

Práctica 4

Jesús Rodríguez Heras
Arantzazu Otal Alberro
Roberto Muras González

22 de mayo de 2019

1. Instalación de la máquina

En esta primera parte vamos a crear el entorno de trabajo que utilizaremos durante la práctica. Para ello vamos a:

- Iniciar una máquina Vagrant con una IP privada.
- Instalar apache.
- Instalar y configurar Webmin.
- Instalar php.

Para facilitar los ejercicios, se aconseja modificar el archivo `host` de la máquina anfitriona (portátil) para que pueda acceder al servidor mediante un nombre de dominio.

Instalación de apache:

```
sudo apt-get install apache2
```

Para instalar y configurar Webmin nos dirigimos a la página de descargas de Webmin para sistemas debian: <http://www.webmin.com/deb.html>

Nos dirigimos al apartado *Using the webmin APT repository*. En el archivo `/etc/apt/sources.list` de la máquina virtual, pegamos lo siguiente:

```
deb https://download.webmin.com/download/repository sarge contrib
```

A continuación, actualizamos los paquetes e instalamos Webmin¹.

```
sudo apt-get update && sudo apt-get install webmin
```

Para instalar php, usamos el siguiente comando:

```
sudo apt-get install php5-common sudo apt-get install  
libapache2-mod-php5
```

A la hora de modificar el fichero `/etc/hosts` de la máquina anfitriona, lo abrimos con:

```
sudo nano /etc/hosts
```

Y añadimos la IP de la máquina virtual al final del archivo:

```
192.168.1.100 manolorg.uca.es
```

Siendo:

- `192.168.1.100` la IP de la máquina virtual.
- `manolorg.uca.es` el dominio con el que vamos a trabajar.

Para comprobar que funciona, escribimos lo siguiente en el navegador de nuestro portátil:

```
manolorg.uca.es
```

¹Si nos da un fallo al hacer `sudo apt-get update` que nos dice que nos falta `apt-transport-https`, hacemos `sudo apt-get install apt-transport-https`.

Y nos debe aparecer la página por defecto de apache.

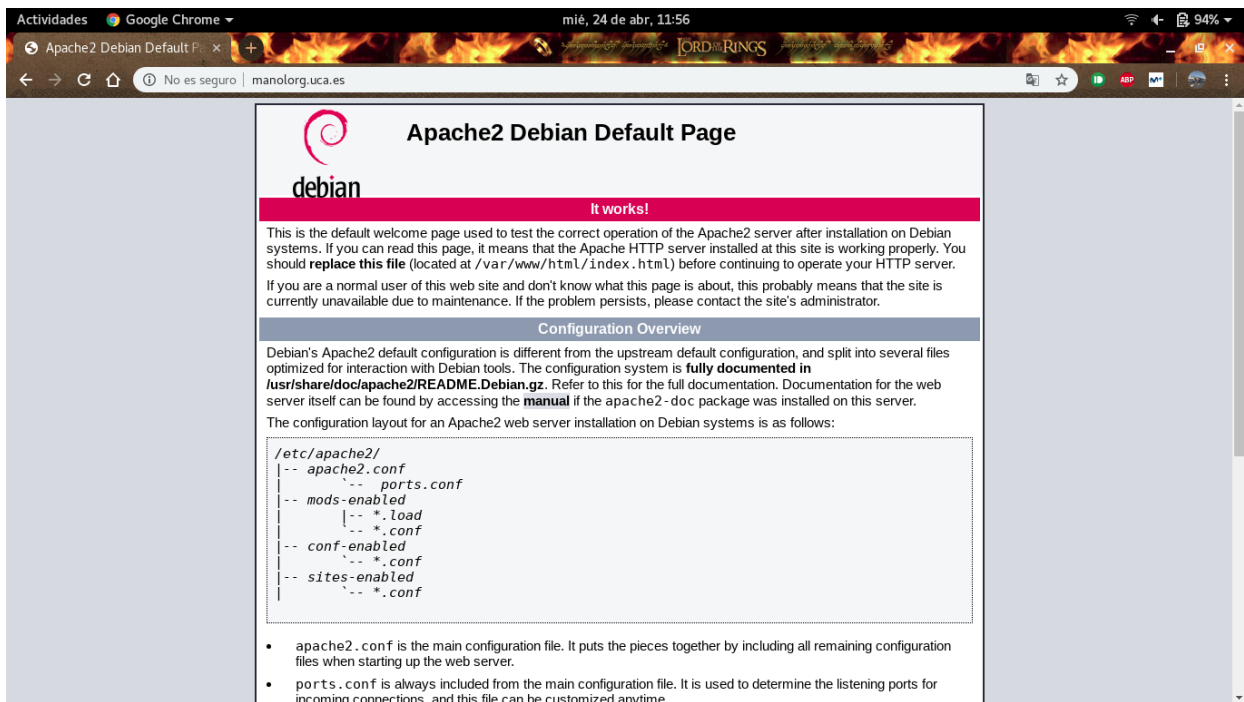


Figura 1: Página inicial de Apache

Para comprobar que Webmin está bien instalado, entramos en la siguiente dirección:

`https://manolorg.uca.es:10000`

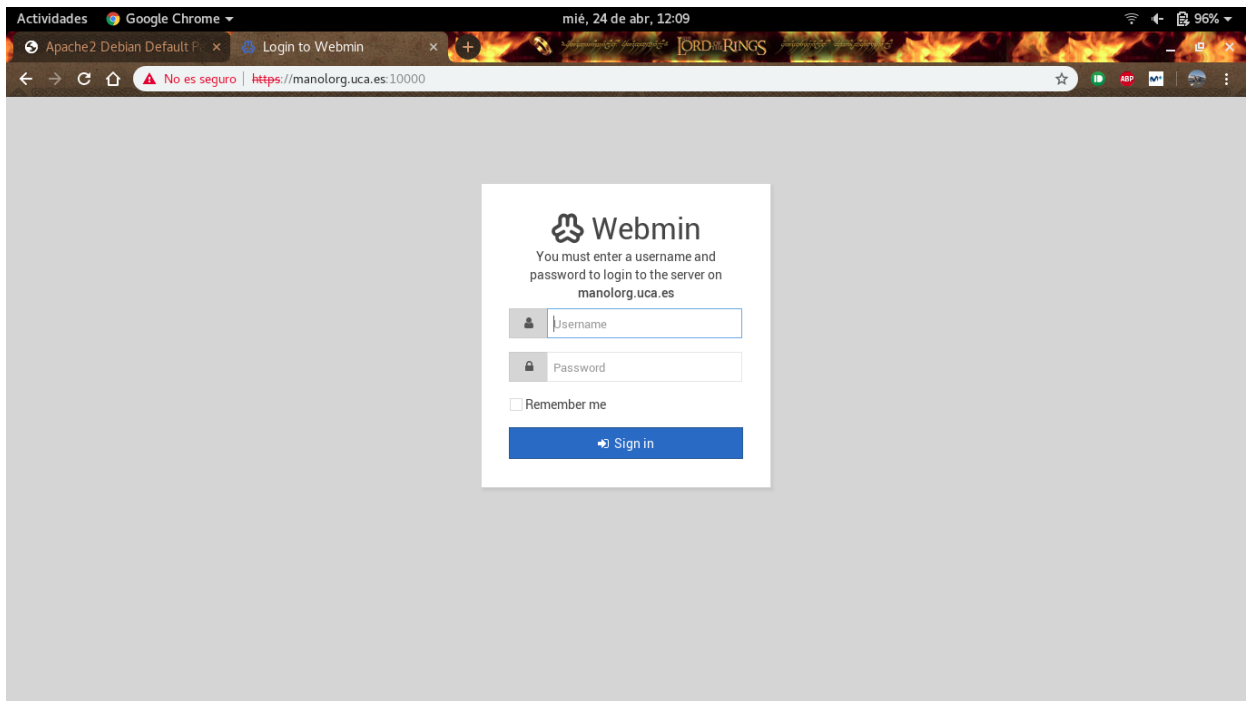


Figura 2: Página principal de Webmin

Para comprobar que php funciona bien cambiamos el archivo `/var/www/html/index.html` por `/var/www/html/index.php` y copiar el siguiente código en su interior:

```
1 <html>
2   <head>
3     <title>Prueba de PHP</title>
4   </head>
5   <body>
6     <?php echo ' <p>Hola_Mundo</p>'; ?>
7   </body>
8 </html>
```

Para ver si funciona, entramos en `manolorg.uca.es`.



Figura 3: Prueba PHP

2. Configuración básica de apache

Como ejemplo, vamos a modelar y configurar el entorno web del grupo de investigación Manolo Research Group.

- La página principal del grupo está bajo el dominio `manolorg.uca.es`. El contenido se almacenará en `/var/www/manolorg`.

Para ello accedemos a webmin y creamos un nuevo host virtual y, en document root ponemos `/var/www/manolorg`.

- Se desea que solo se acceda a la dirección `/docs` solo pueda acceder el usuario `admin` y el usuario `manolo`.

Para ello creamos el directorio `/docs` y lo añadimos como “directory” en webmin. Luego, vamos a “Access control” y añadimos lo siguiente:

- En authentication realm name hay que poner algo (lo que sea).
- En restrict users by login poner “All valid users”.
- En Authentication tipe, lo ponemos en basic.
- En clients must satisfy, ponemos “All access control”.
- En User text file, ponemos el directorio del archivo txt (`acceso.txt`) con las contraseñas de los usuarios.
- En Basic login, en text file.
- Guardamos y volvemos a entrar y podremos ver edit users. Y podremos crear el usuario que queramos.

Debería quedar como en la siguiente imagen²:

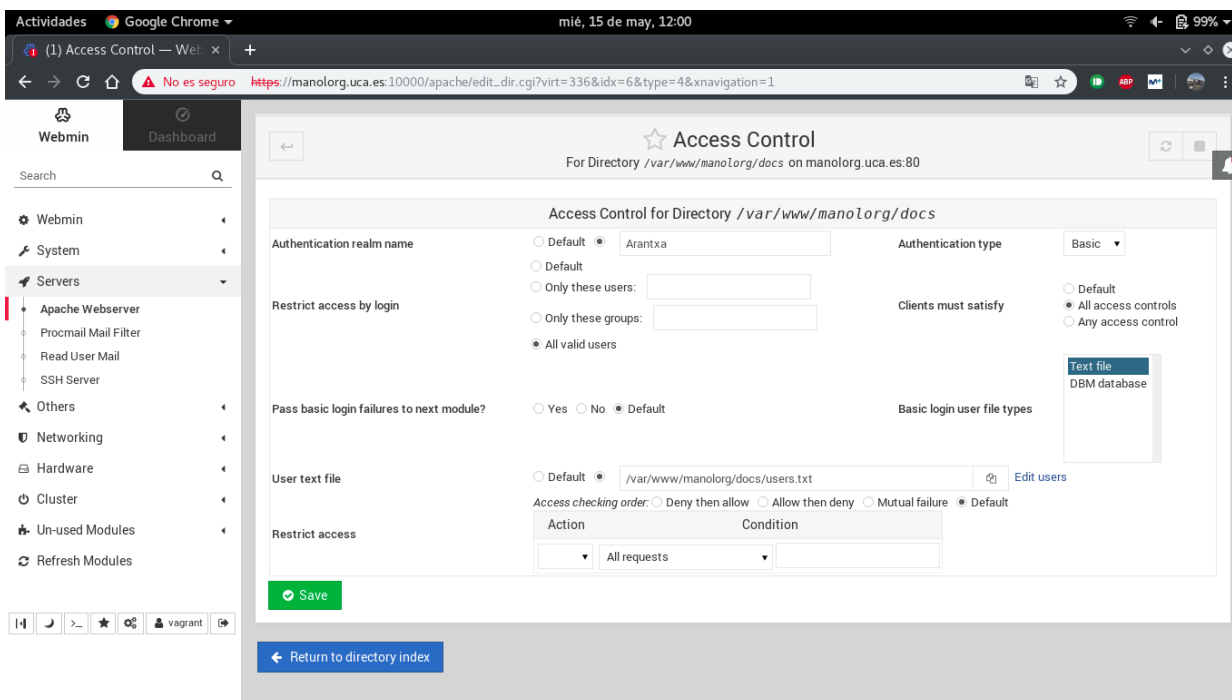


Figura 4: Configuración de control de acceso.

Para crear el usuario con su contraseña, le damos al botón “Edit Users” que hay en “Access Control” y lo ponemos tal como vemos en la siguiente imagen:

²Solo deberíamos tener un fichero de contraseñas, que debería estar en el document root de `manolorg.uca.es`.

