COMANDOS

1. Actualizar los paquetes del sistema

```
root@UbuntuServer:/home/vboxuser# apt update && apt upgrade -y
Hit:1 http://es.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:4 http://es.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:5 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1,484 kB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1,201 kB]
Get:7 http://es.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [286 kB]
Get:8 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:9 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [15.3 kB]
Get:10 http://es.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [2,053 kB]
```

2. Instalamos el servidor web Apache

```
root@UbuntuServer:/home/vboxuser# apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libllvm19 linux-headers-6.14.0-29-generic linux-hwe-6.14-headers-6.14.0-29
  linux-hwe-6.14-tools-6.14.0-29 linux-image-6.14.0-29-generic
  linux-modules-6.14.0-29-generic linux-modules-extra-6.14.0-29-generic
  linux-tools-6.14.0-29-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3
  libaprutil1-ldap libaprutil1t64
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3
  libaprutil1-ldap libaprutil1t64
```

3. Instalamos PHP

```
root@UbuntuServer:/home/vboxuser# apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libllvm19 linux-headers-6.14.0-29-generic linux-hwe-6.14-headers-6.14.0-29
  linux-hwe-6.14-tools-6.14.0-29 linux-image-6.14.0-29-generic
  linux-modules-6.14.0-29-generic linux-modules-extra-6.14.0-29-generic
  linux-tools-6.14.0-29-generic
```

4-5.Iniciamos Apache y verificamos el estado de este mismo

```
root@UbuntuServer:/home/vboxuser# service apache2 start
root@UbuntuServer:/home/vboxuser# systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-10-03 07:43:42 UTC; 8min ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 19200 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 19204 (apache2)
      Tasks: 6 (limit: 4603)
     Memory: 10.6M (peak: 11.4M)
        CPU: 88ms
     CGroup: /system.slice/apache2.service
             ├─19204 /usr/sbin/apache2 -k start
             ├─19208 /usr/sbin/apache2 -k start
             ├─19209 /usr/sbin/apache2 -k start
             ├─19210 /usr/sbin/apache2 -k start
             ├─19211 /usr/sbin/apache2 -k start
             └─19212 /usr/sbin/apache2 -k start

Oct 03 07:43:42 UbuntuServer systemd[1]: Starting apache2.service - The Apache HTTP Se>
Oct 03 07:43:42 UbuntuServer apachectl[19203]: AH00558: apache2: Could not reliably de>
Oct 03 07:43:42 UbuntuServer systemd[1]: Started apache2.service - The Apache HTTP Ser>
lines 1-20/20 (END)
```

6. Creamos un archivo que muestre información de PHP

```
root@UbuntuServer:/home/vboxuser# echo "<?php phpinfo(); ?>" | tee /var/www/html/info.p
hp
<?php phpinfo(); ?>
```

