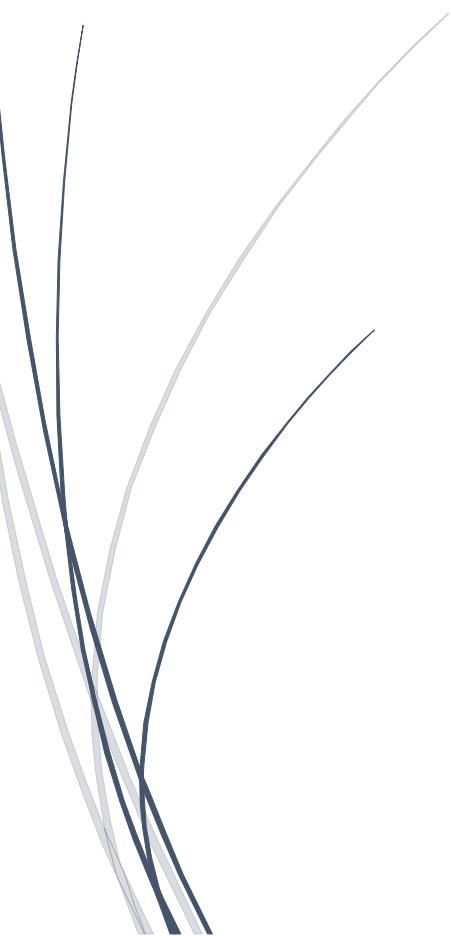




DESPLIEGUE DE APLICACIONES WEB

CONFIGURACIÓN DE APACHE HTTP SERVER



ÍNDICE

INTRODUCCIÓN	3
A. REQUISITOS PARA LA PRÁCTICA	3
INSTALACIÓN DEL SERVIDOR WEB APACHE	4
A. DESPLIEGUE DE NUESTRO PRIMER SITIO WEB EN APACHE	5
B. COMPROBACIÓN DEL FUNCIONAMIENTO EN NUESTRA MÁQUINA CLIENTE	8
C. ARCHIVOS .LOG	9
USO DE LA AUTENTICACIÓN BÁSICA EN APACHE	9
A. CREAR ARCHIVO .htpasswd.....	9
B. CONFIGURACIÓN DEL SITIO WEB PARA QUE TENGA AUTENTICACIÓN BÁSICA	10
C. RESTRICCIÓN DE ACCESO POR IP	13
D. COMBINACIÓN DE RESTRICCIÓN DE IP Y AUTENTICACIÓN BÁSICA	14
E. RESTRICCIÓN DE ACCESO SOBRE UNA PARTE CONCRETA DE LA WEB.....	15
CREACIÓN DE UN PROXY INVERSO EN APACHE	17
A. CONFIGURACIÓN DE LA MÁQUINA QUE ACTÚA DE SERVIDOR WEB	17
B. CONFIGURACIÓN DEL FICHERO webserver.conf	19
C. CONFIGURACIÓN DE LA MÁQUINA QUE ACTUARÁ DE PROXY INVERSO.....	20
D. CONFIGURACIÓN DE LA MÁQUINA CLIENTE Y COMPROBACIÓN DEL FUNCIONAMIENTO	22
BALANCEADOR DE CARGA EN APACHE.....	23
A. CONFIGURACIÓN DEL SERVIDOR WEB 1	23
B. CONFIGURACIÓN DEL SERVIDOR WEB 2	25
C. CONFIGURACIÓN DEL PROXY INVERSO-BALANCEADOR DE CARGA.....	26
D. CONFIGURACIÓN DE LA MÁQUINA CLIENTE	28
CONFIGURACIÓN DEL BALANCEADOR DE CARGA CON SSL.....	32
A. CREACIÓN DE UN CERTIFICADO AUTOFIRMADO	32
B. CONFIGURACIÓN SSL EN EL PROXY INVERSO	33
C. ACCESO DESDE EL NAVEGADOR	35

INTRODUCCIÓN

En anteriores prácticas realizamos la configuración del servidor web NGINX, en esta vamos a hacer las mismas configuraciones pero sobre el servidor web Apache.

Como veremos en los diferentes apartados de esta práctica, la configuración de este servidor no difiere en gran medida a la de NGINX.

A modo de introducción podemos decir que Apache es posiblemente el servidor HTTP más usado y, por tanto, cuenta con una amplia aceptación en la red. Su lanzamiento inicial fue en 1995 y actualmente se encuentra en la versión 2.4.58 lanzada en octubre de 2023. Se trata de un servidor desarrollado y mantenido por una comunidad de usuarios bajo la supervisión de la Apache Software Foundation dentro del proyecto HTTP Server (`httpd`).

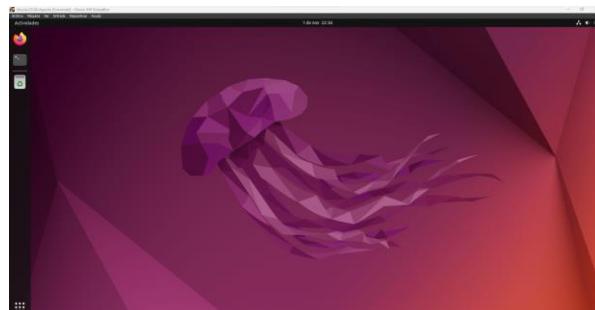
A. REQUISITOS PARA LA PRÁCTICA

Para comenzar con la práctica necesitamos una máquina virtual que tenga instalado el sistema operativo Debian 12. Esta máquina va a realizar la función de servidor.

En nuestro caso vamos a partir con algunos paquetes ya instalados: el paquete tree, el paquete de git y el paquete net-tools.



También debemos contar con una máquina virtual con Ubuntu 22.04 instalado, esta máquina se encargará de hacer las funciones de cliente.



Las características de ambas máquinas son 2GB de RAM y 2 procesadores asignados, así como 20GB de espacio en disco.

INSTALACIÓN DEL SERVIDOR WEB APACHE

Para comenzar a utilizar Apache debemos instalar el paquete necesario en nuestra máquina servidora. Para ello primero vamos a comprobar si existen dependencias a actualizar en nuestra distro de Debian 12:

```
sudo apt-get update
```

```
albertom-servidor@debian-deaw:~$ sudo apt-get update
Obj:1 http://security.debian.org/debian-security bookworm-security InRelease
Obj:2 http://deb.debian.org/debian bookworm InRelease
Obj:3 http://deb.debian.org/debian bookworm-updates InRelease
Leyendo lista de paquetes... Hecho
```

En nuestro caso no tenemos que actualizar ninguna dependencia, en el caso de que tuviéramos alguna deberíamos instalarlas con la opción upgrade de apt-get.

Una vez hecho esto podemos instalar el paquete de apache utilizando el comando:

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

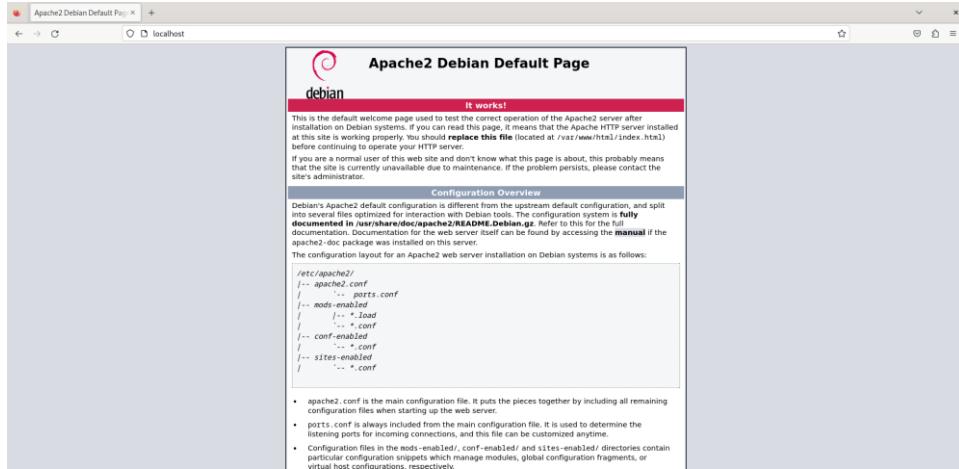
```
albertom-servidor@debian-deaw:~$ sudo apt-get install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  apache2-data apache2-utils
Paquetes sugeridos:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-data apache2-utils
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 577 kB de archivos.
Se utilizarán 1.890 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
```

Si todo ha funcionado correctamente, ahora si comprobamos el estado del servicio debería aparecernos que se encuentra activo:

```
sudo systemctl status apache2.service
```

```
albertom-servidor@debian-deaw:~$ sudo systemctl status apache2.service
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Wed 2023-11-01 20:54:41 CET; 29min ago
     Docs: https://httpd.apache.org/docs/2.4/
 Main PID: 3395 (apache2)
    Tasks: 55 (limit: 2284)
   Memory: 10.7M
      CPU: 416ms
    CGroup: /system.slice/apache2.service
            └─3395 /usr/sbin/apache2 -k start
                ├─3396 /usr/sbin/apache2 -k start
                ├─3397 /usr/sbin/apache2 -k start
```

Si abrimos en el navegador una ventana que nos lleve a `http://localhost` se debería poder ver la siguiente web de bienvenida de Apache2:



Esta instalación nos ha creado un directorio `/etc/apache2/` donde se guardan los archivos de configuración principal de Apache HTTP Server.

```
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/
total 96
drwxr-xr-x  8 root root  4096 nov  1 20:54 .
drwxr-xr-x 134 root root 12288 nov  1 14:11 ..
-IW-I--I--  1 root root  7178 abr 13 2023 apache2.conf
drwxr-xr-x  2 root root  4096 nov  1 20:54 conf-available
drwxr-xr-x  2 root root  4096 nov  1 20:54 conf-enabled
-IW-r--r--  1 root root 1782 jun  9 2022 envvars
-IW-r--r--  1 root root 31063 jun  9 2022 magic
drwxr-xr-x  2 root root 12288 nov  1 20:54 mods-available
drwxr-xr-x  2 root root  4096 nov  1 20:54 mods-enabled
-IW-r--r--  1 root root  274 abr 13 2023 ports.conf
drwxr-xr-x  2 root root  4096 nov  1 20:54 sites-available
drwxr-xr-x  2 root root  4096 nov  1 20:54 sites-enabled
```

A. DESPLIEGUE DE NUESTRO PRIMER SITIO WEB EN APACHE

Vamos a realizar la misma prueba que realizamos con NGINX para comprobar su funcionamiento con un sitio web real, para ello vamos a descargarnos la web estática del correspondiente repositorio de git:

```
sudo git clone https://github.com/cloudacademy/static-website-example
albertom-servidor@debian-deaw:~$ sudo git clone https://github.com/cloudacademy/static-website-example
Clonando en 'static-website-example'...
remote: Enumerating objects: 69, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 69 (delta 8), reused 8 (delta 8), pack-reused 45
Recibiendo objetos: 100% (69/69), 668.08 KiB | 3.75 MiB/S, listo.
Resolviendo deltas: 100% (9/9), listo.
```

Vamos a moverlo al directorio `/var/www` cambiándole el nombre a `web_estatica`:

```
albertom-servidor@debian-deaw:~$ sudo mv static-website-example/ /var/www/web_estatica
albertom-servidor@debian-deaw:~$ ls -la /var/www/
total 16
drwxr-xr-x  4 root root 4096 nov  1 21:34 .
drwxr-xr-x 12 root root 4096 nov  1 11:33 ..
drwxr-xr-x  2 root root 4096 nov  1 20:54 html
drwxr-xr-x  6 root root 4096 nov  1 21:31 web_estatica
```

Al igual que ocurre en NGINX en Apache2 tenemos al usuario del servidor web www-data por lo que debemos darle la propiedad y los permisos necesarios sobre web_estatica para poder acceder al sitio:

```
sudo chown -R www-data:www-data /var/www/web_estatica/
```

```
sudo chmod 755 /var/www/web_estatica/
```

```
albertom-servidor@debian-deaw:~$ sudo chown -R www-data:www-data /var/www/web_estatica/
albertom-servidor@debian-deaw:~$ sudo chmod 755 /var/www/web_estatica/
albertom-servidor@debian-deaw:~$ ls -la /var/www
total 16
drwxr-xr-x  4 root      root      4096 nov  1 21:34 .
drwxr-xr-x 12 root      root      4096 nov  1 11:33 ..
drwxr-xr-x  2 root      root      4096 nov  1 20:54 html
drwxr-xr-x  6 www-data  www-data  4096 nov  1 21:31 web_estatica
```

De esta forma el usuario www-data podrá acceder y gestionar los ficheros del sitio web.

