

Configuración de Proxy inverso

- Actividad 2 -

Nombre: Yumara Vallejo Vallejo

Curso: 2º Desarrollo de Aplicaciones Web

Año: 2024 / 2025

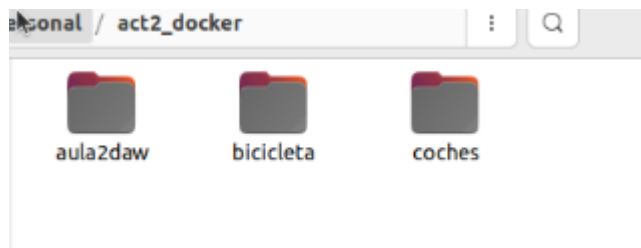
Índice

1. Creación de imágenes, contenedores y configuración de Docker	3
2. Integración de Apache con Docker	7
3. Comprobaciones	11
4. Pruebas de rendimiento	12

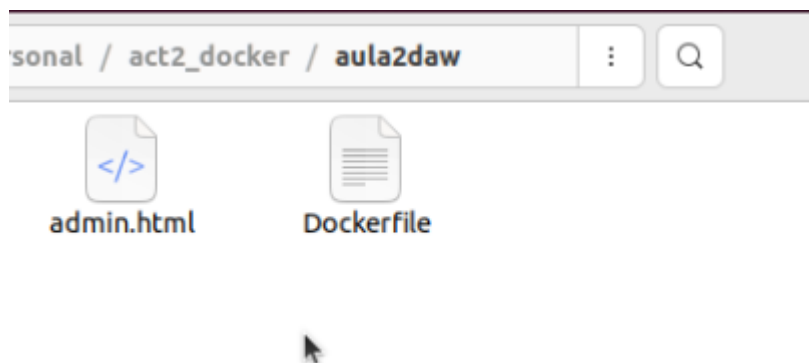
1. Creación de imágenes, contenedores y configuración de Docker

Para este proyecto crearemos una carpeta general para guardar todo, aunque no sea totalmente necesario. Dentro de esta, encontraremos 3 subcarpetas para cada una de las páginas que crearemos, no necesitaremos ningún htaccess porque al copiarlo en el contenedor les pondremos el nombre de index.html, no obstante si no fuese así habríamos de crear un htaccess para cada uno.

Esta será la organización de carpetas:



Dentro de cada carpeta:



Para cada Dockerfile escribiremos lo siguiente (cambiando el nombre del archivo html donde pone admin, por el nombre de cada uno):

```
Abrir  Dockerfile  Guardar
~/act2_docker/aula2daw

1 FROM ubuntu:latest
2
3 RUN apt-get update && apt-get install -y apache2 && apt-get clean
4
5 COPY admin.html /var/www/html/index.html
6
7 EXPOSE 80
8
9 CMD ["apachectl", "-D", "FOREGROUND"]
```

Este Dockerfile nos permitirá tener un contenedor ubuntu con apache2 y copia el archivo html con el nombre de index.html para no tener que añadir un htaccess,

además usará el puerto 80 para que sea accesible desde fuera del contenedor y ejecutará en primer plano apache.

Dentro de los archivos html escribiremos algo así para cada uno pero cambiando el nombre de la web:



```

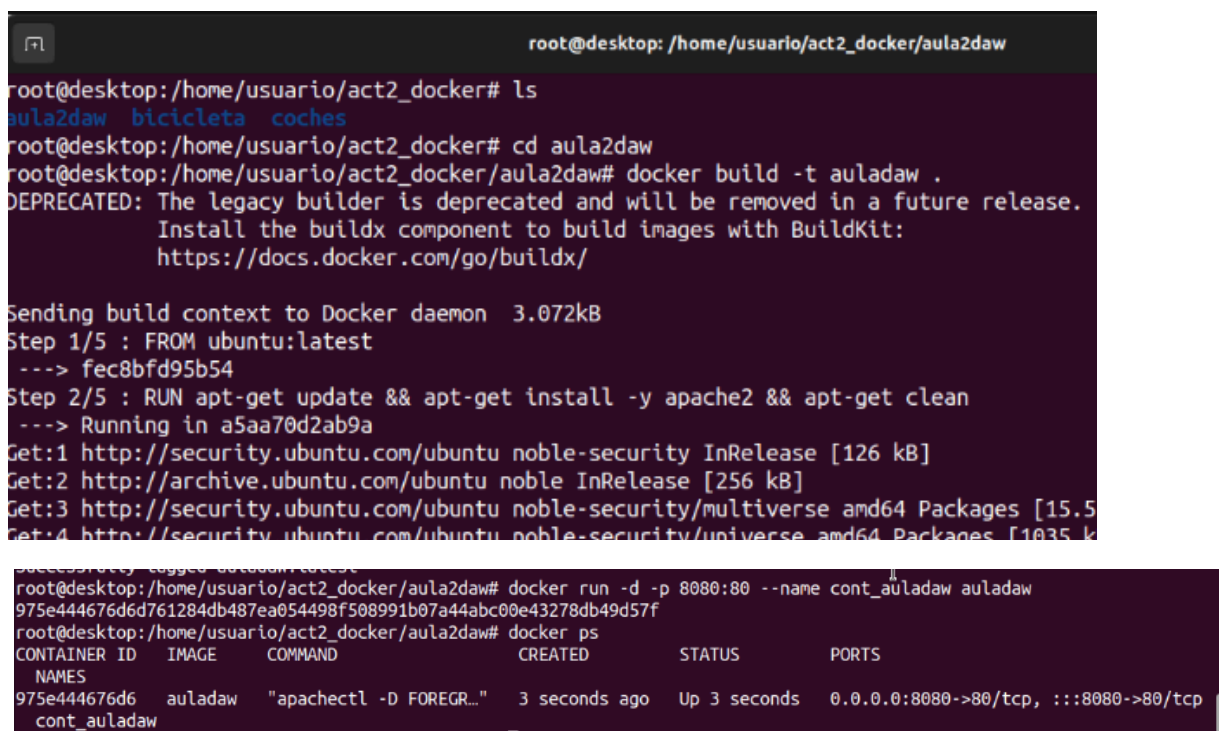
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Mi Coches</title>
5 </head>
6 <body>
7   <p>Funciona www.coches.es. Soy Yumara Vallejo</p>
8 </body>
9 </html>

