

CONFIGURACIÓN AVANZADA DE SERVIDORES WEB Y HTTPS

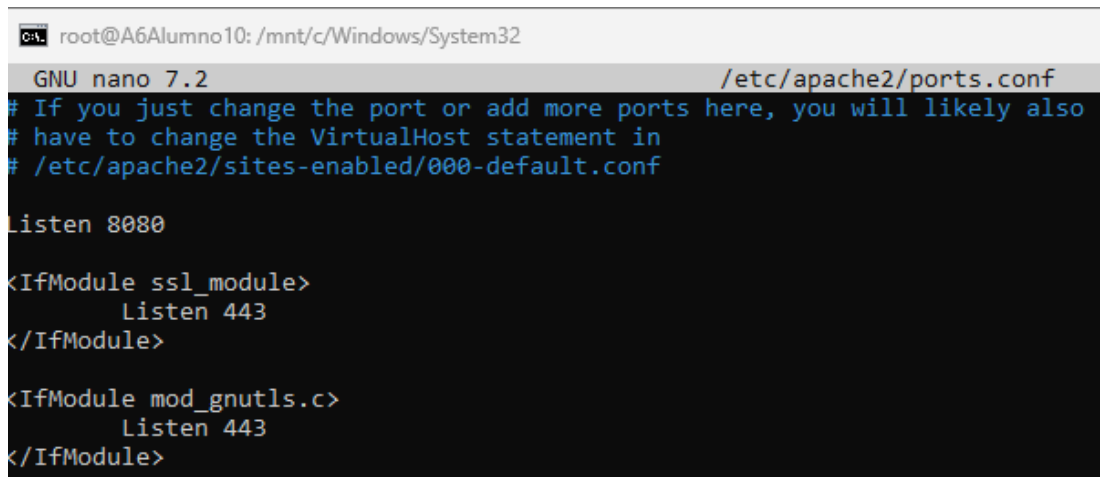
David Arbelaez Mutis

10/10/2025

2º ASIR - Arquitectura en la nube
Prometeo-Caja Mágica

1. INSTALACIÓN Y CONFIGURACIÓN DE APACHE

Se realiza la instalación de apache y la actualización del sistema operativo. Posteriormente se ingresa al archivo de configuración `/etc/apache2/ports.conf` para cambiar el puerto de escucha del 80 al 8080.



```
root@A6Alumno10: /mnt/c/Windows/System32
GNU nano 7.2 /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

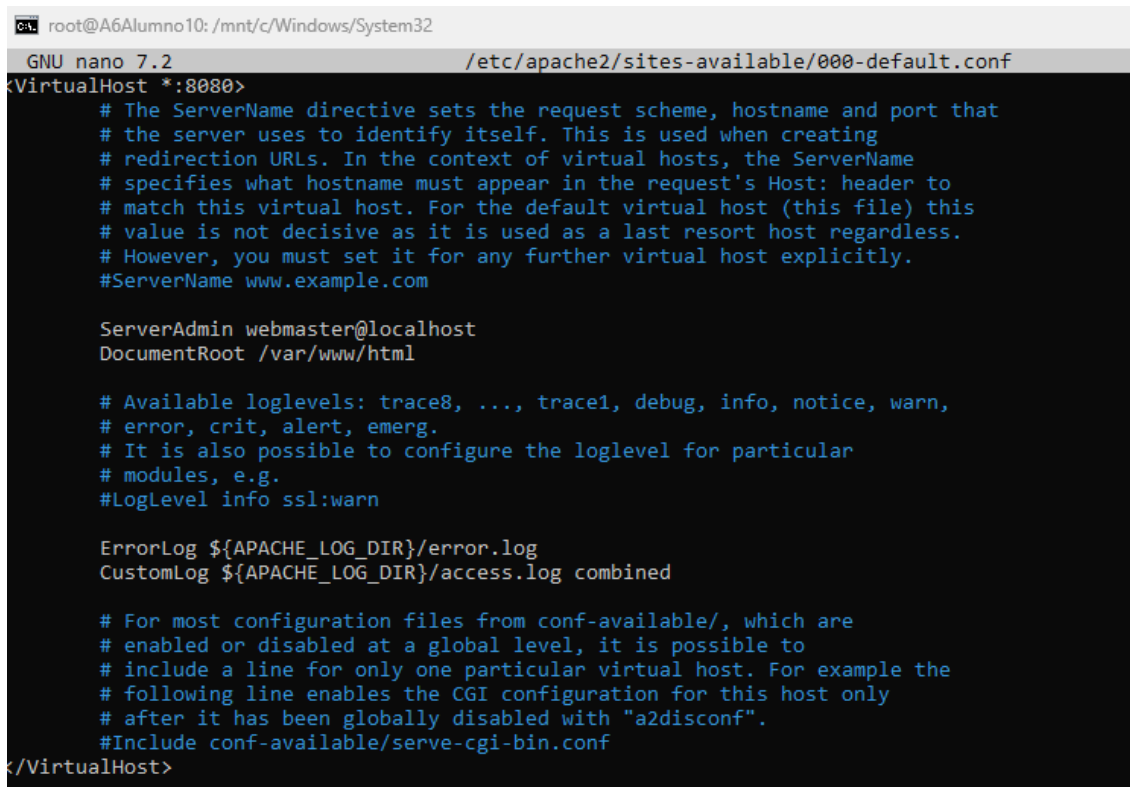
Listen 8080

<IfModule ssl_module>
    Listen 443
</IfModule>

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

<Imagen 1. Cambio del puerto de escucha de apache al 8080>

En la configuración del virtual host de apache se realiza el cambio del puerto al igual que en el paso anterior del 80 al 8080 en la ruta `/etc/apache2/sites-available/000-default.conf`



