

*Autor: Fernando Sánchez Delgado*

## Memoria Práctica 4: Apache

### 1. Introducción (ej 1-3)

Tras ejecutar el script de instalación, hay que añadir al \$PATH el directorio con los ejecutables de Apache, o ir a la carpeta y ejecutar directamente el archivo.

Sin cambiar el puerto donde escucha el servidor, no se inicia al intentar escuchar en el puerto 80, ya que está restringido el acceso. Para poder iniciar el servidor hay que cambiar el puerto en el que el servidor escucha. Para ello en el fichero `https.conf` dentro de la carpeta `httpd/conf`, buscamos la línea 52, y cambiamos la opción *Listen* de 80 a 8080 (o cualquier otro puerto que queramos mientras esté libre).

Tras hacer esto ejecutamos el archivo `apachectl` dentro de la carpeta `httpd/bin`, pasándole el argumento *start*: `./apachectl start`. A continuación, accedemos en el navegador a la dirección **localhost:8080**. Debe salir el mensaje: *“It works!”*.

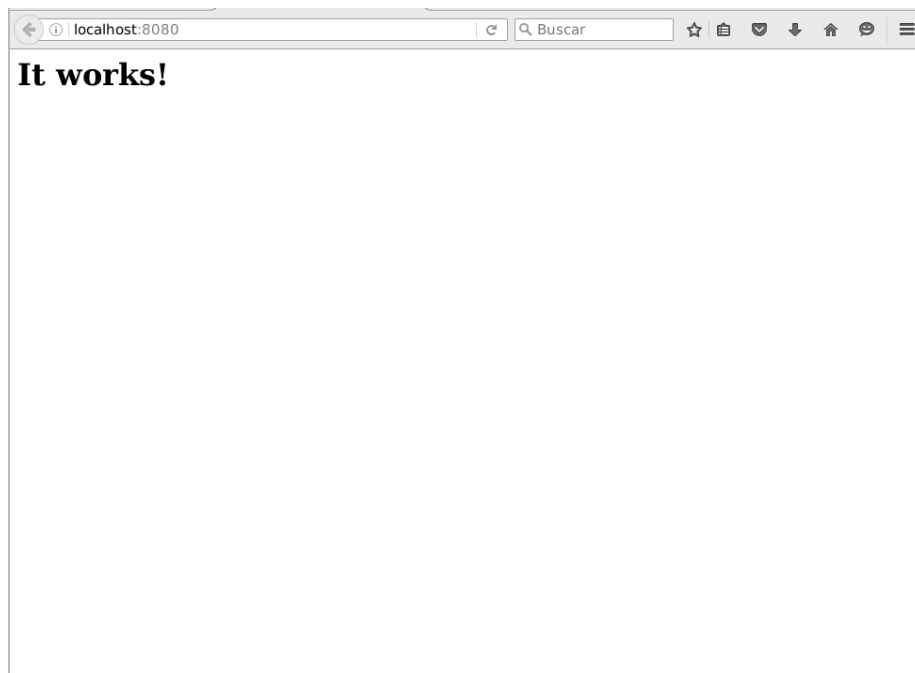


Figure 1: “Página que aparece al iniciar Apache por primera vez”

## 2. Directorio de archivos (ej 4-6)

### Ej4

Para cambiar el directorio donde están los archivos que se sirven, hay que cambiar en las opciones estas dos líneas, situadas en las líneas 214-215:

```
DocumentRoot "/home/i42sadev/httpd/htdocs"  
<Directory "/home/i42sadev/httpd/htdocs">
```

Cambiamos la carpeta en ambas líneas por el directorio deseado, en este caso: */home/i42sadev/httpd-docs*.

### Ej5

El archivo base que se sirve por defecto es *index.html*, y si no lo encuentra devuelve un error de *404 Not Found*, para cambiar el nombre del archivo por defecto se puede editar en el archivo de configuración la línea 248: **DirectoryIndex index.html**. Es posible poner varios archivos separados por espacios, de forma que si no encuentra el primero, use el segundo y así sucesivamente. Para probarlo podemos tener dos ficheros en el directorio **httpd-docs**, uno *index.html* y otro *index.htm*. Si hemos configurado apache de la siguiente forma: **DirectoryIndex index.html index.htm**. Entonces si eliminamos *index.html* automáticamente servirá *index.htm*. Y en el momento que volvamos a crear *index.html* comenzará a servir ese archivo.

