CONFIGURACIÓN AVANZADA DE SERVIDORES WEB Y HTTPS

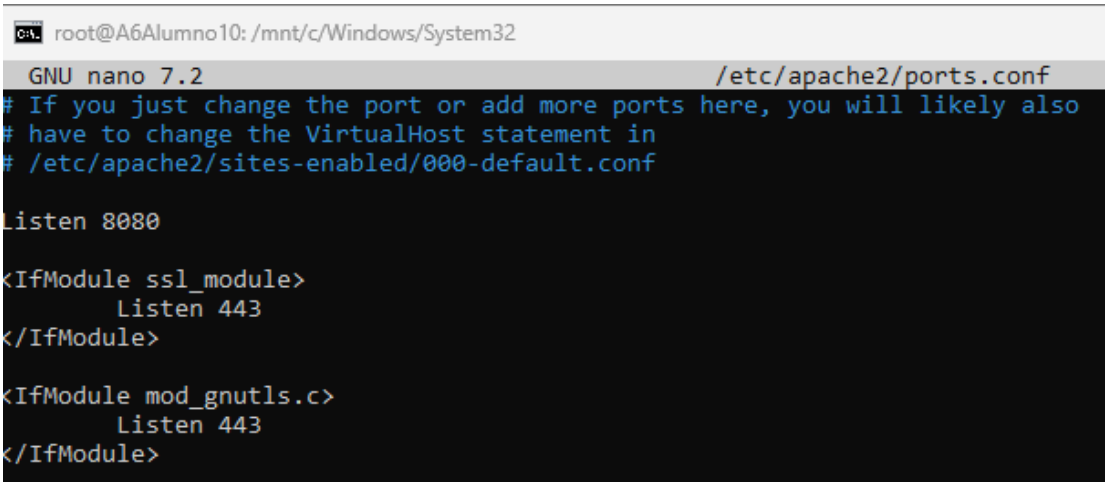
David Arbelaez Mutis

10/10/2025

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

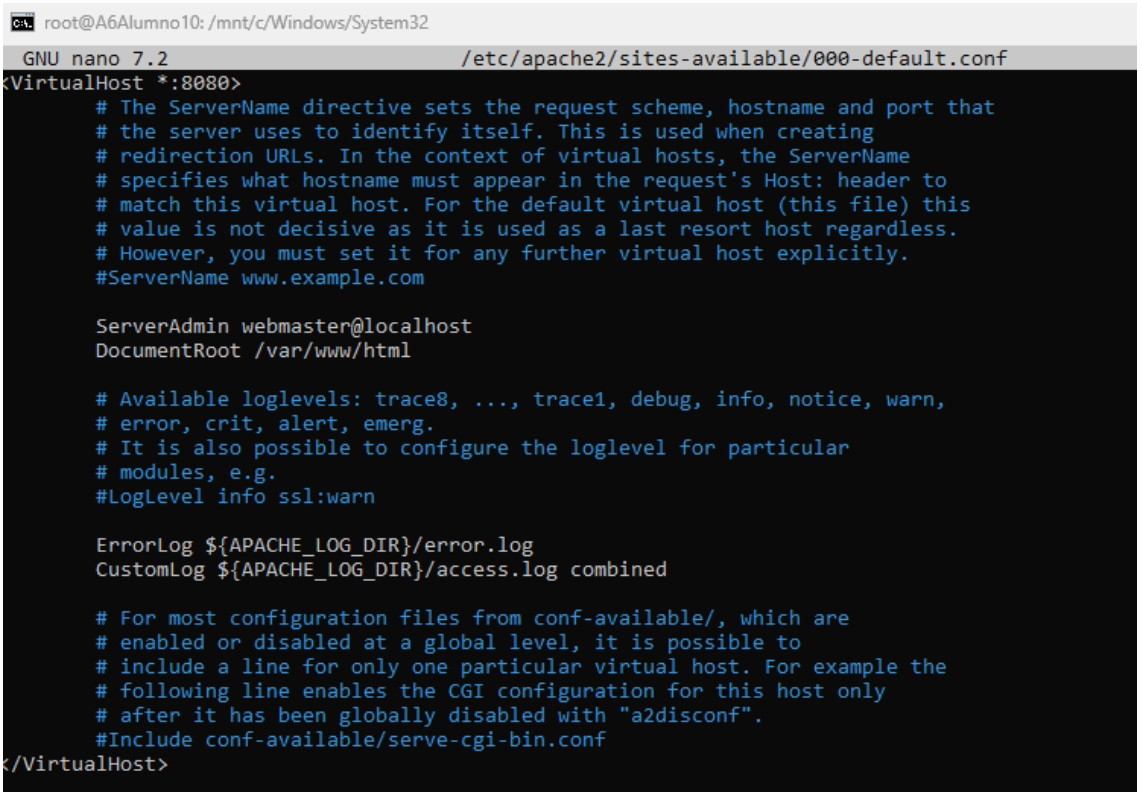
1. INSTALACIÓN Y CONFIGURACIÓN DE APACHE

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.