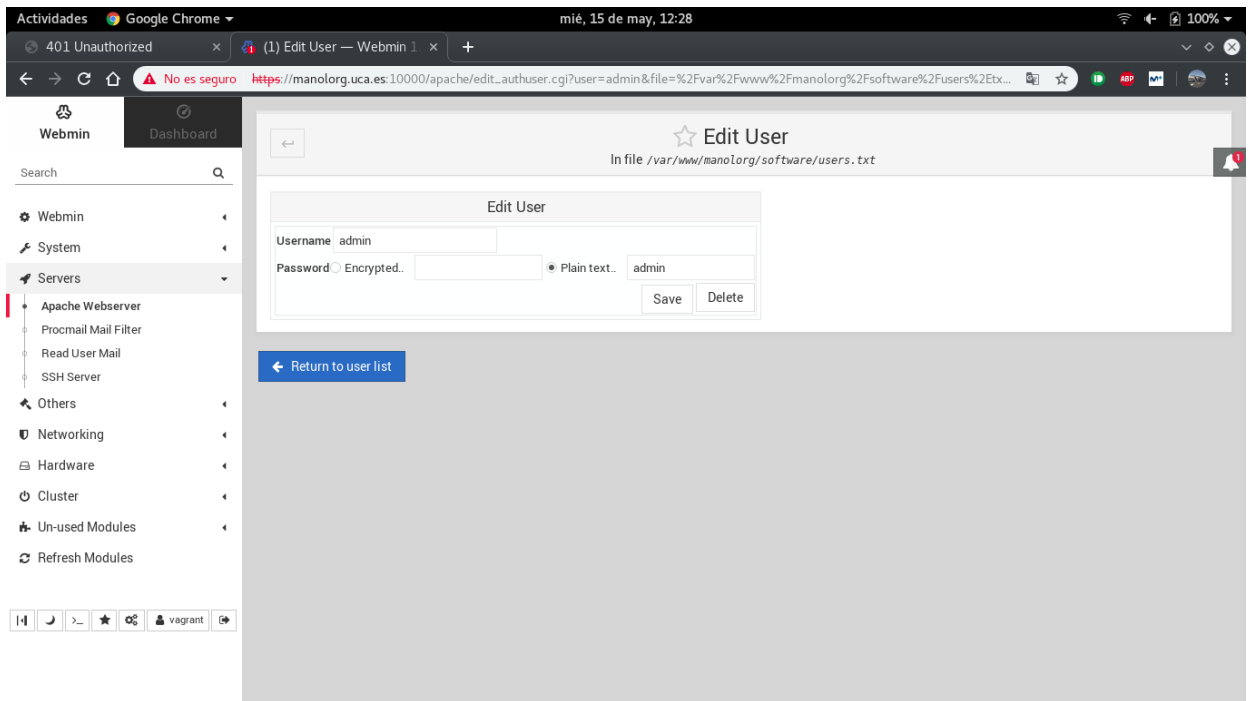


Figura 5: Creación de usuarios.

Para comprobar si funciona entramos a la dirección `manolorg.uca.es/docs` y veremos que nos pide el control de acceso como en la siguiente imagen:

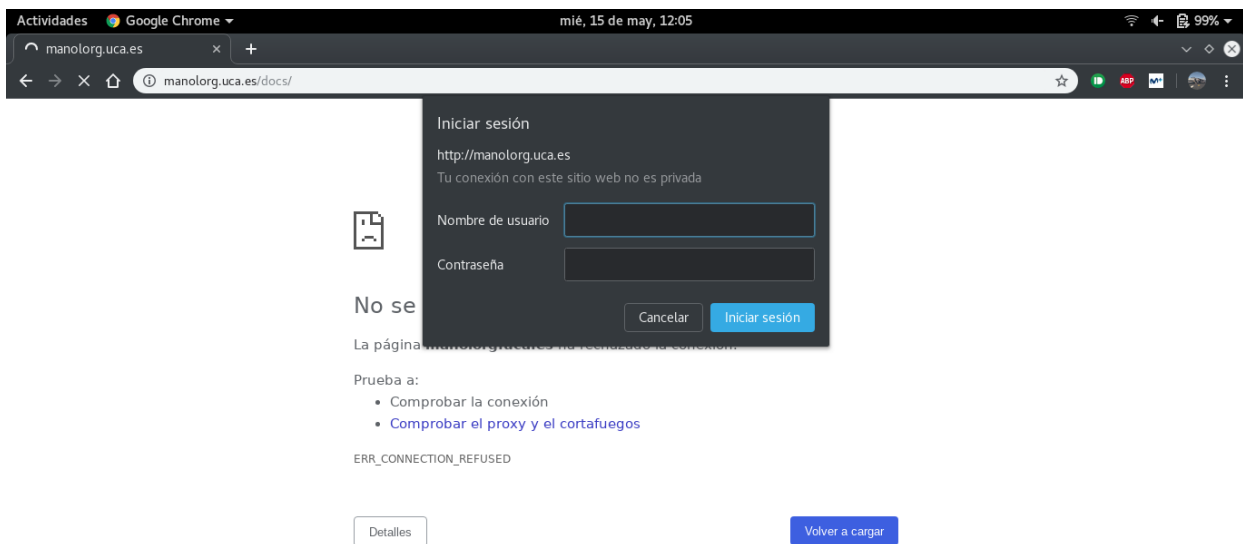


Figura 6: Configuración de control de acceso en `manolorg.uca.es/docs`.

Si ponemos las credenciales creadas anteriormente, nos permitirá el acceso.

- No se mostrarán los directorios, excepto la carpeta `/software` que será solo accesible para el usuario `admin`.

Para que el directorio `software` nos muestre el índice de archivos, tendremos que crear un nuevo “directory” llamado `software`, luego nos iremos al `Document Options` del nuevo directorio. Una vez ahí, tendremos que marcar la opción de “Selected below” y la opción “Generate directory indexes” como “yes”. Quedaría como en la siguiente imagen:

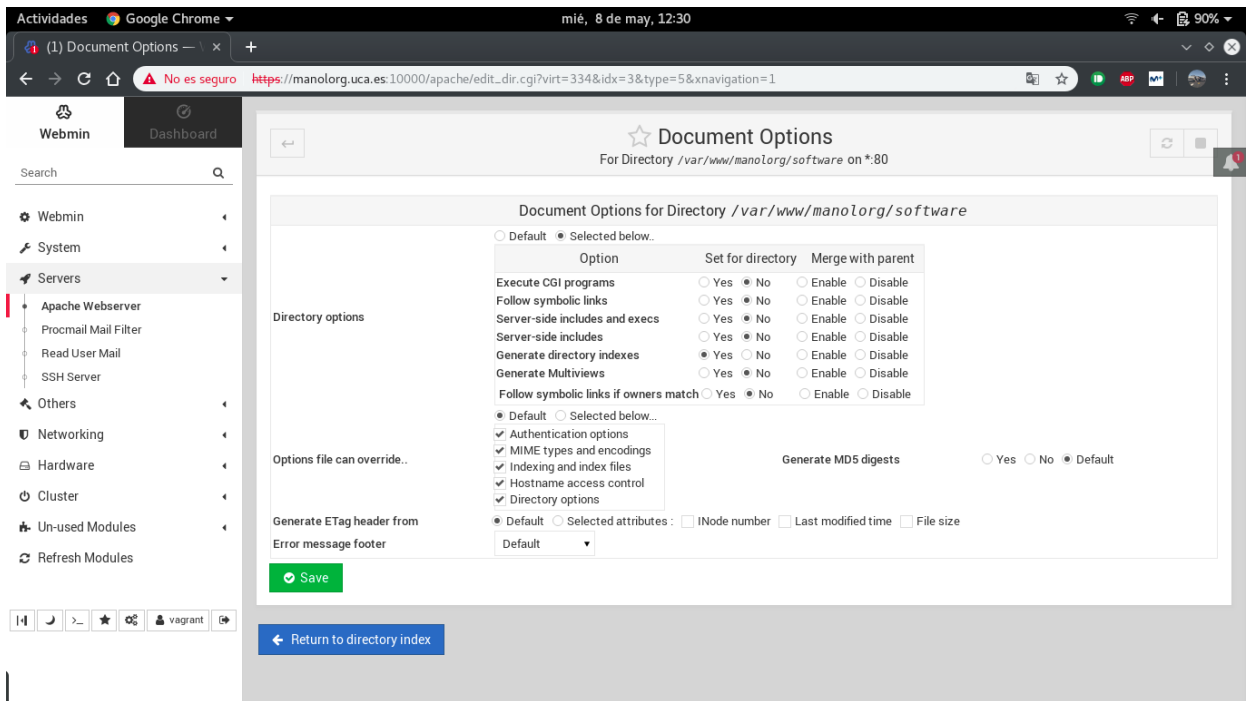


Figura 7: Document Options para software

También debemos configurar que solo tenga acceso el usuario admin en el “Access Control”. Para ello, marcamos la opción “Only these users” como se ve en la siguiente imagen:

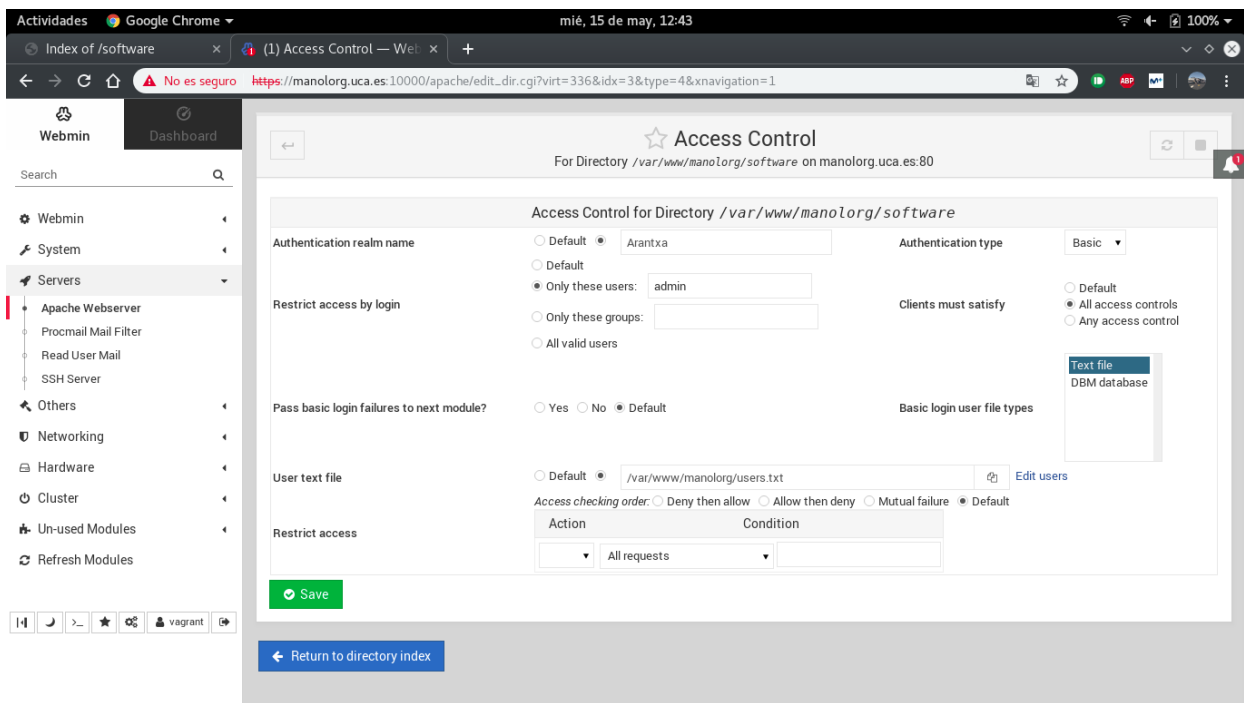


Figura 8: Control de acceso para admin en /software.

- La dirección /sci2s apuntará a la web sci2s.ugr.es.

Para ello creamos el directory de sci2s, entramos dentro y le damos a “Aliases and Redirects” y ponemos la dirección deseada como en la siguiente imagen:

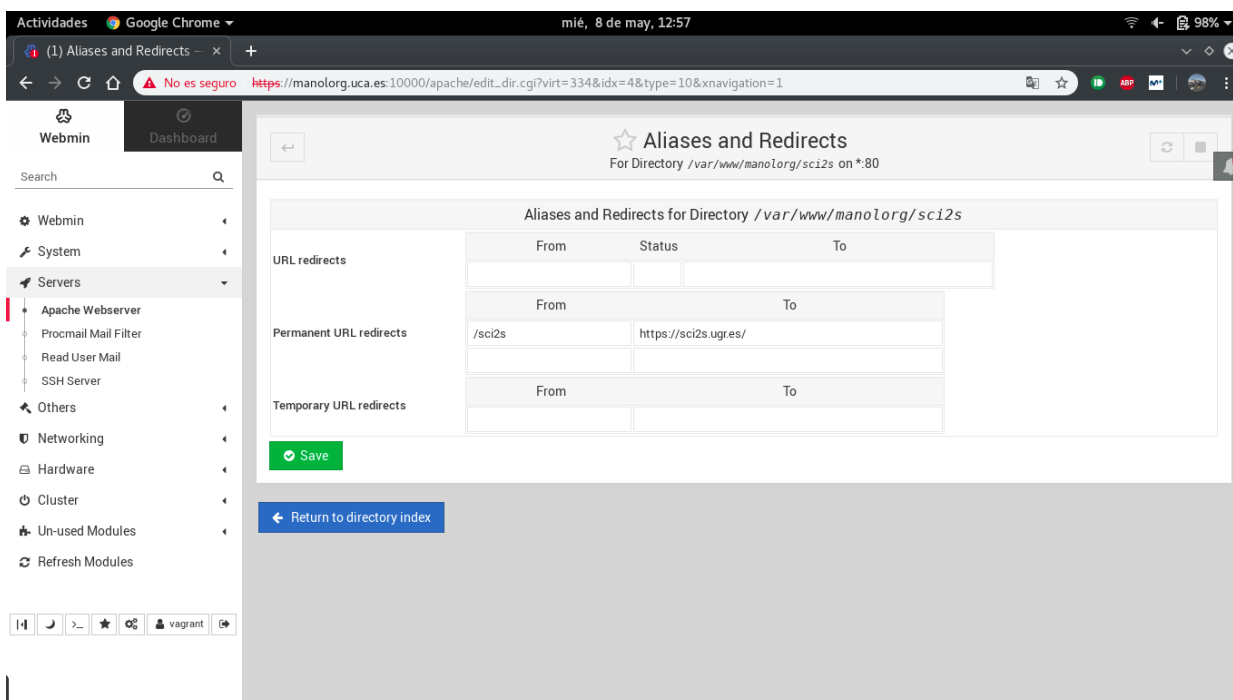


Figura 9: Redirección a sci2s.ugr.es

- El grupo estará creado por diferentes laboratorios, crear un subdominio para el lab1 y el lab2. El contenido se almacenará en `/var/www/lab*`.

Para ello, creamos dos carpetas nuevas desde la consola de la máquina virtual en `/var/www`, llamadas `lab1` y `lab2`.

Para cada uno de los laboratorios, crearemos un virtual host distinto, y, en las opciones de creación, pondremos el document root en `"/var/www/labX"` (siendo X el número del laboratorio), pondremos el puerto 80, como `server name` pondremos `"labX.manolorg.uca.es"` (siendo X el número del laboratorio).

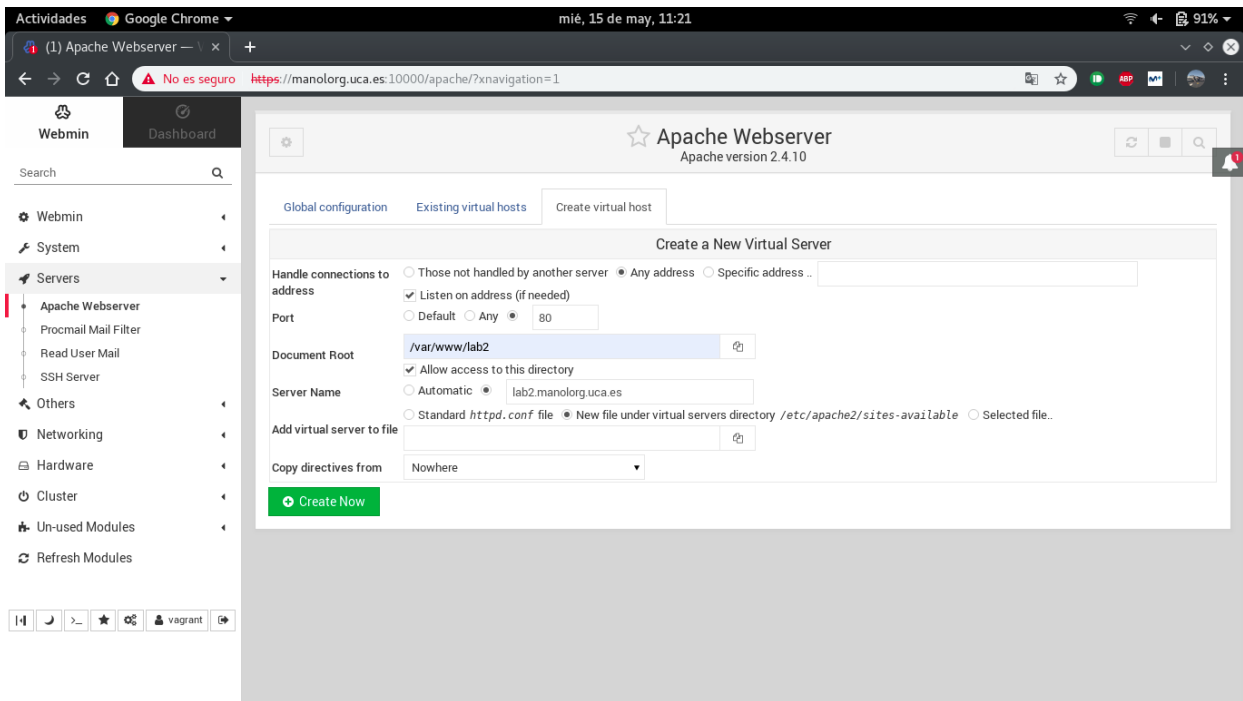


Figura 10: Ejemplo de creación de `lab2.manolorg.uca.es`.

Una vez hecho lo mismo con ambos laboratorios, entramos en `lab1.manolorg.uca.es` y `lab2.manolorg.uca.es` para ver si funcionan ambos sitios, tal como se muestra en la siguiente imagen:

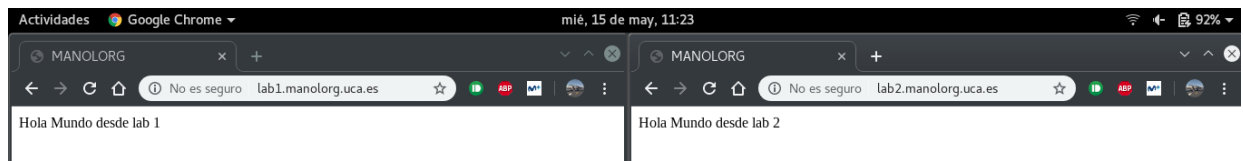


Figura 11: Muestra del funcionamiento de los laboratorios.

Se deberá crear una web sencilla (Hola mundo) en cada uno de los dominios y carpetas. Las páginas deben tener caracteres con tildes, por lo que debemos configurar Apache para que trabaje con UTF8.