### Ej6

Por defecto el directorio raíz incluye la **Indexes** en la línea 228 **Options Indexes FollowSymLinks**. Esta opción es la que permite a Apache crear automáticamente un listado de un directorio cuando no tiene un archivo de índice (ej. *index.html*). Para probar esta opción crearemos un directorio de prueba dentro del directorio raíz. Y dentro creamos unos cuantos ficheros de prueba con el comando **touch test{01..15}.txt**. A continuación accedemos a la ruta **localhost:8080/midirectorio**. Y debe aparecer un listado de todos los ficheros.

Si queremos evitar que se puedan listar los directorios basta con eliminar la opción *Indexes* mencionada antes. Aunque si lo que queremos es que algún directorio en concreto no se pueda mostrar se puede hacer añadiendo lo siguiente al archivo de configuración. Lo cual desactiva la opción solo para este directorio. Devolviendo un error de acceso prohibido (*Forbidden*) cuando se intenta acceder a este directorio.

```
<DirectoryMatch "/midirectorio">  
    Options -Indexes  
</DirectoryMatch>
```



Figure 2: “Índice generado por Apache del directorio /midirectorio”

También se puede mejorar el aspecto visual de los índices descomentando la línea 455: `Include conf/extra/httpd-autoindex.conf`.

### 3. Conexión al servidor (ej7-10)

#### Ej7


Cuando iniciamos apache nos devuelve un mensaje con el siguiente contenido:

```
AH00558: httpd: Could not reliably determine the server's fully
qualified domain name, using 172.16.218.12. Set the 'ServerName'
directive globally to suppress this message
```

Esto hace referencia a que no hemos puesto el nombre del servidor, y por lo tanto tomará como valor por defecto la dirección IP. Para evitar que salga este mensaje debemos configurar el nombre del servidor en la opción *ServerName*. En torno a la línea 190 hay un comentario donde se explica esta opción. Así que ponemos debajo la línea `ServerName localhost:8080`.

#### Ej8

Para modificar el usuario y el grupo bajo los que corre el demonio *httpd* debemos modificar las opciones **User** y **Group**. Estas opciones se encuentran en torno a la línea 160. Como explica el comentario, podemos especificar tanto el nombre como el id del usuario/grupo, teniendo que usar de prefijo `#` en caso de querer usar el id. Además también aclara que para poder usar un usuario o grupo distintos es necesario comenzar la ejecución como *root*.



Name	Last modified	Size	Description
<a href="#">Parent Directory</a>	-		
<a href="#">test01.txt</a>	2016-05-06 13:51	0	
<a href="#">test02.txt</a>	2016-05-06 13:51	0	
<a href="#">test03.txt</a>	2016-05-06 13:51	0	
<a href="#">test04.txt</a>	2016-05-06 13:51	0	
<a href="#">test05.txt</a>	2016-05-06 13:51	0	
<a href="#">test06.txt</a>	2016-05-06 13:51	0	
<a href="#">test07.txt</a>	2016-05-06 13:51	0	
<a href="#">test08.txt</a>	2016-05-06 13:51	0	
<a href="#">test09.txt</a>	2016-05-06 13:51	0	
<a href="#">test10.txt</a>	2016-05-06 13:51	0	
<a href="#">test11.txt</a>	2016-05-06 13:51	0	
<a href="#">test12.txt</a>	2016-05-06 13:51	0	
<a href="#">test13.txt</a>	2016-05-06 13:51	0	
<a href="#">test14.txt</a>	2016-05-06 13:51	0	
<a href="#">test15.txt</a>	2016-05-06 13:51	0	

Figure 3: “Aspecto mejorado del índice autogenerado”