<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



<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 informaciòn 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(); ?>
```

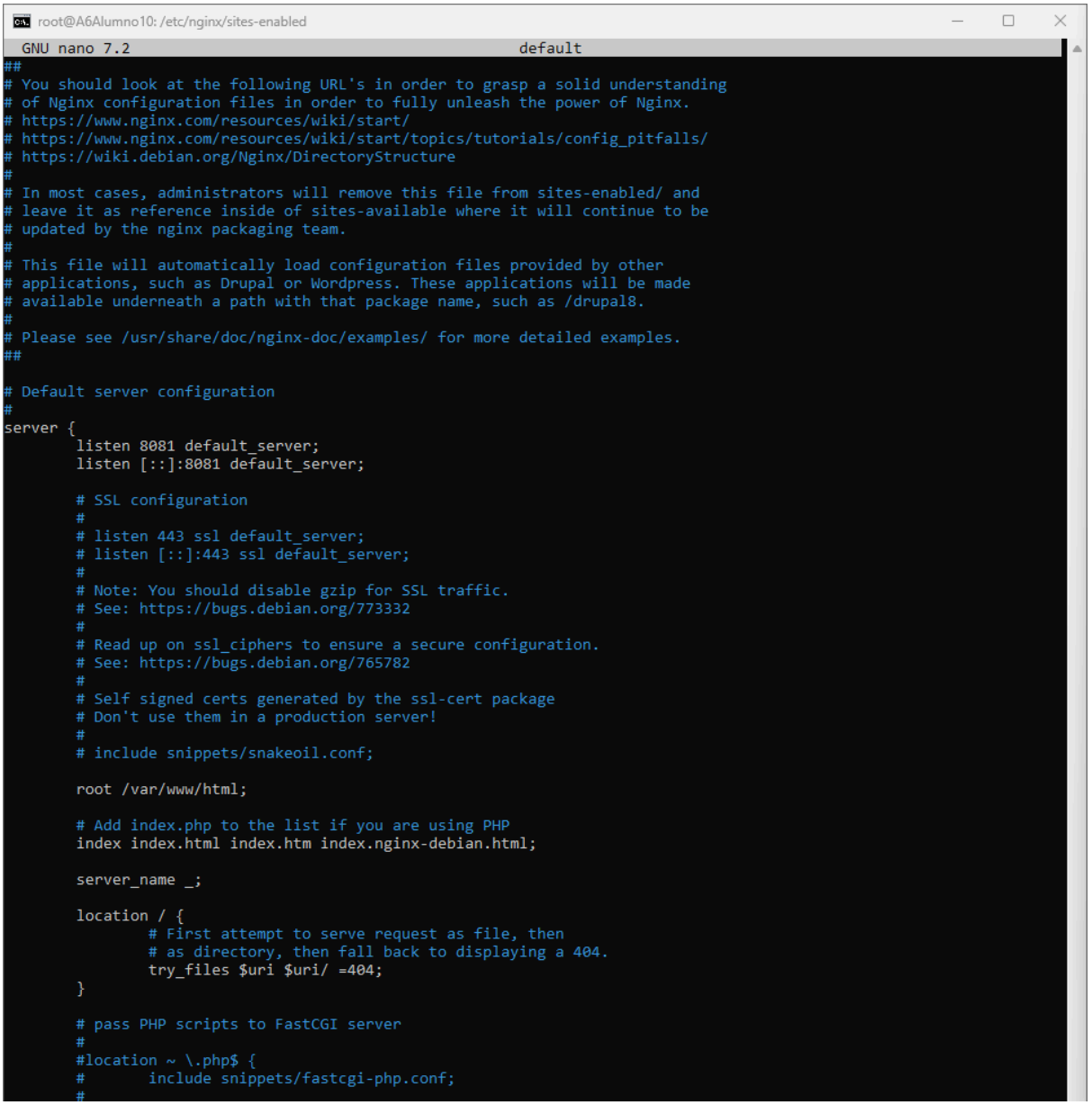

<Imagen 4. Creacion de archivo php.info>

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.