```

Ahora procedemos a crear las imágenes y los contenedores haciendo lo siguiente:
(Dentro de cada carpeta haciendo uso de cd)

sudo docker build -t auladaw .

sudo docker run -d -p 8080:80 --name cnt_auladaw auladaw



```

root@desktop: /home/usuario/act2_docker/auladaw

root@desktop:/home/usuario/act2_docker# ls
auladaw  bicicleta  coches
root@desktop:/home/usuario/act2_docker# cd auladaw
root@desktop:/home/usuario/act2_docker/auladaw# docker build -t auladaw .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
             Install the buildx component to build images with BuildKit:
             https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  3.072kB
Step 1/5 : FROM ubuntu:latest
--> fec8bfd95b54
Step 2/5 : RUN apt-get update && apt-get install -y apache2 && apt-get clean
--> Running in a5aa70d2ab9a
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [15.5 k
Get:4 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1035 k
Successfully tagged auladaw:latest

root@desktop:/home/usuario/act2_docker/auladaw# docker run -d -p 8080:80 --name cnt_auladaw auladaw
975e444676d6761284db487ea054498f508991b07a44abc00e43278db49d57f
root@desktop:/home/usuario/act2_docker/auladaw# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
NAMES
975e444676d6   auladaw   "apachectl -D FOREGR..." 3 seconds ago  Up 3 seconds  0.0.0.0:8080->80/tcp, :::8080->80/tcp
cnt_auladaw

```

sudo docker build -t bicicleta .

sudo docker run -d -p 8081:80 --name cnt_bicicleta bicicleta

```

root@desktop: /home/usuario/act2_docker/bicicleta
root@desktop:/home/usuario/act2_docker/bicicleta# docker build -t bicicleta .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  3.072kB
Step 1/5 : FROM ubuntu:latest
--> fec8bfd95b54
Step 2/5 : RUN apt-get update && apt-get install -y apache2 && apt-get clean
--> Using cache
--> 85b3be75c164
Step 3/5 : COPY bicicletas.html /var/www/html/index.html
--> 37f3b90593b5
Step 4/5 : EXPOSE 80
--> Running in 8f0aaedf52ca
Removing intermediate container 8f0aaedf52ca
--> d2e3230726c4
Step 5/5 : CMD ["apachectl", "-D", "FOREGROUND"]
--> Running in 3e27f24f8728
Removing intermediate container 3e27f24f8728
--> d71c64dfc75d
Successfully built d71c64dfc75d
Successfully tagged bicicleta:latest
root@desktop:/home/usuario/act2_docker/bicicleta# docker run -d -p 8081:80 --name cont_bicicleta bicicleta
df5cbb201af097d0148529c44cee0dc6969b83fc9b906825a5c58dd13cced2c6
root@desktop:/home/usuario/act2_docker/bicicleta# docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
NAMES
df5cbb201af0   bicicleta  "apachectl -D FOREGR..." 4 seconds ago  Up 3 seconds  0.0.0.0:8081->80/tcp, :::8081->80/tcp
p   cont_bicicleta
975e444676d6   auladaw   "apachectl -D FOREGR..." 2 minutes ago  Up 2 minutes  0.0.0.0:8080->80/tcp, :::8080->80/tcp
p   cont_auladaw

```

sudo docker build -t coches .