```
root@A6Alumno10: /mnt/c/Windows/System32
GNU nano 7.2 /etc/apache2/sites-available/000-default.conf
<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 webmaster@localhost
    DocumentRoot /var/www/html

    # 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>
```

<Imagen 2. Cambio de puerto al virtualhost al 8080>

Luego de realizar estos cambios se reinicia el servicio de apache con el comando **systemctl restart apache2** y realizan las pruebas con el comando **netstat -tulpn | grep -E '8080'** con el cual se puede validar que software está utilizando el puerto 8080 y confirmar que la configuración realizada en pasos anteriores esté funcionando.

```
root@A6Alumno10:/mnt/c/Windows/System32# netstat -tulpn | grep -E '8080'
tcp6      0      0 :::8080          :::*              LISTEN     1395/apache2
```

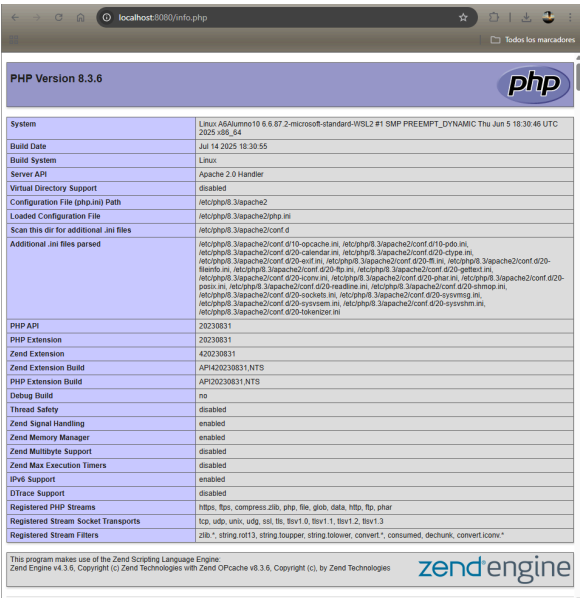
<Imagen 3. Prueba puerto 8080>

Despues de comprobar que el apache esté tomando la configuración que se ha realizado, se procede a instalar php con el comando **sudo apt install php libapache2-mod-php -y** se reinicia el servicio de apache de nuevo y se crea un archivo php que nos va a mostrar la información relacionada a la versión de apache que tenemos instalada con el comando **echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php**