<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 ser>
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.

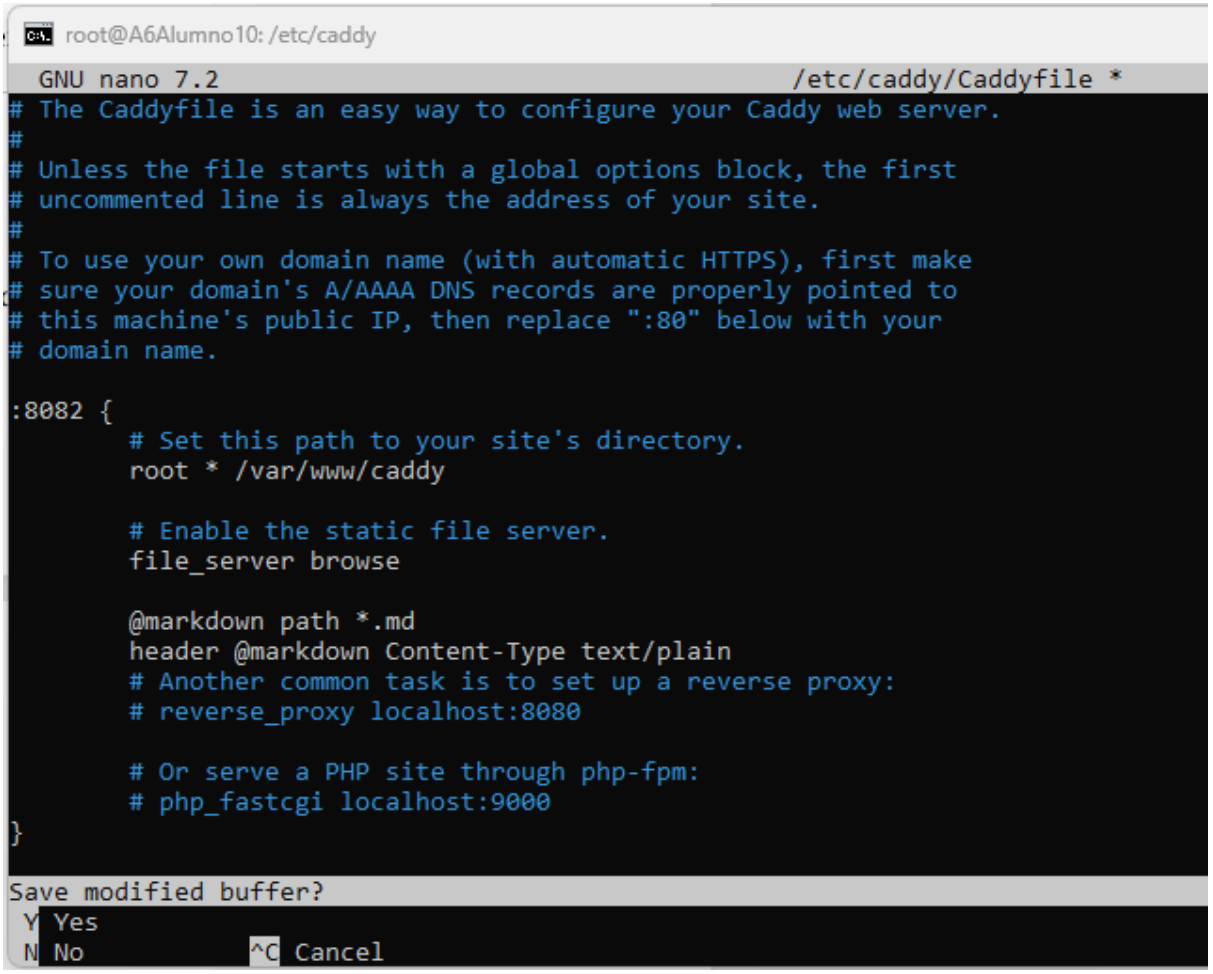
Inicialmente se realiza la instalación de las dependencias necesarias para CADDY con el comando **sudo apt install -y debian-keyring debian-archive-keyring**

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-144x14
4-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**



