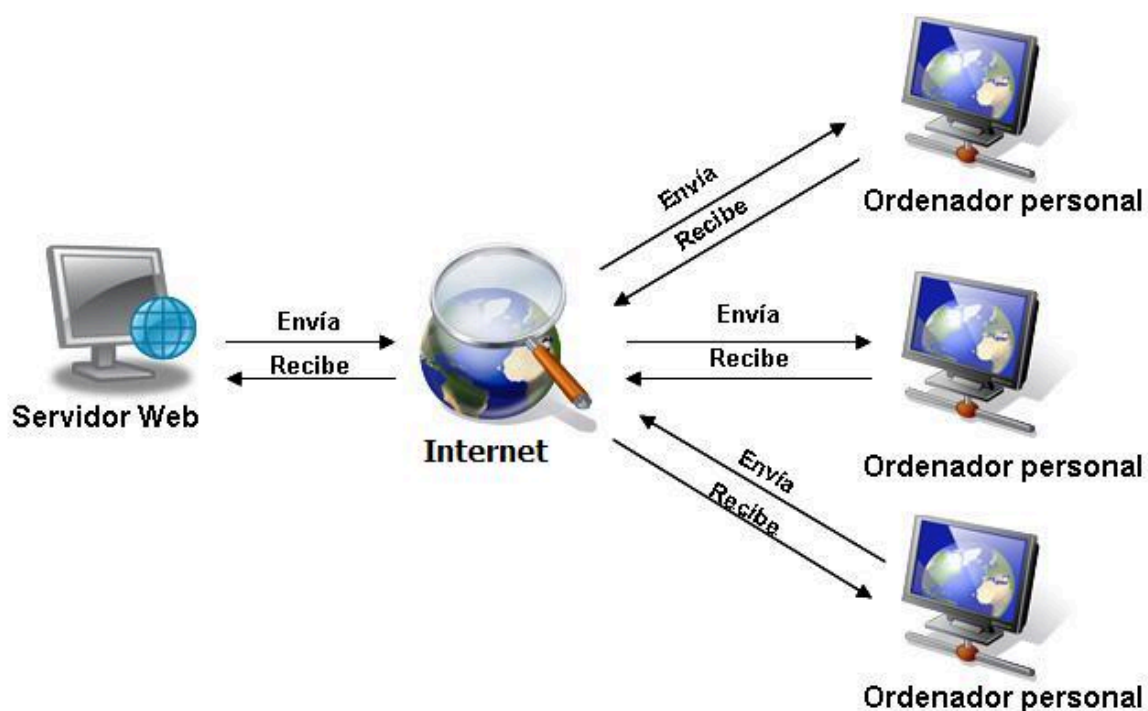


Configuración Avanzada de Servidores WEB y HTTPS



Autor: Diestefano Jiamperi

Asignatura: Arquitectura en la Nube

Tutor: Rubén

Fecha de Entrega: 30/09/2025

ÍNDICE

Servidor Apache con PHP.....	3
Servidor Nginx con HTML.....	8
Servidor Caddy con Archivos Especiales.....	10
Verificación Simultánea de los Tres Servidores.....	14

Servidor Apache con PHP

1. Empezaremos la práctica actualizando el servicio, para mejorar el sistema a las últimas versiones, usaremos `"sudo apt update && sudo apt upgrade -y"`.

```
root@dies-VirtualBox:/home/dies# sudo apt update && sudo apt upgrade -y
Des:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Obj:2 http://es.archive.ubuntu.com/ubuntu noble InRelease
Des:3 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Des:4 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1.2
7 kB]
Des:5 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [2
,5 kB]
Des:6 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadat
```

2. Instalamos Apache2, para ello ponemos `"sudo apt install apache2 -y"`.

```
root@dies-VirtualBox:/home/dies# sudo apt install apache2 -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es nec
esario.
  libllvm19
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
```

3. Configuraremos Apache en el puerto 8080, `"sudo nano /etc/apache2/ports.conf"`. En la configuración cambiaremos **listen 80** por **listen 8080**.

```
root@dies-VirtualBox:/home/dies# sudo 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
# 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>
```