sudo docker run -d -p 8082:80 --name cnt_coches coches

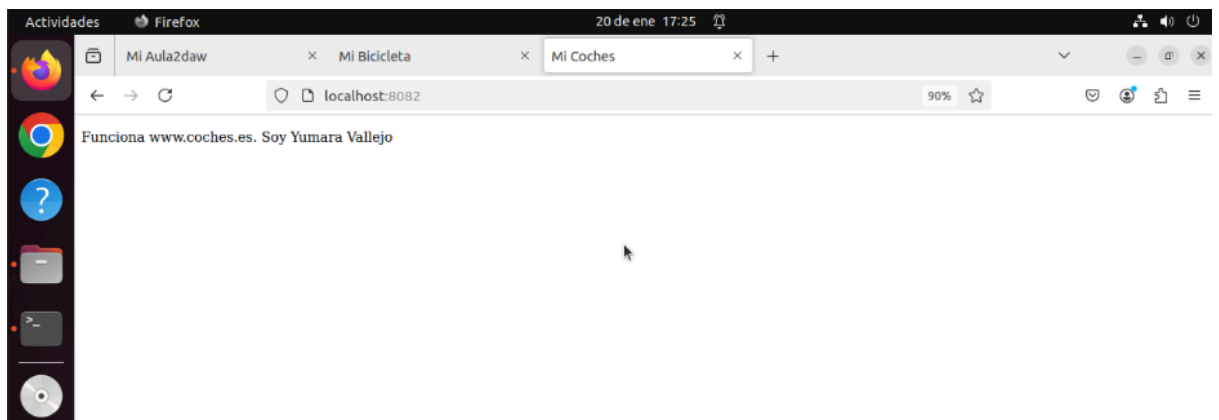
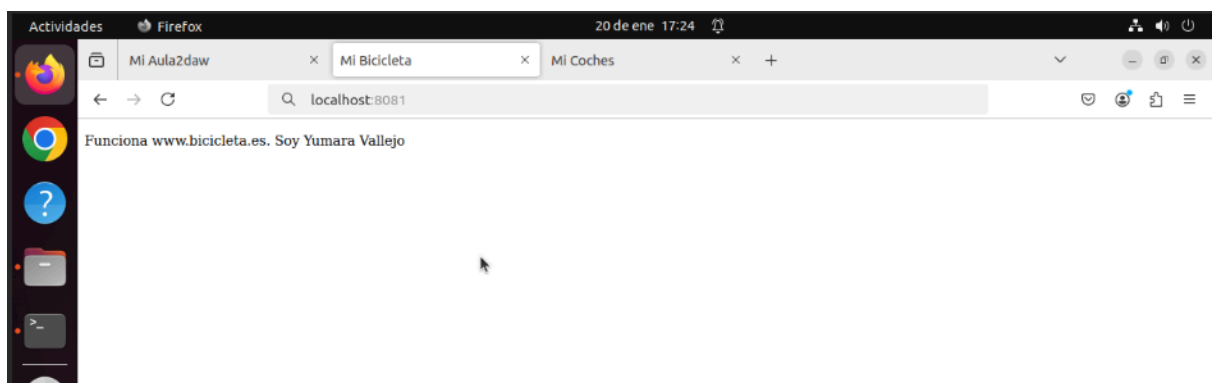
```

root@desktop: /home/usuario/act2_docker/coches
root@desktop:/home/usuario/act2_docker/coches# docker build -t coches .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  3.072kB
Step 1/5 : FROM ubuntu:latest
--> fec8bfd95b54
Step 2/5 : RUN apt-get update && apt-get install -y apache2 && apt-get clean
--> Using cache
--> 85b3be75c164
Step 3/5 : COPY index.html /var/www/html/index.html
--> 775005ca4794
Step 4/5 : EXPOSE 80
--> Running in 10c18b12f7fd
Removing intermediate container 10c18b12f7fd
--> 50b0a4d24efa
Step 5/5 : CMD ["apachectl", "-D", "FOREGROUND"]
--> Running in 7ad1106cae79
Removing intermediate container 7ad1106cae79
--> 5e5b245330e4
Successfully built 5e5b245330e4
Successfully tagged coches:latest
root@desktop:/home/usuario/act2_docker/coches# docker run -d -p 8082:80 --name cnt_coches coches
5cdda1247d8195adb2758494b1a3da6207c6e63487468182482bb0f807e8b472
root@desktop:/home/usuario/act2_docker/coches# docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
NAMES
5cdda1247d81   coches     "apachectl -D FOREGR..." 3 seconds ago  Up 3 seconds  0.0.0.0:8082->80/tcp, :::8082->80/tcp
p   cnt_coches
df5cbb201af0   bicicleta  "apachectl -D FOREGR..." 3 minutes ago  Up 3 minutes  0.0.0.0:8081->80/tcp, :::8081->80/tcp
p   cont_bicicleta

```

Comprobamos que funciona buscando localhost:8080 o el puerto que sea tenemos el 8080, el 8081 y el 8082



2. Integración de Apache con Docker

Si queremos utilizar nombres de dominio para acceder a nuestra web desplegada en un docker, tenemos que utilizar un servidor web con hosts virtual.

En Apache tenemos el concepto de Proxy Reverse, lo que hace es tomar lo que entra en un puerto, y lo reenvía directamente a otro.

Así pues podemos crear un host virtual y redirigir todas las peticiones a ese dominio de un host virtual al contenedor docker creado para ello.