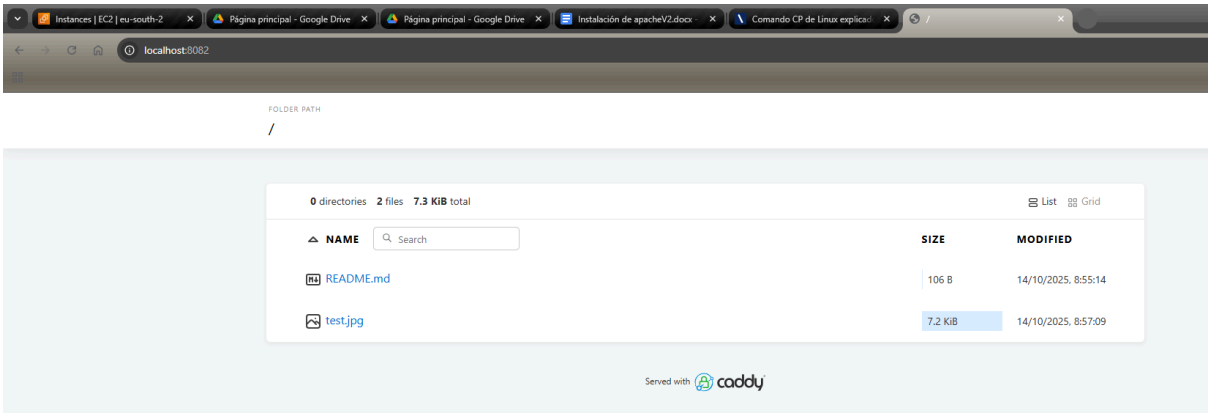
<Imagen 12. Configuración Caddy>

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.

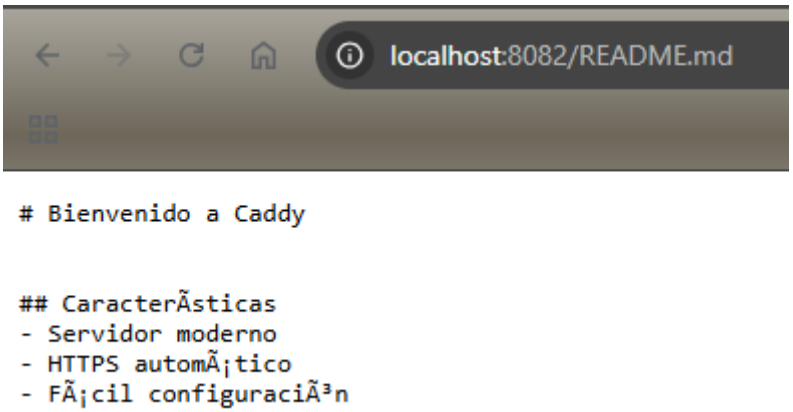
```
root@A6Alumno10:/etc/caddy# sudo netstat -tulpn | grep -E '8082'
tcp6       0      0 :::8082                 :::*                    LISTEN      2606/caddy
```

<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.