4. Modificaremos el archivo de configuración del sitio por defecto para que también use el nuevo puerto, con `"sudo nano /etc/apache2/sites-available/000-default.conf".` Una vez dentro cambiamos **VirtualHost *:80** por **VirtualHost *:8080**.

```
root@dies-VirtualBox:/home/dies# sudo nano /etc/apache2/sites-available/000-default.conf
```

```
GNU nano 7.2 /etc/apache2/sites-available/000-default.conf *
<VirtualHost *:8080>
```

5. Ahora instalamos PHP, con el siguiente comando `"sudo apt install php libapache2-mod-php -y".`

```
root@dies-VirtualBox:/home/dies# sudo apt install php libapache2-mod-php -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
libllvm19
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
libapache2-mod-php8.3 libsodium23 php-common php8.3 php8.3-cli php8.3-common
php8.3-opcache php8.3-readline
Paquetes sugeridos:
```

6. Reiniciamos PHP, así aplicaremos todos los cambios de configuración, usaremos `"sudo systemctl restart apache2".`

```
root@dies-VirtualBox:/home/dies# sudo systemctl restart apache2
```

7. Verificamos su funcionamiento con `"sudo systemctl status apache2"` y observaremos si funciona todo en el puerto 8080 usando `"sudo netstat -tulpn | grep 8080".`

```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-10-15 19:53:23 CEST; 1min 18s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 21400 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
    Main PID: 21403 (apache2)
       Tasks: 6 (limit: 4603)
      Memory: 10.6M (peak: 11.1M)
         CPU: 46ms
```

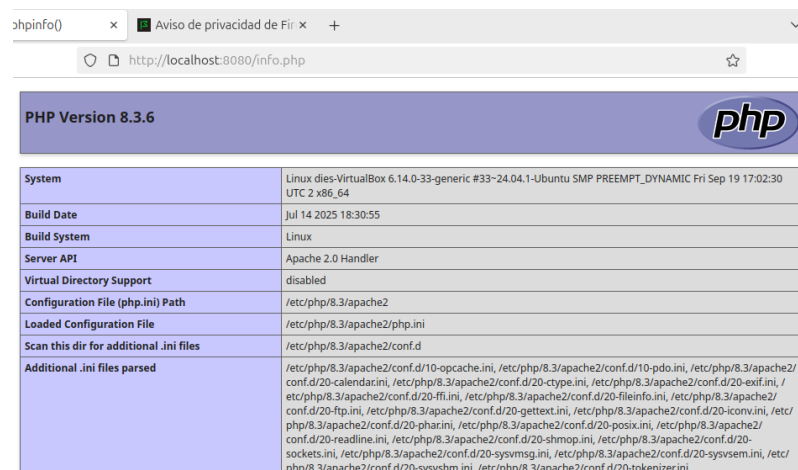
```
root@dies-VirtualBox:/home/dies# sudo netstat -tulpn | grep 8080
tcp6      0      0 :::8080          :::*              ESCUCHAR      21403/a
pache2
```

8. Continuamos creando un archivo `info.php` `"echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php".`

```
root@dies-VirtualBox:/home/dies# echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php
```

9. Accederemos a Apache desde la terminal con `curl http://localhost:8080/info.php`

```
root@dies-VirtualBox:/home/dies# curl http://localhost:8080/info.php
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional-1.0.dtd">
<xhtml xmlns="http://www.w3.org/1999/xhtml"><head>
<style type="text/css">
body {background-color: #fff; color: #222; font-family: sans-serif;}
pre {margin: 0; font-family: monospace;}
a:link {color: #009; text-decoration: none; background-color: #fff;}
a:hover {text-decoration: underline;}
table {border-collapse: collapse; border: 0; width: 934px; box-shadow: 1px 2px 3px rgba(0, 0, 0, 0.2);}
tr td {text-align: center;}
tr td {text-align: left;}
tr th {text-align: center !important;}
tr td, tr th {border: 1px solid #666; font-size: 75%; vertical-align: baseline; padding: 4px; text-align: left;}
```



10. A continuación configuraremos HTTPS con un certificado autofirmado, para ello empezaremos poniendo "`sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt`".

[illegible]

11. Apache necesita que su módulo SSL esté activado para poder manejar conexiones HTTPS, por eso mismo pondremos "`sudo a2enmod ssl`".

```
root@dies-VirtualBox:/home/dies# 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-sig
ned certificates.
```

12. Tenemos que decirle a Apache donde debe encontrar los archivos del certificado que hemos creado, para ello abriremos el archivo de configuración SSL por defecto con "`sudo nano /etc/apache2/sites-available/default-ssl.conf`". Una vez dentro de este archivo verificamos que todo está correctamente y cambiaremos la línea **VirtualHost _default_:443** por **VirtualHost *:8443**. También tenemos que cambiar las certificaciones que viene por defecto por las que hemos creado antes

```
oot@dies-VirtualBox:/home/dies# sudo nano /etc/apache2/sites-available/default-ssl.conf
```

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

    DocumentRoot /var/www/html
```

Las certificaciones son las siguientes

```
SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
```

13. Debemos decirle a Apache que tiene que escuchar el puerto 8443 para ello usaremos "`sudo nano /etc/apache2/ports.conf`" y pondremos debajo de **listen 8080**, **listen 8443**.

```
oot@dies-VirtualBox:/home/dies# sudo 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 like
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 8080
Listen 8443
```

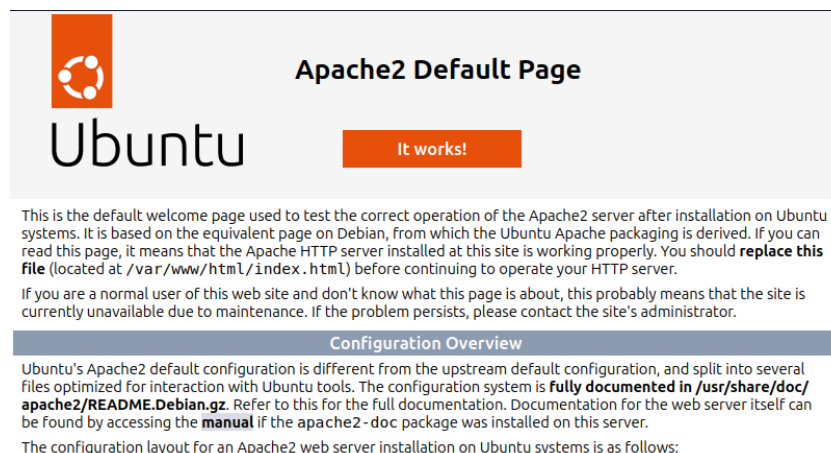
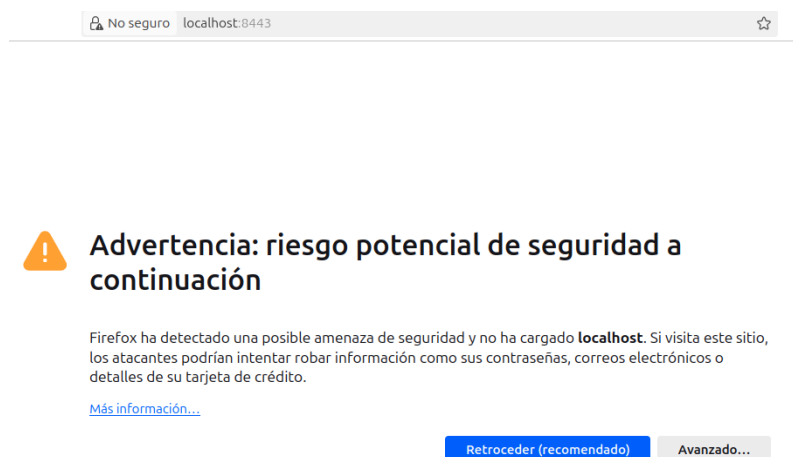
14. Casi terminando tenemos que activar la configuración SSL, con el comando “`sudo a2ensite default-ssl.conf`”.

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

15. Reiniciamos Apache para que se apliquen todos los cambios “`sudo systemctl restart apache2`”.

```
root@dies-VirtualBox:/home/dies# sudo systemctl restart apache2
```

16. Para terminar esta parte, accederemos a “`https://localhost:8443`”, y veremos la pantalla con el candado.



Servidor Nginx con HTML

1. Empezaremos instalando Nginx, "`sudo apt install nginx -y`".