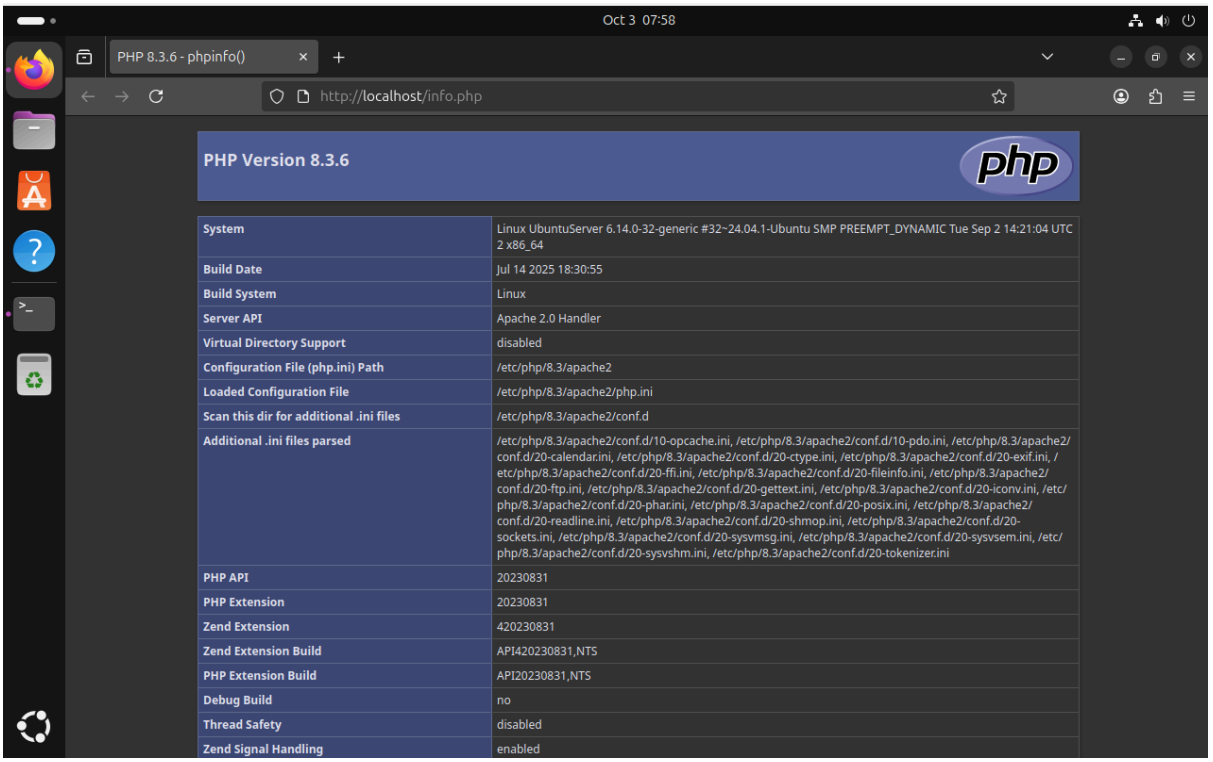
7. Probamos el archivo PHP desde el terminal

```
root@UbuntuServer:/home/vboxuser# curl http://localhost/info.php

Caution: You are using the Snap version of curl.
Due to Snap's sandbox nature, this version has some limitations.
For example, it may not be able to access hidden folders in your home directory
or other restricted areas of the os.
This means you may encounter errors when using snap curl to download files.
For those case, you might want to use the native curl package.
For details, see: https://github.com/boukendesho/curl-snap/issues/1

To stop seeing this message, run the following command:
curl.snap-acked
```

8. Corroboramos que este todo correcto dirigiéndonos al enlace



COMANDO NGINX + HTML

1. Instalamos el servidor web NGINX

```
root@UbuntuServer:/home/vboxuser# apt install nginx -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libllvm19 linux-headers-6.14.0-29-generic linux-hwe-6.14-headers-6.14.0-29
  linux-hwe-6.14-tools-6.14.0-29 linux-image-6.14.0-29-generic
  linux-modules-6.14.0-29-generic linux-modules-extra-6.14.0-29-generic
  linux-tools-6.14.0-29-generic
```

2-3. Iniciamos y verificamos el estado de NGINX

```
root@UbuntuServer:/home/vboxuser# service nginx start
root@UbuntuServer:/home/vboxuser# 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 2025-10-03 08:03:02 UTC; 28s ago
       Docs: man:nginx(8)
    Process: 21251 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on;>
    Process: 21252 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=ex>
   Main PID: 21254 (nginx)
      Tasks: 5 (limit: 4603)
     Memory: 3.7M (peak: 4.3M)
        CPU: 14ms
     CGroup: /system.slice/nginx.service
             ├─21254 "nginx: master process /usr/sbin/nginx -g daemon on; master_proce>
             ├─21255 "nginx: worker process"
             ├─21256 "nginx: worker process"
             ├─21257 "nginx: worker process"
             └─21258 "nginx: worker process"

Oct 03 08:03:02 UbuntuServer systemd[1]: Starting nginx.service - A high performance w>
Oct 03 08:03:02 UbuntuServer systemd[1]: Started nginx.service - A high performance we>
lines 1-19/19 (END)
```