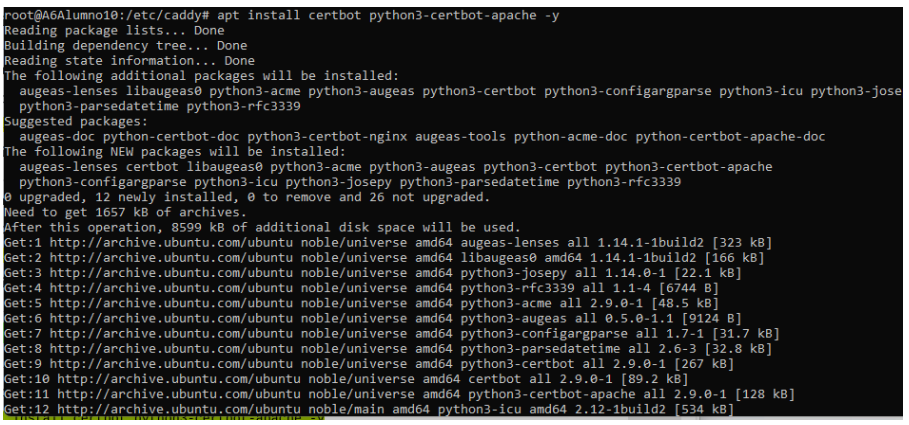
<Imagen 14. Acceso a caddy por web>



<Imagen 15. Acceso a caddy por web>

4. CONFIGURACIÓN DE HTTPS CON CERTBOT EN APACHE

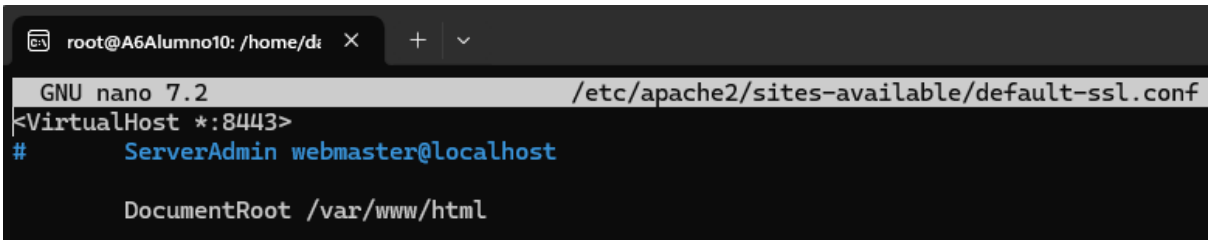
Instalación de CERTBOT con el comando **sudo apt install certbot python3-certbot-apache -y.**



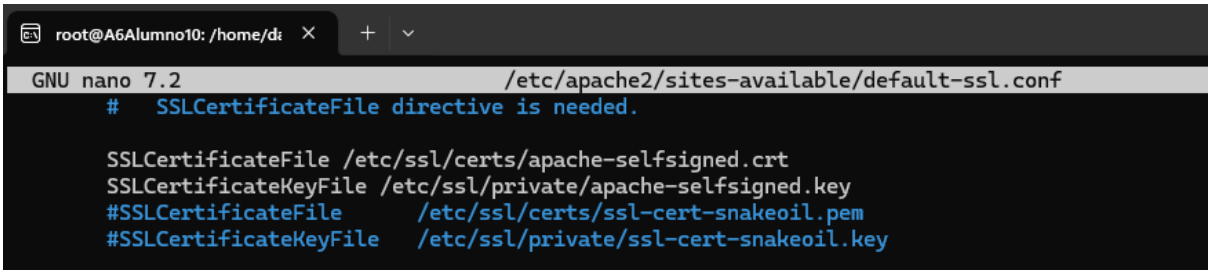
<Imagen 16. Instalación de Cerbot>

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.

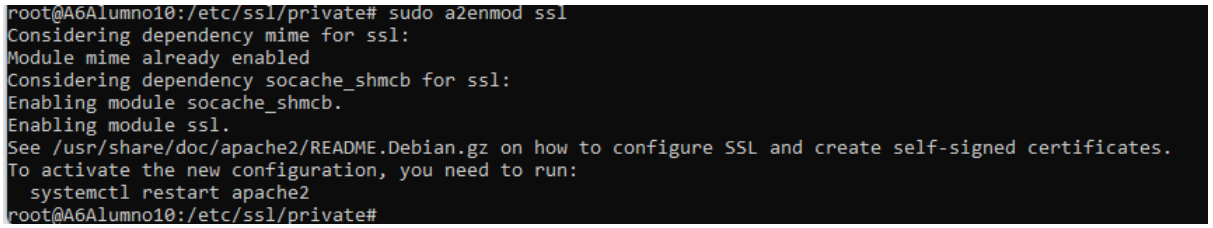


<Imagen 17. Certificado por defecto SSL de apache>



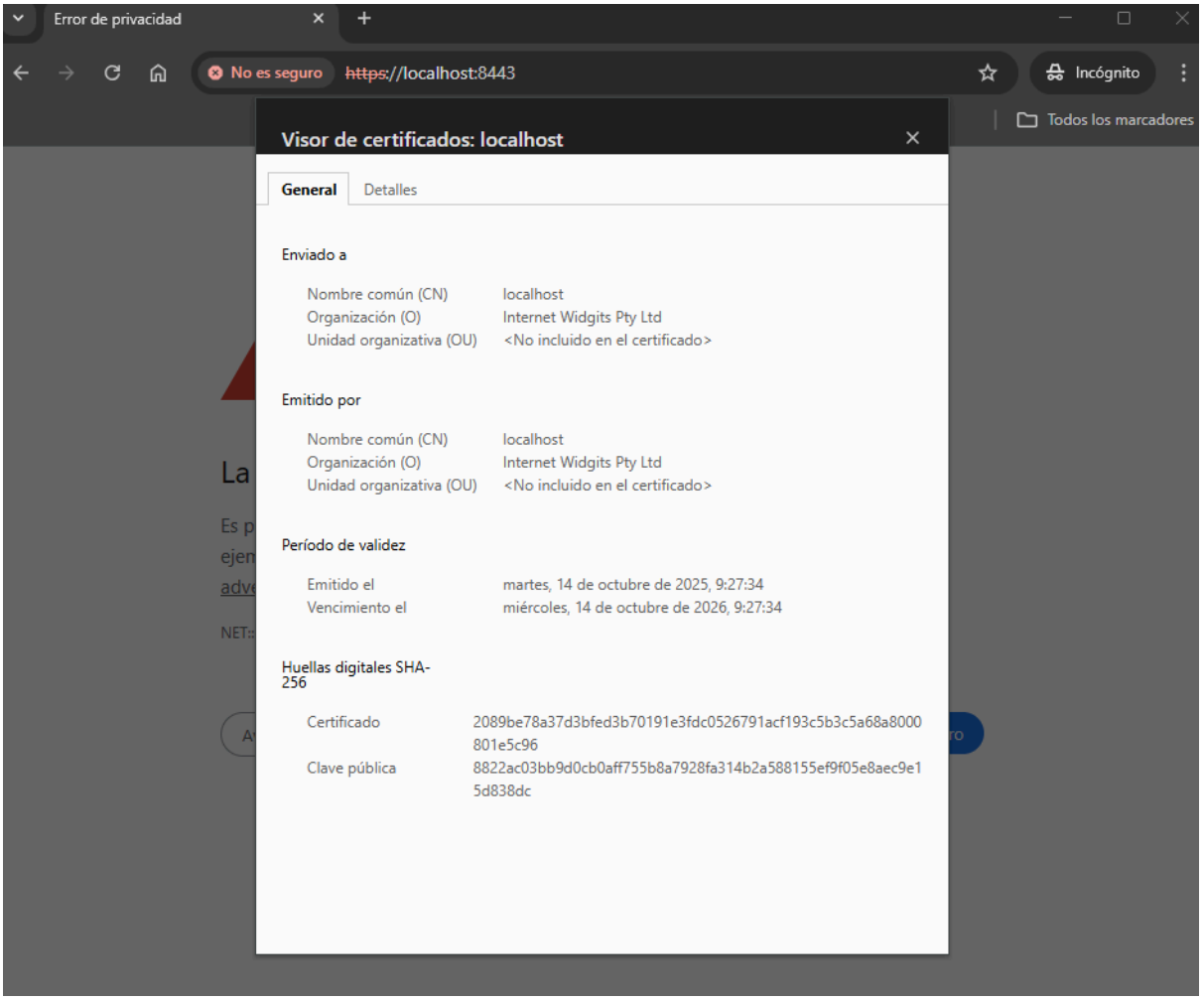
<Imagen 18. Certificado por defecto SSL de apache>

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**



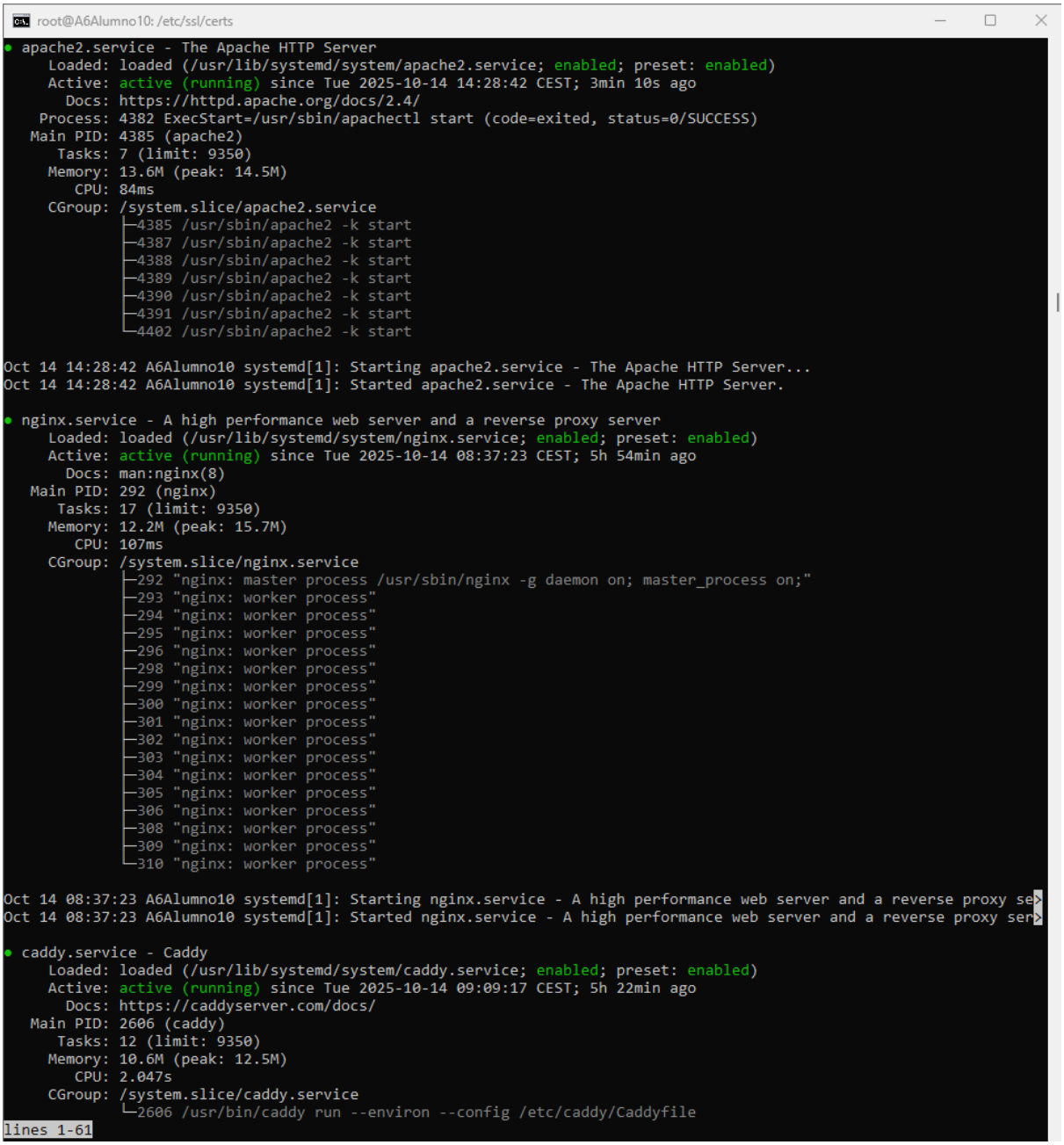
<Imagen 19. Ubicación del archivo index.html>

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



<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>