```
root@A6Alumno10:/mnt/c/Windows/System32# echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php
<?php phpinfo(); ?>
```

<Imagen 4. Creacion de archivo php.info>

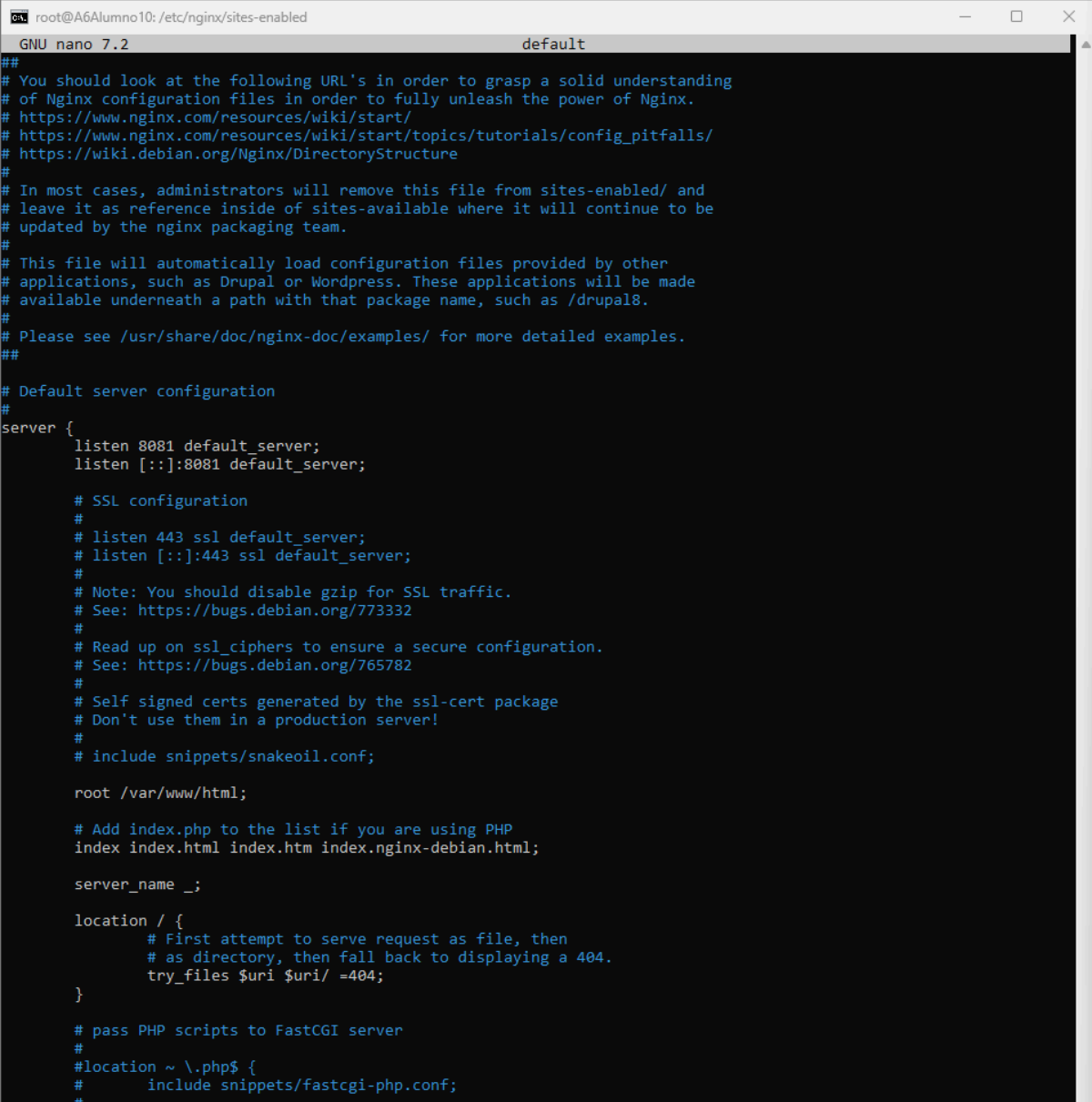
Por último se realiza la prueba de acceso a apache para validar que este cargando la pagina php configurada en el paso anterior, ingresando por el navegador web a **localhost:8080/info.php**



<Imagen 5. visualización del PHPinfo>

2. INSTALACIÓN Y CONFIGURACIÓN DE NGINX

Se instala NGINX con el comando **apt install nginx -y** y posteriormente se accede a la configuración ubicada en `/etc/nginx/sites-enabled` para cambiar el puerto de escucha del 80 al 8081 y garantizar que se ejecute por un puerto diferente al de apache.



```
root@A6Alumno10: /etc/nginx/sites-enabled
GNU nano 7.2                                default
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
    listen 8081 default_server;
    listen [::]:8081 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    root /var/www/html;

    # Add index.php to the list if you are using PHP
    index index.html index.htm index.nginx-debian.html;

    server_name _;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
    }

    # pass PHP scripts to FastCGI server
    #
    #location ~ \.php$ {
    #    include snippets/fastcgi-php.conf;
    #
    #    # ...with the php-fpm user
    #    #include fastcgi-php.conf;
    #}
}
```

<Imagen 6. Cambio de puertos de escucha de NGINX>

Se crea un HTML personalizado para NGINX

```
root@A6Alumno10:/etc/nginx/sites-enabled# echo "<h1>
Servidor Nginx<h1><p>Funcionando en el puerto 8081
</p>" | sudo tee /usr/share/nginx/html/index.html
<h1>Servidor Nginx<h1><p>Funcionando en el puerto 8
081</p>
```

<Imagen 7. Creación del index.html>

Se reinicia el servicio de nginx con el comando **systemctl restart nginx** y se valida el estado posteriormente con **systemctl status nginx**

```
root@A6Alumno10:/etc/nginx/sites-enabled# systemctl restart nginx
root@A6Alumno10:/etc/nginx/sites-enabled# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-10-10 08:52:26 CEST; 27s ago
     Docs: man:nginx(8)
  Process: 1185 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Process: 1187 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
 Main PID: 1188 (nginx)
    Tasks: 17 (limit: 9350)
   Memory: 13.9M (peak: 15.2M)
      CPU: 43ms
   CGroup: /system.slice/nginx.service
           └─1188 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─1189 "nginx: worker process"
               └─1190 "nginx: worker process"
                 └─1191 "nginx: worker process"
                   └─1192 "nginx: worker process"
                     └─1193 "nginx: worker process"
                       └─1194 "nginx: worker process"
                         └─1196 "nginx: worker process"
                           └─1197 "nginx: worker process"
                             └─1198 "nginx: worker process"
                               └─1199 "nginx: worker process"
                                 └─1200 "nginx: worker process"
                                   └─1201 "nginx: worker process"
                                     └─1202 "nginx: worker process"
                                       └─1203 "nginx: worker process"
                                         └─1204 "nginx: worker process"
                                           └─1205 "nginx: worker process"

Oct 10 08:52:26 A6Alumno10 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy se>
Oct 10 08:52:26 A6Alumno10 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy se>
lines 1-31/31 (END)
```

<Imagen 8. Estado del servicio>

Se verifica que servicio está utilizando el puerto 8081 con el comando **sudo netstat -tulpn | grep -E '8081'**

```
root@A6Alumno10:/etc/nginx/sites-enabled# sudo netstat -tulpn
| grep -E '8081'
tcp        0      0 0.0.0.0:8081          0.0.0.0:*
           LISTEN      1188/nginx: master
tcp6       0      0 :::8081              :::*
           LISTEN      1188/nginx: master
```

<Imagen 9. Verificación del servicio que usa el puerto 8081>

Luego se ejecuta el comando **curl http://localhost:8081** y se obtiene la respuesta con el html configurado previamente.

```
root@A6Alumno10:/etc/nginx/sites-enabled# curl http://localhost:8081
<h1>Hola Mundo Desde Nginx</h1><p>Servidor Funcionando Correctamente</p>
root@A6Alumno10:/etc/nginx/sites-enabled#
```

<Imagen 10. Prueba de acceso al index.html con curl localhost>

3. INSTALACIÓN Y CONFIGURACIÓN DE CADDY.

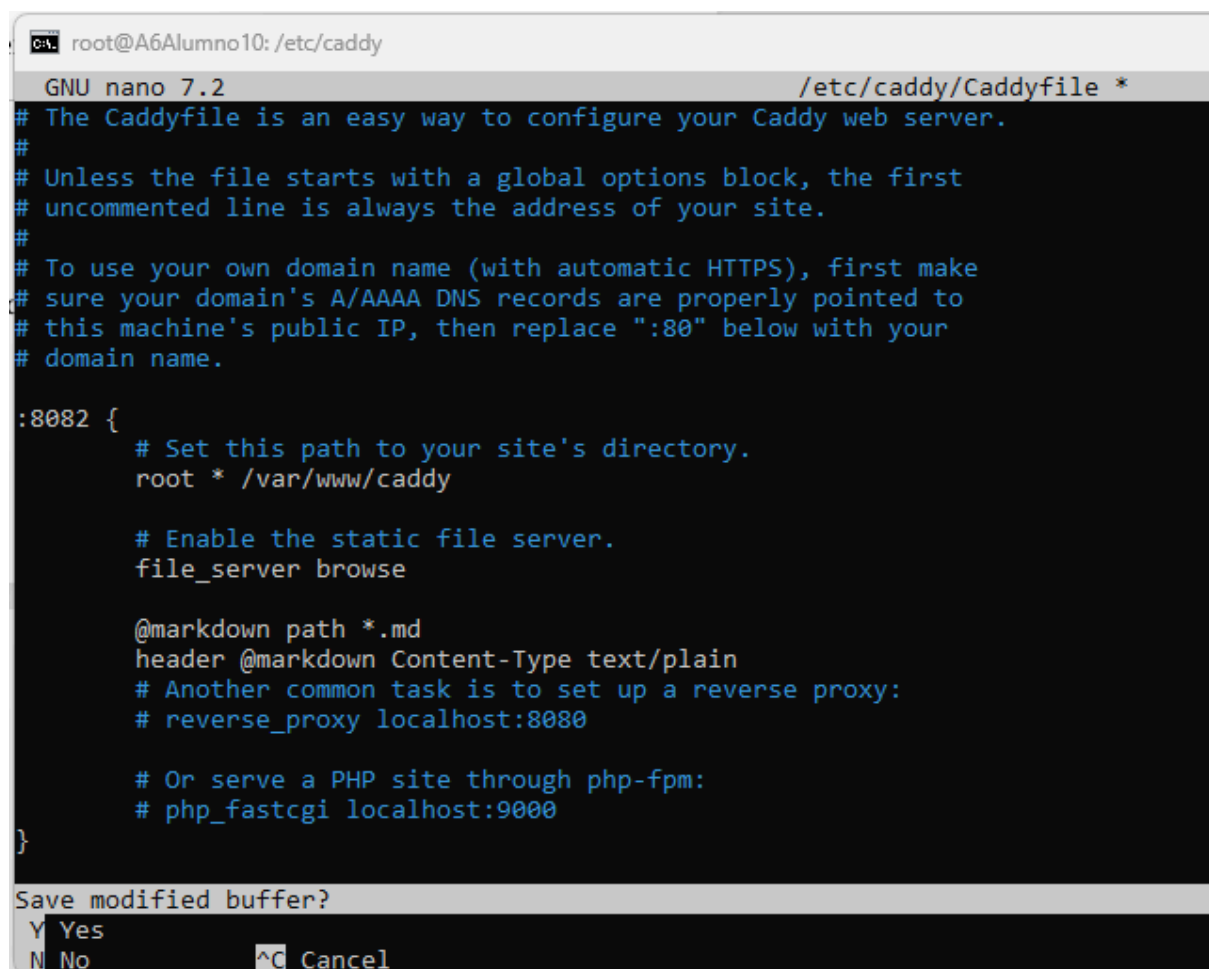
Inicialmente se realiza la instalación de las dependencias necesarias para CADDY con el comando **sudo apt install -y debian-keyring debian-archive-keyring**

apt-transport-https curl. Luego se instalan los repositorios oficiales de CADDY y se instala con **sudo apt update && sudo apt install caddy -y**. Por último se crea un directorio con el comando **sudo mkdir -p /var/www/caddy**

