

ADMINISTRACIÓN DE SISTEMAS Y REDES

Práctica 7

Miguel Méndez Murias
UO287687
Curso 2023/2024

Índice

| | |
|--|----|
| Servidores Web en Linux: Apache..... | 1 |
| 1. Instalación..... | 1 |
| 2. Configuración de las páginas web de los usuarios..... | 5 |
| 3. Configuración del servidor Apache..... | 7 |
| 3.a Ubicación..... | 7 |
| 3.b ServerName..... | 8 |
| 3.c Repositorios..... | 9 |
| 4. Hosts Virtuales..... | 11 |
| 5. Autenticación..... | 12 |
| 6. Servidor Proxy – squid..... | 13 |

Servidores Web en Linux: Apache

1. Instalación

En primer lugar, cambiamos el nombre de la máquina Linux a *linux.as.local* y comprobamos que tenemos acceso a la red pública Internet.

```
U0287687@linux.as.local ~]# hostnamectl set-hostname linux.as.local
U0287687@linux.as.local ~]# uname -a
Linux linux.as.local 5.14.0-362.18.1.el9_3.x86_64 #1 SMP PREEMPT_DYNAMIC Mon Jan 29 07:05:48 EST 2024 x86_64 x86_64 x86_64 GNU/Linux
U0287687@linux.as.local ~]# ping www.google.es
PING www.google.es (142.250.201.67) 56(84) bytes of data:
64 bytes from mad07s25-in-f3.1e100.net (142.250.201.67): icmp_seq=1 ttl=118 time=8.71 ms
64 bytes from mad07s25-in-f3.1e100.net (142.250.201.67): icmp_seq=2 ttl=118 time=10.1 ms
64 bytes from mad07s25-in-f3.1e100.net (142.250.201.67): icmp_seq=3 ttl=118 time=9.89 ms
^C
--- www.google.es ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 8.706/9.555/10.073/0.605 ms
U0287687@linux.as.local ~]#
```

Ahora comprobamos que podemos resolver la dirección *linux.as.local* desde todas las máquinas:

```
PS C:\Users\U0287687> nslookup linux.as.local
Servidor: UnKnown
Address: 192.168.56.101

Nombre: linux.as.local
Address: 192.168.56.100
```

```
U0287687@linux.as.local ~]# nslookup linux.as.local
Server: 192.168.56.101
Address: 192.168.56.101#53

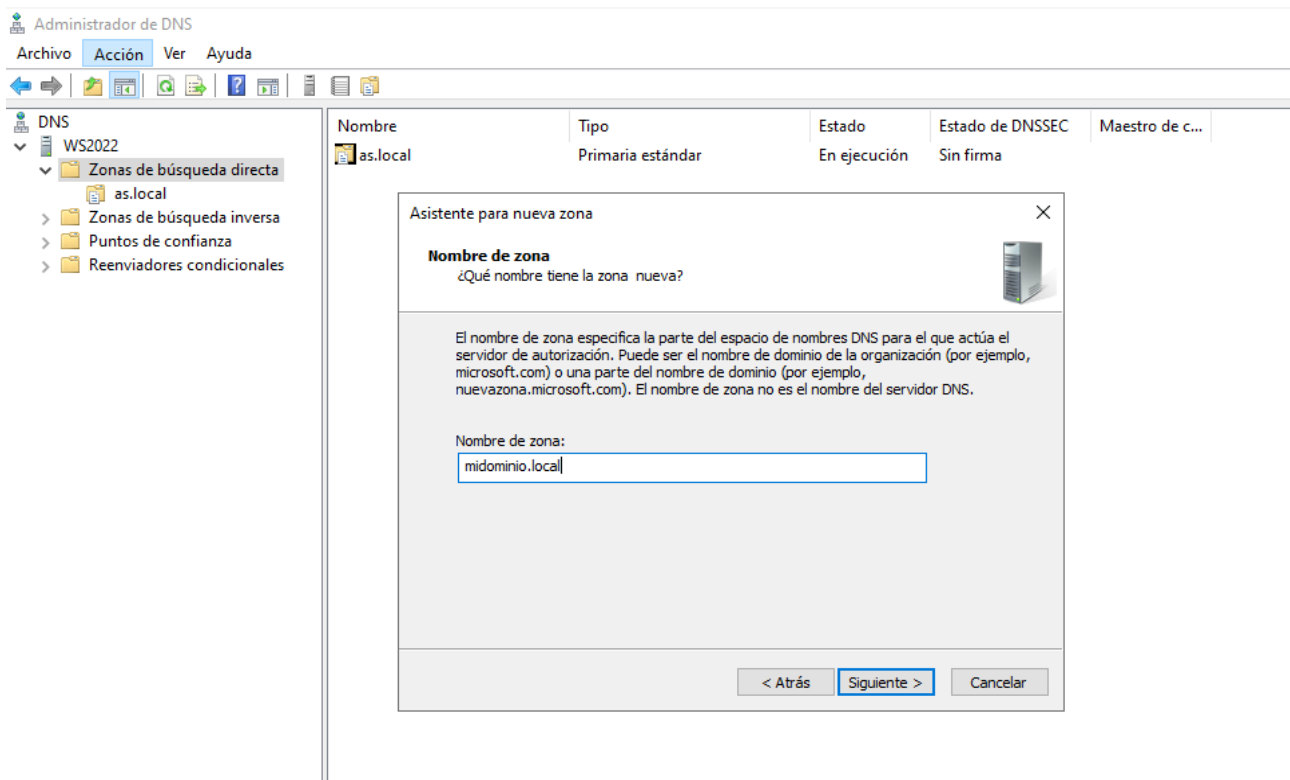
Name: linux.as.local
Address: 192.168.56.100
U0287687@linux.as.local ~]#
```

```
PS C:\Users\Administrador> nslookup linux.as.local
Servidor: UnKnown
Address: 192.168.56.101

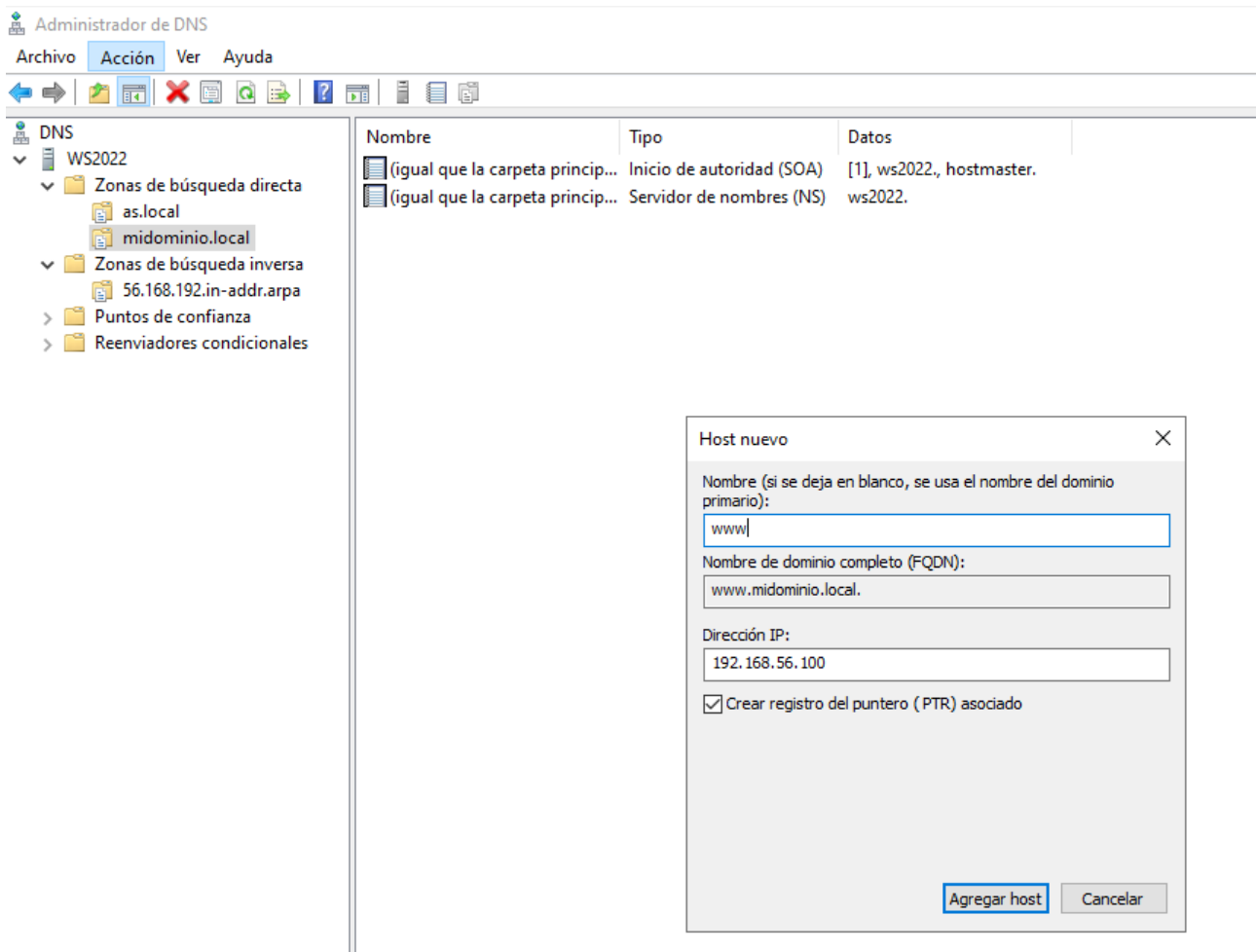
Nombre: linux.as.local
Address: 192.168.56.100

PS C:\Users\Administrador>
```