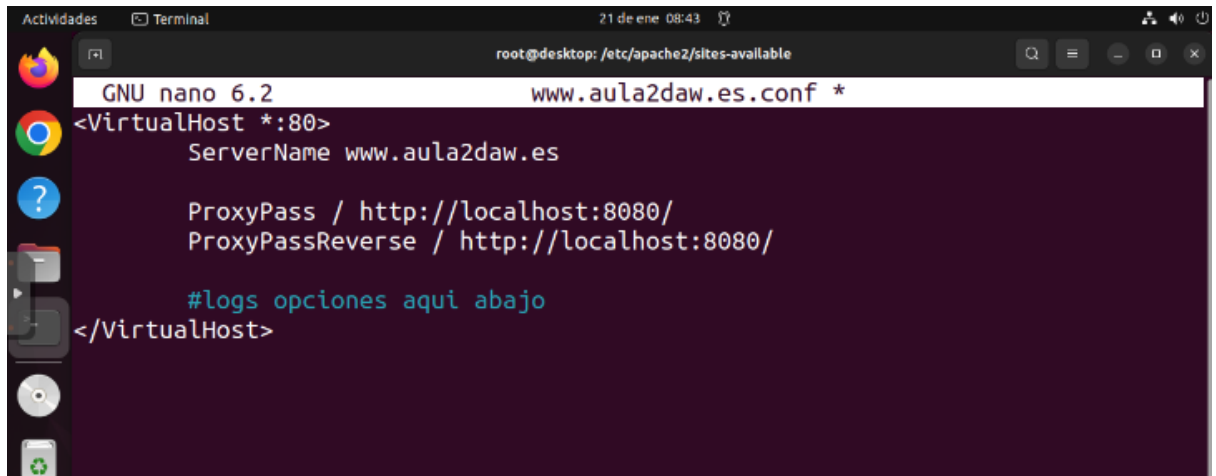
Para empezar tendremos que activar los módulos proxy y proxy_http, el mío ya estaba agregado por lo que no me hace falta, después sería necesario reiniciar apache

```
usuario@desktop: ~  
usuario@desktop:~$ sudo a2enmod proxy  
[sudo] contraseña para usuario:  
Module proxy already enabled  
usuario@desktop:~$ sudo a2enmod proxy_http  
Considering dependency proxy for proxy_http:  
Module proxy already enabled  
Module proxy_http already enabled  
usuario@desktop:~$
```

Para el sitio web de coches.es usaremos https por lo que necesitaremos tener el módulo de ssl activado y crear un certificado usando **sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/cocheses-selfsigned.key -out /etc/ssl/certs/cocheses-selfsigned.crt**

```
root@desktop: /etc/ssl  
++++++  
-----  
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.  
-----  
Country Name (2 letter code) [AU]:ES  
State or Province Name (full name) [Some-State]:Malaga  
Locality Name (eg, city) []:Estepona  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Mi empresa coches  
Organizational Unit Name (eg, section) []:  
Common Name (e.g. server FQDN or YOUR name) []:www.coches.es.  
Email Address []:
```

A continuación, nos dirigimos a la carpeta de configuración de sites available (/etc/apache2/sites-available/) de Apache y crearemos un archivo para el dominio www.aula2daw.es. haciendo en ese directorio `sudo nano www.aula2daw.es.conf`, y otro igual para el de bicicletas

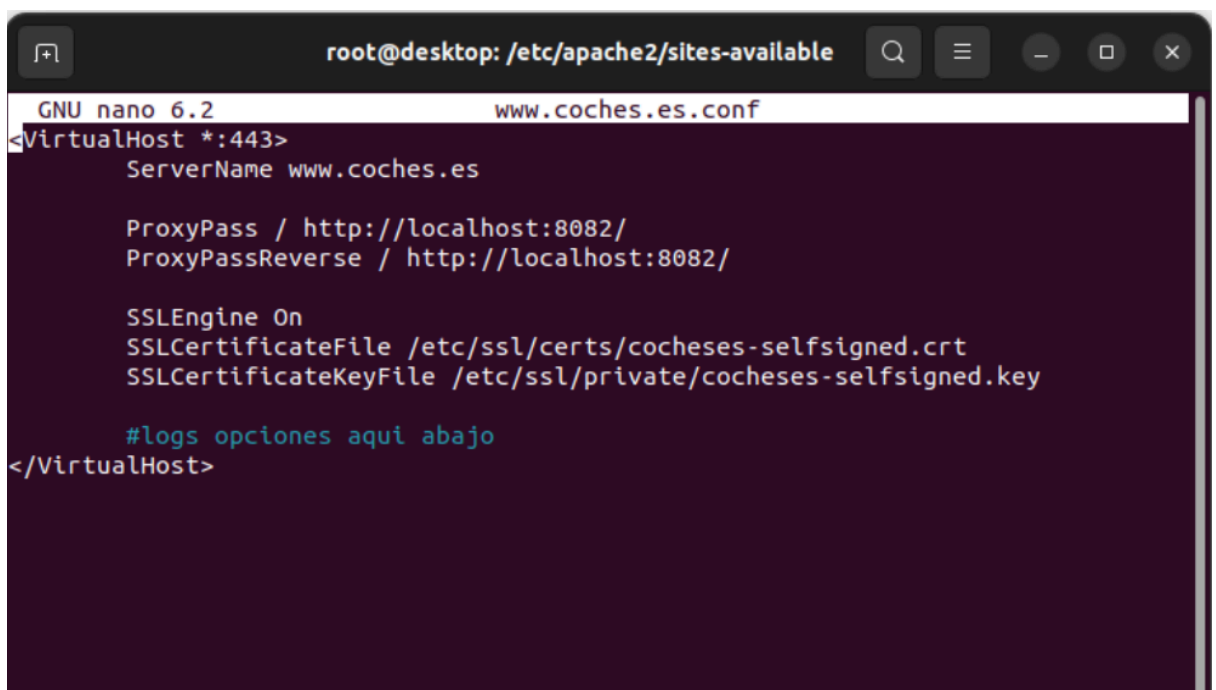


```
GNU nano 6.2 www.aula2daw.es.conf *
<VirtualHost *:80>
    ServerName www.aula2daw.es

    ProxyPass / http://localhost:8080/
    ProxyPassReverse / http://localhost:8080/

    #logs opciones aqui abajo
</VirtualHost>
```