## Ej9

Podemos simular peticiones al servidor mediante el uso del protocolo **telnet**. Para simular una petición estándar a la página principal debemos escribir: **telnet localhost 8080**, y a continuación **GET / HTTP/1.0** seguido de dos veces la tecla *intro*. Esto nos devolverá un mensaje con un contenido similar al siguiente:

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.1 200 OK
Date: Fri, 20 May 2016 10:35:10 GMT
Server: Apache/2.4.20 (Unix)
Last-Modified: Fri, 06 May 2016 11:43:34 GMT
ETag: "9a-5322af7fb2600"
Accept-Ranges: bytes
Content-Length: 154
Connection: close
Content-Type: text/html

<!DOCTYPE html>
<html>
<head>
```

```
<meta charset="UTF-8">
<title>Inicio</title>
</head>
<body>
<h1>Nombre y Apellidos</h1>
<p>Fernando Sánchez Delgado</p>
</body>
</html>
Connection closed by foreign host.
```

Lo primero que hacemos es enviar una petición GET a la raíz del servidor usando el protocolo *HTTP 1.0*. Tras esto el servidor nos devuelve un respuesta, con el status 200 OK usando el protocolo *HTTP 1.1*. Debajo se muestra el resto de la cabecera del mensaje, donde se incluye información como el nombre del servidor **Server: Apache/2.4.20 (Unix)** o el tipo de contenido del mensaje **Content-Type: text/html**. En este caso el contenido es la página *index.html* que habíamos definido antes en la raíz del servidor.

## Ej10

El servidor Apache permite definir páginas personalizadas para los distintos errores que pueden surgir. Uno de los más comunes es el error 404:

```
HTTP/1.1 404 Not Found
Date: Fri, 20 May 2016 10:54:17 GMT
Server: Apache/2.4.20 (Unix)
Content-Length: 202
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

Para configurar las páginas que se devuelven para los distintos tipos de errores, en torno a la línea 420 del fichero *httpd.conf*, Apache ya trae comentados algunos ejemplos. Vamos a definir mensajes para los errores 404 y 501. Para ello añadimos las líneas:

```
ErrorDocument 404 "El URI que has pedido no lo podemos servir"
ErrorDocument 501 "Método no implementado"
```

En caso que prefiramos devolver algo distinto a una cadena de caracteres, podemos especificar una ruta en su lugar, por ejemplo: **ErrorDocument 404 /404.html**.

## 4. Logs y redirecciones (ej11-13)

### Ej11

Las directivas relacionadas con los logs de Apache se a partir de la línea 270 aproximadamente. Por ejemplo `ErrorLog "logs/error_log"`. Se pueden configurar cosas como la localización de los archivos de log, hasta el formato en que deseamos que se realicen los mensajes. Por defecto todos los archivos log de Apache se encuentran en el directorio `httpd/logs/`. En concreto el archivo `access_log` es el que contiene las peticiones al servidor.

```
127.0.0.1 - - [20/May/2016:13:17:55 +0200] "GET / HTTP/1.1" 304 -
127.0.0.1 - - [20/May/2016:13:19:05 +0200] "GET /asdf HTTP/1.1" 404 42
127.0.0.1 - - [20/May/2016:13:19:35 +0200] "GET /midirectorio/test01.txt HTTP/1.1" 200 -
```

Al realizar una petición normal podemos obtener varios tipos de mensajes en el log, en función de la situación que se haya encontrado el servidor. En la primera línea del log de arriba el status es `304` lo cual hace referencia a que el archivo que se pide no ha cambiado desde el último acceso, por lo que se puede usar la copia almacenada en caché. En la tercera línea se accede a un archivo que no está en la caché del navegador, por lo que el servidor responde con un status `200` y el contenido que se ha pedido.

En cuanto a los mensajes resultantes de un acceso erróneo en la segunda línea de arriba vemos un log con el status `404` ya que se ha solicitado un recurso no existente.

### Ej12

Para redireccionar una ruta hacia otra dirección basta con añadir en el fichero de configuración una directiva como la siguiente: `Redirect /uco http://www.uco.es`. De esta forma al acceder a la dirección `localhost:8080/uco`, seremos redireccionados a `http://www.uco.es`. Los comentarios dentro del archivo de configuración correspondientes a esta opción se encuentran en torno a la línea 315.