4. Creamos un HTML

```
root@UbuntuServer:/home/vboxuser# echo "<h1>Hola Mundo desde Nginx</h1><p>Servidor func
ionando correctamente</p>" | tee /var/www/html/index.html
<h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correctamente</p>
```

5. Verificamos en navegador

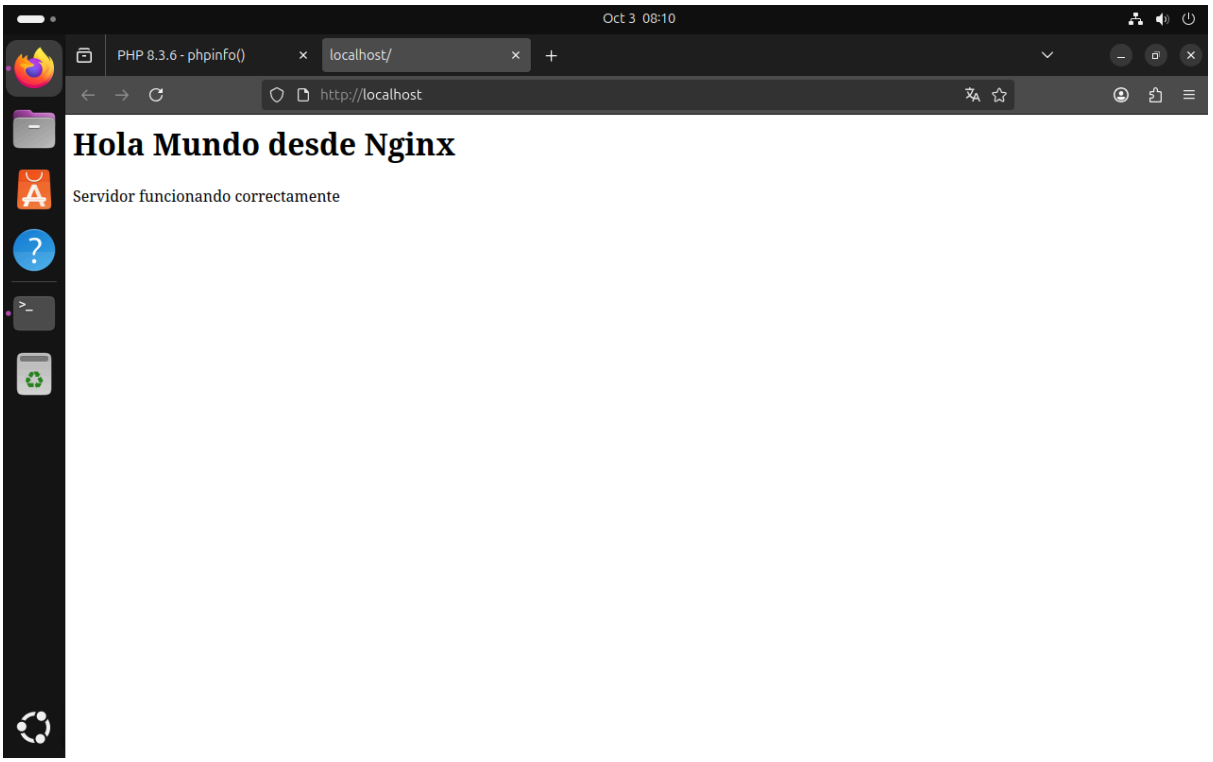
```
root@UbuntuServer:/home/vboxuser# curl http://localhost

Caution: You are using the Snap version of curl.
Due to Snap's sandbox nature, this version has some limitations.
For example, it may not be able to access hidden folders in your home directory
or other restricted areas of the os.
This means you may encounter errors when using snap curl to download files.
For those case, you might want to use the native curl package.
For details, see: https://github.com/boukendesho/curl-snap/issues/1

To stop seeing this message, run the following command:
curl.snap-acked

<h1>Hola Mundo desde Nginx</h1><p>Servidor funcionando correctamente</p>
```

6. Corroboramos que este todo correcto dirigiéndonos al enlace



7. Ver IP de WSL

```
root@UbuntuServer:/home/vboxuser# ip addr show enp0s3 | grep inet
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
    inet6 fd17:625c:f037:2:25df:4815:bc7d:9349/64 scope global temporary dynamic
    inet6 fd17:625c:f037:2:a00:27ff:fe09:66b9/64 scope global dynamic mngtmpaddr
    inet6 fe80::a00:27ff:fe09:66b9/64 scope link
```

INSTALACION Y CONFIGURACION DE CADDY

1. Instalamos las herramientas necesarias para añadir repositorios externos

```
root@UbuntuServer:/home/vboxuser# apt install -y debian-keyring debian-archive-keyring apt-t
ransport-https curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  apt-transport-https curl debian-archive-keyring debian-keyring
```

2. Añadimos un repositorio oficial de caddy a nuestro sistema

```
root@UbuntuServer:/home/vboxuser# curl -1sLf "https://dl.cloudsmith.io/public/caddy/stable/g
pg.key" | gpg --dearmor -o /usr/share/keyrings/caddy-stable-archive-keyring.gpg
```

3. Actualizamos los paquetes e instalamos caddy

```
root@UbuntuServer:/home/vboxuser# curl -1sLf "https://dl.cloudsmith.io/public/caddy/stable/g
pg.key" | gpg --dearmor -o /usr/share/keyrings/caddy-stable-archive-keyring.gpg
root@UbuntuServer:/home/vboxuser# apt update && apt install caddy -y
Hit:1 http://es.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://es.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
20 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

4. Creamos un directorio para caddy

```
root@UbuntuServer:/home/vboxuser# mkdir -p /var/www/caddy
```

5. Creamos un archivo Markdown con contenido de ejemplo

```
root@UbuntuServer:/home/vboxuser# mkdir -p /var/www/caddy
root@UbuntuServer:/home/vboxuser# echo "# Bienvenido a Caddy" | tee /var/www/caddy/README.md
# Bienvenido a Caddy
root@UbuntuServer:/home/vboxuser# echo "" | tee -a /var/www/caddy/README.md
```

```
root@UbuntuServer:/home/vboxuser# echo "Este servidor esta funcionando correctamente." | tee
 -a /var/www/caddy/README.md
Este servidor esta funcionando correctamente.
root@UbuntuServer:/home/vboxuser# echo "" | tee -a /var/www/caddy/README.md
```

```
root@UbuntuServer:/home/vboxuser# echo "## Caracteristicas" | tee -a /var/www/caddy/README.m
d
## Caracteristicas
root@UbuntuServer:/home/vboxuser# echo "- Servidor moderno" | tee -a /var/www/caddy/README.m
d
- Servidor moderno
```

```
root@UbuntuServer:/home/vboxuser# echo "- HTTPS automatico" | tee -a /var/www/caddy/README.m
d
- HTTPS automatico
root@UbuntuServer:/home/vboxuser# echo "- Facil configuracion" | tee -a /var/www/caddy/READM
E.md
- Facil configuracion
```

6. Creamos una imagen de prueba

```
root@UbuntuServer:/home/vboxuser# 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  57560      0 --:--:-- --:--:-- --:--:-- 57671
root@UbuntuServer:/home/vboxuser# mv /tmp/test-image.jpg /var/www/caddy/test.jpg
```

7. Creamos Caddyfile personalizado