```
root@dies-VirtualBox:/home/dies# sudo apt install nginx -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no
liblvm2
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
nginx-common
Paquetes sugeridos:
fcgiwrap nginx-doc
Se instalarán los siguientes paquetes NUEVOS:
nginx nginx-common
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 2 no actualizado
Se necesita descargar 564 kB de archivos.
Se utilizarán 1.596 kB de espacio de disco adicional después de esta opera
Des:1 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx-c
4.0-2ubuntu7.5 [43,4 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx a
ubuntu7.5 [520 kB]
```

2. Procederemos a configurar Nginx en el puerto 8081, con el comando "`sudo nano /etc/nginx/sites-available/default`". En la configuración cambiaremos ***listen 80*** por ***listen 8081***.

```
root@dies-VirtualBox:/home/dies# sudo nano /etc/nginx/sites-available/default
```

```
server {
    listen 8081 default_server;
    listen [::]:8081 default_server;
```

3. Crearemos una página HTML personalizada, usando "`echo "<h1>Servidor Nginx</h1><p> Funcionando en puerto 8081</p>" | sudo tee /usr/share/nginx/html/index.html`".

```
root@dies-VirtualBox:/home/dies# echo "<h1>Servidor Nginx</h1><p>Funcionando en puerto
8081</p>" | sudo tee /usr/share/nginx/html/index.html
```

4. Reiniciamos Nginx, así aplicaremos todos los cambios de configuración, usaremos "`sudo systemctl restart nginx`".

```
root@dies-VirtualBox:/home/dies# sudo systemctl restart nginx
```


5. Verificamos su estado y miraremos si funciona en el puerto 8081, “`sudo systemctl status nginx`”.

```
root@dies-VirtualBox:/home/dies# 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 Wed 2025-10-15 21:20:05 CEST; 26s ago
     Docs: man:nginx(8)
   Process: 24736 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on;
   Process: 24741 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=ex>
 Main PID: 24742 (nginx)
    Tasks: 5 (limit: 4603)
   Memory: 3.8M (peak: 4.2M)
      CPU: 19ms
```

6. Accederemos a “`http://localhost:8081`” y verificaremos que Nginx muestra bien el HTML. También podemos acceder desde la terminal con “`curl http://localhost:8081`”.

```
root@dies-VirtualBox:/home/dies# curl http://localhost:8081
<h1>Servidor Nginx</h1><p>Funcionando en puerto 8081</p>
```



Servidor Caddy con Archivos Especiales

1. Comenzaremos instalando las dependencias necesarias, con “`sudo apt install -y debian-keyring debian-archive-keyring apt-transport-https curl`” posteriormente añadiremos los repositorios externos.

```
root@dies-VirtualBox:/home/dies# sudo apt install -y debian-keyring debian-archive-keyring apt-transport-https curl
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
curl ya está en su versión más reciente (8.5.0-2ubuntu10.6).
El paquete indicado a continuación se instaló de forma automática y ya no es necesario que lo instale.
*
  libllvm19
Utilice «sudo apt autoremove» para eliminarlo.
```

2. Agregaremos el repositorio oficial de Caddy mediante el comando “`curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | sudo gpg --dearmor -o /usr/share/keyrings/caddy-stable-archive-keyring.gpg`” este añade la clave de seguridad y “`curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/debian.deb.txt' | sudo tee /etc/apt/sources.list.d/caddy-stable.list`” este añade la fuente del software.

```
root@dies-VirtualBox:/home/dies# curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | sudo gpg --dearmor -o /usr/share/keyrings/caddy-stable-archive-keyring.gpg
```

```
root@dies-VirtualBox:/home/dies# 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
```

3. Actualizaremos la lista de paquetes e instalaremos Caddy, “`sudo apt update && sudo apt install caddy -y`”.

```
root@dies-VirtualBox:/home/dies# sudo apt update && sudo apt install caddy -y
Des:1 https://dl.cloudsmith.io/public/caddy/stable/deb/debian any-version InRelease [14,8 kB]
Obj:2 http://es.archive.ubuntu.com/ubuntu noble InRelease
Des:3 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Des:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Obj:5 http://es.archive.ubuntu.com/ubuntu noble-backports InRelease
Des:6 http://es.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1.489 kB]
Des:7 https://dl.cloudsmith.io/public/caddy/stable/deb/debian any-version/main amd64 Packages [4.329 B]
```

4. Ahora crearemos un directorio específico para los archivos de Caddy, `sudo mkdir -p /var/www/caddy`.