```
root@A6Alumno10:/mnt/c/Windows/System32# curl -o /tmp/test-image.jpg "https://www.python.org/static/apple-touch-icon-144x144-precomposed.png"
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 7382  100 7382    0     0  171k    0 --:--:-- --:--:-- --:--:--  175k
root@A6Alumno10:/mnt/c/Windows/System32# sudo mv /tmp/test-image.jpg /var/www/caddy/test.jpg
```

<Imagen 11. Creación de la imagen de prueba en CADDY>

Se accede al archivo de configuración de caddy en la ruta **/etc/caddy/Caddyfile** para cambiar los puertos por defecto al 8082 y la ruta a la que se creó anteriormente **/var/www/caddy**



```
root@A6Alumno10:/etc/caddy
GNU nano 7.2 /etc/caddy/Caddyfile *
# The Caddyfile is an easy way to configure your Caddy web server.
#
# Unless the file starts with a global options block, the first
# uncommented line is always the address of your site.
#
# To use your own domain name (with automatic HTTPS), first make
# sure your domain's A/AAAA DNS records are properly pointed to
# this machine's public IP, then replace ":80" below with your
# domain name.

:8082 {
    # Set this path to your site's directory.
    root * /var/www/caddy

    # Enable the static file server.
    file_server browse

    @markdown path *.md
    header @markdown Content-Type text/plain
    # Another common task is to set up a reverse proxy:
    # reverse_proxy localhost:8080

    # Or serve a PHP site through php-fpm:
    # php_fastcgi localhost:9000
}

Save modified buffer?
Y Yes
N No ^C Cancel
```

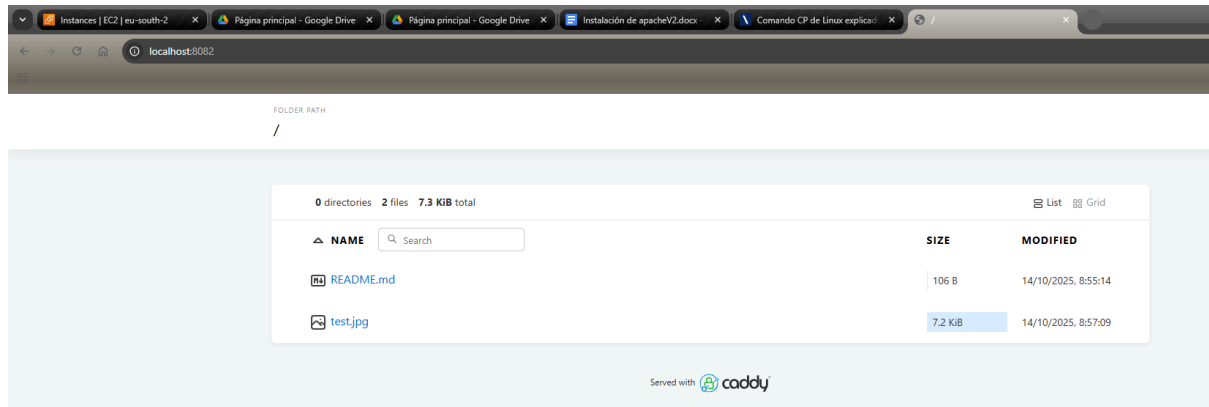
<Imagen 12. Configuración Caddy>

Se realiza la prueba de que el puerto 8082 si este asociado al servicio de Caddy con el comando **sudo netstat -tulpn | grep -E '8082'** y se confirma que si está asociado el puerto al servicio.

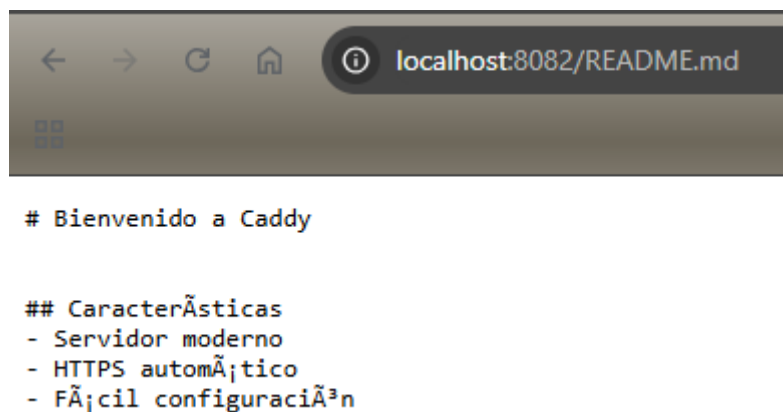
```
root@A6Alumno10:/etc/caddy# sudo netstat -tulpn | grep -E '8082'
tcp6        0      0 :::8082          :::*              LISTEN      2606/caddy
```

<Imagen 13. Uso del puerto 8082>

Finalmente se reinicia el servicio de CADDY y se realiza el acceso por web a la URL **localhost:8082** para comprobar que es accesible.



<Imagen 14. Acceso a caddy por web>



<Imagen 15. Acceso a caddy por web>

4. CONFIGURACIÓN DE HTTPS CON CERTBOT EN APACHE

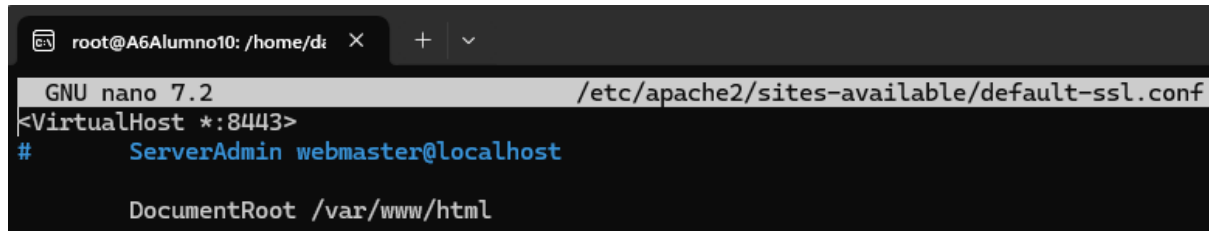
Instalación de CERTBOT con el comando **sudo apt install certbot python3-certbot-apache**.