Además, para el de coches como necesita de un certificado para el ssl tendremos el siguiente (ojo con el puerto que también ha sido cambiado a 443)



```
GNU nano 6.2 www.coches.es.conf
<VirtualHost *:443>
    ServerName www.coches.es

    ProxyPass / http://localhost:8082/
    ProxyPassReverse / http://localhost:8082/

    SSLEngine On
    SSLCertificateFile /etc/ssl/certs/cocheses-selfsigned.crt
    SSLCertificateKeyFile /etc/ssl/private/cocheses-selfsigned.key

    #logs opciones aqui abajo
</VirtualHost>
```

Ojo con cambiar los puertos de proxy y proxy reverse a cada uno de los creados, además del nombre del server (8080, 8081 y 8082)

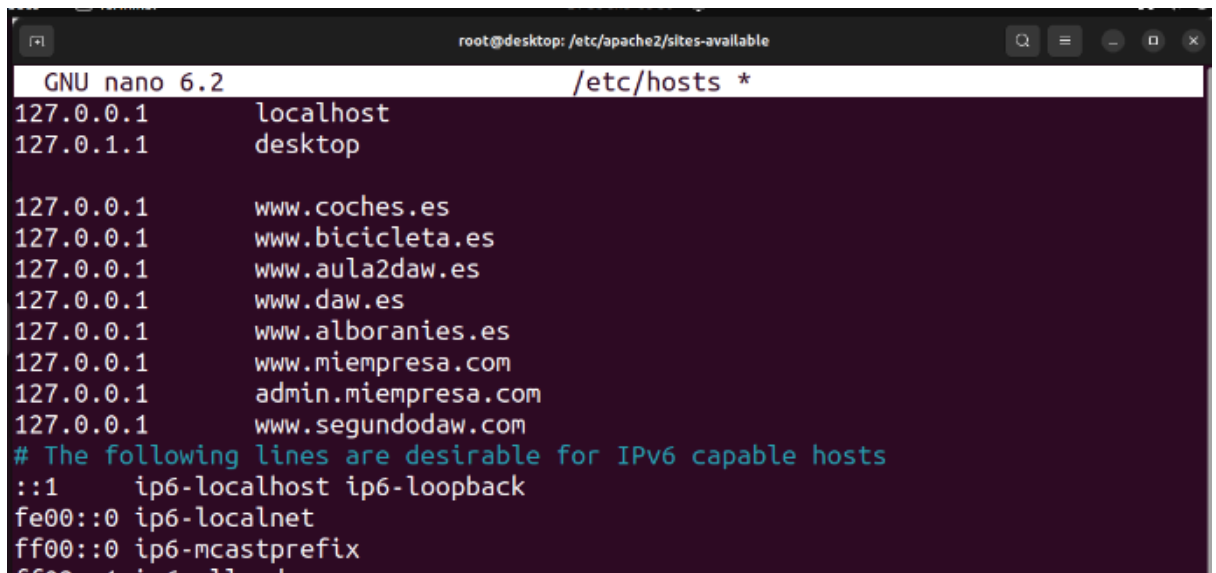


```
root@desktop:/etc/apache2/sites-available# ls
000-default.conf      www.bicicleta.es.conf  www.daw.es.conf
admin.miempresa.com.conf  www.bicileta.com.conf  www.miempresa.com.conf
default-ssl.conf        www.coches.com.conf    www.segundodaw.com.conf
www.aula2daw.es.conf    www.coches.es.conf
root@desktop:/etc/apache2/sites-available#
```