```
root@dies-VirtualBox:/home/dies# sudo mkdir -p /var/www/caddy
```

5. Continuamos creando un archivo de Markdown de prueba, `echo "# Bienvenido a Caddy" | sudo tee /var/www/caddy/README.md` `echo "" | sudo tee -a /var/www/caddy/README.md` `echo "Este servidor está funcionando correctamente." | sudo tee -a /var/www/caddy/README.md` `echo "" | sudo tee -a /var/www/caddy/README.md` `echo "## Características" | sudo tee -a /var/www/caddy/README.md` `echo "- Servidor moderno" | sudo tee -a /var/www/caddy/README.md` `echo "- HTTPS automático" | sudo tee -a /var/www/caddy/README.md` `echo "- Fácil configuración" | sudo tee -a /var/www/caddy/README.md`.

```
/README.md
echo "" | sudo tee -a /var/www/caddy/README.md
echo "Este servidor está funcionando correctamente." | sudo tee -a
/var/www/caddy/README.md
echo "" | sudo tee -a /var/www/caddy/README.md
echo "## Características" | sudo tee -a /var/www/caddy/README.md
echo "- Servidor moderno" | sudo tee -a /var/www/caddy/README.md
echo "- HTTPS automático" | sudo tee -a /var/www/caddy/README.md
echo "- Fácil configuración" | sudo tee -a /var/www/caddy/README.md
# Bienvenido a Caddy

Este servidor está funcionando correctamente.
bash: /var/www/caddy/README.md: Permiso denegado

## Características
- Servidor moderno
- HTTPS automático
- Fácil configuración
```

6. Descargamos una imagen de prueba para verificar que Caddy sirve archivos estáticos, usaremos `curl -o /tmp/test-image.jpg "https://www.python.org/static/apple-touch-icon-144x144-precomposed.png"` y `sudo mv /tmp/test-image.jpg /var/www/caddy/test.jpg`.

```
root@dies-VirtualBox:/home/dies# curl -o /tmp/test-image.jpg "https://www.python.org/s
tatic/apple-touch-icon-144x144-precomposed.png"
sudo mv /tmp/test-image.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  156k      0 --:--:-- --:--:-- --:--:-- 160k
```

7. Procederemos crear Caddyfile personalizado, con el comando "`sudo nano /etc/caddy/Caddyfile`" abriremos el archivo de configuración y escribiremos lo siguiente "`:8082 { root * /var/www/caddy file_server browse @markdown path *.md header @markdown Content-Type text/plain }`".

```
root@dies-VirtualBox:/home/dies# sudo nano /etc/caddy/Caddyfile
```

```
GNU nano 7.2 /etc/caddy/Caddyfile *
:8082 {
  root * /var/www/caddy
  file_server browse
  @markdown path *.md
  header @markdown Content-Type text/plain
}
```

8. Reiniciamos Caddy para aplicar la configuración "`sudo systemctl restart caddy`".

```
root@dies-VirtualBox:/home/dies# sudo systemctl restart caddy
```

9. Revisamos su estado "`sudo systemctl status caddy`".