```
  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

        # Another common task is to set up a reverse proxy:
        # reverse_proxy localhost:8080
        @markdown path *.md
        header @markdown Content-Type text/plain
        # Or serve a PHP site through php-fpm:
        # php_fastcgi localhost:9000
}
```

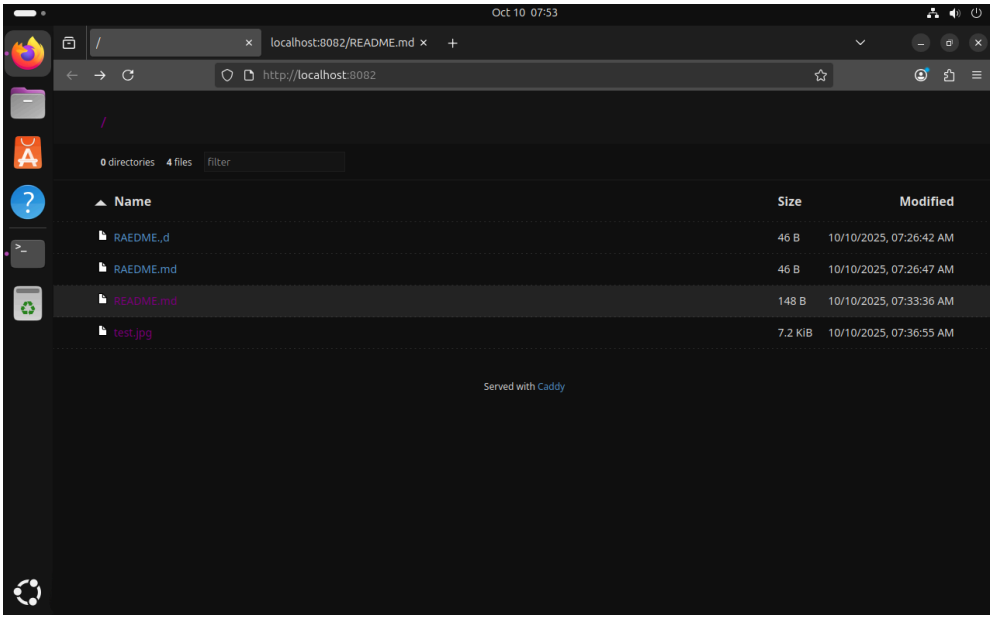
8. Reiniciamos caddy

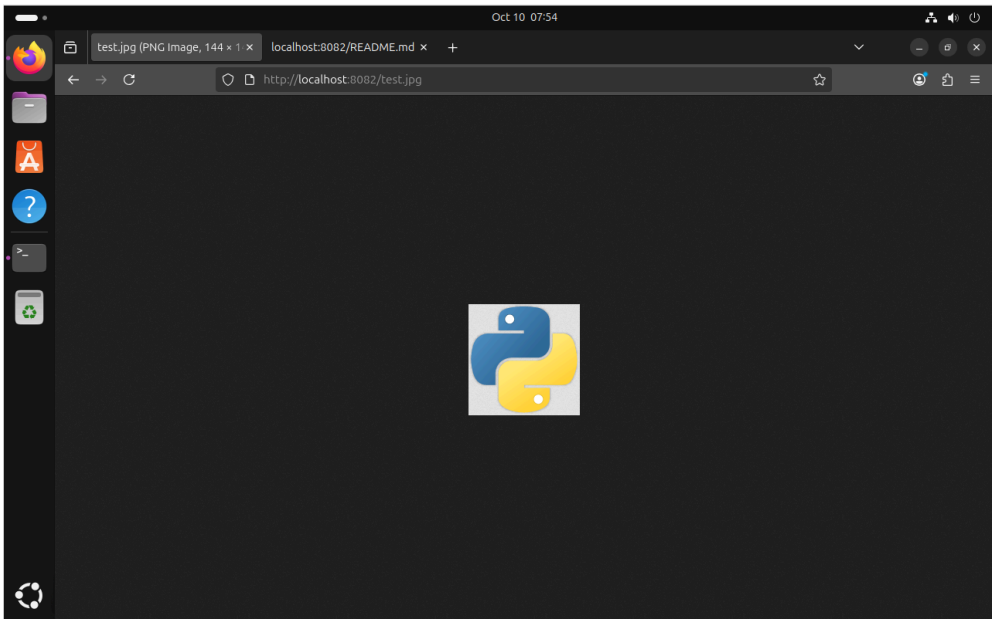
```
root@UbuntuServer:/home/vboxuser# systemctl restart caddy
```

9. Verificamos estado del caddy

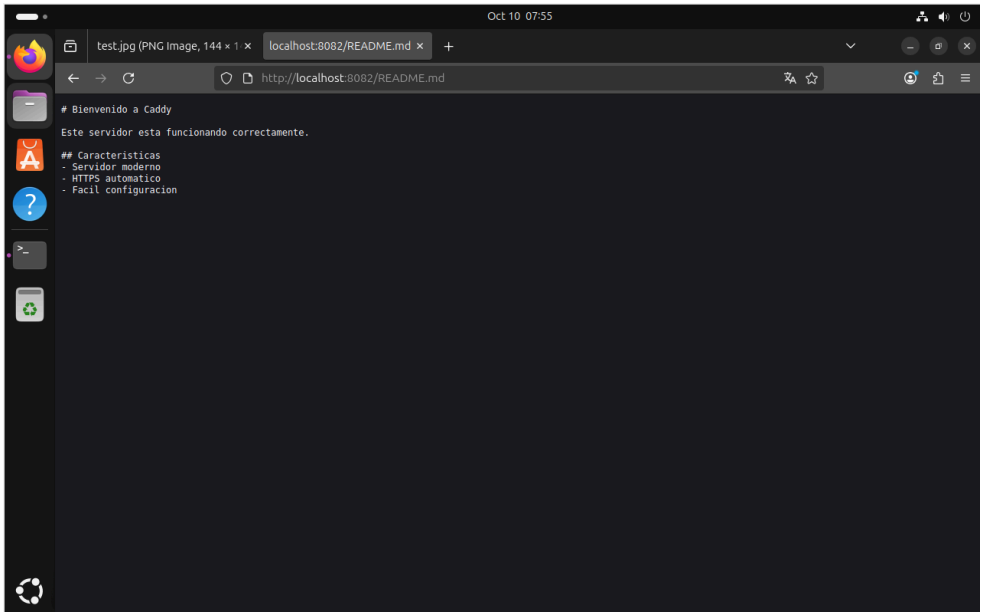
```
root@UbuntuServer:/home/vboxuser# systemctl status caddy
● caddy.service - Caddy
     Loaded: loaded (/usr/lib/systemd/system/caddy.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-10-10 07:46:52 UTC; 2min 7s ago
       Docs: https://caddyserver.com/docs/
   Main PID: 5192 (caddy)
      Tasks: 9 (limit: 4603)
     Memory: 7.1M (peak: 8.3M)
        CPU: 50ms
     CGroup: /system.slice/caddy.service
             └─5192 /usr/bin/caddy run --environ --config /etc/caddy/Caddyfile
```

10. Probamos caddy desde terminal





11. Probamos archivo Markdown



CONFIGURACIÓN DE HTTPS CON CERTBOT EN APACHE

1. Instalamos Certbot y el plugin de Apache