```
root@A6Alumno10:/etc/caddy# apt install certbot python3-certbot-apache -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  Augeas-lenses libaugeas0 python3-acme python3-augeas python3-certbot python3-configargparse python3-icu python3-jose
  python3-parsedatetime python3-rfc3339
Suggested packages:
  Augeas-doc python-certbot-doc python3-certbot-nginx Augeas-tools python-acme-doc python-certbot-apache-doc
The following NEW packages will be installed:
  Augeas-lenses libaugeas0 python3-acme python3-augeas python3-certbot python3-certbot-apache
  python3-configargparse python3-icu python3-jose python3-parsedatetime python3-rfc3339
0 upgraded, 12 newly installed, 0 to remove and 26 not upgraded.
Need to get 1657 kB of archives.
After this operation, 8599 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble/universe amd64 Augeas-lenses all 1.14.1-1build2 [323 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble/universe amd64 libaugeas0 amd64 1.14.1-1build2 [166 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-josepy all 1.14.0-1 [22.1 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-rfc3339 all 1.1-4 [6744 B]
Get:5 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-acme all 2.9.0-1 [48.5 kB]
Get:6 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-augeas all 0.5.0-1.1 [9124 B]
Get:7 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-configargparse all 1.7-1 [31.7 kB]
Get:8 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-parsedatetime all 2.6-3 [32.8 kB]
Get:9 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-certbot all 2.9.0-1 [267 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble/universe amd64 certbot all 2.9.0-1 [89.2 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble/universe amd64 python3-certbot-apache all 2.9.0-1 [128 kB]
Get:12 http://archive.ubuntu.com/ubuntu noble/main amd64 python3-icu amd64 2.12-1build2 [534 kB]
```

<Imagen 16. Instalación de Cerbot>

Con el CERBOT vamos a generar un certificado SSL autofirmado para asociar a las solicitudes por https de apache. se dejan los valores del certificado con el CN de localhost, pais España y ciudad Madrid con el comando **sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt**

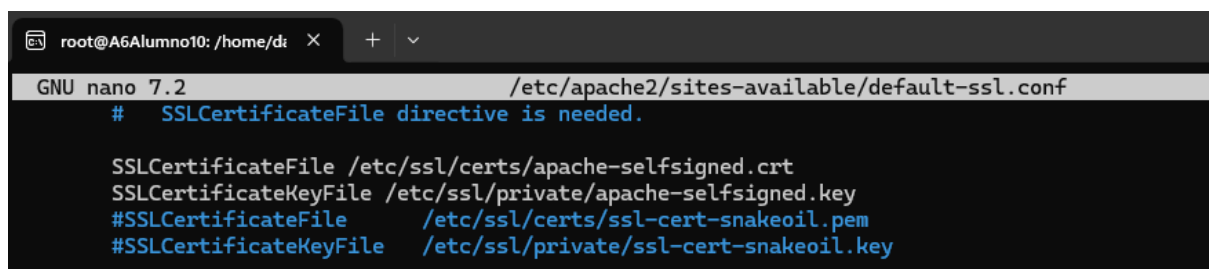
Posteriormente se configura el certificado de apache en la ruta **/etc/apache2/sites-available/default-ssl.conf** y se cambia el puerto certificado que hay por defecto, por el que hemos generado en el paso anterior.