A continuación, añadimos la www.midominio.local con la misma dirección que la máquina Linux. Para ello, primero añadimos una nueva zona al servidor DNS: *midominio.local*



Creamos el host www con la misma dirección IP que la máquina Linux:



Una vez hecho esto, las máquina deberían poder resolver la dirección www.midominio.local. En la siguiente captura se muestra como esto se cumple lo esperado:

```
PS C:\Users\U0287687> nslookup www.midominio.local
Servidor: UnKnown
Address: 192.168.56.101

Nombre: www.midominio.local
Address: 192.168.56.100
```

```
[U0287687@linux.as.local ~]$ nslookup www.midominio.local
Server: 192.168.56.101
Address: 192.168.56.101#53

Name: www.midominio.local
Address: 192.168.56.100
```

Procedemos ahora a realizar la instalación de Apache en la máquina Linux y activamos el servicio:

```
[U0287687@linux.as.local ~]$ dnf install httpd
Error al cargar el complemento "config_manager": '*prog'
AlmaLinux 9 - AppStream 3.3 kB/s | 4.1 kB 00:01
AlmaLinux 9 - AppStream 3.4 MB/s | 9.4 MB 00:02
AlmaLinux 9 - BaseOS 7.2 kB/s | 3.8 kB 00:00
AlmaLinux 9 - BaseOS 2.1 MB/s | 4.8 MB 00:02
AlmaLinux 9 - Extras 6.9 kB/s | 3.8 kB 00:00
AlmaLinux 9 - Extras 18 kB/s | 17 kB 00:00
Dependencias resueltas.
=====
Paquete Arquitectura Versión Repositorio Tam.
-----
Instalando:
httpd x86_64 2.4.57-5.el9 appstream 46 k
Instalando dependencias:
almaLinux-logos-httpd noarch 90.5.1-1.1.el9 appstream 18 k
apr x86_64 1.7.0-12.el9_3 appstream 122 k
apr-util x86_64 1.6.1-23.el9 appstream 94 k
apr-util-bdb x86_64 1.6.1-23.el9 appstream 12 k
httpd-core x86_64 2.4.57-5.el9 appstream 1.4 M
httpd-filesystem noarch 2.4.57-5.el9 appstream 12 k
httpd-tools x86_64 2.4.57-5.el9 appstream 80 k
mailcap noarch 2.1.49-5.el9 baseos 32 k
Instalando dependencias débiles:
apr-util-openssl x86_64 1.6.1-23.el9 appstream 14 k
mod_http2 x86_64 1.15.19-5.el9 appstream 148 k
mod_lua x86_64 2.4.57-5.el9 appstream 60 k
Resumen de la transacción
=====
Instalar 12 Paquetes
Tamaño total de la descarga: 2.0 M
Tamaño instalado: 6.0 M
¿Está de acuerdo [s/N]? s
```