```
root@dies-VirtualBox:/home/dies# sudo systemctl status caddy
● caddy.service - Caddy
   Loaded: loaded (/usr/lib/systemd/system/caddy.service; enabled; preset: enabled)
   Active: active (running) since Wed 2025-10-15 23:03:40 CEST; 23s ago
     Docs: https://caddyserver.com/docs/
   Main PID: 27886 (caddy)
    Tasks: 9 (limit: 4603)
   Memory: 11.7M (peak: 12.2M)
      CPU: 181ms
   CGroup: /system.slice/caddy.service
           └─27886 /usr/bin/caddy run --environ --config /etc/caddy/Caddyfile
```

10. Ahora empezaremos con las verificaciones, empezamos capturando el listado de archivos "`http://localhost:8082`".



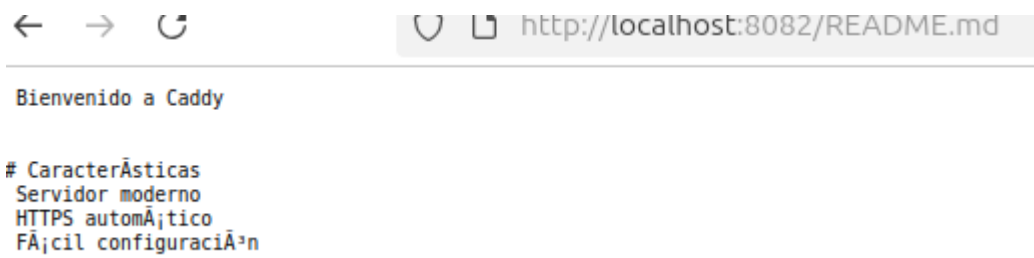
http://localhost:8082

0 directories 2 files 7.3 KiB total

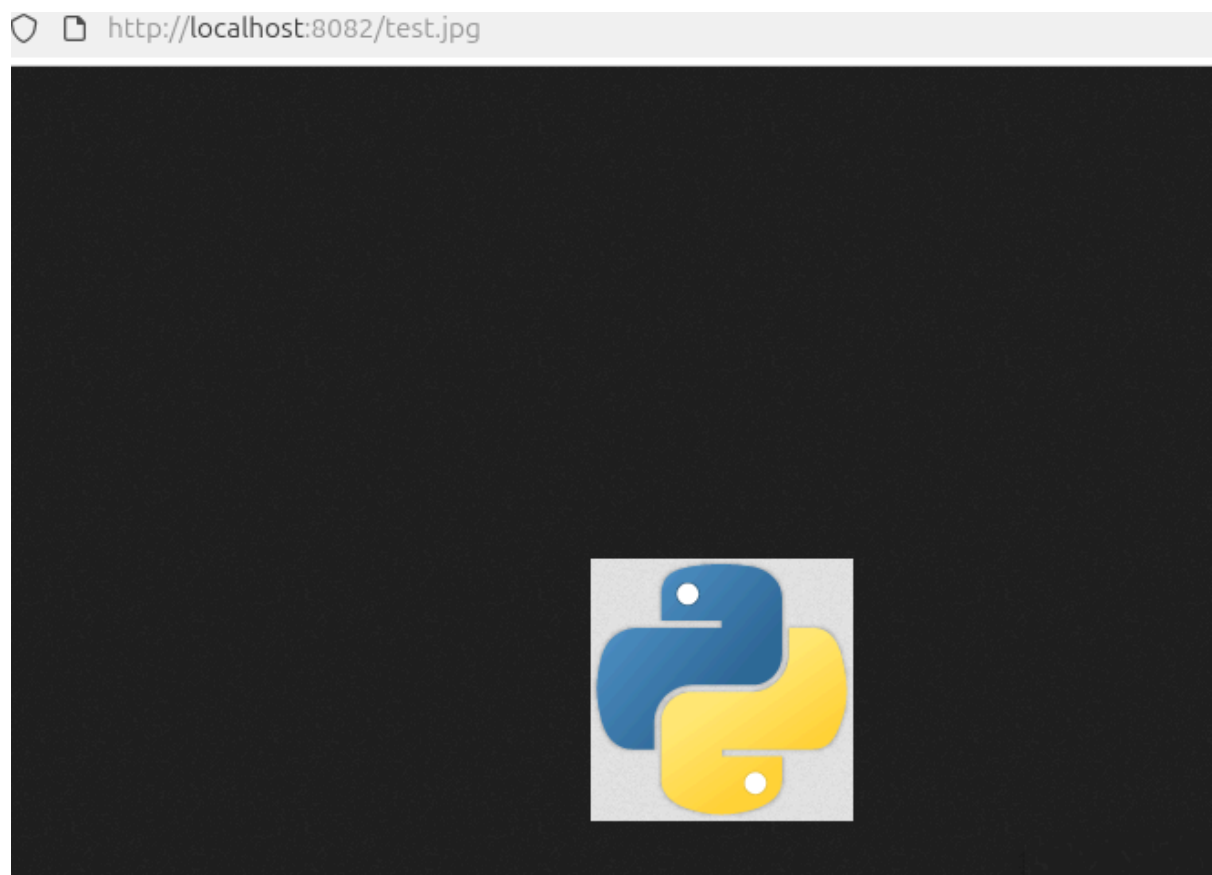
NAME	SIZE	MODIFIED
README.md	106 B	15/10/2025, 22:50:01
test.jpg	7.2 KiB	15/10/2025, 22:53:59

Served with caddy

11. Continuaremos capturando el contenido, "<http://localhost:8082/README.md>".



12. Finalizamos capturando la imagen, "<http://localhost:8082/test.jpg>".



Verificación Simultánea de los Tres Servidores

1. Ejecutaremos un comando para que muestre el estado de los tres servicios simultáneamente, con "`sudo systemctl status apache2 nginx caddy`".