```
root@A6Alumno10: /home/di X + v
GNU nano 7.2 /etc/apache2/sites-available/default-ssl.conf
<VirtualHost *:8443>
#       ServerAdmin webmaster@localhost

       DocumentRoot /var/www/html
```

<Imagen 17. Certificado por defecto SSL de apache>

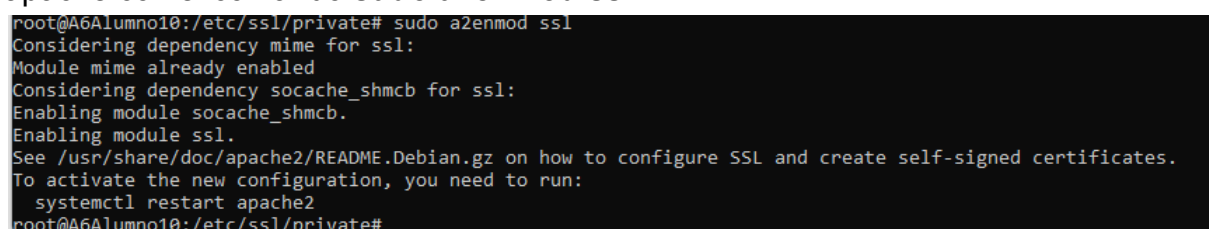


```
root@A6Alumno10: /home/di X + v
GNU nano 7.2 /etc/apache2/sites-available/default-ssl.conf
#       SSLCertificateFile directive is needed.

       SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
       SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
#SSLCertificateFile      /etc/ssl/certs/ssl-cert-snakeoil.pem
#SSLCertificateKeyFile    /etc/ssl/private/ssl-cert-snakeoil.key
```

<Imagen 18. Certificado por defecto SSL de apache>

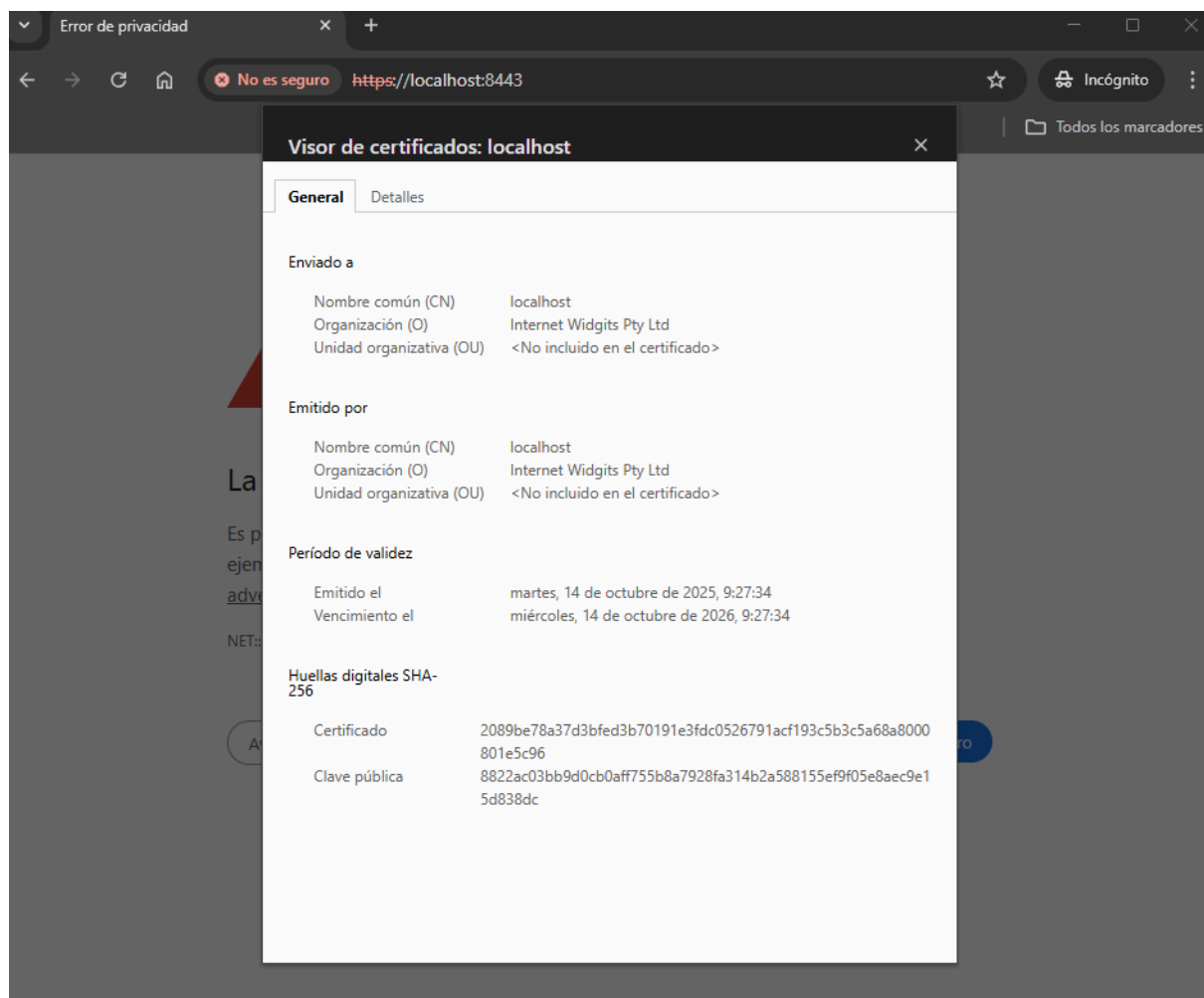
Ya se tiene configurado el servidor de apache para que responda por el puerto 8443 y para que tome el certificado SSL ahora, se procede a habilitar el módulo SSL en apache con el comando **sudo a2enmod ssl**