Ahora vamos a configurar el sitio web y lo haremos sobre el directorio /etc/apache2/ y, al igual que pasaba con NGINX tenemos tanto un directorio sites-available/ como un directorio sites-enabled/ que tienen la misma finalidad.

Si observamos el estado actual del directorio sites-available/ veremos que tenemos dos ficheros:

```
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/sites-available/
total 20
drwxr-xr-x  2 root      root      4096 nov  1 21:38 .
drwxr-xr-x  8 root      root      4096 nov  1 20:54 ..
-rw-r--r--  1 root      root    1286 abr 13 2023 000-default.conf
-rw-r--r--  1 root      root    6195 abr 13 2023 default-ssl.conf
```

A diferencia de NGINX y como veremos en su correspondiente apartado, en Apache se utiliza un archivo para las conexiones HTTP y otro archivo para las conexiones seguras HTTPS a través de SSL.

Por ahora vamos a centrarnos en la conexión por defecto y para ello vamos a hacer una copia del archivo 000-default.conf:

```
sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-
available/estatica.conf
```

```
albertom-servidor@debian-deaw:~$ sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/estatica.conf
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/sites-available/
total 24
drwxr-xr-x  2 root      root      4096 nov  1 21:52 .
drwxr-xr-x  8 root      root      4096 nov  1 20:54 ..
-rw-r--r--  1 root      root    1286 abr 13 2023 000-default.conf
-rw-r--r--  1 root      root    6195 abr 13 2023 default-ssl.conf
-rw-r--r--  1 root      root    1286 nov  1 21:52 estatica.conf
```

Y vamos a proceder a su configuración para lo cual usaremos nano al igual que hicimos en las prácticas de NGINX:

```
sudo nano /etc/apache2/sites-available/estatica.conf
```

```
albertom-servidor@debian-deaw:~$ sudo nano /etc/apache2/sites-available/estatica.conf
```

En este archivo tenemos las siguientes partes:

```
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin admin@correo.com
    ServerName www.estatica.es
    DocumentRoot /var/www/web_estatica

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

- <VirtualHost *:80>. Esta sección de host virtual coincide con cualquier solicitud que se realice por el puerto 80, el puerto HTTP predeterminado.
- ServerAdmin. En esta directiva tenemos que incluir un correo electrónico para que el administrado del sitio pueda recibir correos electrónicos.
- ServerName. Establece el nombre de dominio.
- DocumentRoot. Ubicación del directorio raíz del sitio web.
- ErrorLog. Establece el fichero .log en el que se almacenarán los logs de error. Por defecto es el archivo error.log del directorio /var/log/apache2/.
- CustomLog. Establece el fichero .log en el que se almacenarán los logs de acceso. Por defecto es el archivo access.log del directorio /var/log/apache2/.

NOTA: Adicional a esto en el archivo /etc/apache2/ports.conf podemos configurar los puertos por defecto tanto para HTTP como para HTTPS:

```
GNU nano 7.2
# If you just change the port
# have to change the VirtualH
# /etc/apache2/sites-enabled/

Listen 80

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>
```

Una vez hecho esto tenemos el archivo de sites-available/ completado y debemos de realizar un enlace simbólico en nuestro sites-enabled/ pero a diferencia de NGINX en apache tenemos un comando propio tanto para

crear enlaces simbólicos (a2ensite) como para destruirlos (a2dissite) si estos archivos a los que hagamos referencia tienen la extensión .conf, vamos a crear un enlace simbólico de nuestro archivo estatica.conf.

sudo a2ensite estatica.conf

```
albertom-servidor@debian-deaw:~$ sudo a2ensite estatica.conf
Enabling site estatica.
To activate the new configuration, you need to run:
    systemctl reload apache2
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 nov  1 22:10 .
drwxr-xr-x 8 root root 4096 nov  1 20:54 ..
lrwxrwxrwx 1 root root   35 nov  1 20:54 000-default.conf -> ../sites-available/000-default.conf
lrwxrwxrwx 1 root root   32 nov  1 22:10 estatica.conf -> ../sites-available/estatica.conf
```

Y reiniciamos el servicio de apache con el comando correspondiente de systemctl:

sudo systemctl restart apache2.service

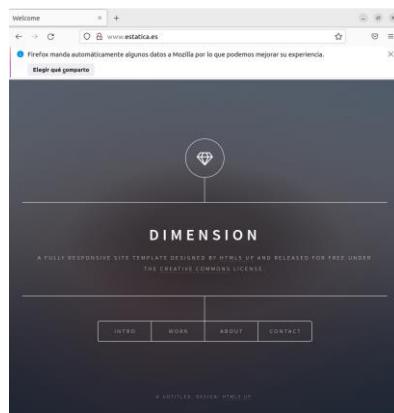
```
albertom-servidor@debian-deaw:~$ sudo systemctl restart apache2.service
albertom-servidor@debian-deaw:~$ sudo systemctl status apache2.service
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Wed 2023-11-01 22:14:22 CET; 7s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 4852 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 4857 (apache2)
   Tasks: 55 (limit: 2284)
    Memory: 18.8M
      CPU: 58ms
     CGroup: /system.slice/apache2.service
             ├─4857 /usr/sbin/apache2 -k start
             ├─4858 /usr/sbin/apache2 -k start
             └─4859 /usr/sbin/apache2 -k start
```

B. COMPROBACIÓN DEL FUNCIONAMIENTO EN NUESTRA MÁQUINA CLIENTE

Ahora vamos a utilizar una máquina virtual cliente con un Ubuntu 22.04 instalado. En su archivo /etc/hosts colocamos la IP del servidor junto con la dirección www.estatica.es.

```
GNU nano 6.2                               /etc/hosts
127.0.0.1           localhost
127.0.1.1          albertomcliente-VirtualBox
10.0.2.24          www.estatica.es
```

Si ahora accedemos a www.estatica.es desde el navegador accederemos a la web creada:



C. ARCHIVOS .LOG

Si ahora accedemos al log de acceso del servidor Apache podremos ver que se ha registrado este acceso:

```
sudo tail /var/log/apache2/access.log
```

```
albertom-servidor@debian-deaw:~$ sudo tail /var/log/apache2/access.log
10.0.2.15 - - [01/Nov/2023:22:31:51 +0100] "GET /images/pic01.jpg HTTP/1.1" 200 1
10.0.2.15 - - [01/Nov/2023:22:31:51 +0100] "GET /images/pic03.jpg HTTP/1.1" 200 5
10.0.2.15 - - [01/Nov/2023:22:31:51 +0100] "GET /assets/js/jquery.min.js HTTP/1.1
10.0.2.15 - - [01/Nov/2023:22:31:51 +0100] "GET /assets/js/util.js HTTP/1.1" 200
10.0.2.15 - - [01/Nov/2023:22:31:51 +0100] "GET /assets/js/main.js HTTP/1.1" 200
10.0.2.15 - - [01/Nov/2023:22:31:51 +0100] "GET /assets/css/font-awesome.min.css
```

Se han recibido múltiples solicitudes de tipo GET desde una dirección IP 10.0.2.15 que coincide con la dirección IP de nuestra máquina virtual Ubuntu:

```
alberton-cliente@albertoMartinezPerez:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel
    link/ether 08:00:27:fa:b9:2b brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic nopref
        valid_lft 364sec preferred_lft 364sec
    inet6 fe80::9b8b:8a6d:51d9:6bb8/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

USO DE LA AUTENTICACIÓN BÁSICA EN APACHE

Ahora vamos a ver cómo se ejecuta una autenticación básica en apache. Lo primero que debemos hacer es activar el modo de autenticación básica con el siguiente comando:

```
sudo a2enmod auth_basic
```

```
albertom-servidor@debian-deaw:~$ sudo a2enmod auth_basic
Considering dependency authn_core for auth_basic:
Module authn_core already enabled
Module auth_basic already enabled
```

A. CREAR ARCHIVO .htpasswd

Para hacerlo vamos a usar la utilidad htpasswd que nos va a generar un archivo .htpasswd en /etc/apache2. Vamos a crear un usuario llamado alberto

```
sudo htpasswd -c /etc/apache2/.htpasswd alberto
```

```
albertom-servidor@debian-deaw:~$ sudo htpasswd -c /etc/apache2/.htpasswd alberto
New password:
Re-type new password:
Adding password for user alberto
```

Y creamos un segundo usuario martinez (en este caso debemos quitar la opción -c del comando):

```
sudo htpasswd /etc/apache2/.htpasswd martinez
```

```
albertom-servidor@debian-deaw:~$ sudo htpasswd /etc/apache2/.htpasswd martinez
New password:
Re-type new password:
Adding password for user martinez
```

Podemos hacer un cat del archivo .htpasswd para ver lo que se ha guardado:

```
sudo cat /etc/apache2/.htpasswd
```

```
albertom-servidor@debian-deaw:~$ sudo cat /etc/apache2/.htpasswd
alberto:$apr1$Uu3bZKKA$KZfItyryoEJ4PPA8qNYu3.
martinez:$apr1$wkaKhMSo$Rd2Rc2zCxivU7itz3kN8u/
```

B. CONFIGURACIÓN DEL SITIO WEB PARA QUE TENGA AUTENTICACIÓN BÁSICA

En esta sección de la práctica vamos a usar el sitio web www.academia.es que usamos en diferentes prácticas de NGINX.

Debemos asegurarnos de tener el sitio configurado como tenemos el sitio de www.estatica.es

Es decir:

- Tener el directorio creado en /var/www/ con la propiedad y los permisos correspondientes:

```
albertom-servidor@debian-deaw:~$ ls -la /var/www
total 20
drwxr-xr-x  5 root      root      4096 nov  2 19:41 .
drwxr-xr-x 12 root      root      4096 nov  1 11:33 ..
drwxr-xr-x  7 www-data  www-data  4096 mar 28 2019 fich_academia
drwxr-xr-x  2 root      root      4096 nov  1 20:54 html
drwxr-xr-x  6 www-data  www-data  4096 nov  1 21:31 web_estatica
```

- Tener el fichero de configuración en /etc/apache2/sites-available/:

```
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/sites-available/
total 28
drwxr-xr-x  2 root root 4096 nov  2 19:49 .
drwxr-xr-x  8 root root 4096 nov  2 19:05 ..
-rw-r--r--  1 root root 1286 abr 13 2023 000-default.conf
-rw-r--r--  1 root root 1436 nov  2 19:49 academia.conf
-rw-r--r--  1 root root 6195 abr 13 2023 default-ssl.conf
-rw-r--r--  1 root root 1319 nov  1 22:20 estatica.conf
```

- Generar el enlace simbólico (con el comando a2ensite) en el directorio /etc/apache2/sites-enabled:

```
albertom-servidor@debian-deaw:~$ sudo a2ensite academia.conf
Enabling site academia.
To activate the new configuration, you need to run:
  systemctl reload apache2
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/sites-enabled/
total 8
drwxr-xr-x  2 root root 4096 nov  2 19:52 .
drwxr-xr-x  8 root root 4096 nov  2 19:05 ..
lrwxrwxrwx  1 root root   35 nov  1 20:54 000-default.conf -> ../../sites-available/000-default.conf
lrwxrwxrwx  1 root root   32 nov  2 19:52 academia.conf -> ../../sites-available/academia.conf
lrwxrwxrwx  1 root root   32 nov  1 22:10 estatica.conf -> ../../sites-available/estatica.conf
```

Una vez hecho lo anterior debemos cambiar la configuración del archivo /etc/apache2/sites-available/academia.conf:

```
sudo nano etc/apache2/sites-available/academia.conf
```

Tenemos dos opciones para solicitar autenticación para el acceso: a través de la etiqueta <Directory> o a través de la etiqueta <Location>. En ambos casos debemos crear un bloque de código al archivo (dentro del bloque <VirtualHost>).

La principal diferencia entre ellas es que la primera afecta únicamente sobre objetos del sistema de archivos (por ejemplo, directorios) mientras que la segunda sólo afecta a URLs. También se diferencian en que en <Directory> debemos explicitar toda la ruta del fichero que queremos afectar (por ejemplo, <Directory /var/www/fich_academia/>) mientras que con <Location> se afecta en base al directorio del directorio del DocumentRoot (por ejemplo, <Location /contact.html>).