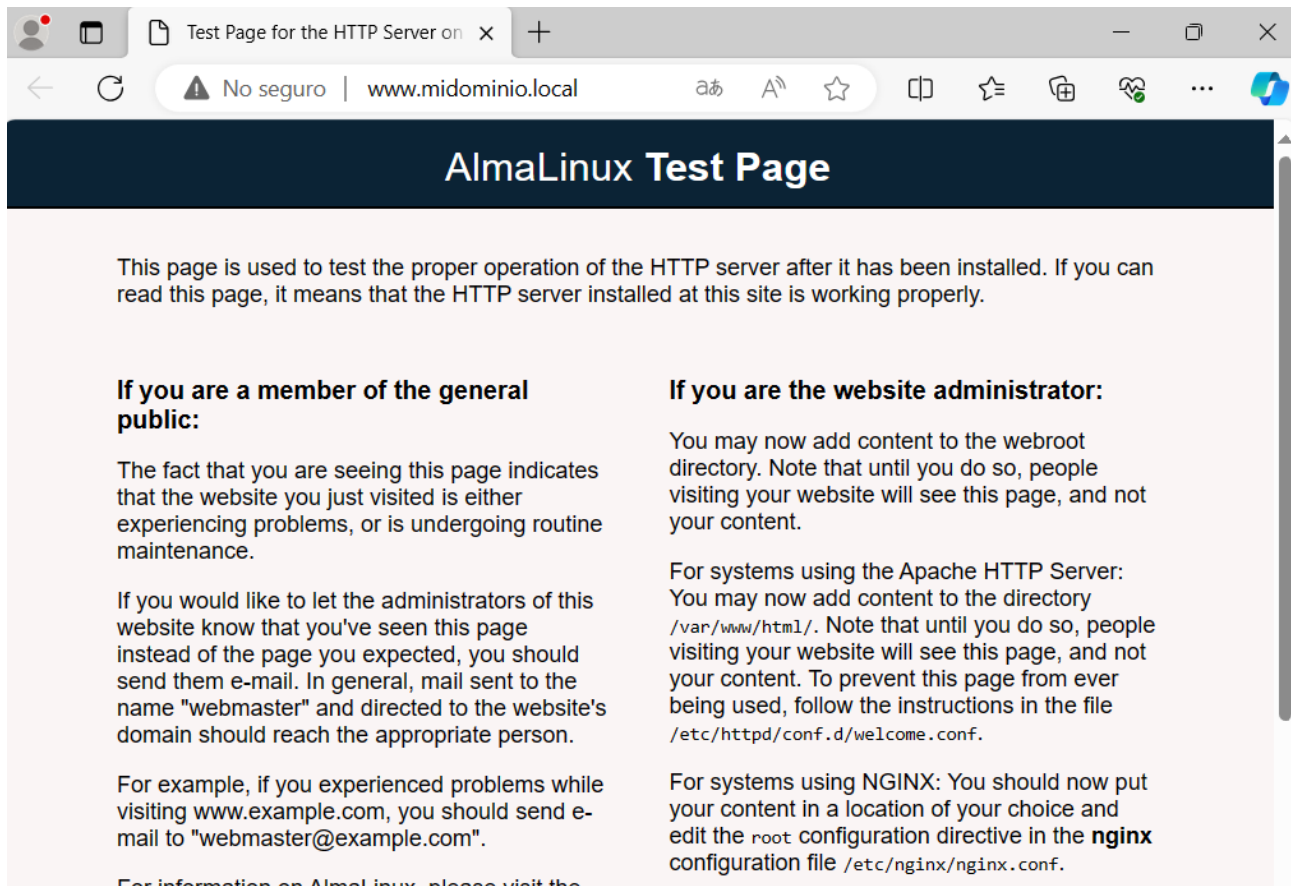
```
[U0287687@linux.as.local ~]$ systemctl start httpd
[U0287687@linux.as.local ~]$ systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service + /usr/lib/systemd/system/httpd.service.
[ 2368.959518] systemd-rc-local-generator[2037]: /etc/rc.d/rc.local is not marked executable, skipping.
[U0287687@linux.as.local ~]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: active (running) since Thu 2024-03-14 10:01:19 CET; 24s ago
     Docs: man:httpd.service(8)
  Main PID: 1809 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/sec"
    Tasks: 213 (limit: 10931)
   Memory: 25.2M
      CPU: 117ms
    CGroup: /system.slice/httpd.service
            └─1809 /usr/sbin/httpd -DFOREGROUND
              └─1810 /usr/sbin/httpd -DFOREGROUND
                └─1811 /usr/sbin/httpd -DFOREGROUND
                  └─1812 /usr/sbin/httpd -DFOREGROUND
                    └─1813 /usr/sbin/httpd -DFOREGROUND

mar 14 10:01:19 linux.as.local systemd[1]: Starting The Apache HTTP Server...
mar 14 10:01:19 linux.as.local httpd[1809]: Server configured, listening on: port 80
mar 14 10:01:19 linux.as.local systemd[1]: Started The Apache HTTP Server.
[U0287687@linux.as.local ~]$
```

Añadimos una nueva regla al firewall para permitir las conexiones *http*:

```
[U0287687@linux.as.local ~]# firewall-cmd --zone=internal --permanent --add-service=http
success
[U0287687@linux.as.local ~]# firewall-cmd --reload
success
```

Tras realizar este proceso, si entramos en el navegador desde la máquina W10 y buscamos <http://www.midominio.local> deberíamos ver la página de prueba de AlmaLinux.