```
root@dies-VirtualBox:/home/dies# sudo 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 Thu 2025-10-16 08:50:17 CEST; 13min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 1244 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 1289 (apache2)
    Tasks: 6 (limit: 4601)
   Memory: 18.9M (peak: 19.4M)
      CPU: 321ms
```

```
● 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 Thu 2025-10-16 08:50:19 CEST; 13min ago
     Docs: man:nginx(8)
   Process: 1454 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; >
```


```
● caddy.service - Caddy
   Loaded: loaded (/usr/lib/systemd/system/caddy.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-10-16 08:50:20 CEST; 13min ago
     Docs: https://caddyserver.com/docs/
  Main PID: 1451 (caddy)
    Tasks: 9 (limit: 4601)
   Memory: 48.5M (peak: 49.4M)
      CPU: 416ms
   CGroup: /system.slice/caddy.service
```

2. Seguiremos con el comando `netplan`, para mostrar los cuatro puertos en uso, usaremos "`sudo netstat -tulpn | grep -E '8080|8081|8082|8443'`".

```
root@dies-VirtualBox:/home/dies# sudo netstat -tulpn | grep -E '8080|8081|8082|8443'
tcp        0      0 0.0.0.0:8081          0.0.0.0:*            ESCUCHAR    1466/ng
inx: master
tcp6       0      0 :::8080             :::*                  ESCUCHAR    1289/ap
ache2
tcp6       0      0 :::8081             :::*                  ESCUCHAR    1466/ng
inx: master
tcp6       0      0 :::8082             :::*                  ESCUCHAR    1451/ca
ddy
tcp6       0      0 :::8443             :::*                  ESCUCHAR    1289/ap
ache2
```

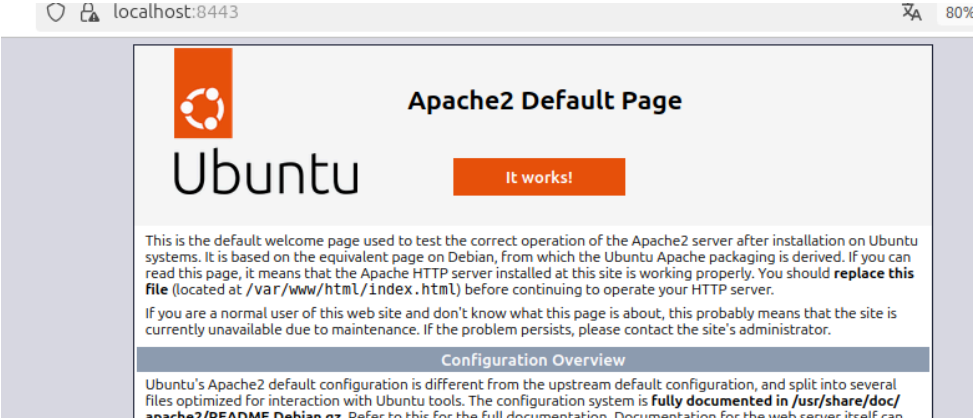
3. Abriremos las cuatro pestañas del navegador simultáneamente, Apache HTTP, Apache HTTPS, Nginx y Caddy.

- **Apache HTTP:** <http://localhost:8080/info.php>



PHP Version 8.3.6	
System	Linux dies-VirtualBox 6.14.0-33-generic #33~24.04.1-Ubuntu SMP PREEMPT_DYNAMIC Fri Sep 19 17:02:30 UTC 2 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

- **Apache HTTPS:** <https://localhost:8443>



Apache2 Default Page

Ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in** `/usr/share/doc/apache2/README.Debian.gz`. Refer to this for the full documentation. Documentation for the web server itself can

- **Nginx:** <http://localhost:8081>



Servidor Nginx

Funcionando en puerto 8081

- **Caddy:** <http://localhost:8082>

http://localhost:8082

80%

FOLDER PATH

/

0 directories 2 files 7.3 KiB total

ListGrid

NAME	SIZE	MODIFIED
 README.md	106 B	15/10/2025, 22:50:01
 test.jpg	7.2 KiB	15/10/2025, 22:53:59

PHP 8.3.6 - phpinfo()

×

Apache2 Ubuntu Defaul

×

localhost:8081/

×

/

×