Para ello, usaremos el siguiente código en el fichero `/var/www/manolorg`:

```

1 <html>
2 <head>
3   <title>MANOLORG</title>
4 </head>
5 <body>
6   <p>Página de inicio de manolorg.</p>
7 </body>
8 </html>

```

Por defecto, no funciona la codificación UTF8, como podemos ver en la siguiente imagen:

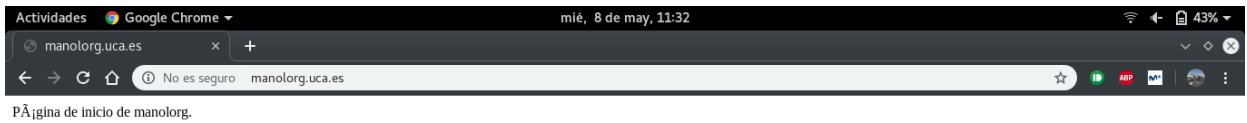


Figura 12: Sin UTF8

Para activar el UTF8, nos dirigimos a la edición de directivas globales (en la pestaña "global configuration") y añadimos la siguiente directiva al final³:

```
adddefaultcharset utf-8
```

³Si estamos usando Debian/Jessie debemos cambiar el archivo `/etc/apache2/conf-enabled/charset.conf` y descomentar la instrucción que dice: `#AddDefaultCharset UTF-8`.

Una vez añadida la directiva mencionada, podemos ver como sí que funciona la codificación UTF8:

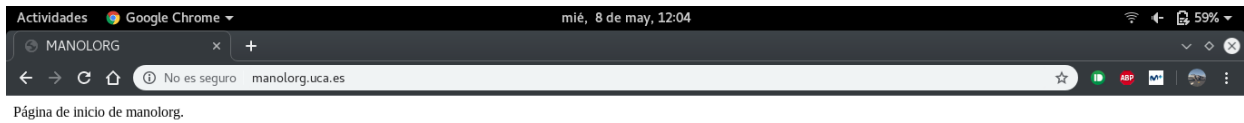


Figura 13: Con UTF8

3. Caché

Las imágenes (png, jpg) se cachearán para agilizar la web (hay muchas fotos). Esto se hará por medio del plugin expires.

Para ello, iremos a “Configure Apache Modules” y seleccionamos la opción “Expires”.

Para descargar la imagen usar el siguiente comando:

```
wget dirección
```

Y la ubicaremos en el “Document root” del servidor⁴.

Siendo *dirección*, la dirección de la imagen.

Para configurar la expiración de la caché, nos dirigimos al “Edit directives” del virtual host (*manolorg.uca.es*) y añadimos en el apartado de dicho directorio el siguiente código:

```
ExpiresActive On
ExpiresByType image/jpeg "access plus 3 minutes"
```

Para comprobar si funciona abrimos una consola en nuestro ordenador y ejecutamos el comando:

```
curl --head manolorg.uca.es/UCA.jpg
```

⁴La hemos renombrado como *UCA.jpg* para que sea más corto.

Una vez hecho eso, obtenemos lo siguiente:

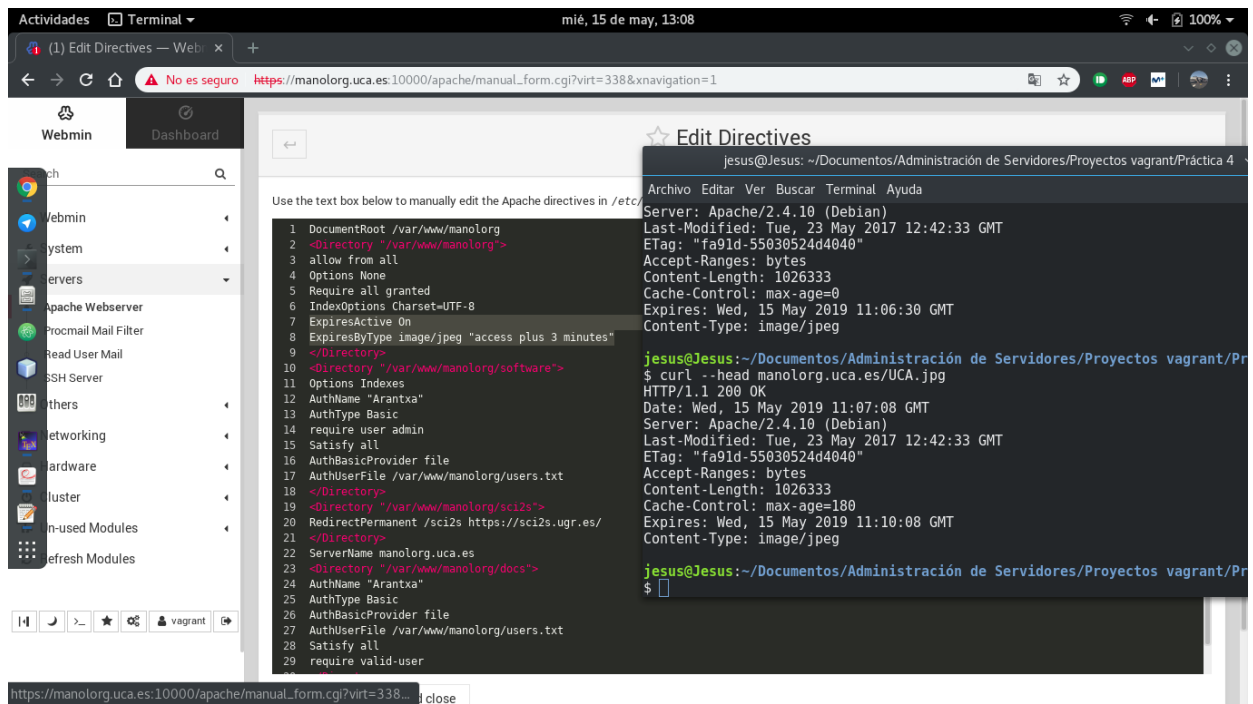


Figura 14: Configuración y resultado del expires.

4. Redirección

- Hay un problema en la base de datos con los nombres, y algunos ficheros aparecen erróneamente como .jpeg de forma errónea. Se desea que en el subdirectorio /imagenes todos los ficheros con extensión .jpeg se busquen con extensión .jpg.

Para ello, creamos el subdirectorio /imagenes en el directorio /var/www/manolorg.

A continuación, creamos el directory /imagenes en webmin.

En “Global configuration” nos vamos a “Configure Apache Modules” y activamos el módulo rewrite.

Luego, nos dirigimos al “Edit directives” de manolorg y añadimos al final lo siguiente:

```
RewriteEngine On
RewriteRule "^imagenes/(.*)jpeg" "/imagenes/$1.jpg" [R]
Es un acento circunflejo.
```

Entonces, al buscar `http://manolorg.uca.es/imagenes/UCA.jpeg`, nos mostrará `http://manolorg.uca.es/imagenes/UCA.jpg` y nos cambiará la dirección como podemos ver en la siguiente imagen.



Figura 15: Imagen del subdirectorio imágenes.