Vamos a cambiar la página de inicio que vemos al visitar <http://www.midominio.local>. Para ello creamos el archivo `/var/www/html/index.html` con el contenido mostrado en la siguiente captura.

```
<!DOCTYPE html>
<head>
  <meta charset="utf-8" />
  <title>Servidor AS</title>
</head>
<body>
  <h1>Administración de Sistemas y Redes</h1>
</body>
</html>
```

Si accedemos de nuevo a la url anterior desde la máquina W10 deberíamos ver la nueva página.



2. Configuración de las páginas web de los usuarios

En primer lugar, modificamos el archivo `/etc/httpd/conf.d/userdir.conf` para que quede como en la siguiente captura:

```
#
# UserDir: The name of the directory that is appended onto a user's home
# directory if a ~user request is received.
#
# The path to the end user account 'public_html' directory must be
# accessible to the webserver userid. This usually means that ~userid
# must have permissions of 711, ~userid/public_html must have permissions
# of 755, and documents contained therein must be world-readable.
# Otherwise, the client will only receive a "403 Forbidden" message.
#
<IfModule mod_userdir.c>
    #
    # UserDir is disabled by default since it can confirm the presence
    # of a username on the system (depending on home directory
    # permissions).
    #
    # UserDir disabled

    #
    # To enable requests to ~/user/ to serve the user's public_html
    # directory, remove the "UserDir disabled" line above, and uncomment
    # the following line instead:
    #
    # UserDir public_html
</IfModule>

#
# Control access to UserDir directories. The following is an example
# for a site where these directories are restricted to read-only.
#
<Directory "/home/*/public_html">
    AllowOverride FileInfo AuthConfig Limit Indexes
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    Require method GET POST OPTIONS
</Directory>
```

Cambiamos los permisos del directoria `/home/asuser`:

```
[U0287687@linux.as.local ~]# chmod 711 /home/asuser/
```

Ejecuta el siguiente comando para permitir que Apache pueda leer contenidos localizados en los directorios de inicio de los usuarios locales:

```
[U0287687@linux.as.local ~]# setsebool -P httpd_read_user_content on
[ 4754.452899] SELinux: Converting 411 SID table entries...
[ 4754.477974] SELinux: policy capability network_peer_controls=1
[ 4754.477978] SELinux: policy capability open_perms=1
[ 4754.477978] SELinux: policy capability extended_socket_class=1
[ 4754.477979] SELinux: policy capability always_check_network=0
[ 4754.477979] SELinux: policy capability cgroup_seclabel=1
[ 4754.477980] SELinux: policy capability nnp_nosuid_transition=1
[ 4754.477980] SELinux: policy capability genfs_seclabel_symlinks=1
[U0287687@linux.as.local ~]#
```

Habilitamos el uso de los directorios `~/public_html` de los usuarios:

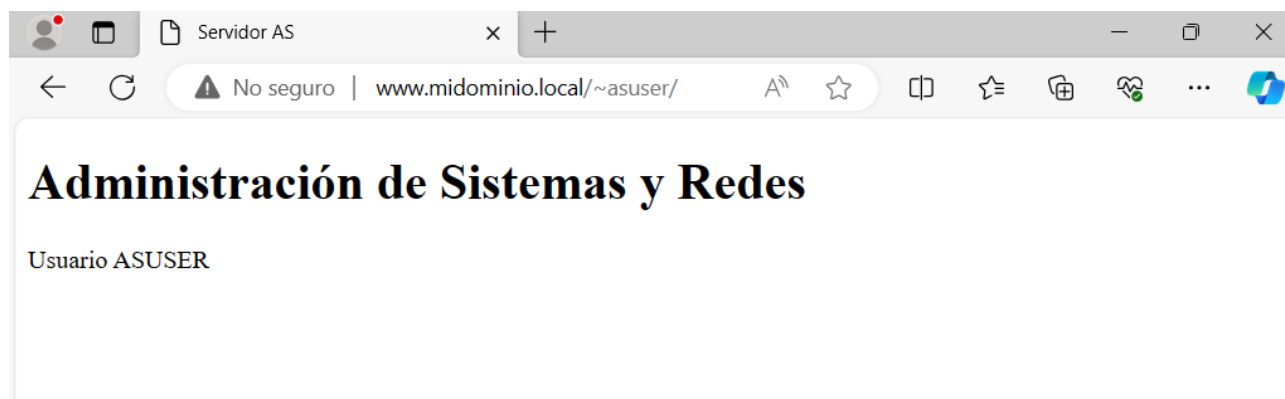
```
[U0287687@linux.as.local ~]# setsebool -P httpd_enable_homedirs on
[ 4882.205750] SELinux: Converting 411 SID table entries...
[ 4882.212208] SELinux: policy capability network_peer_controls=1
[ 4882.212212] SELinux: policy capability open_perms=1
[ 4882.212212] SELinux: policy capability extended_socket_class=1
[ 4882.212213] SELinux: policy capability always_check_network=0
[ 4882.212213] SELinux: policy capability cgroup_seclabel=1
[ 4882.212214] SELinux: policy capability nnp_nosuid_transition=1
[ 4882.212214] SELinux: policy capability genfs_seclabel_symlinks=1
[U0287687@linux.as.local ~]#
```