Por ello hemos decidido usar <Directory> para la autenticación que vamos a hacer sobre la raíz del sitio y <Location> para la autenticación específica sobre una parte de nuestra web y que vemos al final de esta sección:

```
<Directory /var/www/fich_academia/>
    AuthType Basic
    AuthName "Zona restringida"
    AuthUserFile /etc/apache2/.htpasswd
    Require valid-user
</Directory>
```

Sobre el directorio raíz del sitio web estamos usando las siguientes directivas:

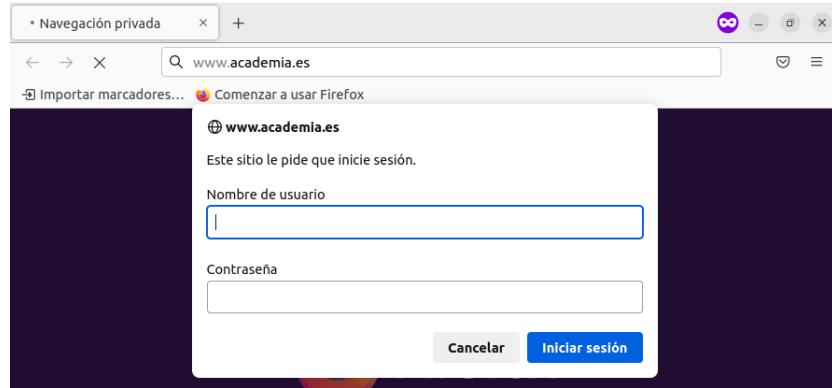
- AuthType: El tipo de autenticación que queremos utilizar en este caso vamos a usar la básica.
- AuthName: Nombre que tendrá la zona a la que vamos a controlar el acceso.
- AuthUserFile: La dirección donde hemos almacenado el archivo con los usuarios y contraseñas.
- Require: Establece que se requiere un usuario válido para acceder al directorio.

Una vez completada esta configuración debemos reiniciar el servicio de apache.

Vamos a comprobar el funcionamiento en nuestra máquina cliente Ubuntu, como siempre lo primero que debemos hacer es añadir el nuevo dominio al archivo /etc/hosts:

```
GNU nano 6.2
127.0.0.1      localhost
127.0.1.1      albertomcliente-VirtualBox
10.0.2.15       www.estatica.es
10.0.2.15       www.academia.es
```

Si ahora accedemos al sitio web www.academia.es nos aparecerá el siguiente mensaje:



Si iniciamos sesión de manera correcta, el acceso se guardará en el log de acceso, vamos a hacer inicio de sesión con el usuario 'alberto'.

Se nos permite el acceso a la web:



Y esta acción queda reflejado en el access.log del servidor:

```
sudo tail /var/log/apache2/access.log
```

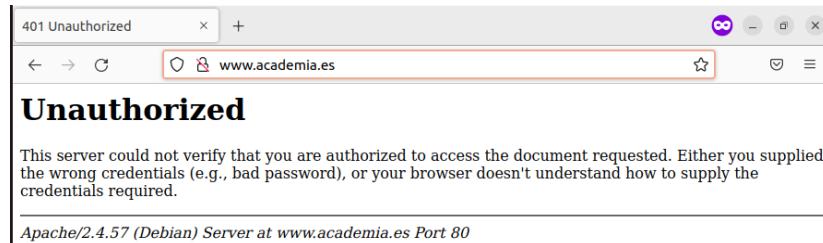
```
albertom-servidor@debian-deaw:~$ sudo tail /var/log/apache2/access.log
10.0.2.25 - alberto [02/Nov/2023:20:39:58 +0100] "GET /js/custom.js HTTP/1.1" 200 1388
10.0.2.25 - alberto [02/Nov/2023:20:39:58 +0100] "GET /css/animate.css HTTP/1.1" 200 4
10.0.2.25 - alberto [02/Nov/2023:20:39:58 +0100] "GET /css/font-awesome.min.css HTTP/1
10.0.2.25 - alberto [02/Nov/2023:20:39:58 +0100] "GET /css/responsiveslides.css HTTP/1
10.0.2.25 - alberto [02/Nov/2023:20:39:58 +0100] "GET /css/magnific-popup.css HTTP/1.1
10.0.2.25 - alberto [02/Nov/2023:20:39:58 +0100] "GET /css/flaticon.css HTTP/1.1" 200
10.0.2.25 - alberto [02/Nov/2023:20:39:58 +0100] "GET /css/timeline.css HTTP/1.1" 200
10.0.2.25 - alberto [02/Nov/2023:20:39:59 +0100] "GET /images/heading_main_border.png
```

De la misma manera si introducimos mal la contraseña este evento quedará almacenado en el archivo error.log, vamos a provocar un fallo de contraseña con el usuario 'martinez':

```
sudo tail /var/log/apache2/error.log
```

```
albertom-servidor@debian-deaw:~$ sudo tail -1 /var/log/apache2/error.log
[Thu Nov 02 20:43:40.437779 2023] [auth_basic:error] [pid 5358:tid 140067238524608] [client 10.0.2.25:42848] AH01617: user martinez: authentication failure for "/": Password Mismatch
```

En el caso de que pulsemos “cancelar” en la ventana desplegable de contraseña se nos desplegará el error 401 Unauthorized:



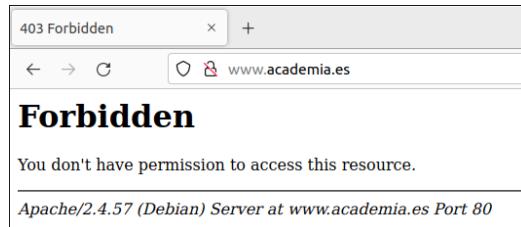
C. RESTRICCIÓN DE ACCESO POR IP

Al igual que en NGINX también podemos restringir el acceso por IP a nuestro sitio web. Vamos a restringir el acceso a la IP de la MV cliente y a permitir el acceso de la IP de la MV servidora:

```
sudo nano etc/apache2/sites-available/academia.conf
```

```
<Directory /var/www/fich_academia/>
    Order allow,deny
    Deny from 10.0.2.25
    Allow from 10.0.2.15
</Directory>
```

Si ahora intentamos acceder desde el navegador de nuestra máquina cliente obtendremos el error 403 Forbidden:



Quedando esta acción almacenada en el log de errores:

```
sudo tail /var/log/apache2/error.log
```

```
albertom-servidor@debian-deaw:~$ sudo tail -1 /var/log/apache2/error.log
[Thu Nov 02 21:22:00.860556 2023] [access_compat:error] [pid 7132:tid 139727340553920] [client 10.0.2.25:58254]
```

Sin embargo, desde el navegador de la máquina servidora podremos acceder sin problemas:



Esta acción se almacena en el log de accesos:

```
sudo tail /var/log/apache2/access.log
```

```
albertom-servidor@debian-deaw:~$ sudo tail -1 /var/log/apache2/access.log
10.0.2.15 - - [02/Nov/2023:21:23:20 +0100] "GET /js/jquery.min.js HTTP/1.1" 200
```

D. COMBINACIÓN DE RESTRICCIÓN DE IP Y AUTENTICACIÓN BÁSICA

Al igual que en NGINX podemos combinar la restricción de IP con la autenticación básica haciendo uso de la directiva Satisfy que, de nuevo, vuelve a tener dos valores posibles “All” para obligar al usuario que quiera acceder a completar ambas cosas o “Any” siendo necesario completar sólo una de ellas.

Para ello debemos modificar nuestro archivo de configuración:

```
sudo nano etc/apache2/sites-available/academia.conf
```

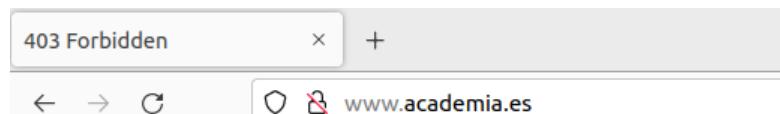
Si utilizamos la directiva con all:

```
<Directory /var/www/fich_academia/>
    Satisfy all

    AuthType Basic
    AuthName "Zona restringida"
    AuthUserFile /etc/apache2/.htpasswd
    Require valid-user

    Order allow,deny
    Deny from 10.0.2.25
    Allow from 10.0.2.15
</Directory>
```

Reiniciamos el servicio y probamos a intentar conectarnos. Como la directiva se encuentra en any seguiremos sin poder acceder a la web desde nuestra máquina Ubuntu porque nuestra IP se encuentra vetada:



Forbidden

You don't have permission to access this resource.

Apache/2.4.57 (Debian) Server at www.academia.es Port 80

De nuevo esto quedará almacenado en el log de errores:

```
albertom-servidor@debian-deaw:~$ sudo tail -1 /var/log/apache2/error.log
[Thu Nov 02 21:33:45.715939 2023] [access_compat:error] [pid 7480:tid 139912368019136] [client 10.0.2.25:38778]
```

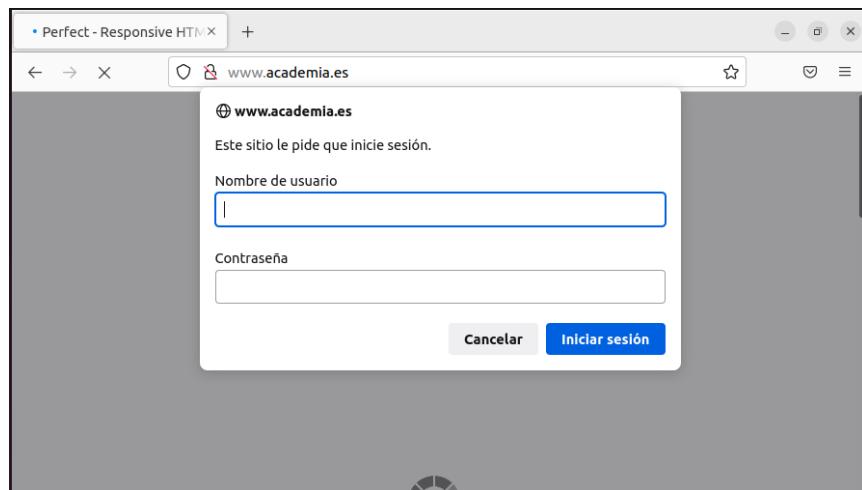
En cambio, si asignamos el valor “any” a la directiva Satisfy:

```
<Directory /var/www/fich_academia/>
    Satisfy any

    AuthType Basic
    AuthName "Zona restringida"
    AuthUserFile /etc/apache2/.htpasswd
    Require valid-user

    Order allow,deny
    Deny from 10.0.2.25
    Allow from 10.0.2.15
</Directory>
```

Reiniciamos el servicio y probamos a conectarnos en el cliente. Aunque la IP de la máquina cliente sigue vetada, se nos solicita el introducir usuario y contraseña:



Si iniciamos sesión correctamente podremos acceder a la web y esta acción quedará registrada en el log de accesos:

```
sudo tail /var/log/apache2/access.log
```

```
albertom-servidor@debian-deaw:~$ sudo tail -9 /var/log/apache2/access.log
10.0.2.25 - alberto [02/Nov/2023:21:40:16 +0100] "GET /js/jquery.magnific-popup.min.js HTTP/1.1" 200 10198 "
10.0.2.25 - alberto [02/Nov/2023:21:40:16 +0100] "GET /js/isotope.min.js HTTP/1.1" 200 596 "
10.0.2.25 - alberto [02/Nov/2023:21:40:16 +0100] "GET /js/form-validator.min.js HTTP/1.1" 200
10.0.2.25 - alberto [02/Nov/2023:21:40:16 +0100] "GET /js/jquery.pogo-slider.min.js HTTP/1.1" 200 30485 "
10.0.2.25 - alberto [02/Nov/2023:21:40:16 +0100] "GET /js/slider-index.js HTTP/1.1" 200 2947 "
10.0.2.25 - alberto [02/Nov/2023:21:40:16 +0100] "GET /js/smoothscroll.js HTTP/1.1" 200 21
10.0.2.25 - alberto [02/Nov/2023:21:40:16 +0100] "GET /js/contact-form-script.js HTTP/1.1" 200 21
10.0.2.25 - alberto [02/Nov/2023:21:40:16 +0100] "GET /js/images-loaded.min.js HTTP/1.1" 200 21
```

E. RESTRICCIÓN DE ACCESO SOBRE UNA PARTE CONCRETA DE LA WEB

Vamos a solicitar autenticación básica si el usuario intenta acceder a la sección “About” de la web.