- Modificar la condición anterior para que sólo cambie las extensiones a .jpg cuando no exista el fichero con la extensión .jpeg.

Para ello tendremos que añadir una condición al “Edit directives” del “Global configuration” poniéndole lo siguiente:

```
RewriteCond%Document_root%Request_Filename !-f
```

Luego, probamos a entrar en: <http://manolorg.uca.es/imagenes/UCA2.jpeg> y nos mostrará la “jpeg” aunque exista la “jpg”.



Figura 16: Imagen jpeg.

- El dominio `manolorg.uca.es` tiene diferentes páginas para cada usuario. Estas páginas se sirven de forma dinámica mediante `php`. Así, la URL `manolorg.uca.es/personalWeb.php?user=CoboMJ` mostrará la página personal del usuario CoboMJ donde se podría ver el listado de sus publicaciones. Necesitamos convertir esta URL en una URL más amigable del tipo `manolo.uca.es/team/CoboMJ`. Para poder probar el funcionamiento, se deberá crear la página `php` `personalWeb.php` que lo único que hará es mostrar un mensaje con el parámetro pasado por `get`.

Para ello, crearemos el fichero `personalWeb.php` en `/var/www/manolorg/` con un `get`:

```
1  <?php
2  echo 'Hola ' . htmlspecialchars($_GET["user"]) . '!';
3  ?>
```

Ahora probamos a entrar en `http://manolorg.uca.es/personalWeb.php?user=JAR` y veremos lo siguiente:

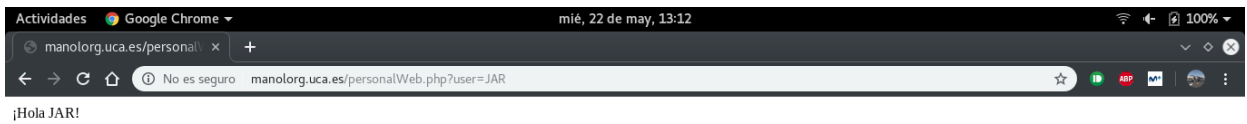


Figura 17: Comprobación php.

Ahora, añadimos la siguiente línea en el “Edit directives” de “Global configuration”:

```
RewriteRule "^team/(.*)" "/personalWeb.php?user=$1"
```

Para comprobar que funciona nos dirigimos a la siguiente dirección:
`manolorg.uca.es/team/JAR2`, y nos sale lo siguiente:

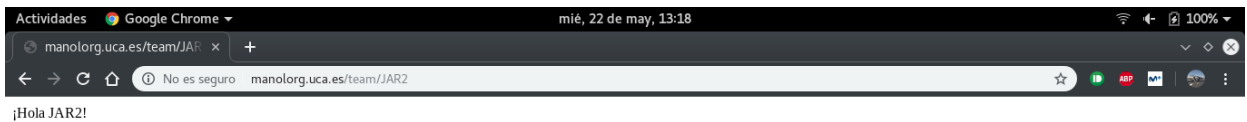


Figura 18: Comprobación de php con redirección.

5. Balanceo de carga

Dado que esperamos mucho servicios, las páginas se servirán mediante un balanceo de carga, entre los servidores `nodo01.manolorg.uca.es`, `nodo02.manolorg.uca.es`, y `nodo03.manolorg.uca.es`.

El servidor `nodo02` gestionará la mitad de peticiones que `nodo01`. El último es un servidor de respaldo que funcionará sólo en el caso de que los anteriores fallen.

Para ello, haremos cuatro máquinas virtuales e instalaremos webmin en la máquina donde se encuentra el balanceo de carga y modificaremos el “`index.html`” de las otras tres máquinas virtuales con el siguiente código:

```
1 <html>
2 <h1>NODO 1</h1>
3 </html>
```

Para configurar el balanceo de carga tendremos que habilitar los siguientes módulos de apache:

- `lbmethod_bybusyness`
- `lbmethod_byrequests`
- `lbmethod_bytraffic`
- `lbmethod_heartbeat`
- `proxy_balancer`
- `proxy_html`
- `proxy_http`

Luego, nos dirigimos al host virtual por defecto de la máquina que actúa como balanceador y, en “Edit directives” ponemos lo siguiente:

```
ProxyPass / balancer://hotcluster/
<Proxy balancer://hotcluster>
BalancerMember http://192.168.1.11 loadfactor=2
BalancerMember http://192.168.1.12 loadfactor=1
BalancerMember http://192.168.1.13 status=+H
ProxySet lbmethod=bytraffic
</Proxy>
```

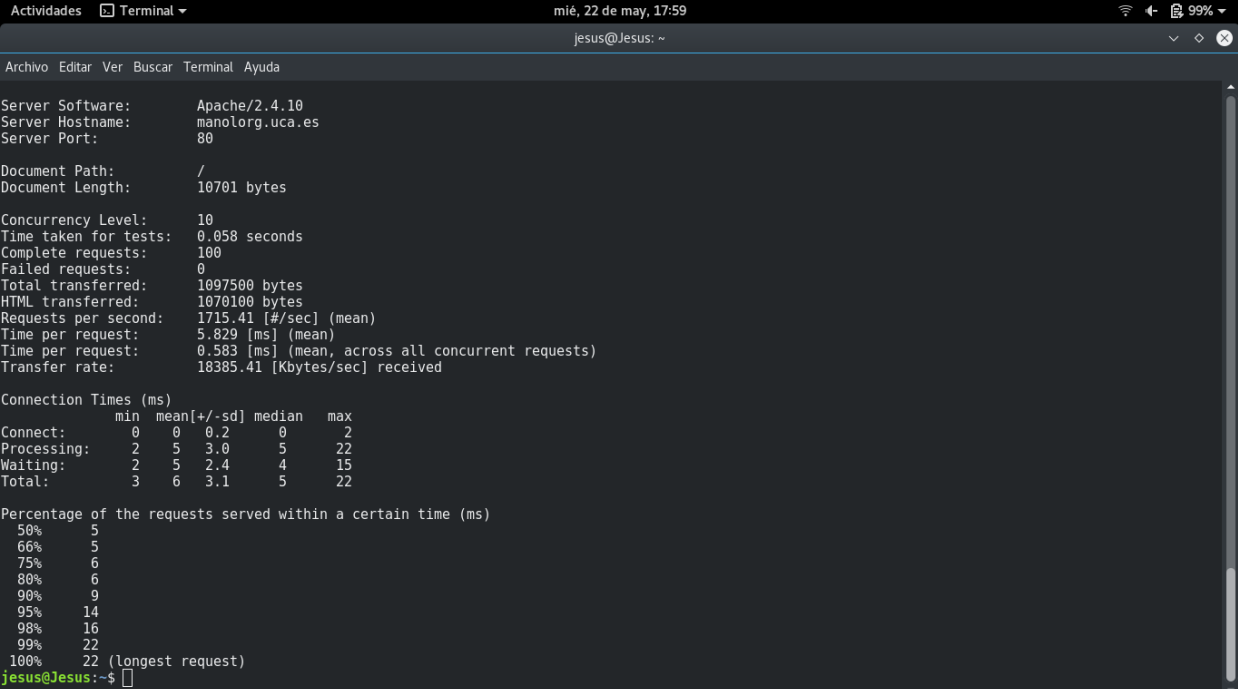
El `loadfactor` es para que el nodo1 reciba el doble de peticiones que el nodo2
Una vez configurado, debemos comprobar el rendimiento que obtenemos. Para ello:

- Probar, usando alguna herramienta de medición, el número de peticiones por segundo sin usar balanceo de carga.