Ahora, iniciamos sesión como *asuser* y creamos un archivo *index.html* en la carpeta *public_html* de su directorio personal, asegurándonos de que esta tenga los permisos correctos:

```
[U0287687@linux.as.local ~]# su asuser
[U0287687@linux.as.local root]# cd
[U0287687@linux.as.local ~]# mkdir public_html
[U0287687@linux.as.local ~]# chmod 755 -R public_html/
[U0287687@linux.as.local ~]# cd public_html/
[U0287687@linux.as.local public_html]# vi index.html
```

Reiniciamos el servicio *httpd* y comprobamos que podamos acceder desde el navegador de la máquina W10 a la página personal del usuario *asuser*:

```
[U0287687@linux.as.local ~]# systemctl restart httpd
```



3. Configuración del servidor Apache

3.a Ubicación

Creamos la carpeta `/as/web` y copiamos en ella el archivo `index.html` (lo editamos ligeramente para apreciar diferencias en el despliegue y comprobar que el proceso es correcto).

```
[U0287687@linux.as.local ~]$ mkdir /as
[U0287687@linux.as.local ~]$ mkdir /as/web
[U0287687@linux.as.local ~]$ cp /var/www/html/index.html /as/web
```

Editamos el archivo `/etc/httpd/conf/httpd.conf` y cambiamos la directiva `DocumentRoot` para que busque en `/as/web` en vez de en `/var/www/html`. Sustituimos también la sección `<Directory "/var/www">` por `<Directory "/as/web">`:

```
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/as/web"
#
# Relax access to content within /var/www.
#
<Directory "/as/web">
    Options FollowSymLinks
    AllowOverride None
    # Allow open access:
    Require all granted
</Directory>
```

Reiniciamos el servicio `httpd` y asignamos el contexto `httpd_sys_content_t`:

```
[U0287687@linux.as.local web]$ systemctl restart httpd
[U0287687@linux.as.local web]$ chcon -R -h -t httpd_sys_content_t /as/web
```

Accedemos nuevamente a la url a través del navegador de la máquina W10 y comprobamos que se muestra la que está en el directorio `/as/web`:



3.b ServerName

Modificamos las directivas (en el archivo `/etc/httpd/conf/httpd.conf`) `ServerAdmin` y `ServerName` para que se correspondan con mi email y con el nombre de dominio www.midominio.local. Cambiamos también el puerto de escucha del servidor del 80 al 9999. Por último, abrimos en el firewall el puerto 9999.

```
#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed. This address appears on some server-generated pages, such
# as error documents. e.g. admin@your-domain.com
#
ServerAdmin uo287687@uniovi.es
#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
ServerName www.midominio.local:9999
```

```
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on a specific IP address, but note that if
# httpd.service is enabled to run at boot time, the address may not be
# available when the service starts. See the httpd.service(8) man
# page for more information.
#
#Listen 12.34.56.78:80
Listen 9999
```

```
[UO287687@linux.as.local web]# firewall-cmd --zone=internal --add-port=9999/tcp --permanent
success
[UO287687@linux.as.local web]# firewall-cmd --reload
success
```

Reiniciamos el servicio `httpd` y comprobamos que podamos acceder desde el navegador de la máquina W10 a la página por el puerto 9999.



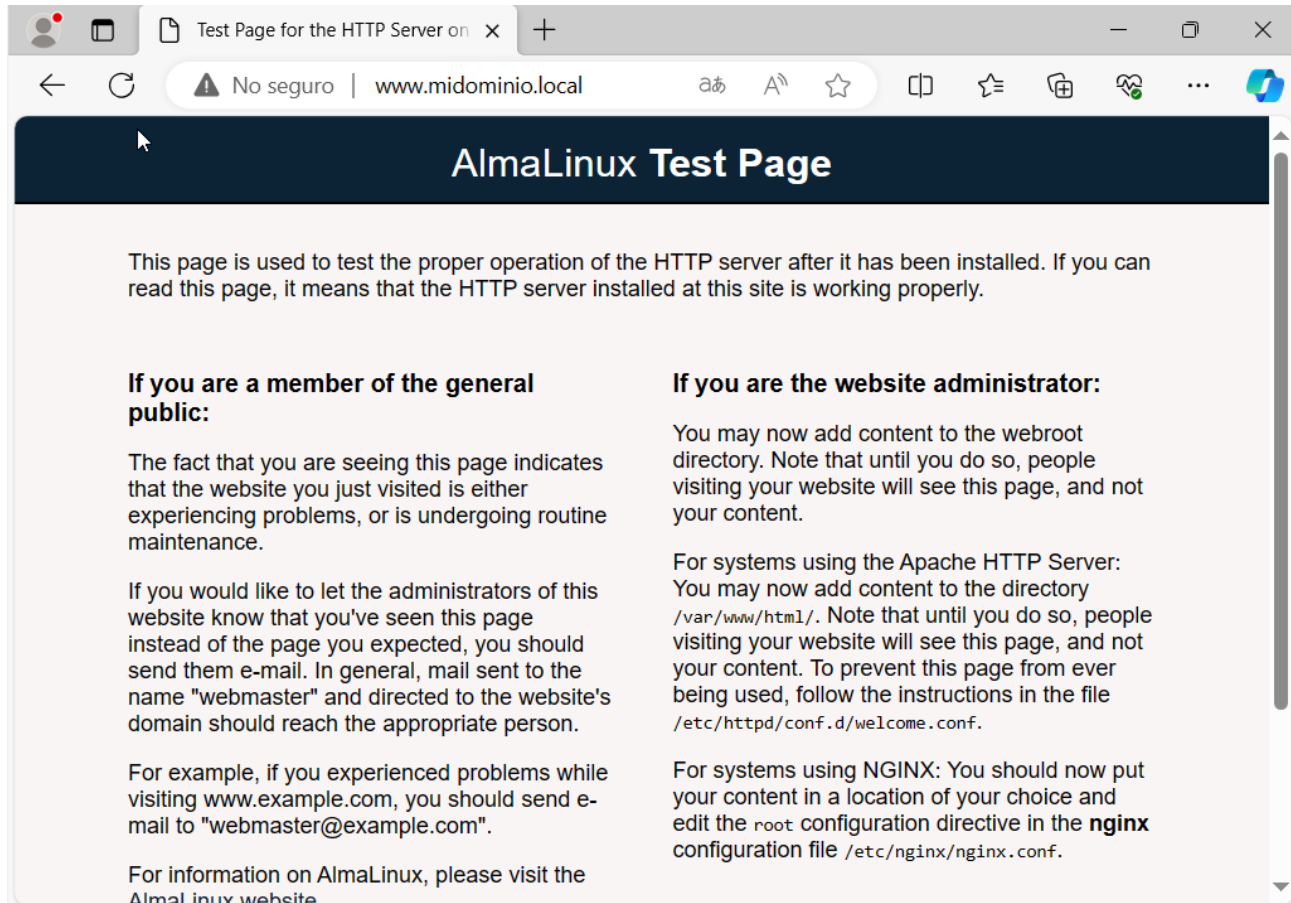
Finalmente, dejamos el servidor escuchando de nuevo en el puerto 80.