Una vez creadas todas habría que activar los sites y reiniciar apache


```
root@desktop:/etc/apache2/sites-available# a2ensite www.aula2daw.es.conf
Enabling site www.aula2daw.es.
To activate the new configuration, you need to run:
    systemctl reload apache2
root@desktop:/etc/apache2/sites-available# a2ensite www.bicicleta.es.conf
Enabling site www.bicicleta.es.
To activate the new configuration, you need to run:
    systemctl reload apache2
root@desktop:/etc/apache2/sites-available# a2ensite www.coches.es.conf
Enabling site www.coches.es.
To activate the new configuration, you need to run:
    systemctl reload apache2
root@desktop:/etc/apache2/sites-available# service apache2 restart
root@desktop:/etc/apache2/sites-available#
```

Ahora tenemos que modificar el archivo de /etc/hosts haciéndole nano para poder poner que se abran con nuestra ip (como localhost)



```
GNU nano 6.2 /etc/hosts *
127.0.0.1    localhost
127.0.1.1    desktop

127.0.0.1    www.coches.es
127.0.0.1    www.bicicleta.es
127.0.0.1    www.aula2daw.es
127.0.0.1    www.daw.es
127.0.0.1    www.alboranies.es
127.0.0.1    www.miempresa.com
127.0.0.1    admin.miempresa.com
127.0.0.1    www.segundodaw.com
# The following lines are desirable for IPv6 capable hosts
::1          ip6-localhost ip6-loopback
fe00::0      ip6-localnet
ff00::0      ip6-mcastprefix
ff02::1      ip6-allnodes
```

Debemos tener activado apache por lo que cambiaré su puerto de escucha para que no haya fallos, ya que si está en el 8080 no nos dejará activar el apache a causa del contenedor en la 8080 activo, es decir que habría un conflicto de puertos

```
root@desktop: ~
GNU nano 6.2 /etc/apache2/ports.conf *
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80
Listen 8084

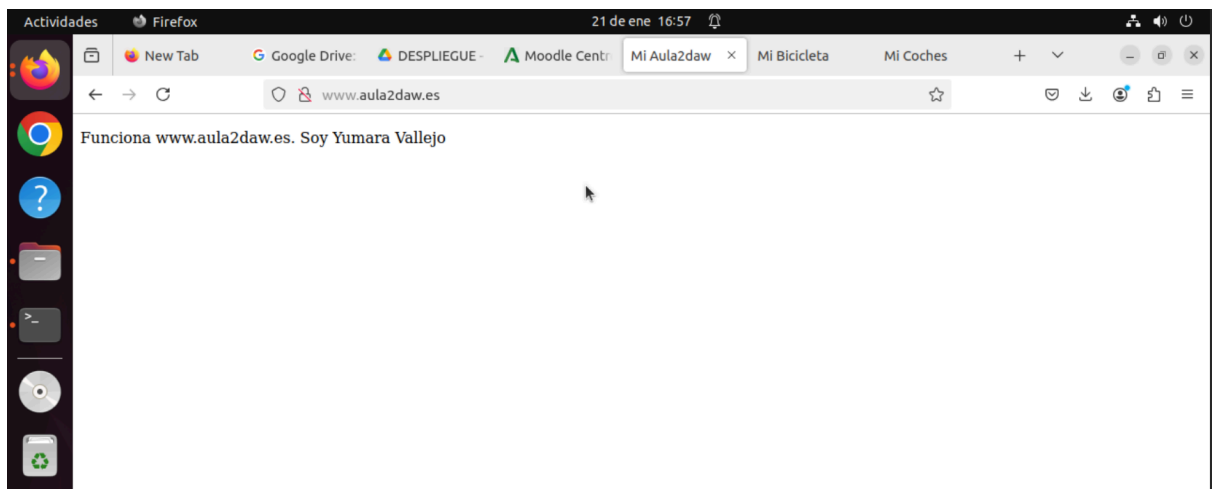
<IfModule ssl_module>
    Listen 443
</IfModule>

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

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

3. Comprobaciones

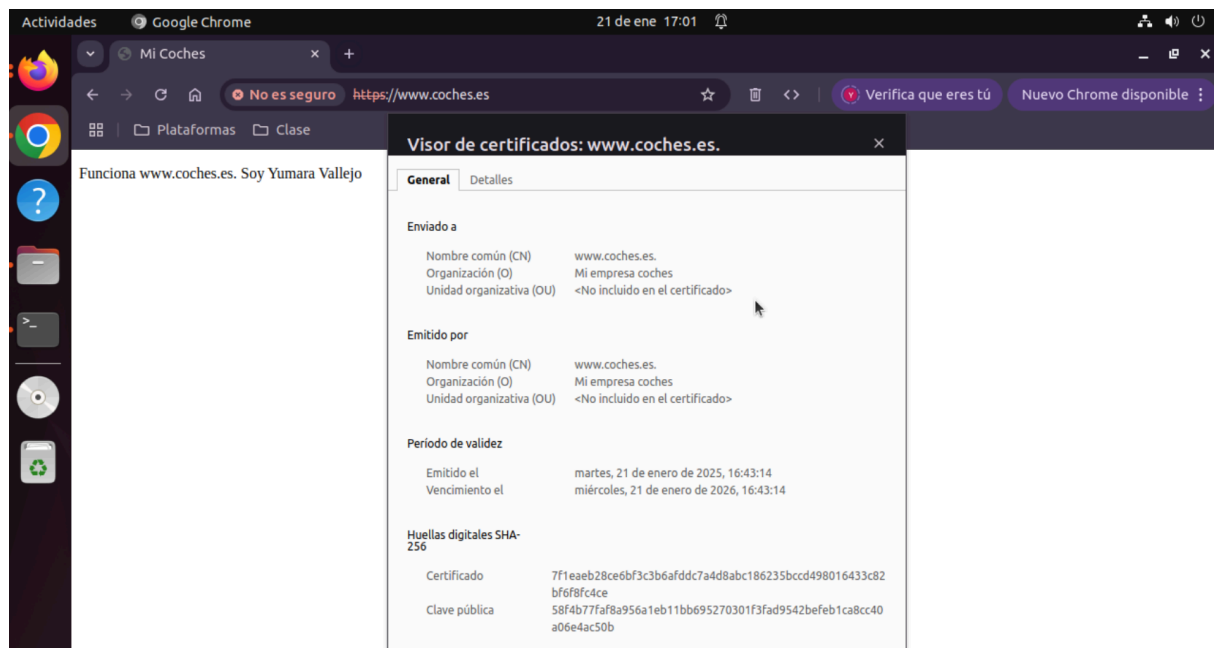
<http://www.aula2daw.es>



<http://www.bicicletas.es>



<https://www.coches.es> Este lo mostraré desde otro navegador para ver los detalles del certificado



4. Comprobaciones en una máquina externa

Para poder conectarnos desde otra máquina, primero necesitamos conocer la dirección ip de la máquina a la que nos vamos a conectar. Haciendo uso del comando **ip a**