Si utilizamos el comando `tree` podremos ver cómo se estructura nuestro directorio del sitio web:

```
sudo tree /var/www/fich_academia/
```

```
albertom-servidor@debian-deaw:~$ sudo tree /var/www/fich_academia/
/var/www/fich_academia/
├── about.html
├── contact.html
└── css
    ├── animate.css
    ├── bootstrap.min.css
    ├── custom.css
    ├── flashy.min.css
    ├──扁平化.css
    ├── font-awesome.min.css
    ├── magnific-popup.css
    ├── pogo-slider.min.css
    ├── responsive.css
    └── responsiveslides.css
```

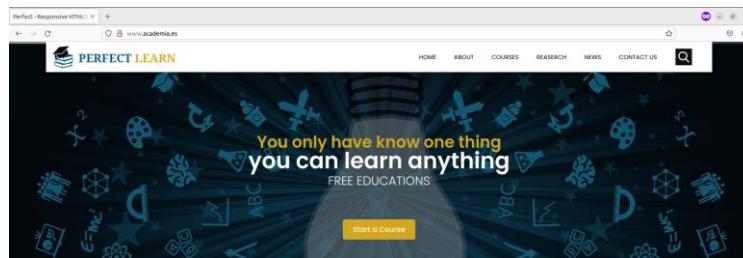
Modificamos el archivo de configuración del sitio:

```
sudo nano etc/apache2/sites-available/academia.conf
```

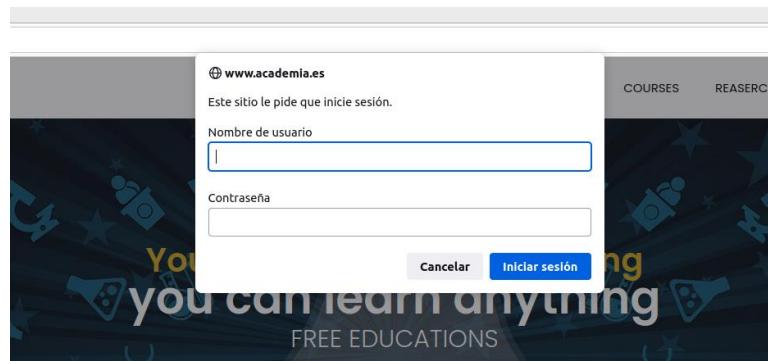
Borramos el `<Directory>` que teníamos creado y creamos un `<Location>` que afecte al archivo `about.html` que, como hemos visto en el `tree` cuelga directamente de `/var/www/fich_academia` que es nuestro `DocumentRoot`.

```
<Location /about.html>
    AuthType Basic
    AuthName "Zona restringida"
    AuthUserFile /etc/apache2/.htpasswd
    Require valid-user
</Location>
```

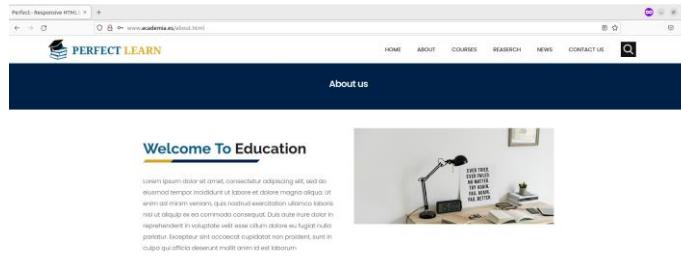
Tras reiniciar el servicio probamos a acceder a la web y podremos acceder sin problemas a la web principal:



Pero si hacemos clic en `about` nos pedirá autenticarnos:



Si nos autenticamos de forma correcta podremos acceder:



Esta acción queda registrada en el log de accesos:

```
albertom-servidor@debian-deaw:~$ sudo tail -13 /var/log/apache2/access.log
10.0.2.25 - alberto [02/Nov/2023:22:04:11 +0100] "GET /about.html HTTP/1.1" 200 3842 "http://www.academia.es/"
```

CREACIÓN DE UN PROXY INVERSO EN APACHE

En esta parte de la práctica vamos a crear un segundo servidor Apache para que funcione como proxy inverso de forma que filtre las conexiones cliente-servidor, el cliente ya nunca se conectará directamente al servidor sino al proxy que se hará pasar por el servidor y que al recibir solicitudes del cliente realizará peticiones al servidor.

A. CONFIGURACIÓN DE LA MÁQUINA QUE ACTÚA DE SERVIDOR WEB

Vamos a configurar la máquina que hemos estado usando hasta ahora para utilizarla como servidor web. Lo que vamos a hacer en primer lugar es eliminar todos los sitios web salvo el correspondiente a estática, por tanto:

- En el directorio `/var/www` sólo debemos tener el directorio `web_estatica/`, tenemos que eliminar el `html/` (que tiene el sitio web base de Apache) y `fich_academia/` (que creamos en la sección anterior para comprobar la autenticación). Para ello usamos dos comandos:

```
sudo rm -Rf /var/www/html/
```

```
sudo rm -Rf /var/www/fich_academia/
```

```
albertom-servidor@debian-deaw:~$ ls -la /var/www/
total 20
drwxr-xr-x  5 root      root      4096 nov  2 19:41 .
drwxr-xr-x 12 root      root      4096 nov  1 11:33 ..
drwxr-xr-x  7 www-data  www-data  4096 mar 28  2019 fich_academia
drwxr-xr-x  2 root      root      4096 nov  1 20:54 html
drwxr-xr-x  6 www-data  www-data  4096 nov  1 21:31 web_estatica
albertom-servidor@debian-deaw:~$ sudo rm -Rf /var/www/html/
albertom-servidor@debian-deaw:~$ sudo rm -Rf /var/www/fich_academia/
albertom-servidor@debian-deaw:~$ ls -la /var/www/
total 12
drwxr-xr-x  3 root      root      4096 nov  3 07:29 .
drwxr-xr-x 12 root      root      4096 nov  1 11:33 ..
drwxr-xr-x  6 www-data  www-data  4096 nov  1 21:31 web_estatica
```

Además, vamos a renombrar el directorio web_estatica/ a webserver/

```
sudo mv /var/www/web_estatica /var/www/webserver/
```

```
albertom-servidor@debian-deaw:~$ sudo mv /var/www/web_estatica/ /var/www/webserver/
albertom-servidor@debian-deaw:~$ ls -la /var/www/
total 12
drwxr-xr-x  3 root      root      4096 nov  3 07:30 .
drwxr-xr-x 12 root      root      4096 nov  1 11:33 ..
drwxr-xr-x  6 www-data  www-data  4096 nov  1 21:31 webserver
```

- b) Eliminar los enlaces simbólicos de /etc/apache2/sites-enabled/. Vamos a romper los enlaces simbólicos de los sitios web pero en este caso no lo vamos a hacer con unlink como en NGINX sino con el comando a2dissite:

```
sudo a2dissite 000-default.conf
```

```
sudo a2dissite academia.conf
```

```
sudo a2dissite estatica.conf
```

```
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 nov  2 19:52 .
drwxr-xr-x 8 root root 4096 nov  2 20:35 ..
lrwxrwxrwx 1 root root   35 nov  1 20:54 000-default.conf -> ../sites-available/000-default.conf
lrwxrwxrwx 1 root root   32 nov  2 19:52 academia.conf -> ../sites-available/academia.conf
lrwxrwxrwx 1 root root   32 nov  1 22:10 estatica.conf -> ../sites-available/estatica.conf
albertom-servidor@debian-deaw:~$ sudo a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
albertom-servidor@debian-deaw:~$ sudo a2dissite academia.conf
Site academia disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
albertom-servidor@debian-deaw:~$ sudo a2dissite estatica.conf
Site estatica disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

- c) Eliminamos los archivos de configuración 000-default.conf, academia.conf y default-ssl.conf del directorio /etc/apache2/sites-available/

```
sudo rm /etc/apache2/sites-available/000-default.conf
```

```
sudo rm /etc/apache2/sites-available/academia.conf
```

```
sudo rm /etc/apache2/sites-available/default-ssl.conf
```

```
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/sites-available/
total 28
drwxr-xr-x 2 root root 4096 nov  3 07:38 .
drwxr-xr-x 8 root root 4096 nov  2 20:35 ..
-rw-r--r-- 1 root root 1286 abr 13 2023 000-default.conf
-rw-r--r-- 1 root root 1466 nov  2 22:37 academia.conf
-rw-r--r-- 1 root root 6195 abr 13 2023 default-ssl.conf
-rw-r--r-- 1 root root 1319 nov  1 22:20 estatica.conf
albertom-servidor@debian-deaw:~$ sudo rm /etc/apache2/sites-available/000-default.conf
albertom-servidor@debian-deaw:~$ sudo rm /etc/apache2/sites-available/academia.conf
albertom-servidor@debian-deaw:~$ sudo rm /etc/apache2/sites-available/default-ssl.conf
```

Y renombramos el fichero estatica.conf a webserver.conf:

```
sudo mv /etc/apache2/sites-available/estatica.conf /etc/apache2/sites-available/webserver.conf
```

```
albertom-servidor@debian-deaw:~$ sudo mv /etc/apache2/sites-available/estatica.conf /etc/apache2/sites-available/webserver.conf
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/sites-available/
total 12
drwxr-xr-x 2 root root 4096 nov  3 07:43 .
drwxr-xr-x 8 root root 4096 nov  2 20:35 ..
-rw-r--r-- 1 root root 1319 nov  1 22:20 webserver.conf
```

- d) Eliminamos cualquier referencia que tuviéramos en el archivo /etc/hosts (salvo las 2 básicas):

```
GNU nano 7.2
127.0.0.1      localhost
127.0.1.1      debian-deaw
```

B. CONFIGURACIÓN DEL FICHERO webserver.conf

En el archivo de configuración debemos explicitar el puerto de escucha 8080 (Listen 8080) y, además, cambiar el puerto de escucha del VirtualHosts pasando de *:80 a *:8080 y, además debemos modificar el serverName (pasará a ser webserver) y la dirección del DocumentRoot (/var/www/webserver):

```
Listen 8080

<VirtualHost *:8080>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin admin@correo.com
    ServerName webserver
    DocumentRoot /var/www/webserver/

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

Ahora debemos crear el enlace simbólico en /etc/apache2/sites-enabled/ para ello usamos el comando:

```
sudo a2ensite webserver.conf
```

```
albertom-servidor@debian-deaw:~$ sudo a2ensite webserver.conf
Enabling site webserver.
To activate the new configuration, you need to run:
    systemctl reload apache2
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 nov  4 16:41 .
drwxr-xr-x 8 root root 4096 nov  2 20:35 ..
lrwxrwxrwx 1 root root   33 nov  4 16:41 webserver.conf -> ../sites-available/webserver.conf
```

Reiniciamos el servicio de Apache y con esto habrá finalizado la configuración de nuestra máquina servidora.

C. CONFIGURACIÓN DE LA MÁQUINA QUE ACTUARÁ DE PROXY INVERSO

Debemos hacer una clonación de la máquina servidora. Hay que tener en cuenta que esta clonación se debe hacer con la opción de generar nuevas direcciones MAC ya que de otra forma la máquina se quedará sin una IP propia.