3.c Repositorios

Renombramos el archivo `/as/web/index.html` a `/as/web/indice.html`.

```
[U0287687@linux.as.local web]# mv index.html indice.html
```

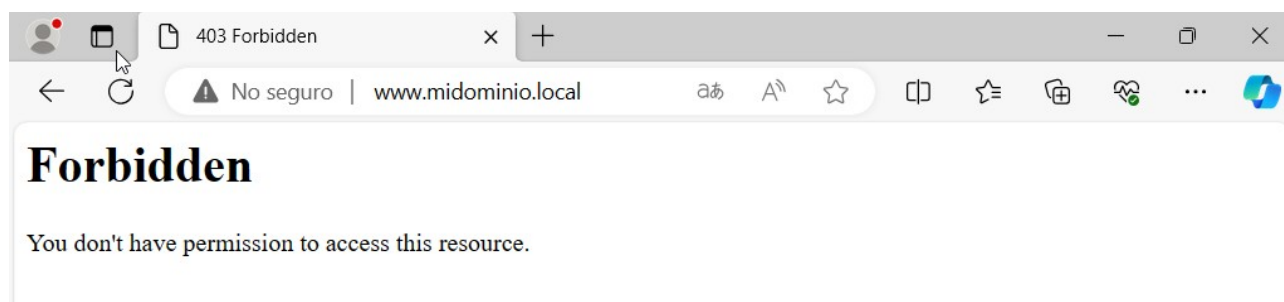
Si accedemos de nuevo a la url <http://www.midominio.local> vemos que se vuelve a cargar la página por defecto:



Comentamos todas las líneas del archivo `/etc/httpd/conf.d/welcome.conf` para desactivar la presentación de la página por defecto en caso de que no encuentre un archivo `index.html`.

```
##
## This configuration file enables the default "Welcome" page if there
## is no default index page present for the root URL. To disable the
## Welcome page, comment out all the lines below.
##
## NOTE: if this file is removed, it will be restored on upgrades.
##
##<LocationMatch "^/+?$">
##     Options -Indexes
##     ErrorDocument 403 /.noindex.html
##</LocationMatch>
##
##<Directory /usr/share/httpd/noindex>
##     AllowOverride None
##     Require all granted
##</Directory>
##
##Alias /.noindex.html /usr/share/httpd/noindex/index.html
##Alias /poweredby.png /usr/share/httpd/icons/apache_pb3.png
##Alias /system_noindex_logo.png /usr/share/httpd/icons/system_noindex_logo.png
```

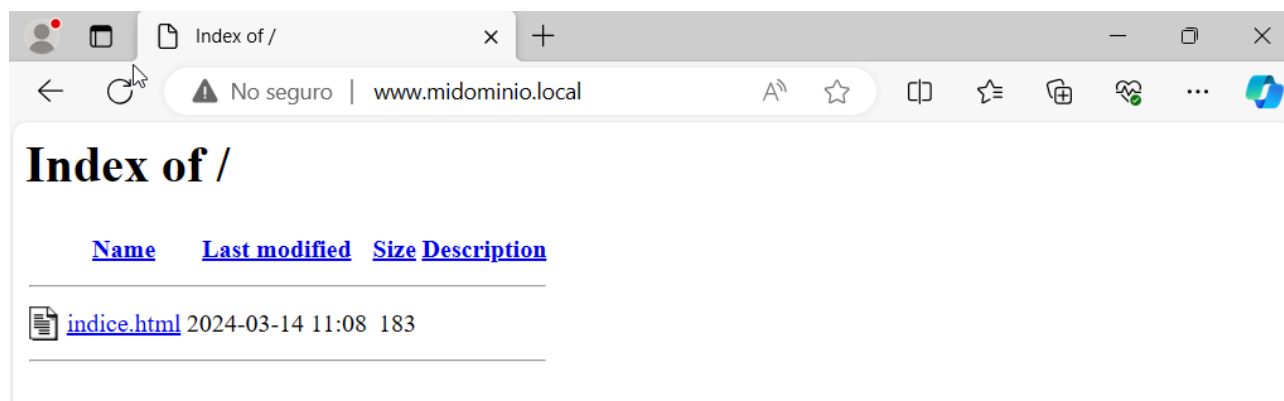
Si reiniciamos el servicio *httpd* y volvemos a acceder a la url en el navegador vemos que no tenemos permisos:



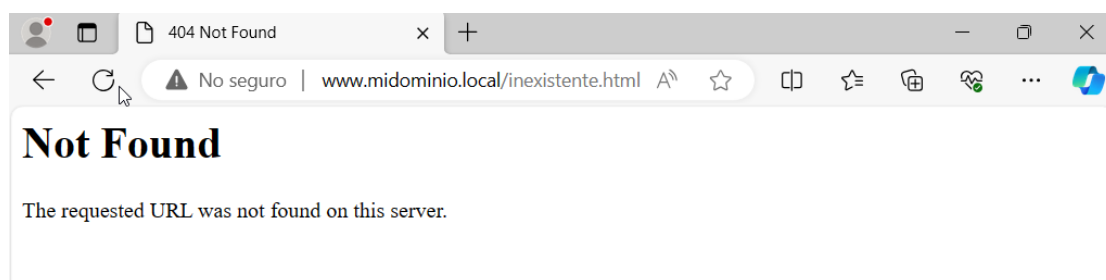
A continuación, modificamos la siguiente sección del archivo */etc/httpd/conf/httpd.conf*

```
<Directory "/as/web">
    Options Indexes FollowSymLinks
    AllowOverride None
    # Allow open access:
    Require all granted
</Directory>
```

Reiniciamos el servicio *httpd* y vemos que podemos acceder al contenido de los directorios en el navegador:



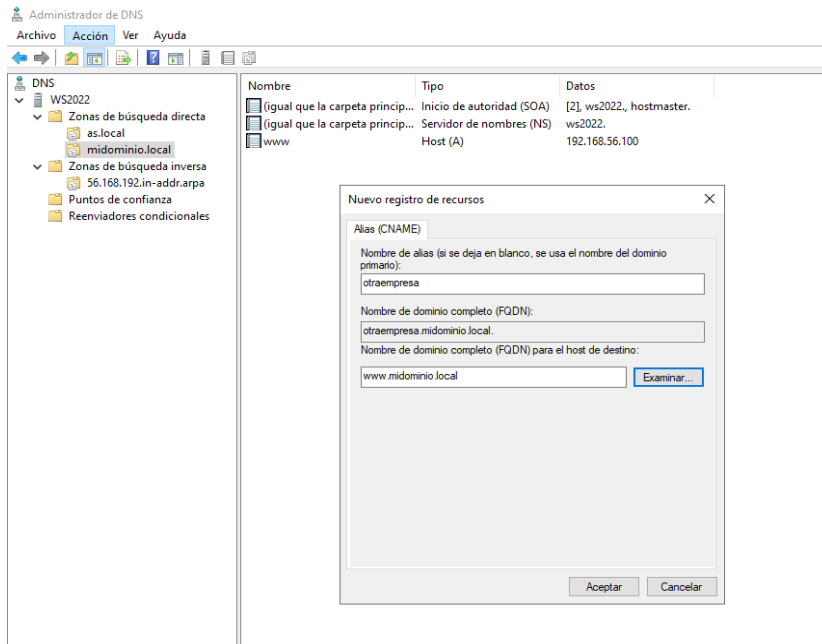
Si accedemos a una página inexistente podemos ver que se registra la petición en */var/log/httpd/access_log* y se puede ver que el archivo no se ha encontrado.



```
[root@localhost ~]# tail -n 5 /var/log/httpd/access_log
192.168.56.111 - - [14/Mar/2024:12:24:30 +0100] "GET /icons/blank.gif HTTP/1.1" 200 148 "http://www.midominio.local/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36 Edg/122.0.0.0"
192.168.56.111 - - [14/Mar/2024:12:24:30 +0100] "GET /icons/text.gif HTTP/1.1" 200 229 "http://www.midominio.local/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36 Edg/122.0.0.0"
192.168.56.111 - - [14/Mar/2024:12:25:22 +0100] "-" 408 - "-" "-"
192.168.56.111 - - [14/Mar/2024:12:25:42 +0100] "GET /inexistente.html HTTP/1.1" 404 196 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36 Edg/122.0.0.0"
192.168.56.111 - - [14/Mar/2024:12:26:34 +0100] "-" 408 - "-" "-"
```

4. Hosts Virtuales

Creamos un nuevo alias para www.midominio.local llamado *otraempresa.midominio.local*:



Creamos el directorio `/as/web/otraempresa`, creamos un archivo `index.html` y añadimos lo siguiente en el archivo `/etc/httpd/conf/httpd.conf` para crear un host virtual:

```
# All of these directives may appear inside <VirtualHost> containers,
# in which case these default settings will be overridden for the
# virtual host being defined.
##
<VirtualHost *:80>
    ServerName otraempresa.midominio.local
    DocumentRoot /as/web/otraempresa

    <Directory /as/web/otraempresa>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
    </Directory>
</VirtualHost>
```

Reiniciamos el servicio `httpd` y comprobamos que se muestre la página de *otraempresa* en el navegador con la url <http://otraempresa.midominio.local>:



5. Autenticación

Configuramos el acceso autorizado para la web *otraempresa.midominio.local*.

Para ello añadimos la directiva *AllowOverride AuthConfig* en la sección *Directory* de */as/web/otraempresa*:

```
<VirtualHost *:80>
    ServerName otraempresa.midominio.local
    DocumentRoot /as/web/otraempresa

    <Directory /as/web/otraempresa>
        Options Indexes FollowSymLinks
        AllowOverride AuthConfig_
        Require all granted
    </Directory>
</VirtualHost>
```