```
root@UbuntuServer:/home/vboxuser# apt install certbot python3-certbot-apache -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  augeas-lenses libaugeas0 python3-acme python3-augeas python3-certbot
  python3-configargparse python3-icu python3-josepy python3-openssl python3-parsedatetime
  python3-rfc3339
```

2. Verificar dominio o usar localhost

```
root@UbuntuServer:/home/vboxuser# 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]:ES
State or Province Name (full name) [Some-State]:Madrid
Locality Name (eg, city) []:Madrid
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:localhost
Email Address []:camilosc.0718@gmail.com
```

3. Habilitamos módulo SSL en Apache

```
root@UbuntuServer:/home/vboxuser# 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 ce
rtificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@UbuntuServer:/home/vboxuser# systemctl restart apache2
```

4. Habilitamos configuración SSL para apache

```
root@UbuntuServer:/home/vboxuser# nano /etc/apache2/sites-available/default-ssl.conf
```

```
SSLEngine on

#   A self-signed (snakeoil) certificate can be created by instal
#   the ssl-cert package. See
#   /usr/share/doc/apache2/README.Debian.gz for more info.
#   If both key and certificate are stored in the same file, only
#   SSLCertificateFile directive is needed.
SSLCertificateFile      /etc/ssl/certs/apache-selfsigned.crt
SSLCertificateKeyFile   /etc/ssl/private/apache-selfsigned.key
```

5. Cambiamos puerto SSL

```
root@UbuntuServer:/home/vboxuser# nano /etc/apache2/ports.conf
```

```
  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
Listen 8443

<IfModule ssl_module>
        Listen 443
</IfModule>

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

6. Modificamos VirtualHost SSL

```
root@UbuntuServer:/home/vboxuser# nano /etc/apache2/sites-available/default-ssl.conf
```

```
  GNU nano 7.2              /etc/apache2/sites-available/default-ssl.conf
<VirtualHost *:8443>
```

7. Habilitamos sitio SSL

```
root@UbuntuServer:/home/vboxuser# a2ensite default-ssl.conf
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@UbuntuServer:/home/vboxuser# system reload apache2
```

8. Verificamos HTTPS

```
Date: Fri, 10 Oct 2025 08:52:24 GMT
Server: Apache/2.4.58 (Ubuntu)
Last-Modified: Fri, 03 Oct 2025 08:07:43 GMT
ETag: "49-6403c98a87e43"
Accept-Ranges: bytes
Content-Length: 73
Vary: Accept-Encoding
Content-Type: text/html
```

Hola Mundo desde Nginx

Servidor funcionando correctamente

VERIFICACIÓN FINAL DE LOS TRES SERVIDORES

1. Verificar que todos los servidores están activos

```
root@UbuntuServer:/home/vboxuser# systemctl status apache2 nginx caddy
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-10-10 08:52:03 UTC; 8min ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 12974 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 12977 (apache2)
      Tasks: 7 (limit: 4603)
     Memory: 13.0M (peak: 13.2M)
        CPU: 145ms
     CGroup: /system.slice/apache2.service
             ├─12977 /usr/sbin/apache2 -k start
             ├─12979 /usr/sbin/apache2 -k start
             ├─12980 /usr/sbin/apache2 -k start
             ├─12981 /usr/sbin/apache2 -k start
             ├─12982 /usr/sbin/apache2 -k start
             ├─12983 /usr/sbin/apache2 -k start
             └─13980 /usr/sbin/apache2 -k start

Oct 10 08:52:02 UbuntuServer systemd[1]: Starting apache2.service - The Apache HTTP Server...
Oct 10 08:52:03 UbuntuServer apachectl[12976]: AH00558: apache2: Could not reliably determin>
Oct 10 08:52:03 UbuntuServer 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 Fri 2025-10-10 08:59:58 UTC; 14s ago
       Docs: man:nginx(8)
    Process: 14030 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code>
    Process: 14032 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, >
   Main PID: 14033 (nginx)
      Tasks: 5 (limit: 4603)
     Memory: 3.7M (peak: 4.3M)
```

2. Verificar puertos en uso

```
root@UbuntuServer:/home/vboxuser# netstat -tulpn | grep -E "8080|8081|8082|8443"
tcp6       0      0 :::8443                 :::*                    LISTEN      12977/apache2

tcp6       0      0 :::8082                 :::*                    LISTEN      5192/caddy

tcp6       0      0 :::8080                 :::*                    LISTEN      12977/apache2
```