En esta máquina no necesitamos ningún archivo salvo el básico de configuración por lo que vamos a eliminar de /var/www/ el directorio que creamos en el paso anterior para el servidor web:

```
sudo rm -Rf /var/www/webserver/
```

```
albertom-servidor@debian-deaw:~$ sudo rm -Rf /var/www/webserver/
albertom-servidor@debian-deaw:~$ ls -la /var/www/
total 8
drwxr-xr-x 2 root root 4096 nov  4 16:37 .
drwxr-xr-x 12 root root 4096 nov  1 11:33 ..
```

También vamos a romper el enlace simbólico de /etc/apache2/sites-enabled:

```
albertom-servidor@debian-deaw:~$ sudo a2dissite webserver.conf
Site webserver disabled.
To activate the new configuration, you need to run:
    systemctl reload apache2
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 nov  4 16:45 .
drwxr-xr-x 8 root root 4096 nov  2 20:35 ..
```

Para que funcione el proxy debemos activar 2 modificaciones apache2:

```
sudo a2enmod proxy
```

```
sudo a2enmod proxy_http
```

```
albertom-servidor@debian-deaw:~$ sudo a2enmod proxy
Enabling module proxy.
To activate the new configuration, you need to run:
    systemctl restart apache2
albertom-servidor@debian-deaw:~$ sudo a2enmod proxy_http
Considering dependency proxy for proxy_http:
Module proxy already enabled
Enabling module proxy_http.
To activate the new configuration, you need to run:
    systemctl restart apache2
```

Vamos a renombrar también el archivo /etc/apache/sites-available/webserver.conf a estatica.conf:

```
albertom-servidor@debian-deaw:~$ sudo mv /etc/apache2/sites-available/webserver.conf /etc/apache2/sites-available/estatica.conf
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/sites-available/
total 12
drwxr-xr-x 2 root root 4096 nov  4 16:46 .
drwxr-xr-x 8 root root 4096 nov  2 20:35 ..
-rw-r--r-- 1 root root 1313 nov  4 16:18 estatica.conf
```

Y lo vamos a configurar:

```
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin admin@correo.com
    ServerName www.estatica.es

    ProxyPass / http://webserver:8080/
    ProxyPassReverse / http://webserver:8080/

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

Debemos añadir dos directivas:

- ProxyPass: especifica el mapeo de peticiones entrantes al servidor backend. Se utiliza el valor "/" para referirse a todas las solicitudes.
- ProxyPassReverse: es necesaria para asegurarse de ello y que las cabeceras Location: generadas en el backend se modifican para apuntar al proxy inverso, en lugar del propio backend.

Añadimos la dirección del servidor al archivo /etc/hosts del proxy:

```
GNU nano 7.2
127.0.0.1      localhost
127.0.1.1      debian-deaw
10.0.2.26       webserver
```

Y creamos el enlace simbólico en el directorio sites-enabled/ con el comando:

```
sudo a2ensite estatica.conf
```

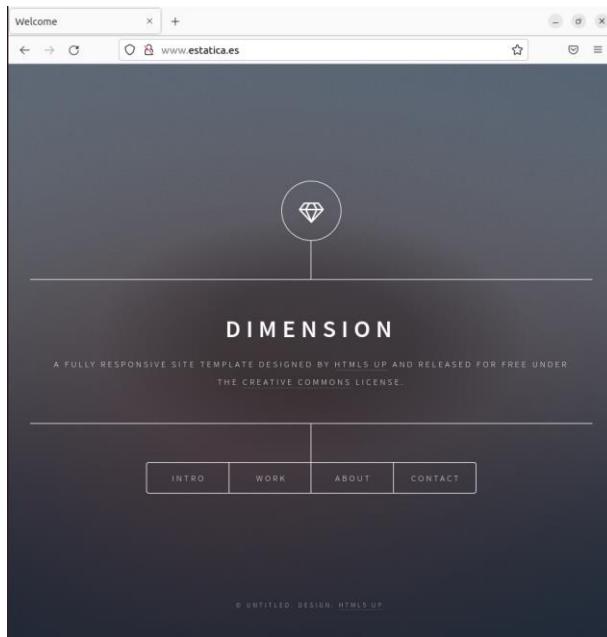
```
albertom-servidor@debian-deaw:~$ sudo a2ensite estatica.conf
Enabling site estatica.
To activate the new configuration, you need to run:
    systemctl reload apache2
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 nov  4 16:52 .
drwxr-xr-x 8 root root 4096 nov  2 20:35 ..
lrwxrwxrwx 1 root root   32 nov  4 16:52 estatica.conf -> ../sites-available/estatica.conf
```

D. CONFIGURACIÓN DE LA MÁQUINA CLIENTE Y COMPROBACIÓN DEL FUNCIONAMIENTO

En la máquina servidora debemos hacer una modificación del archivo /etc/hosts para que la IP del proxy se traduzca a partir de su dirección de dominio (www.estatica.es):

```
127.0.0.1      localhost
127.0.1.1      albertomcliente-VirtualBox
10.0.2.30      www.estatica.es
```

Ahora si accedemos a la dirección desde el navegador web accederemos al sitio web:



Para comprobar que estamos realizando el acceso a través del proxy podemos acceder a los logs de acceso:

```
sudo tail /var/log/apache2/access.log
```

- Log de acceso del proxy: En él veremos que la petición se realiza desde la dirección IP de nuestra máquina cliente:

```
albertom-servidor@debian-deaw:~$ sudo tail /var/log/apache2/access.log
10.0.2.28 - - [04/Nov/2023:19:10:19 +0100] "GET /assets/js/main.js HTTP/1.1" 200 2554 "http://www.estatica.es/"
10.0.2.28 - - [04/Nov/2023:19:10:19 +0100] "GET /assets/js/jquery.min.js HTTP/1.1" 200 33626 "http://www.estatica.es/"
10.0.2.28 - - [04/Nov/2023:19:10:19 +0100] "GET /images/pic01.jpg HTTP/1.1" 200 10352 "http://www.estatica.es/"
10.0.2.28 - - [04/Nov/2023:19:10:19 +0100] "GET /images/pic03.jpg HTTP/1.1" 200 9984 "http://www.estatica.es/"
10.0.2.28 - - [04/Nov/2023:19:10:19 +0100] "GET /images/pic02.jpg HTTP/1.1" 200 9191 "http://www.estatica.es/"
10.0.2.28 - - [04/Nov/2023:19:10:19 +0100] "GET /assets/css/font-awesome.min.css HTTP/1.1" 200 7003 "http://www.
```

- Log de acceso del servidor: En él veremos cómo las diferentes peticiones se han realizado desde la IP del proxy y no de la máquina cliente:

```
albertom-servidor@debian-deaw:~$ sudo tail /var/log/apache2/access.log
10.0.2.30 - - [04/Nov/2023:19:10:18 +0100] "GET /assets/js/main.js HTTP/1.1" 200 2554 "http://www.estatica.es/"
10.0.2.30 - - [04/Nov/2023:19:10:18 +0100] "GET /assets/js/jquery.min.js HTTP/1.1" 200 33626 "http://www.estatica.es/"
10.0.2.30 - - [04/Nov/2023:19:10:18 +0100] "GET /images/pic01.jpg HTTP/1.1" 200 10352 "http://www.estatica.es/"
10.0.2.30 - - [04/Nov/2023:19:10:18 +0100] "GET /images/pic02.jpg HTTP/1.1" 200 9191 "http://www.estatica.es/"
10.0.2.30 - - [04/Nov/2023:19:10:18 +0100] "GET /images/pic03.jpg HTTP/1.1" 200 9984 "http://www.estatica.es/"
10.0.2.30 - - [04/Nov/2023:19:10:18 +0100] "GET /assets/css/font-awesome.min.css HTTP/1.1" 200 7003 "http://www.
```

BALANCEADOR DE CARGA EN APACHE

Vamos a crear ahora un balanceador de carga a partir del proxy inverso que hemos generado en la anterior sección.

A. CONFIGURACIÓN DEL SERVIDOR WEB 1

En esta sección vamos a utilizar unos sitios web nuevos por lo que vamos a empezar eliminando toda referencia a la web estática de la anterior página:

- Eliminamos el directorio webserver/ de /var/www:

```
sudo rm -Rf /var/www/webserver/
```

```
albertom-servidor@debian-deaw:~$ sudo rm -Rf /var/www/webserver/
albertom-servidor@debian-deaw:~$ ls -la /var/www/
total 8
drwxr-xr-x 2 root root 4096 nov  4 19:34 .
drwxr-xr-x 12 root root 4096 nov  1 11:33 ..
```

- Destruimos el enlace simbólico de sites-enabled/:

```
sudo a2dissite webserver.conf
```

```
albertom-servidor@debian-deaw:~$ sudo a2dissite webserver.conf
Site webserver disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 nov  4 19:36 .
drwxr-xr-x 8 root root 4096 nov  2 20:35 ..
```

- Renombramos el archivo webserver.conf de sites-available/ a webserver1.conf:

```
sudo mv /etc/apache2/sites-available/webserver.conf
/etc/apache2/sites-available/webserver1.conf
```

```
albertom-servidor@debian-deaw:~$ sudo mv /etc/apache2/sites-available/webserver.conf /etc/apache2/sites-available/webserver1.conf
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/sites-available/
total 12
drwxr-xr-x 2 root root 4096 nov  4 19:38 .
drwxr-xr-x 8 root root 4096 nov  2 20:35 ..
-rw-r--r-- 1 root root 1326 nov  4 18:11 webserver1.conf
```

Una vez hecho esto vamos a crear un nuevo directorio en /var/www/ al que vamos a llamar webserver1/:

```
sudo mkdir /var/www/webserver1/
```

```
albertom-servidor@debian-deaw:~$ sudo mkdir /var/www/webserver1/
albertom-servidor@debian-deaw:~$ ls -la /var/www/
total 12
drwxr-xr-x 3 root root 4096 nov  4 19:39 .
drwxr-xr-x 12 root root 4096 nov  1 11:33 ..
drwxr-xr-x 2 root root 4096 nov  4 19:39 webserver1
```

Y en su interior vamos a crear un archivo index.html muy básico que nos sirva como información de que nos encontramos en el servidor web número 1:

```
sudo nano /var/www/webserver1/index.html
```

```
GNU nano 7.2
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Prueba de balanceo de carga con Apache</title>
</head>
<body>
    <h2>Este es el servidor web 1</h2>
    <p>Comprueba el balanceo de carga con Apache recargando esta página.</p>
</body>
</html>
```

Guardamos el fichero y lo siguiente será darle la propiedad del directorio webserver1/ al usuario www-data. Así mismo debemos modificar los permisos:

```
sudo chown -R www-data:www-data /var/www/webserver1/
```

```
sudo chmod -R 755 /var/www/webserver1/
```

```
albertom-servidor@debian-deaw:~$ sudo chown -R www-data:www-data /var/www/webserver1/
albertom-servidor@debian-deaw:~$ sudo chmod -R 755 /var/www/webserver1/
albertom-servidor@debian-deaw:~$ ls -la /var/www/
total 12
drwxr-xr-x  3 root      root      4096 nov  4 19:39 .
drwxr-xr-x 12 root      root      4096 nov  1 11:33 ..
drwxr-xr-x  2 www-data  www-data  4096 nov  4 19:44 webserver1
```

Vamos a habilitar dos nuevos modificadores de apache que nos van a permitir crear cabeceras:

```
sudo a2enmod headers
```

```
sudo a2enmod rewrite
```

```
albertom-servidor@debian-deaw:~$ sudo a2enmod headers
Enabling module headers.
To activate the new configuration, you need to run:
  systemctl restart apache2
albertom-servidor@debian-deaw:~$ sudo a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
  systemctl restart apache2
```

Por último, vamos a configurar el archivo webserver1.conf que hemos generado más arriba. Debemos establecer estos valores en los campos ServerName y DocumentRoot:

```
ServerName webserver1
DocumentRoot /var/www/webserver1/
```

Y añadimos una nueva cabecera que informe de que nos encontramos en el servidor web 1:

```
Header set Host Servidor_Web1_albertoM
```

Una vez hecho esto creamos el enlace simbólico en sites-enabled/:

```
sudo a2ensite webserver1.conf
```

```
albertom-servidor@debian-deaw:~$ sudo a2ensite webserver1.conf
Enabling site webserver1.
To activate the new configuration, you need to run:
    systemctl reload apache2
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 nov  4 19:47 .
drwxr-xr-x 8 root root 4096 nov  2 20:35 ..
lwxrwxrwx 1 root root   34 nov  4 19:47 webserver1.conf -> ../sites-available/webserver1.conf
```

Reiniciamos el servicio Apache y con esto habrá terminado la configuración de la máquina servidora número 1.

B. CONFIGURACIÓN DEL SERVIDOR WEB 2

Ahora vamos a configurar el segundo servidor, para facilitar la tarea vamos a realizar una clonación del anterior servidor, pero generando nuevas MAC para que pueda tener su propia IP.

Una vez hecha esta clonación debemos cambiar ciertas cosas en las configuraciones:

- Renombrar el directorio webserver1/ de /var/www/ a webserver2/:

```
sudo mv /var/www/webserver1 /var/www/webserver2
```

```
albertom-servidor@debian-deaw:~$ sudo mv /var/www/webserver1 /var/www/webserver2
albertom-servidor@debian-deaw:~$ ls -la /var/www/
total 12
drwxr-xr-x  3 root      root      4096 nov  4 19:59 .
drwxr-xr-x 12 root      root      4096 nov  1 11:33 ..
drwxr-xr-x  2 www-data  www-data  4096 nov  4 19:44 webserver2
```

- Modificar el archivo index.html de este directorio para que informe de que estamos en el servidor 2:

```
GNU nano 7.2                                     /var/www/webserver2/index.html *
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Prueba de balanceo de carga con Apache</title>
</head>
<body>
    <h2>Este es el servidor web 2</h2>
    <p>Comprueba el balanceo de carga con Apache recargando esta página.</p>
</body>
</html>
```

- Romper el enlace simbólico de sites-enabled/:

```
sudo a2dissite webserver1.conf
```

```
albertom-servidor@debian-deaw:~$ sudo a2dissite webserver1.conf
Site webserver1 disabled.
To activate the new configuration, you need to run:
    systemctl reload apache2
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 nov  4 20:01 .
drwxr-xr-x 8 root root 4096 nov  2 20:35 ..
```

- Cambiar el nombre del fichero webserver1.conf de sites-available/ a webserver2.conf:

```
sudo mv /etc/apache2/sites-available/webserver1.conf /etc/
apache2/sites-available/webserver2.conf
```

```
albertom-servidor@debian-deaw:~$ sudo mv /etc/apache2/sites-available/webserver1.conf /etc/apache2/sites-available/webserver2.conf
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/sites-available/
total 12
drwxr-xr-x 2 root root 4096 nov  4 20:04 .
drwxr-xr-x 8 root root 4096 nov  2 20:35 ..
-rw-r--r-- 1 root root 1328 nov  4 19:47 webserver2.conf
```

Dentro de este fichero vamos a modificar el ServerName y el DocumentRoot y que hagan referencia a webserver2:

```
ServerName webserver2
DocumentRoot /var/www/webserver2/
```

También modificamos la cabecera para que informe sobre el servidor web 2:

```
Header set Host Servidor_Web2_albertoM
```

Completados todos los pasos previos podemos crear el enlace simbólico en sites-enabled/:

```
sudo a2ensite webserver2.conf
albertom-servidor@debian-deaw:~$ sudo a2ensite webserver2.conf
Enabling site webserver2.
To activate the new configuration, you need to run:
  systemctl reload apache2
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 nov  4 20:10 .
drwxr-xr-x 8 root root 4096 nov  2 20:35 ..
lrwxrwxrwx 1 root root   34 nov  4 20:10 webserver2.conf -> ../sites-available/webserver2.conf
```

Finalizamos la configuración del servidor realizando un reinicio del servicio Apache.

C. CONFIGURACIÓN DEL PROXY INVERSO-BALANCEADOR DE CARGA

En el proxy debemos también de eliminar todo rastro de la web estática así que empezaremos eliminando el enlace simbólico:

```
sudo a2dissite estatica.conf
```

```
albertom-servidor@debian-deaw:~$ sudo a2dissite estatica.conf
Site estatica disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 nov  4 20:21 .
drwxr-xr-x 8 root root 4096 nov  2 20:35 ..
```

Completado esto vamos a modificar el nombre del archivo estatica.conf de sites-available/ por balanceo.conf:

```
sudo mv /etc/apache2/sites-available/estatica.conf /etc/apache2/sites-available/balanceo.conf
```

```
albertom-servidor@debian-deaw:~$ sudo mv /etc/apache2/sites-available/estatica.conf /etc/apache2/sites-available/balanceo.conf
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/sites-available/
total 12
drwxr-xr-x 2 root root 4096 nov 4 20:23 .
drwxr-xr-x 8 root root 4096 nov 2 20:35 ..
-rw-r--r-- 1 root root 1371 nov 4 19:09 balanceo.conf
```

Antes de configurar el archivo debemos añadir 2 nuevos módulos:

```
sudo a2enmod proxy_balancer
```

```
sudo a2enmod lbmethod_byrequests
```

```
albertom-servidor@debian-deaw:~$ sudo a2enmod proxy_balancer
Considering dependency proxy for proxy_balancer:
Module proxy already enabled
Considering dependency alias for proxy_balancer:
Module alias already enabled
Considering dependency slotmem_shm for proxy_balancer:
Enabling module slotmem_shm.
Enabling module proxy_balancer.
To activate the new configuration, you need to run:
    systemctl restart apache2
albertom-servidor@debian-deaw:~$ sudo a2enmod lbmethod_byrequests
Considering dependency proxy_balancer for lbmethod_byrequests:
Considering dependency proxy for proxy_balancer:
Module proxy already enabled
Considering dependency alias for proxy_balancer:
Module alias already enabled
Considering dependency slotmem_shm for proxy_balancer:
Module slotmem_shm already enabled
Module proxy_balancer already enabled
Enabling module lbmethod_byrequests.
To activate the new configuration, you need to run:
    systemctl restart apache2
```

Y configuramos el archivo para que nos quede de esta forma:

```
<VirtualHost *:80>
    <Proxy balancer://miSetDeServidores>
        BalancerMember http://webserver1:8080/
        BalancerMember http://webserver2:8080/
        ProxySet lbmethod=byrequests
    </Proxy>

    ServerAdmin admin@correo.com
    ServerName www.balanceo.es

    Header set Host2 Proxy_Inverso_albertoM

    ProxyPass / balancer://miSetDeServidores/
    ProxyPassReverse / balancer://miSetDeServidores/

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Creamos un bloque <Proxy> que nos servirá para definir los servidores con los que vamos a hacer balanceo de carga (BalancerMember) y el método que utilizaremos para balancear la carga (ProxySet).

Además, cambiamos el ServerName a www.balanceo.es, añadimos una cabecera que informe de que se reciben los datos del proxy inverso (añadir los módulos headers y rewrite si es necesario) y en las directivas relativas al ProxyPass cambiamos la dirección webserver:8080 por la del balanceador que hemos creado en el bloque Proxy.

Una vez hecho esto debemos modificar el archivo /etc/hosts:

```
GNU nano 7.2
127.0.0.1      localhost
127.0.1.1      debian-deaw
10.0.2.31       webserver1
10.0.2.32       webserver2
```

Creamos el enlace simbólico en sites-enabled/:

```
sudo a2ensite balanceo.conf
```

```
albertom-servidor@debian-deaw:~$ sudo a2ensite balanceo.conf
Enabling site balanceo.
To activate the new configuration, you need to run:
    systemctl reload apache2
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 nov  4 20:42 .
drwxr-xr-x 8 root root 4096 nov  2 20:35 ..
lrwxrwxrwx 1 root root   32 nov  4 20:42 balanceo.conf -> ../sites-available/balanceo.conf
```

Reiniciamos el servicio apache para que los cambios surtan efecto y con esto finalizamos la configuración del proxy-balanceador.

D. CONFIGURACIÓN DE LA MÁQUINA CLIENTE

Vamos a configurar el archivo de /etc/hosts para que se pueda acceder a la web de www.balanceo.es:

```
GNU nano 6.2
127.0.0.1      localhost
127.0.1.1      albertomcliente-VirtualBox
10.0.2.33       www.balanceo.es
```

Si ahora accedemos a la web desde un navegador podremos:

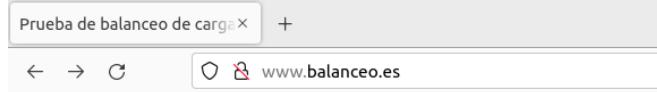
- Acceder al sitio correspondiente al servidor web 1:

The screenshot shows a browser window with the following content:

- Address bar:** Prueba de balanceo de carga > www.balanceo.es
- Page Content:**

Este es el servidor web 1
 Comprueba el balanceo de carga con Apache recargando esta página.
- Network tab:**
 - Host:** Servidor_Web1_albertoM
 - Host2:** Proxy_Inverso_albertoM
 - Keep-Alive:** timeout=5, max=100
 - Last-Modified:** Sat, 04 Nov 2023 18:44:15 GMT
 - Server:** Apache/2.4.57 (Debian)
 - Vary:** Accept-Encoding

- Acceder al sitio correspondiente al servidor web 2:



Este es el servidor web 2

Comprueba el balanceo de carga con Apache recargando esta página.

Cabeceras de la respuesta (398 B)	
Accept-Ranges: bytes	
Connection: Keep-Alive	
Content-Encoding: gzip	
Content-Length: 245	
Content-Type: text/html	
Date: Sat, 04 Nov 2023 20:04:27 GMT	
ETag: "15c-60958406f6dc0-gzip"	
Host: Servidor_Web2_albertoM	
Host2: Proxy_Inverso_albertoM	
Keep-Alive: timeout=5, max=100	
Last-Modified: Sat, 04 Nov 2023 19:01:43 GMT	
Server: Apache/2.4.57 (Debian)	
Vary: Accept-Encoding	

Podemos comprobar en los logs de acceso cómo funciona el balanceador de carga:

```
sudo tail /var/log/apache2/access.log
```

- Log del proxy: Las llamadas se hacen desde la IP de la máquina cliente:


```
albertom-servidor@debian-dewar:~$ sudo tail /var/log/apache2/access.log
10.0.2.34 - - [04/Nov/2023:21:05:11 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:05:12 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:05:13 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:05:14 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:05:14 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:05:15 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:05:16 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:05:17 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:05:17 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:05:18 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
```
- Log del servidor web 1: Las llamadas se hacen desde la IP del proxy:


```
albertom-servidor@debian-dewar:~$ sudo tail /var/log/apache2/access.log
10.0.2.33 - - [04/Nov/2023:20:50:38 +0100] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:20:50:40 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:20:51:44 +0100] "GET // HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:02:20 +0100] "GET // HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:05:10 +0100] "GET // HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:05:11 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:05:13 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:05:14 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:05:16 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:05:18 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
```
- Log del servidor web 2: Al igual que en el anterior todas las llamadas se hacen desde la IP del proxy:


```
albertom-servidor@debian-dewar:~$ sudo tail /var/log/apache2/access.log
10.0.2.33 - - [04/Nov/2023:20:51:19 +0100] "GET // HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:20:52:47 +0100] "GET // HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:20:53:13 +0100] "GET / HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:04:27 +0100] "GET // HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:05:11 +0100] "GET // HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:05:12 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:05:14 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:05:16 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:05:17 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:05:18 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
```

Ahora vamos a hacer comprobamos de que el balanceo funciona de forma correcta y que en caso de que uno de los servidores caiga, todas las solicitudes son atendidas por el otro servidor.

Vamos a empezar parando el servicio de apache en el servidor web número uno:

```
sudo systemctl stop apache2.service
```

```
albertom-servidor@debian-deaw:~$ sudo systemctl stop apache2.service
albertom-servidor@debian-deaw:~$ sudo systemctl status apache2.service
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: inactive (dead) since Sat 2023-11-04 21:11:57 CET; 6s ago
     Duration: 10min 5.046s
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 2982 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Process: 3124 ExecStop=/usr/sbin/apachectl graceful-stop (code=exited, status=0/SUCCESS)
 Main PID: 2986 (code=exited, status=0/SUCCESS)
      CPU: 176ms
```

Si ahora hacemos varias llamadas a www.balanceo.es veremos como siempre accedemos al servidor web 2:



En los logs podemos comprobar como todas las solicitudes del cliente que han llegado al proxy:

```
albertom-servidor@debian-deaw:~$ sudo tail /var/log/apache2/access.log
10.0.2.34 - - [04/Nov/2023:21:12:42 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:12:43 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:12:44 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:12:44 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:12:44 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:12:45 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:12:46 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:12:46 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:12:47 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:12:48 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:12:48 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
```

Han sido procesadas por el servidor 2:

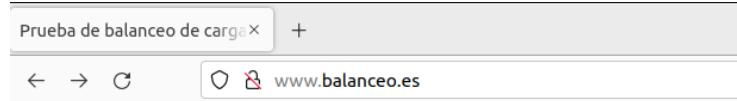
```
albertom-servidor@debian-deaw:~$ sudo tail /var/log/apache2/access.log
10.0.2.33 - - [04/Nov/2023:21:12:42 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:12:43 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:12:44 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:12:45 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:12:46 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:12:47 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:12:48 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:12:48 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
```

Y ninguna por el servidor 1:

```
albertom-servidor@debian-deaw:~$ sudo tail /var/log/apache2/access.log
10.0.2.33 - - [04/Nov/2023:20:50:38 +0100] "GET // HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:20:50:40 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:20:51:44 +0100] "GET // HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:02:20 +0100] "GET // HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:05:10 +0100] "GET // HTTP/1.1" 200 612 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:05:11 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:05:13 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:05:14 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:05:16 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:05:18 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
```

Vamos a repetir la prueba pero ahora paralizando el servidor dos. Primero antes de comenzar la prueba debemos volver a iniciar el servidor web uno para que pueda haber peticiones.

Una vez que hayamos hecho esto veremos como todas las solicitudes se dirigen al servidor web 1:



Este es el servidor web 1

Comprueba el balanceo de carga con Apache recargando esta página.