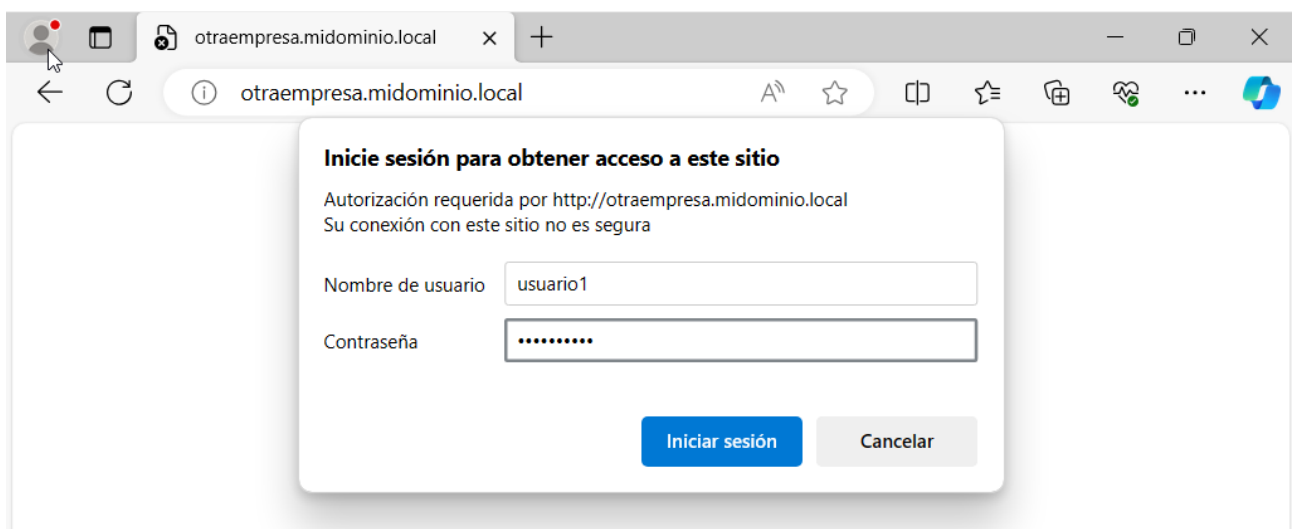
Creamos el archivo */as/web/otraempresa/.htaccess* con el siguiente contenido:

```
AuthType Basic
AuthName "Área Restringida"
AuthUserFile /etc/httpd/password.file
AuthGroupFile /dev/null
Require valid-user
```

A continuación, creamos dos usuarios con sus respectivas contraseña:

```
[U0287687@linux.as.local otraempresa]# htpasswd -c /etc/httpd/password.file usuario1
New password:
Re-type new password:
Adding password for user usuario1
[U0287687@linux.as.local otraempresa]# htpasswd /etc/httpd/password.file usuario2
New password:
Re-type new password:
Adding password for user usuario2
```

Por último reiniciamos el servicio *httpd*, accedemos a la url <http://otraempresa.midominio.local> y vemos que nos pide que nos registremos.



6. Servidor Proxy – squid

En primer lugar instalamos el servidor proxy para web con caché llamado *Squid*.

```
[U0287687@linux.as.local ~]# dnf install squid
```

Lo iniciamos y configuramos el arranque automático:

```
[U0287687@linux.as.local ~]# systemctl start squid
[U0287687@linux.as.local ~]# systemctl enable squid
Created symlink /etc/systemd/system/multi-user.target.wants/squid.service → /usr/lib/systemd/system/squid.service.
[14668.057239] systemd-rc-local-generator[6183]: /etc/rc.d/rc.local is not marked executable, skipping.
```

Editamos el archivo `/etc/squid/squid.conf` y establecemos el rango IP en los que se puede encontrar el servidor:

```
acl localnet src 192.168.56.0/24
acl localnet src 0.0.0.1-0.255.255.255 # RFC 1122 "this" network (LAN)
acl localnet src 10.0.0.0/8 # RFC 1918 local private network (LAN)
acl localnet src 100.64.0.0/10 # RFC 6598 shared address space (CGN)
acl localnet src 169.254.0.0/16 # RFC 3927 link-local (directly plugged) machines
acl localnet src 172.16.0.0/12 # RFC 1918 local private network (LAN)
acl localnet src 192.168.0.0/16 # RFC 1918 local private network (LAN)
acl localnet src fc00::/7 # RFC 4193 local private network range
acl localnet src fe80::/10 # RFC 4291 link-local (directly plugged) machines
```

Descomentamos la línea donde se define el almacenamiento de la memoria caché:

```
# Uncomment and adjust the following to add a disk cache directory.
cache_dir ufs /var/spool/squid 100 16 256
```

Añadimos una nueva regla en el firewall para permitir conexiones de *squid*:

```
[U0287687@linux.as.local ~]# firewall-cmd --zone=internal --add-service=squid
success
```

Configuramos la máquina de W10 para que se conecte al proxy de Linux:

Configuración

Proxy

Configuración manual del proxy

Usa un servidor proxy para conexiones Ethernet o Wi-Fi. Esta configuración no se aplica a conexiones VPN.

Usar servidor proxy

☒ Activado

Dirección: Puerto:

Usar el servidor proxy excepto para direcciones que empiecen con las siguientes entradas. Usa el punto y coma (;) para separar las entradas.

☐ No usar el servidor proxy para direcciones locales (intranet)

Por último, accedemos a una página web y comprobamos que la conexión se realizó a través del proxy viendo el contenido del archivo `/var/log/squid/access.log`:

```
[U0287687@linux.as.local ~]$ tail -f /var/log/squid/access.log
1718428733.889 75 192.168.56.111 TCP_MISS/503 4835 GET http://ipv6.msftconnecttest.com/connecttest.txt - HIER_DIRECT/2a02:26f0:9f00:7::215:270b text/html
1718428733.835 183 192.168.56.111 TCP_MISS/200 386 GET http://www.msftconnecttest.com/connecttest.txt - HIER_DIRECT/96.16.84.22 text/plain
1718428749.569 3 192.168.56.111 TCP_MISS/304 338 GET http://otraempresa.midominio.local/ - HIER_DIRECT/192.168.56.100 -
1718428750.516 1 192.168.56.111 TCP_MISS/304 338 GET http://otraempresa.midominio.local/ - HIER_DIRECT/192.168.56.100 -
1718428758.244 2 192.168.56.111 TCP_MISS/401 741 GET http://www.midominio.local/ - HIER_DIRECT/192.168.56.100 text/html
1718428767.882 1 192.168.56.111 TCP_MISS/200 561 GET http://www.midominio.local/ - HIER_DIRECT/192.168.56.100 text/html
```