```

usuario@desktop: ~
usuario@desktop:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    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: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether bc:24:11:29:cc:13 brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 172.26.103.24/16 brd 172.26.255.255 scope global dynamic noprefixroute ens18
        valid_lft 7083sec preferred_lft 7083sec
    inet6 fe80::c891:cb0b:9889:3469/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:3f:d9:d6:95 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
4: br-9e56063b0e2c: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default

```

La ip está en el número 2, donde pone ens18 y en mi caso es la ip 172.26.103.24/16.

A continuación, en la máquina donde vamos a hacer la conexión, editaremos el archivo hosts dentro de /etc para añadir la ip de la máquina donde queremos conectarnos y escribiremos por cada una de nuestras webs la ip con la ruta

```

root@yumara-ALDA-PRO: /home/yumara
GNU nano 7.2 /etc/hosts
127.0.0.1 localhost
127.0.1.1 yumara-ALDA-PRO

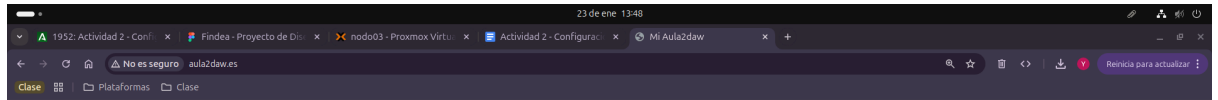
172.26.103.24 www.aula2daw.es
172.26.103.24 www.bicicleta.es
172.26.103.24 www.coches.es
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

```

Ahora solo tendríamos que buscar desde nuestra máquina

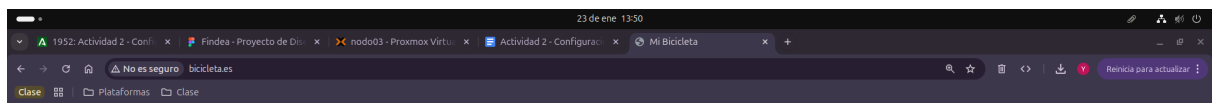
Yumara Vallejo Vallejo

2º DAW



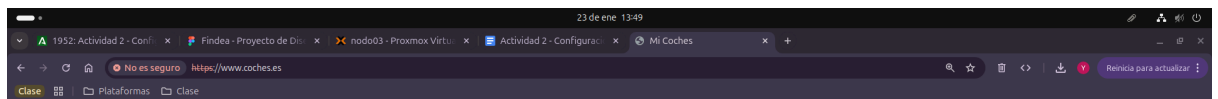
Funciona www.aula2daw.es. Soy Yumara Vallejo

```
root@yumara-ALDA-PRO:/home/yumara
2: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 50:eb:f6:95:05:c4 brd ff:ff:ff:ff:ff:ff
    inet 172.26.102.4/16 brd 172.26.255.255 scope global dynamic noprefixroute enp3s0
        valid_lft 3921sec preferred_lft 3921sec
    inet6 fe80::80a6:2518:26ee:3ac9/64 scope link noprefixroute
```



Funciona www.bicicleta.es. Soy Yumara Vallejo

```
root@yumara-ALDA-PRO:/home/yumara
2: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 50:eb:f6:95:05:c4 brd ff:ff:ff:ff:ff:ff
    inet 172.26.102.4/16 brd 172.26.255.255 scope global dynamic noprefixroute enp3s0
        valid_lft 3921sec preferred_lft 3921sec
    inet6 fe80::80a6:2518:26ee:3ac9/64 scope link noprefixroute
```



Funciona www.coches.es. Soy Yumara Vallejo