### Ej13

Para crear un host virtual que dependa de la dirección con la que el cliente accede basta con añadir al fichero de configuración el siguiente contenido:

```
<VirtualHost *:8080>
    DocumentRoot "/home/i42sadev/httpd-docs/"
    ServerName localhost

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

```

        CustomLog "logs/local-access.log" common

</VirtualHost>

<VirtualHost *:8080>
    DocumentRoot "/home/i42sadev/httpd-docs/midirectorio"
    ServerName 172.16.218.12

    ErrorLog "logs/ip-error.log"
    CustomLog "logs/ip-access.log" common

</VirtualHost>

```

Al lado de *VirtualHost* ponemos *\*:8080* para asegurarnos de que siguen escuchando en el puerto 8080. A continuación ponemos la nueva ruta para la raíz de cada una de las direcciones y ponemos el nombre del servidor que queremos que tenga. Para personalizar los log a cada uno de los *virtual host* basta con añadir en cada uno las líneas *ErrorLog* y *CustomLog*. Para probar si ha funcionado solo basta acceder tras reiniciar Apache a las direcciones de cada *virtual host* para comprobar que los cambios han surgido efecto.

#### Ej14

Para que Apache muestre la documentación, en la ruta por defecto */manual*, es necesario descomentar la línea `Include conf/extra/httpd-manual.conf` en torno a la línea 494. A continuación abrimos el archivo de configuración `httpd/conf/extra/httpd-manual.conf` y observamos los módulos situados en el comentario de la cabecera bajo *Required modules*. Ahora hay que asegurarse de que estos módulos están incluidos para eso buscamos en *httpd.conf* los módulos y descomentamos su línea si aún no lo está. En nuestro caso parece que sólo el módulo `mod_negotiation` no está incluido. Tras esto ya podemos acceder a la dirección */manual* para acceder a la documentación de Apache.

#### Ej15

Para controlar recursos restringidos mediante los medios que incluye Apache debemos crear un archivo *.htaccess* en la raíz del directorio que queremos proteger (*.htaccess sirve para realizar configuración a nivel de directorio*). El contenido del fichero será el siguiente:

```

AuthUserFile /home/fernan/.htpasswd
AuthName CarpetaSecreta
AuthType Basic

require user usuario1 usuario2 usuario3

```

En la última línea, la sentencia **require user** permite especificar cuáles son los usuarios que tienen permitido el acceso. La primera línea contiene el directorio donde se encuentra el archivo con los usuarios y contraseñas (*es recomendable ponerlo fuera de la raíz del servidor*). Ahora creamos el archivo `.htpasswd`, para ello usamos el comando `htpasswd -cb <ruta-archivo> <nombre-usuario> <contraseña-usuario>`. Esto solo se debe realizar para el primer usuario, el resto deben añadirse sin la opción `c`. Por último, se debe cambiar en la configuración del directorio dentro de `httpd.conf` la opción **AllowOverride none** a **AllowOverride All**, de forma que pueda hacer efecto la configuración del archivo `.htaccess`.

## Ej16

Para controlar el rango de direcciones que acceden al servidor web, vamos a la parte de `httpd.conf` donde se configura el directorio raíz, y añadimos dentro de la directiva `<Directory>` asociada:

```
Order Allow,Deny
Allow from <ip-maquina>/24
```

De esta forma estamos restringiendo el acceso a todos los clientes que no pertenezcan a la misma subred que el servidor. La directiva **Order** fija el orden en que se evalúan las directivas **Allow** y **Deny**. En el caso de que quisieramos restringir el acceso para todas las direcciones IP, basta con añadir la directiva **Deny from all**.

Cuando se intente acceder a alguna dirección de nuestro servidor nos devolverá un error *Forbidden*. Hay que tener en cuenta que al restringir el acceso a la subred, ya no podemos acceder por medio de localhost, ya que es la IP `127.0.0.1` la cual no pertenece a la subred. Basta añadir localhost después, para que siga funcionando.