```
root@A6Alumno10:/etc/ssl/private# sudo a2enmod ssl
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
root@A6Alumno10:/etc/ssl/private#
```

<Imagen 19. Ubicación del archivo index.html>

Por último se realiza el acceso a <https://localhost:8443> y se verifica que el certificado SSL corresponda al que configuramos en pasos anteriores y que la conexión se haga por https



<Imagen 20. Acceso por https y certificado SSL>

Finalmente se realiza una prueba de que los servicios se están ejecutando en simultáneo cada uno con su respectivo puerto con los comandos

```
root@A6Alumno10: /etc/ssl/certs
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-10-14 14:28:42 CEST; 3min 10s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 4382 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 4385 (apache2)
    Tasks: 7 (limit: 9350)
   Memory: 13.6M (peak: 14.5M)
      CPU: 84ms
   CGroup: /system.slice/apache2.service
           └─4385 /usr/sbin/apache2 -k start
             └─4387 /usr/sbin/apache2 -k start
               └─4388 /usr/sbin/apache2 -k start
                 └─4389 /usr/sbin/apache2 -k start
                   └─4390 /usr/sbin/apache2 -k start
                     └─4391 /usr/sbin/apache2 -k start
                       └─4402 /usr/sbin/apache2 -k start

Oct 14 14:28:42 A6Alumno10 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Oct 14 14:28:42 A6Alumno10 systemd[1]: Started apache2.service - The Apache HTTP Server.

● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-10-14 08:37:23 CEST; 5h 54min ago
     Docs: man:nginx(8)
   Main PID: 292 (nginx)
    Tasks: 17 (limit: 9350)
   Memory: 12.2M (peak: 15.7M)
      CPU: 107ms
   CGroup: /system.slice/nginx.service
           └─292 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─293 "nginx: worker process"
               └─294 "nginx: worker process"
                 └─295 "nginx: worker process"
                   └─296 "nginx: worker process"
                     └─298 "nginx: worker process"
                       └─299 "nginx: worker process"
                         └─300 "nginx: worker process"
                           └─301 "nginx: worker process"
                             └─302 "nginx: worker process"
                               └─303 "nginx: worker process"
                                 └─304 "nginx: worker process"
                                   └─305 "nginx: worker process"
                                     └─306 "nginx: worker process"
                                       └─308 "nginx: worker process"
                                         └─309 "nginx: worker process"
                                           └─310 "nginx: worker process"

Oct 14 08:37:23 A6Alumno10 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy se
Oct 14 08:37:23 A6Alumno10 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy ser

● caddy.service - Caddy
   Loaded: loaded (/usr/lib/systemd/system/caddy.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-10-14 09:09:17 CEST; 5h 22min ago
     Docs: https://caddyserver.com/docs/
   Main PID: 2606 (caddy)
    Tasks: 12 (limit: 9350)
   Memory: 10.6M (peak: 12.5M)
      CPU: 2.047s
   CGroup: /system.slice/caddy.service
           └─2606 /usr/bin/caddy run --environ --config /etc/caddy/Caddyfile

lines 1-61
```

<Imagen 21. Ubicación del archivo index.html>

```
root@A6Alumno10:/etc/ssl/certs# sudo netstat -tulpn | grep -E '8080|8081|8082|8443'
tcp        0      0 0.0.0.0:8081          0.0.0.0:*            LISTEN     292/nginx: master p
tcp6       0      0 :::8080              :::*                  LISTEN     4385/apache2
tcp6       0      0 :::8081              :::*                  LISTEN     292/nginx: master p
tcp6       0      0 :::8082              :::*                  LISTEN     2606/caddy
tcp6       0      0 :::8443              :::*                  LISTEN     4385/apache2
root@A6Alumno10:/etc/ssl/certs#
```

<Imagen 22. Prueba del acceso por navegador a localhost:8080>