Para ello, comentamos las líneas que habilitan el balanceo de carga en webmin y, con todos los servidores encendidos y sin balanceo de carga, lanzaremos el siguiente comando desde nuestra terminal:

```
ab -n 100 -c 10 http://manolorg.uca.es/
```

Y obtendremos el siguiente resultado:



```
Server Software: Apache/2.4.10
Server Hostname: manolorg.uca.es
Server Port: 80
Document Path: /
Document Length: 10701 bytes
Concurrency Level: 10
Time taken for tests: 0.058 seconds
Complete requests: 100
Failed requests: 0
Total transferred: 1097500 bytes
HTML transferred: 1070100 bytes
Requests per second: 1715.41 [#/sec] (mean)
Time per request: 5.829 [ms] (mean)
Time per request: 0.583 [ms] (mean, across all concurrent requests)
Transfer rate: 18385.41 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median  max
Connect:  0    0  0.2   0    2
Processing:  2    5  3.0   5   22
Waiting:  2    5  2.4   4   15
Total:  3    6  3.1   5   22

Percentage of the requests served within a certain time (ms)
 50%    5
 66%    5
 75%    6
 80%    6
 90%    9
 95%   14
 98%   16
 99%   22
100%   22 (longest request)
jesus@Jesus:~$
```

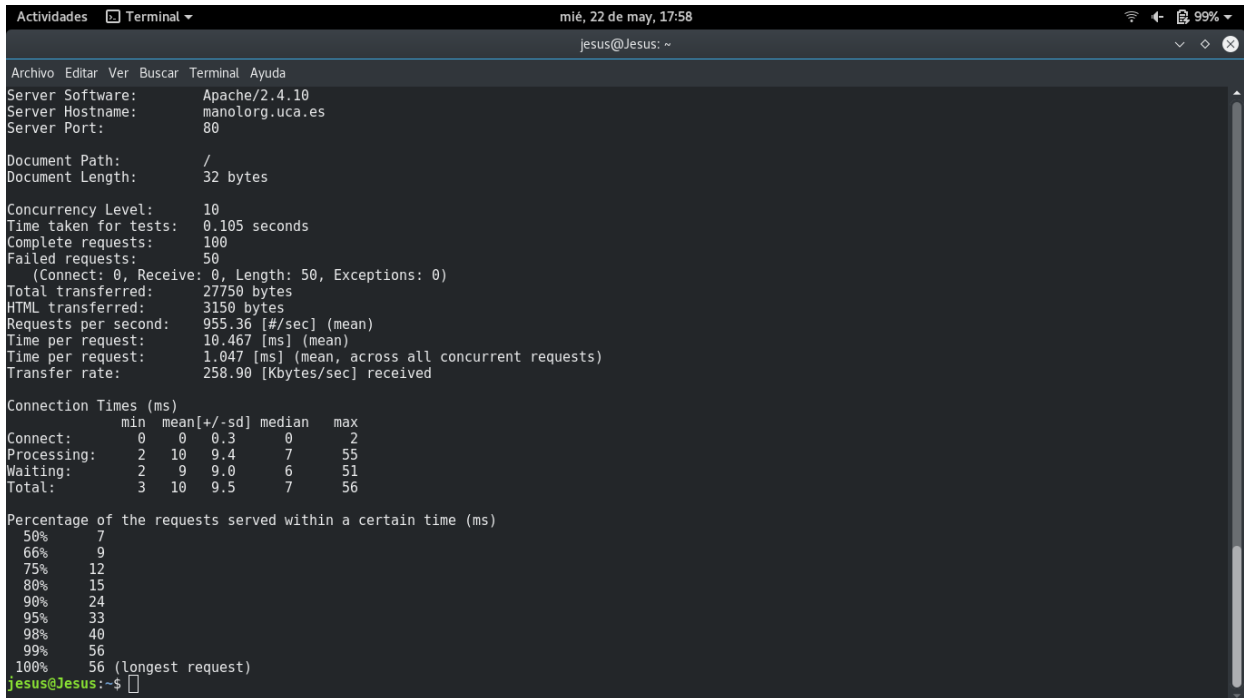
Figura 19: Sin balanceo.

- Indicar, usando la misma herramienta, usando el balanceo de carga anterior.

Para ello, con todos los servidores encendidos y con balanceo de carga lanzaremos el siguiente comando desde nuestra terminal:

```
ab -n 100 -c 10 http://manolorg.uca.es/
```

Y obtendremos el siguiente resultado:



```
Actividades Terminal mié, 22 de may, 17:58 99%
jesus@Jesus: ~
Archivo Editar Ver Buscar Terminal Ayuda
Server Software: Apache/2.4.10
Server Hostname: manolorg.uca.es
Server Port: 80
Document Path: /
Document Length: 32 bytes
Concurrency Level: 10
Time taken for tests: 0.105 seconds
Complete requests: 100
Failed requests: 50
  (Connect: 0, Receive: 0, Length: 50, Exceptions: 0)
Total transferred: 27750 bytes
HTML transferred: 3150 bytes
Requests per second: 955.36 [#/sec] (mean)
Time per request: 10.467 [ms] (mean)
Time per request: 1.047 [ms] (mean, across all concurrent requests)
Transfer rate: 258.90 [Kbytes/sec] received

Connection Times (ms)
      min  mean[+/-sd] median  max
Connect:    0     0  0.3      0    2
Processing:  2    10  9.4      7   55
Waiting:    2     9  9.0      6   51
Total:      3    10  9.5      7   56

Percentage of the requests served within a certain time (ms)
 50%    7
 66%    9
 75%   12
 80%   15
 90%   24
 95%   33
 98%   40
 99%   56
100%   56 (longest request)
jesus@Jesus:~$
```

Figura 20: Con balanceo.