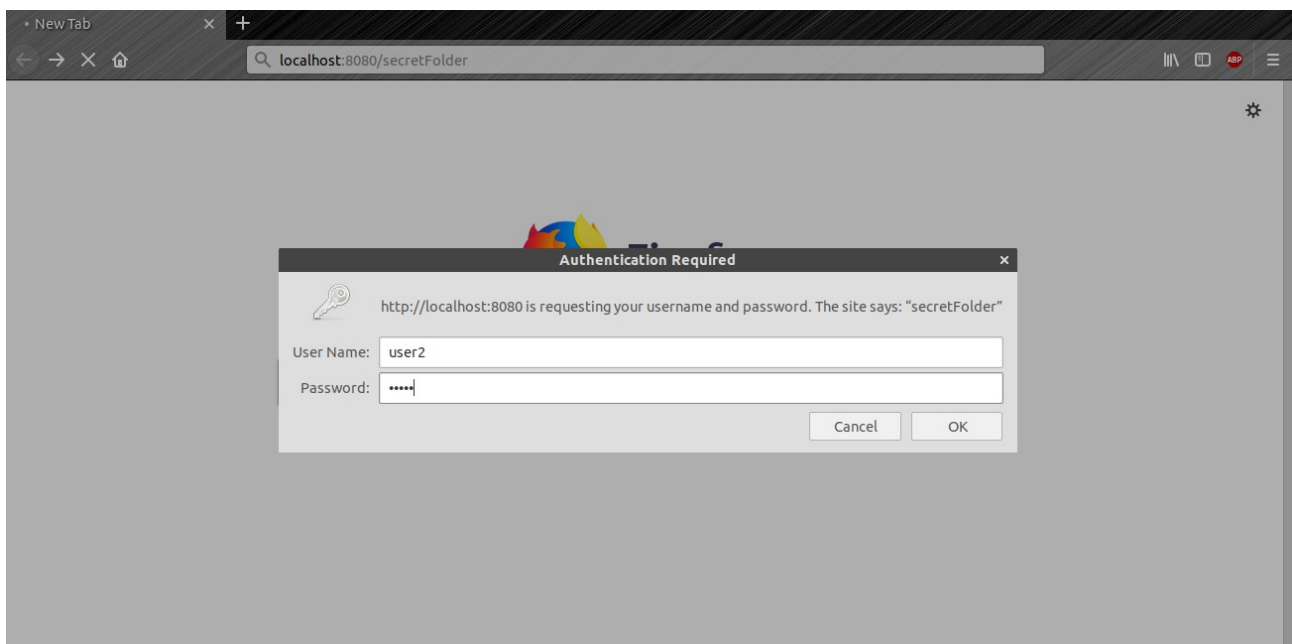
require user user1 user2 user3
```

- c. Debes crear un directorio, que se llamará *secretFolder*, de forma que, para acceder a él, habrá que autenticarse. Los usuarios que tendrán acceso a ese directorio serán: *user1*, *user2* y *user3*; y la contraseña, para todos será: *entra*. Explica los pasos realizados para conseguirlo, y el contenido de los archivos creados.

Se creará el directorio incluyendo el siguiente código en el archivo *http.conf*:

```
<Directory "/home/alex/httpd/htdocs/secretFolder">
    Options +Indexes
    AuthType Basic
    AuthName secretFolder
    AuthUserFile /home/alex/httpd/bin/.htpasswd
    Require valid-user
</Directory>
```

Tras esto se podrá acceder a la carpeta protegida *secretFolder*:



Si se el usuario identifica adecuadamente, es decir con cualquiera de los usuario antes autorizados, se podrá acceder a dicho directorio:



16. 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.

Al añadir las siguientes líneas al archivo *httpd.conf* se puede restringir o permitir el acceso a ciertas ip.

```
Order allow,deny
Deny from all
Allow from 127.0.1.1/24
```

En el caso de poner en la directiva *Order allow,deny* se leerá primero la directiva *allow* permitiendo el acceso a todos los usuarios cuya ip coincida con esta y negando el acceso a todos los demás por defecto.

En el caso en el que aparezca *deny,allow* se comprobará primero si la *ip* del solicitante coincide con la especificada en *deny*, si esto es así se le prohibirá la entrada, si no se permitirá la entrada por defecto.

En el caso de acceso denegado se podrá ver en el navegador el siguiente mensaje:

