SEGUNDA PRACTICA

sudo apt update && sudo apt upgrade -y

```
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ sudo apt update && sudo apt upgrade -y
Hit:1 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:3 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
17 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
The following packages will be upgraded:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras libnss-systemd libpam-systemd libsystemd-shared libsystem
  systemd-timesyncd udev
```

sudo apt install apache2 -y

```
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ sudo apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.58-1ubuntu8.8).
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

sudo nano /etc/apache2/ports.conf

cambiamos el listen 80 por 8080

```
# 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>
```

```
 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>
```

Aqui igual

```
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ sudo apt install php libapache2-mod-php -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
php is already the newest version (2:8.3+93ubuntu2).
libapache2-mod-php is already the newest version (2:8.3+93ubuntu2).
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$
```

instalamos php

```
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ sudo systemctl restart apache2
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$
```

reiniciamos apache

```
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ sudo systemctl restart apache2
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ sudo systemctl status apache2
● 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 09:01:26 CEST; 17s ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 2942 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 2945 (apache2)
      Tasks: 6 (limit: 9352)
     Memory: 12.9M (peak: 14.2M)
        CPU: 31ms
     CGroup: /system.slice/apache2.service
             ├─2945 /usr/sbin/apache2 -k start
             ├─2947 /usr/sbin/apache2 -k start
             ├─2948 /usr/sbin/apache2 -k start
             ├─2949 /usr/sbin/apache2 -k start
             ├─2950 /usr/sbin/apache2 -k start
             └─2951 /usr/sbin/apache2 -k start

Oct 14 09:01:26 A6Alumno19 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Oct 14 09:01:26 A6Alumno19 systemd[1]: Started apache2.service - The Apache HTTP Server.
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$
```

verificamos si esta activo

http://localhost:8080/info.php

PHP Version 8.3.6

System	Linux A6Alumno19 6.6.87.2-microsoft-standard-WSL2 #1 SMP PREEMPT_DYNAMIC Thu Jun 5 18:30:46 UTC 2025 x86_64
Build Date	Jul 14 2025 18:30:55
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.3/apache2
Loaded Configuration File	/etc/php/8.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.3/apache2/conf.d
Additional .ini files parsed	/etc/php/8.3/apache2/conf.d/10-opcache.ini, /etc/php/8.3/apache2/conf.d/10-pdo.ini, /etc/php/8.3/apache2/conf.d/20-calendar.ini, /etc/php/8.3/apache2/conf.d/20-ctype.ini, /etc/php/8.3/apache2/conf.d/20-exif.ini, /etc/php/8.3/apache2/conf.d/20-ffi.ini, /etc/php/8.3/apache2/conf.d/20-fileinfo.ini, /etc/php/8.3/apache2/conf.d/20-ftp.ini, /etc/php/8.3/apache2/conf.d/20-gettext.ini, /etc/php/8.3/apache2/conf.d/20-iconv.ini, /etc/php/8.3/apache2/conf.d/20-phar.ini, /etc/php/8.3/apache2/conf.d/20-posix.ini, /etc/php/8.3/apache2/conf.d/20-readline.ini, /etc/php/8.3/apache2/conf.d/20-shmop.ini, /etc/php/8.3/apache2/conf.d/20-sockets.ini, /etc/php/8.3/apache2/conf.d/20-sysvmsg.ini, /etc/php/8.3/apache2/conf.d/20-sysvsem.ini, /etc/php/8.3/apache2/conf.d/20-sysvshm.ini, /etc/php/8.3/apache2/conf.d/20-tokenizer.ini
PHP API	20230831
PHP Extension	20230831
Zend Extension	420230831
Zend Extension Build	API420230831,NTS
PHP Extension Build	API20230831,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
Zend Max Execution Timers	disabled
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v4.3.6, Copyright (c) Zend Technologies with Zend OPcache v8.3.6, Copyright (c), by Zend Technologies

comprobamos que funciona

```
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ sudo apt install certbot python3-certbot-apache openssl -y
Reading package lists... Done
Building dependency tree... Done
```

instalamos certbot y dependencias ssl

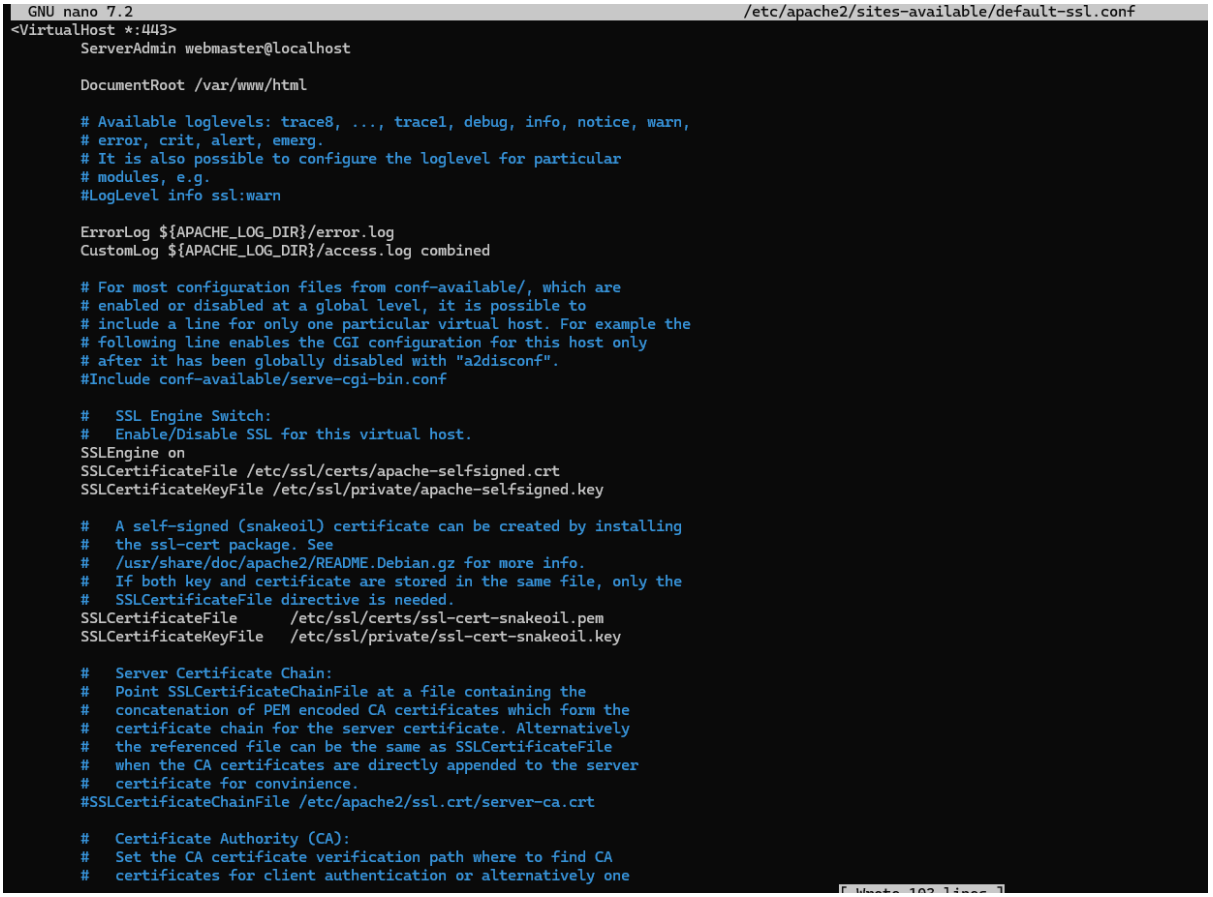
```
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
-keyout /etc/ssl/private/apache-selfsigned.key \
-out /etc/ssl/certs/apache-selfsigned.crt
.++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.....+.++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++
.....+.........++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++*.+....+.....+..........+...........+.+...+.
......+.........+..+............+.+.....+....+...........+.+.........+.........+..+...+...+....+.....+....+.....+......+.+..
++++++++++++++++++++++++++++++
-----
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.
-----
Country Name (2 letter code) [AU]:
```

creamos el certificado y rellenamos los campos

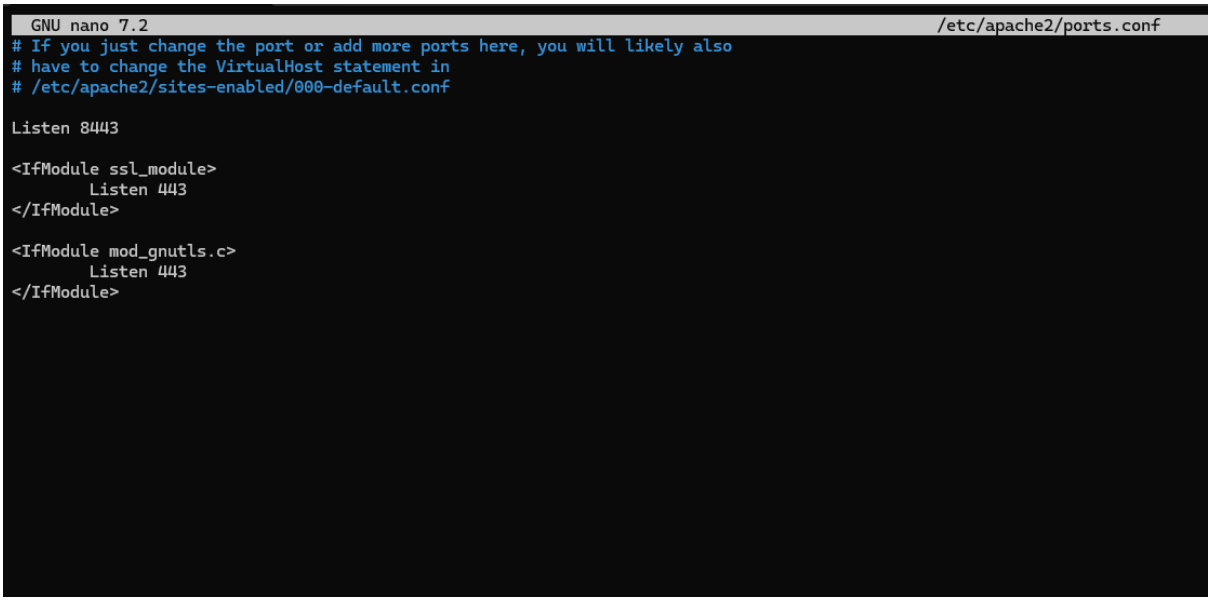
```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:manolo
String too long, must be at most 2 bytes long
Country Name (2 letter code) [AU]:manolin
String too long, must be at most 2 bytes long
Country Name (2 letter code) [AU]:manolito
String too long, must be at most 2 bytes long
Country Name (2 letter code) [AU]:mo
State or Province Name (full name) [Some-State]:Spain
Locality Name (eg, city) []:Spain
Organization Name (eg, company) [Internet Widgits Pty Ltd]:prometeo
Organizational Unit Name (eg, section) []:prometeo
Common Name (e.g. server FQDN or YOUR name) []:manolo
Email Address []:grecurco@gmail.com
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$
```

```
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ 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
```

activamos ssl



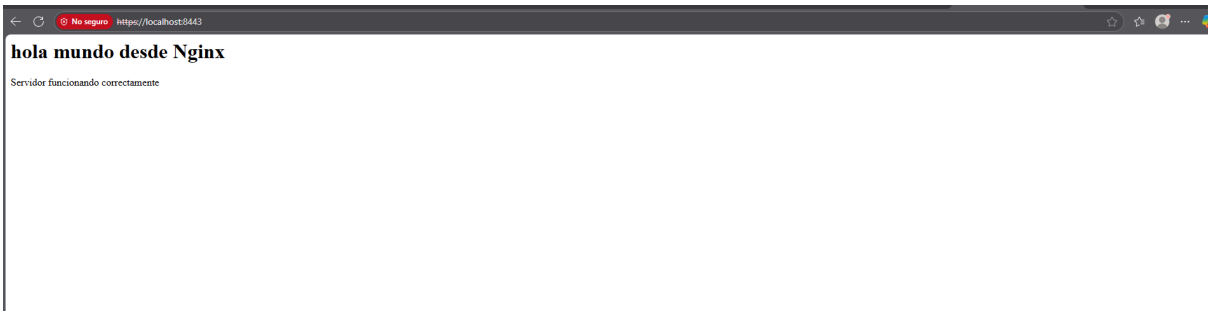
cambiamos el archivo https

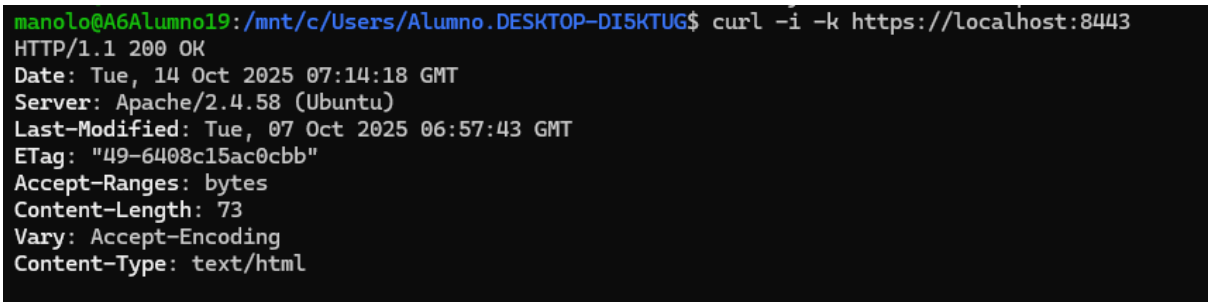


cambiamos el puerto a 8443

activamos el sitio

lo reiniciamos





PARTE 2

```
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ sudo apt install nginx -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.24.0-2ubuntu7.5).
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Instalamos nginx

configuramos el puerto 80 al 8081

```
 GNU nano 7.2                                                                 /etc/nginx/sites-available/
##
# 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 [::]:80 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 _;
```

```
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ echo "<h1>Servidor Nginx</h1><p>Funcionando en puerto 8081</p>" | sudo tee /usr/share/nginx/html/index.html
<h1>Servidor Nginx</h1><p>Funcionando en puerto 8081</p>
```

creamos la pagina html

```
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ sudo systemctl restart nginx
```

reiniciamos

```
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ sudo 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 Tue 2025-10-14 09:18:25 CEST; 22s ago
       Docs: man:nginx(8)
    Process: 3497 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 3499 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 3500 (nginx)
      Tasks: 13 (limit: 9352)
     Memory: 9.9M (peak: 11.0M)
        CPU: 43ms
     CGroup: /system.slice/nginx.service
             ├─3500 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─3501 "nginx: worker process"
             ├─3502 "nginx: worker process"
             ├─3503 "nginx: worker process"
             ├─3504 "nginx: worker process"
             ├─3506 "nginx: worker process"
             ├─3507 "nginx: worker process"
             ├─3508 "nginx: worker process"
             ├─3509 "nginx: worker process"
             ├─3510 "nginx: worker process"
             ├─3511 "nginx: worker process"
             ├─3512 "nginx: worker process"
             └─3513 "nginx: worker process"

Oct 14 09:18:25 A6Alumno19 systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Oct 14 09:18:25 A6Alumno19 systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
```

comprobamos que funciona

```
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ curl http://localhost:8081
<h1>hola mundo desde Nginx</h1><p>Servidor funcionando correctamente</p>
```

recibimos respuesta

PARTE 3

```
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ sudo apt install -y debian-keyring debian-archive-keyring apt-transport-https curl
Reading package lists... Done
```

Instalamos las dependencias

curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | sudo gpg --dearmor -o /usr/share/keyrings/caddy-stable-archive-keyring.gpg

curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/debian.deb.txt' | sudo tee /etc/apt/sources.list.d/caddy-stable.list

```
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | sudo gpg --dearmor -o /usr/share/keyrings/caddy-stable-archive-keyring.gpg

curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/debian.deb.txt' | sudo tee /etc/apt/sources.list.d/caddy-stable.list
# Source: Caddy
# Site: https://github.com/caddyserver/caddy
# Repository: Caddy / stable
# Description: Fast, multi-platform web server with automatic HTTPS


deb [signed-by=/usr/share/keyrings/caddy-stable-archive-keyring.gpg] https://dl.cloudsmith.io/public/caddy/stable/deb/debian any-version main

deb-src [signed-by=/usr/share/keyrings/caddy-stable-archive-keyring.gpg] https://dl.cloudsmith.io/public/caddy/stable/deb/debian any-version main
```

Agregamos el repositorio de caddy

```
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ sudo apt update && sudo apt install caddy -y
Hit:1 https://download.docker.com/linux/ubuntu jammy InRelease
```

instalamos caddy

```
Processing triggers for man-db (2.12.0-4build2) ...
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ sudo mkdir -p /var/www/caddy
echo "# Bienvenido a Caddy" | sudo tee /var/www/caddy/README.md
echo "Este servidor está funcionando correctamente." | sudo tee -a /var/www/caddy/README.md
# Bienvenido a Caddy
Este servidor está funcionando correctamente.
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$
```

creamos los directorios y archivos

```
Este servidor está funcionando correctamente.
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$ curl -o /tmp/test.jpg "https://www.python.org/static/apple-touch-icon-144x144-precomposed.png"
sudo mv /tmp/test.jpg /var/www/caddy/test.jpg
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  7382  100  7382    0     0   193k      0 --:--:-- --:--:-- --:--:--  194k
manolo@A6Alumno19:/mnt/c/Users/Alumno.DESKTOP-DI5KTUG$
```

descargamos una imagen de prueba

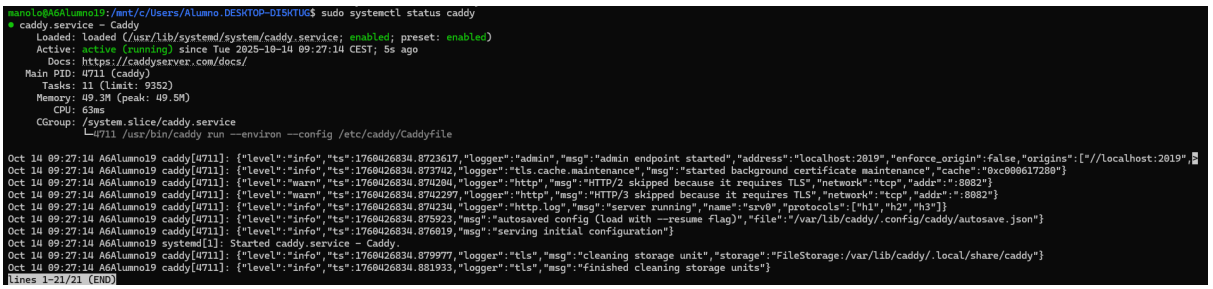
```
 GNU nano 7.2                                                    /etc/caddy/Caddyfile
# The Caddyfile is an easy way to configure your Caddy web server.
#
# Unless the file starts with a global options block, the first
# uncommented line is always the address of your site.
#
# To use your own domain 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.

:80 {
        # Set this path to your site's directory.
        root * /usr/share/caddy

        # Enable the static file server.
        file_server

        # 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
}

# Refer to the Caddy docs for more information:
# https://caddyserver.com/docs/caddyfile
```

configuramos caddyfile

```
  GNU nano 7.2
# 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.

:80 {
        # Set this path to your site's directory.
        root * /usr/share/caddy

        # Enable the static file server.
        file_server

        # 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
}
:8082 {
    root * /var/www/caddy
    file_server browse

    @markdown path *.md
    header @markdown Content-Type text/plain
}

# Refer to the Caddy docs for more information:
# https://caddyserver.com/docs/caddyfile
```

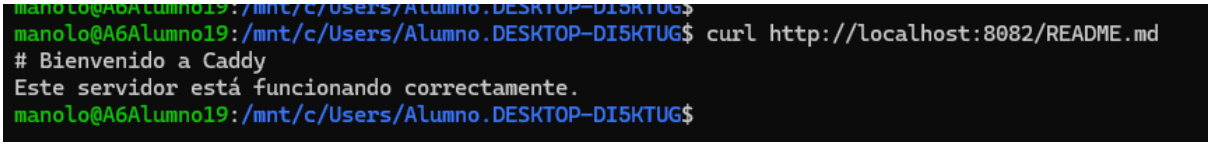
```
:8082 {
    root * /var/www/caddy
    file_server browse

    @markdown path *.md
    header @markdown Content-Type text/plain
}
```

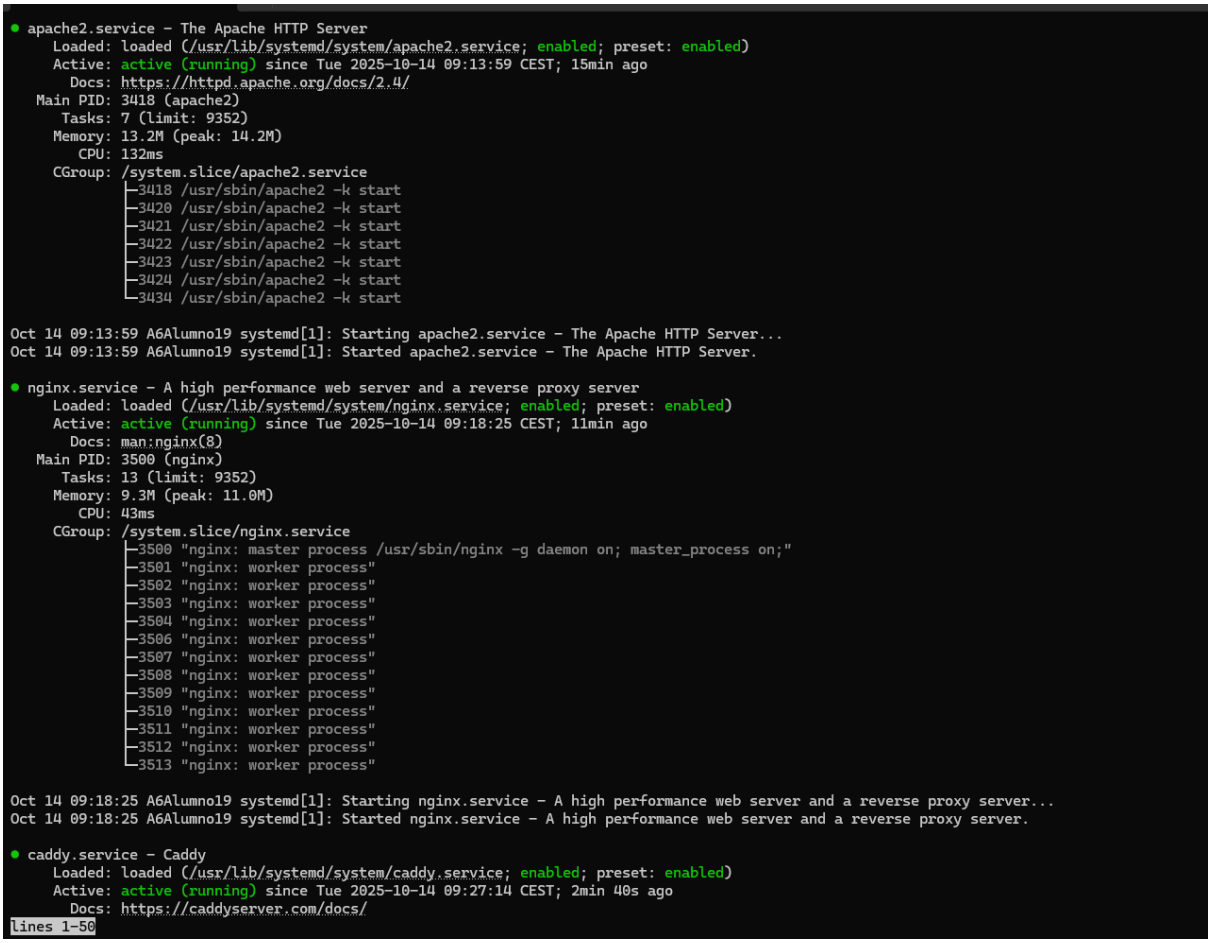
lo dejamos asi



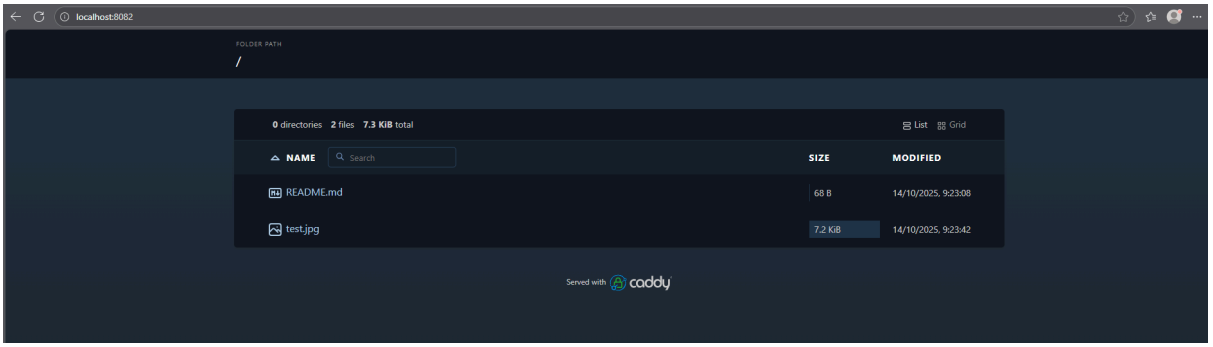
comprobamos que funciona

funciona

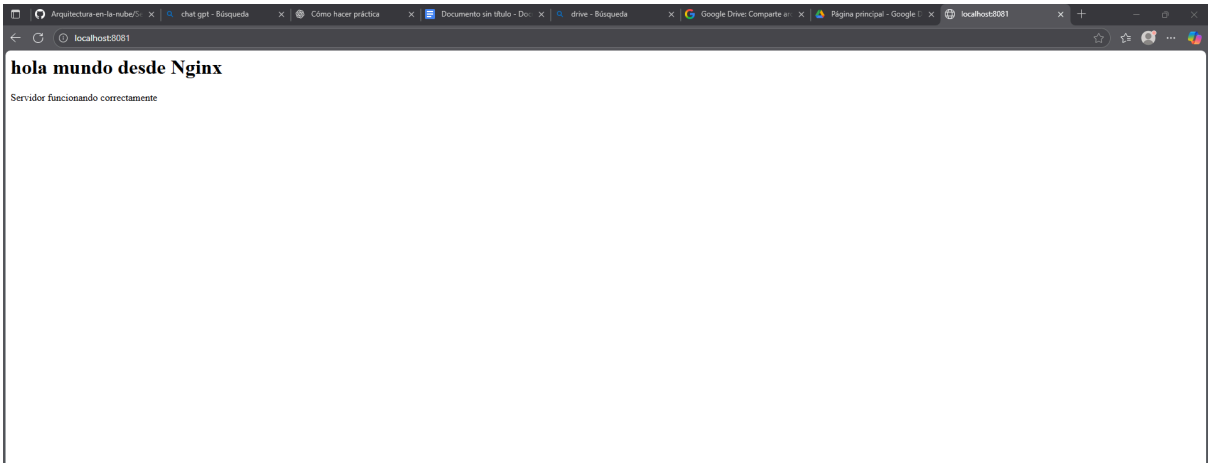




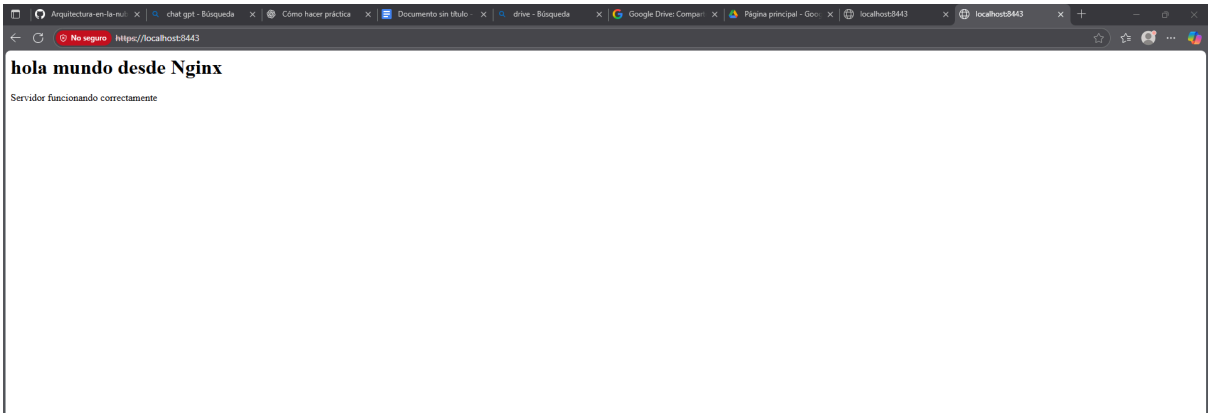
comprobamos que todos los servidores estan corriendo correctamente



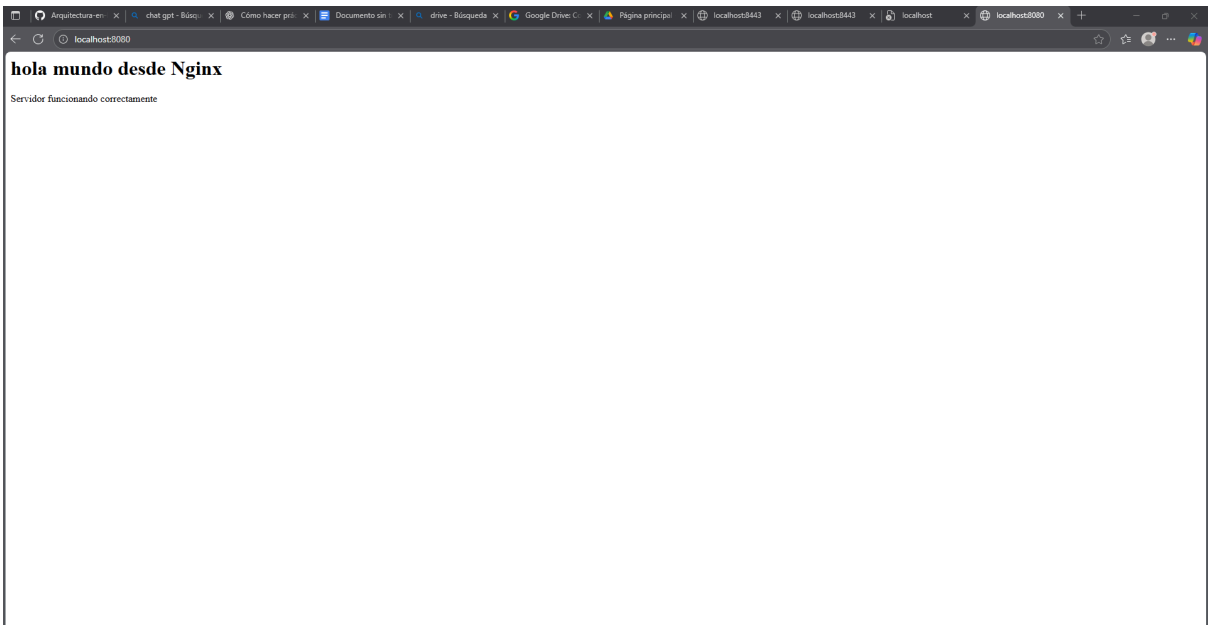
caddy funciona



nginx funciona



apache https funciona



apache con 8080 funciona