```
root@yumara-ALDA-PRO:/home/yumara
2: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 50:eb:f6:95:05:c4 brd ff:ff:ff:ff:ff:ff
    inet 172.26.102.4/16 brd 172.26.255.255 scope global dynamic noprefixroute enp3s0
        valid_lft 3921sec preferred_lft 3921sec
    inet6 fe80::80a6:2518:26ee:3ac9/64 scope link noprefixroute
```

5. Pruebas de rendimiento

Para las pruebas de rendimiento haremos uso de Apache Bench por lo que habrá que instalarlo si no se tiene con **sudo apt install apache2-utils**

Usaremos esto para cada una: **ab -n 1000 -c 10 <http://www.nombre.es/>** Con esto podremos obtener tiempo total de respuesta, tasa de solicitudes por segundo, tiempo promedio por solicitud, tiempo de respuesta mínimo y máximo y transferencia de datos...

ab -n 1000 -c 10 <http://www.aula2daw.es/>

```

root@desktop:/etc/apache2# ab -n 1000 -c 10 http://www.aula2daw.es/
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking www.aula2daw.es (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests

Server Software:      Apache/2.4.58
Server Hostname:      www.aula2daw.es
Server Port:          80

Document Path:        /
Document Length:      142 bytes

Concurrency Level:    10
Time taken for tests:  1.016 seconds
Complete requests:    1000
Failed requests:       0
Total transferred:    412000 bytes
HTML transferred:     142000 bytes
Requests per second:  984.33 [#/sec] (mean)
Time per request:     10.159 [ms] (mean)
Time per request:     1.016 [ms] (mean, across all concurrent requests)
Transfer rate:        396.04 [Kbytes/sec] received

Connection Times (ms)
  min   mean[+/-sd] median   max
Connect:    0      0  0.1    0      1
Processing:  1     10  4.9    9     30
Waiting:    1      8  4.5    7     30
Total:      2     10  4.9    9     30

Percentage of the requests served within a certain time (ms)
 50%    9
 66%   11
 75%   13
 80%   14
 90%   17
 95%   19
 98%   22
 99%   25
100%   30 (longest request)
root@desktop:/etc/apache2#

```

ab -n 1000 -c 10 http://www.bicicletas.es/

```

root@desktop:/etc/apache2# ab -n 1000 -c 10 http://www.bicicletas.es/
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking www.bicicletas.es (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests

Server Software:      Apache
Server Hostname:      www.bicicletas.es
Server Port:          80

Document Path:        /
Document Length:       0 bytes

Concurrency Level:     10
Time taken for tests:   5.636 seconds
Complete requests:      1000
Failed requests:         883
  (Connect: 0, Receive: 0, Length: 883, Exceptions: 0)
Non-2xx responses:      1000
Total transferred:      363702 bytes
HTML transferred:       175717 bytes
Requests per second:    177.42 [#/sec] (mean)
Time per request:       56.364 [ms] (mean)
Time per request:       5.636 [ms] (mean, across all concurrent requests)
Transfer rate:          63.02 [Kbytes/sec] received

Connection Times (ms)
      min   mean[+/-sd] median   max
Connect:    22    25   2.4    25    44
Processing:  23    30   8.4    28   100
Waiting:    23    28   5.4    26    76
Total:      46    55   9.0    53   126

Percentage of the requests served within a certain time (ms)
 50%    53
 66%    55
 75%    57
 80%    59
 90%    65
 95%    69
 98%    87
 99%    95
100%   126 (longest request)

```

ab -n 1000 -c 10 <https://www.coches.es/> los sitios web que usan ssl suelen ser ligeramente más lentos por el tiempo que se necesita para establecer la conexión SSL.

```

root@desktop:/etc/apache2# ab -n 1000 -c 10 http://www.coches.es/
This is ApacheBench, Version 2.3 <$Revision: 1879490 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking www.coches.es (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests


Server Software:      Apache/2.4.52
Server Hostname:      www.coches.es
Server Port:          80

Document Path:        /
Document Length:      80 bytes

Concurrency Level:    10
Time taken for tests:  0.914 seconds
Complete requests:    1000
Failed requests:       0
Total transferred:    349000 bytes
HTML transferred:     80000 bytes
Requests per second:  1093.52 [#/sec] (mean)
Time per request:     9.145 [ms] (mean)
Time per request:     0.914 [ms] (mean, across all concurrent requests)
Transfer rate:        372.69 [Kbytes/sec] received


Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0    0   0.1      0      1
Processing:      1    9   6.5      8     46
Waiting:         0    7   5.7      5     46
Total:           1    9   6.5      8     46


Percentage of the requests served within a certain time (ms)
 50%    8
 66%   10
 75%   12
 80%   14
 90%   18
 95%   21
 98%   27
 99%   31
100%   46 (longest request)

```

6. Github

Esta actividad puede ser descargada a su vez desde Github

[Enlace a actividad de Github](#)