Si ahora acudimos a los logs de acceso podremos ver que todas las solicitudes recibidas por el proxy:

```
albertom-servidor@debian-dew:~$ sudo tail /var/log/apache2/access.log
10.0.2.34 - - [04/Nov/2023:21:17:17 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:17:18 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:17:19 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:17:20 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:17:21 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:17:22 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:17:23 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:17:24 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:17:25 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:17:26 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:17:27 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:17:28 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:17:29 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:17:30 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:17:31 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:17:32 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:17:33 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.34 - - [04/Nov/2023:21:17:34 +0100] "GET / HTTP/1.1" 200 642 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
```

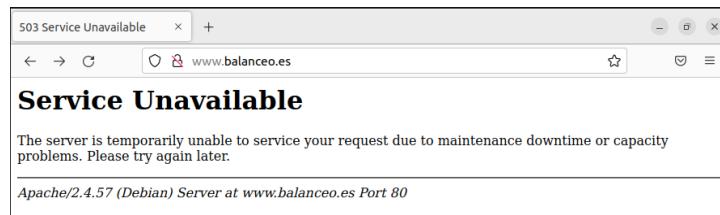
Han sido atendidas por el servidor web número 1:

```
albertom-servidor@debian-dew:~$ sudo tail /var/log/apache2/access.log
10.0.2.33 - - [04/Nov/2023:21:17:18 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:17:18 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:17:19 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:17:20 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:17:21 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:17:22 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:17:23 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:17:24 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:17:25 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:17:26 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:17:27 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:17:28 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:17:29 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:17:30 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:17:31 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:17:32 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:17:33 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:17:34 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
```

Y que ninguna ha llegado hasta el servidor web número 2:

```
albertom-servidor@debian-dew:~$ sudo tail /var/log/apache2/access.log
10.0.2.33 - - [04/Nov/2023:21:12:42 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:12:43 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:12:44 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:12:44 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:12:45 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:12:45 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:12:46 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:12:46 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:12:47 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:12:48 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.33 - - [04/Nov/2023:21:12:48 +0100] "GET // HTTP/1.1" 200 611 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
```

Ahora vamos a mostrar qué ocurriría en el caso de una caída de ambos servidores, para ello volvemos a paralizar el servidor web 1. Y si intentamos acceder al sitio web obtendremos el siguiente resultado:



Apache/2.4.57 (Debian) Server at www.balanceo.es Port 80

Como respuesta obtenemos un código de error 503. Algo que queda registrado en los logs de error del proxy:

```
sudo tail /var/log/apache2/error.log
```

```
[Sat Nov 04 21:24:09 2023] [proxy:error] [pid 3287:tid 14003313861768] [client 10.0.2.34:33444] AH01114: HTTP: failed to make connection to backend: webserver2
[Sat Nov 04 21:24:20 2023] [proxy_error] [pid 3287:tid 140033164039872] [111]Connection refused: AH00957: http: attempt to connect to 10.0.2.31:8088 (webserver1:8080) failed
[Sat Nov 04 21:24:20 2023] [proxy_error] [pid 3287:tid 140033164039872] AH00959: ap_proxy_connect_backend disabling worker for (webserver1:8080) for 60s
[Sat Nov 04 21:24:20 2023] [proxy_error] [pid 3287:tid 140033164039872] [client 10.0.2.34:33444] AH01114: HTTP: failed to make connection to backend: webserver1
[Sat Nov 04 21:24:21 2023] [proxy_error] [pid 3287:tid 14003313861768] [111]Connection refused: AH00957: http: attempt to connect to 10.0.2.31:8088 (webserver1:8080) failed
[Sat Nov 04 21:24:21 2023] [proxy_error] [pid 3287:tid 14003313861768] AH00959: ap_proxy_connect_backend disabling worker for (webserver1:8080) for 60s
[Sat Nov 04 21:24:21 2023] [proxy_error] [pid 3287:tid 14003313861768] [client 10.0.2.34:44846] AH01114: HTTP: failed to make connection to backend: webserver1
[Sat Nov 04 21:24:21 2023] [proxy_error] [pid 3287:tid 14003313861768] AH00959: ap_proxy_connect_backend disabling worker for (webserver1:8080) for 60s
[Sat Nov 04 21:24:21 2023] [proxy_error] [pid 3287:tid 14003313861768] AH00959: ap_proxy_connect_backend disabling worker for (webserver2:8080) for 60s
[Sat Nov 04 21:24:21 2023] [proxy_error] [pid 3287:tid 14003313861768] AH00959: ap_proxy_connect_backend disabling worker for (webserver2:8080) for 60s
[Sat Nov 04 21:24:21 2023] [proxy_error] [pid 3287:tid 14003313861768] AH00959: ap_proxy_connect_backend disabling worker for (webserver2:8080) for 60s
[Sat Nov 04 21:24:21 2023] [proxy_error] [pid 3287:tid 14003313861768] AH00959: ap_proxy_connect_backend disabling worker for (webserver2:8080) for 60s
```

CONFIGURACIÓN DEL BALANCEADOR DE CARGA CON SSL

Para finalizar esta guía de configuración del servidor web Apache vamos a hacer una configuración SSL para nuestros sitios web HTTPS. Esta conexión HTTPS sólo se realizará entre el cliente y el proxy por lo que vamos a realizar una reconfiguración de nuestra máquina proxy-balanceador de carga.

A. CREACIÓN DE UN CERTIFICADO AUTOFIRMADO

Vamos a comenzar creándonos un certificado autofirmado propio. Para ello primero vamos a crear un nuevo directorio llamado ssl/ en nuestro directorio /etc/apache2/:

```
sudo mkdir /etc/apache2/ssl
```

```
albertom-servidor@debian-deaw:~$ sudo mkdir /etc/apache2/ssl
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/
total 104
drwxr-xr-x  9 root root  4096 nov  4 21:36 .
drwxr-xr-x 135 root root 12288 nov  4 20:34 ..
-rw-r--r--  1 root root  7178 abr 13 2023 apache2.conf
drwxr-xr-x  2 root root  4096 nov  1 20:54 conf-available
drwxr-xr-x  2 root root  4096 nov  1 20:54 conf-enabled
-rw-r--r--  1 root root  1782 jun  9 2022 envvars
-rw-r--r--  1 root root     93 nov  2 19:07 .htpasswd
-rw-r--r--  1 root root 31063 jun  9 2022 magic
drwxr-xr-x  2 root root 12288 nov  1 20:54 mods-available
drwxr-xr-x  2 root root  4096 nov  4 20:50 mods-enabled
-rw-r--r--  1 root root   274 abr 13 2023 ports.conf
drwxr-xr-x  2 root root  4096 nov  4 21:05 sites-available
drwxr-xr-x  2 root root  4096 nov  4 20:42 sites-enabled
drwxr-xr-x  2 root root  4096 nov  4 21:36 ssl
```

Para crear el certificado y la clave de forma simultánea debemos usar el siguiente comando:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/server.key -out /etc/apache2/ssl/server.crt
```

```
albertson-Server:~/Desktop$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/server.key -out /etc/apache2/ssl/server.crt
-----  
You are about to be asked to enter information that will be incorporated  
into your certificate request.  
What you are about to enter is what is called a Distinguished Name or a DN.  
There are quite a few fields but you can leave some blank  
For some fields there will be a default value  
If you enter ...,, the field will be left blank  
-----
```

Y rellenamos el formulario con nuestros datos:

Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Comunidad de Madrid
Locality Name (eg, city) []:Coslada
Organization Name (eg, company) [Internet Widgits Pty Ltd]:IES Luis Braille
Organizational Unit Name (eg, section) []:2DAW - DEAW - ALBERTO
Common Name (e.g. server FQDN or YOUR name) []:balanceo
Email Address []:miCorreo@email.com

Esto nos ha creado tanto una clave como un certificado:

```
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/ssl/
total 16
drwxr-xr-x 2 root root 4096 nov  4 21:39 .
drwxr-xr-x 9 root root 4096 nov  4 21:36 ..
-rw-r--r-- 1 root root 1533 nov  4 21:39 server.crt
-rw----- 1 root root 1704 nov  4 21:37 server.key
```

Este es el aspecto que tiene la clave creada:

```
albertom-servidor@debian-deaw:~$ sudo cat /etc/apache2/ssl/server.key
-----BEGIN PRIVATE KEY-----
MIIEQTAhBAAgBqkhhkIG9w0BAQEFASCBKcwggSjAgEEAoIBAQDX/rUWKhrc4zu
tV7dEJBKmZccyXm98c98U6R6MvAgh+R/1H+YUOQo2kSpIyCBrFxHi
b1Cw-31W7vcvOLxmInNjmcUT53502F1kxEHyM9Jn2qFbJplcmPhvibosYhlc
D09J/HMC1EPK14oeidQruldHw7mdWeXv6WvC1gtB43a08cJ4beGAvir
mBagYCVjv/c1ctrGRJp1Pk52Q0a1406q8QDXWtQkDNuqCgcGmVmnb01S1MIXU0Mj
3tgndJUA1NhFc1k8ko1qfP7Y75bvclWbzbj9t+bgdknomsZ2d8YU2+EN4Kx9eU
1Kk8o5DxAgMAAEcgEA00k14kysnuF6EwFhuhlus70829pFBavgkuyafkxN885bH
6iDpe5K0m8jdu//ZxeGZL14e4XH50C+HFM2sLYQ9D.JN9f+ufckZgfaWLFsvo
dGw5SVQxvHITJGDFsJn3q1gb13kw20+u+v9.JM2ze9yhnxs52zwBKTKLxeoK/X
/sicEG5J6332Yv1Lo1k15zDw1x881UpTy3340Bzis1a0r/bf73cxhQbCEVKKh
zQR4enBqkb5e+cypgl2akQvPv49wu/K1t85opvzCN47mbbksth3VNDRIA/qt
UUWkyEFjvolegTtkz1XmkedVjV8fVfP+JqEy6uANiwKBg0DBhCwtTqflbaut5zn
fVTQ/dbe6asAPlwot1zpRv5C/yWQ61Xza65xQFvUXO+su.PmCirz/01yEQRyBh4
jG68Y2KunqBW/phLnCE17yNidmLZPx9d02Q93ghe66czNc6+Ov8k197b2w9
eh0eB161r1475nJh0fnuAewKBgDa+YzLwK4CxK69JaN357BpvzFDAarC8Q8U
3SYr02taTVLnx/b571zqbldspfAC24|S1NygsfmsZmu22Tx111xCUeCN6+w
9Y90SPK+Qne0B1SPxPz1P6xDulLzZmQngv+s4I+L1t150JDf4UfLgbhiqHTz7seV
PQe9zxmtEwKbgBawct4d3hjgJvokbnvCeazWKByjAqklmFuQkIRqu14qebozrkP
zQgOMX1+u64nInAp1QuScXbm55wPQG5k8G7Gw921Ea3t1Mrf30RczfPwUev+BY
M1sIMgivcvb5@0Hn9uDvBHFSc1euMc1z6jVaEQ0w05veUay/MRs3MKRAoGAc1a4
9XU9anEcxxK6d2df0BjsjZEUGKOr13BCEPE7d2vz+B2Fky+01C7NViuy42E4/
GES183z6u0wLwQfOFUnzgrw14XY1zFKDADUcpEfbwaoDU5hNjlkLdx7Y2P8bj
cSyS9w0dub1Gck/mx9R1vD+d50+0Zf6ec3RNf0cgYEarln0xp.8eut0usenVm0fa
mH1kzovio1cyhuopttJ9FEahveoymh1xe2MfxuMBR6P1/+F7z0n1vvvdqEq2
JgwLk0g0GVxRwpk8cgVbh/nzbgPwoC1KMQi51LFWTktqWpR0kV07W6cyVB3UJ
quoUM1rZvwk2lmuIa632cLU=
-----END PRIVATE KEY-----
```

Y este es el aspecto del certificado:

```
albertom-servidor@debian-deaw:~$ sudo cat /etc/apache2/ssl/server.crt
-----BEGIN CERTIFICATE-----
MIIEPzCCAegAwIBAgIUSv9gY1ARSIOXGW32hAx6v1pMuwDQYJKoZIhvvcNAQEL
BQAwgA4xCzAjBgNVBAYTAKJWMRMwGgDQVQKDBB/RMgTHVpcyBcmFpbGxLMR4w
MRAwDgYDVQQHODad3NsYWRMRkwFwYDVQKDBB/RMgTHVpcyBcmFpbGxLMR4w
HAYDVQQLDBUyREFXIC0g0REVBVyaATIEFMQkVSv8xETAPBgNvBAMCGjhBGFUV2V
MSEwhWYKj0ZhvCNaqKfhJtauNcnJ1b0BlbwFpbC5jb20WhhcnMjMxM7A0MjAz
OTEzWhCNMj0XMTAzMjAzOTEzWzCBjELMAKGA1UEBhMCRMxMHDaBgNVBAgME0Nv
bXVuahZC8kZSBnvYRyawQxEDAOBgNVBACmB0NvcxZxZGEgTAXBgNVBAgME0Nv
UyBMDw1zIEJyYnlsbgUxHjAcBgNVBAsMFTEJQvQcgLSBERUFxIC0gQUxCRVJUTZER
MA8GAIUEAwIYmf-sYw5jZw8xITAfbgkhhG9w0BCEWEm1p029ycmVnQGVYnW1s
LmNvbtCCAS1wDQYKoZihvCNaqEBQADggEPADCCAOocggEBAnf+trYkr+tzj061
VN3UQkEqlxzIBz6tVVE9jEMytz3xTpEy8Ah5H/Uen5145CjaRKnVhwF/EciJv
Vxb7fVbu9y84teV2U2OYJRPhd17YVSCqf1z1ymba0Nsmtay+G-JujiieVm
70n8e0wLwLQ08LqCh6j1Cu4tx0fLDwPATPxpsq9u1c0Hj+BoPxwmPgF4ZUCu+Y
FqBgIm/Yv0v2sZEmKes+Tn2AHr3j1qzxNdZNCR01SoIzwaZwUAPV1whd7QyPe
2Cd0lQDU2cv2UrySlw1rx9M9jvkG9yJzhv0O1t5uBSeib113wShn403grH150i
QzyjkPECAwEAaaNTMFEwHQYDVR0BBYEFTIRBNdw3DSRdt901jMWN02uHypXWB8G
A1udIwQYMBAfIRBNdw3DSRdt901jMWN02uHypXMA8GAIUdEwB/0QFMABAf8w
DQYJKoZihvCNaqELBQADggEBAJBotaUVSe5ohg563aBgT1/LX17Df7TRMS5gnrpU
qZkbjB8VQ8J6n5xzFkH3RT5vLcF0TCjmY570rGe2wsb101jv82275d0lVfDCLrw
KzTD28FQ/kqdj0xp/5q2XZnNPKmfHnzC9bn6U+nveMLCVkGZkjojwCazAGf0M
9zz4/zzo5JvQDwYQTAkCcPkm1B27w16o405yznnzt1vboh0@0hyffHxQ9dD
52FM2As0zR1z0b0Qs1CMGJyPW+y29pdzuNbTzHafcs1QEnfmj3Z5L0xRSb3ixjb
VnCoX0d0bTiimMS219m06zuw4q7GulJuL5hDwAgd1B1etE=
-----END CERTIFICATE-----
```

B. CONFIGURACIÓN SSL EN EL PROXY INVERSO

Antes de comenzar con la configuración debemos tener activado el módulo SSL de Apache, para ello usamos el siguiente comando:

```
sudo a2enmod ssl
```

```
albertom-servidor@debian-deaw:~$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
    systemctl restart apache2
```

Una vez hecho esto debemos crear un archivo en /etc/apache2/sites-available/ con el nombre de balanceo-ssl.conf y dentro debemos colocar la siguiente información:

```
<VirtualHost *:443>
    <Proxy balancer://miSetDeServidores>
        BalancerMember http://webserver1:8080/
        BalancerMember http://webserver2:8080/
        ProxySet lbmethod=byrequests
    </Proxy>

    ServerAdmin admin@correo.com

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/https_access.log combined

    SSLEngine on

    SSLCertificateFile      /etc/apache2/ssl/server.crt
    SSLCertificateKeyFile   /etc/apache2/ssl/server.key

    <FilesMatch "\.(?:cgi|shtml|phtml|php)$">
        SSLOptions +StdEnvVars
    </FilesMatch>
    <Directory /usr/lib/cgi-bin>
        SSLOptions +StdEnvVars
    </Directory>

    ProxyPass / balancer://miSetDeServidores/
    ProxyPassReverse / balancer://miSetDeServidores/
</VirtualHost>
```

El puerto de escucha es el 443 que es el puerto por defecto en HTTPS, volvemos a tener el balanceo de carga y añadimos el SSLEngin on para que funcione el SSL y además dos nuevas directivas SSLCertificateFile que tiene asociado el certificado que hemos creado y SSLCertificateKeyFile que tiene asociada la ruta a la clave que acabamos de crear.

Debemos crear un enlace simbólico en el directorio /etc/apache/sites-enabled/:

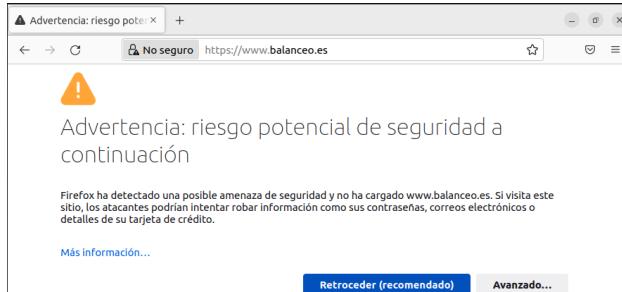
```
sudo a2ensite balanceo-ssl.conf
```

```
albertom-servidor@debian-deaw:~$ sudo a2ensite balanceo-ssl.conf
Enabling site balanceo-ssl.
To activate the new configuration, you need to run:
    systemctl reload apache2
albertom-servidor@debian-deaw:~$ ls -la /etc/apache2/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 nov  4 22:09 .
drwxr-xr-x 9 root root 4096 nov  4 21:36 ..
lrwxrwxrwx 1 root root   32 nov  4 20:42 balanceo.conf -> ../sites-available/balanceo.conf
lrwxrwxrwx 1 root root   36 nov  4 22:09 balanceo-ssl.conf -> ../sites-available/balanceo-ssl.conf
```

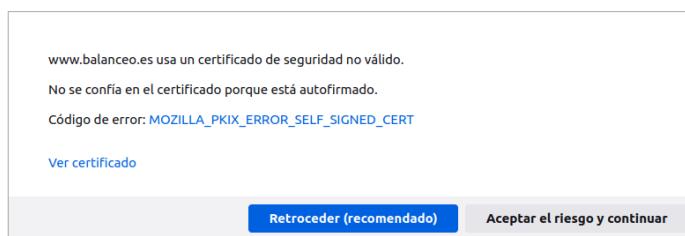
Y por último debemos reiniciar el servicio Apache para que estos cambios tengan efecto.

C. ACCESO DESDE EL NAVEGADOR

Ahora si accedemos al sitio <https://www.balanceo.es> desde el navegador de nuestra máquina cliente recibiremos el siguiente aviso:



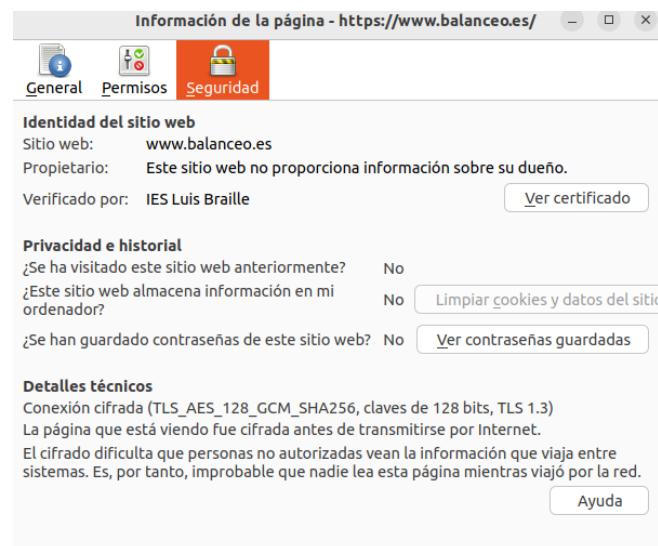
Si aceptamos el certificado:



Podremos acceder a la web:



Para consultar el certificado debemos hacer clic en el candado de la barra de navegación > Conexión no segura > Más información, eso hará que aparezca una nueva ventana:



Si hacemos clic en “Ver certificado” podremos ver el certificado que hemos creado:

Certificado

Nombre del asunto	
País	ES
Estado/Provincia	Comunidad de Madrid
Localidad	Coslada
Organización	IES Luis Braille
Unidad organizativa	2DAW - DEAW - ALBERTO
Nombre común	balanceo
Dirección de correo electrónico	miCorreo@email.com

Nombre del emisor	
País	ES
Estado/Provincia	Comunidad de Madrid
Localidad	Coslada
Organización	IES Luis Braille
Unidad organizativa	2DAW - DEAW - ALBERTO
Nombre común	balanceo
Dirección de correo electrónico	miCorreo@email.com

Validez	
No antes	Sat, 04 Nov 2023 20:39:13 GMT
No después	Sun, 03 Nov 2024 20:39:13 GMT

Información de clave pública	
Algoritmo	RSA
Tamaño de la clave	2048
Exponente	65537
Módulo	D7:FE:B5:16:24:AF:EB:73:8C:EE:B5:54:DD:D4:42:41:2A:66:5C:73:20:16:71:EA...

Los diferentes accesos HTTPS que hagamos se van a registrar en el archivo log que hemos creado al hacer el archivo balanceo-ssl.conf:

```
sudo tail /var/log/apache2/https_access.log
```

```
albertom-servidor:~# sudo tail /var/log/apache2/https_access.log
10.0.2.36 - - [04/Nov/2023:22:40:02 +0100] "GET /favicon.ico HTTP/1.1" 404 512 "https://www.balanceo.es/" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.36 - - [04/Nov/2023:22:47:48 +0100] "GET / HTTP/1.1" 200 633 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.36 - - [04/Nov/2023:22:47:49 +0100] "GET / HTTP/1.1" 200 633 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.36 - - [04/Nov/2023:22:47:50 +0100] "GET / HTTP/1.1" 200 633 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.36 - - [04/Nov/2023:22:49:52 +0100] "GET / HTTP/1.1" 200 953 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.36 - - [04/Nov/2023:22:49:53 +0100] "GET / HTTP/1.1" 200 633 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.36 - - [04/Nov/2023:22:49:53 +0100] "GET / HTTP/1.1" 200 633 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.36 - - [04/Nov/2023:22:49:54 +0100] "GET / HTTP/1.1" 200 633 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.36 - - [04/Nov/2023:22:49:55 +0100] "GET / HTTP/1.1" 200 633 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
10.0.2.36 - - [04/Nov/2023:22:49:55 +0100] "GET / HTTP/1.1" 200 633 "-" "Mozilla/5.0 (X11; Linux x86_64; rv:99.0) Gecko/20100101 Firefox/99.0"
```

El problema es que si ahora intentamos acceder a <http://www.balanceo.es> podemos hacerlo sin problema.

Prueba de balanceo de carga +

← → ⌂ www.balanceo.es

Este es el servidor web 1

Comprueba el balanceo de carga con Apache recargando esta página.

Para evitar que esto pueda ocurrir vamos a modificar el archivo balanceo.conf de nuestro proxy-balanceador:

```
<VirtualHost *:80>
    <Proxy balancer://miSetDeServidores>
        BalancerMember http://webserver1:8080/
        BalancerMember http://webserver2:8080/
        ProxySet lbmethod=byrequests
    </Proxy>

    ServerAdmin admin@correo.com
    ServerName www.balanceo.es

    Header set Host2 Proxy_Inverso_albertoM

    ProxyPass / balancer://miSetDeServidores/
    ProxyPassReverse / balancer://miSetDeServidores/

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/http_access.log combined

    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule ^/(.*) https://www.balanceo.es/$1 [R,L]
</VirtualHost>
```

Añadimos unas líneas que provocan una reescritura de la URL en el caso de intentar acceder por HTTP y ahora siempre se accederá por HTTPS a nuestro sitio web.

Además, hemos cambiado el log de acceso para que se registren estos intentos de acceso por HTTP.

Reiniciamos el servicio Apache para que estos cambios tengan efecto y comprobamos su funcionamiento en la máquina cliente intentando acceder a <http://www.balanceo.es>:



- Log de accesos por HTTP:

```
albertom-servidor@debian-deaw:~$ sudo tail -2 /var/log/apache2/http_access.log
10.0.2.36 - - [04/Nov/2023:22:57:36 +0100] "GET / HTTP/1.1" 302 538 "-" "Mozilla/5.0
```

- Log de accesos por HTTPS:

```
albertom-servidor@debian-deaw:~$ sudo tail -2 /var/log/apache2/https_access.log
10.0.2.36 - - [04/Nov/2023:22:57:36 +0100] "GET / HTTP/1.1" 200 2438 "-" "Mozilla/5.0 (X11;
10.0.2.36 - - [04/Nov/2023:22:57:36 +0100] "GET /favicon.ico HTTP/1.1" 404 512 "https://www
```

Como podemos en el log de HTTP recibimos un código de estado 302 o Found, el cual es un código de redirección HTTP que indica que el recurso solicitado ha sido movido temporalmente a la URL dada por las cabeceras Location.