

TEMARIO: ARQUITECTURA EN LA NUBE - CONFIGURACIÓN AVANZADA DE SERVIDORES WEB Y HTTPS CONCEPTOS BÁSICOS PARA LA CLASE

Las Herramientas que Vamos a Usar

Apache (*httpd*):

Es el servidor web más veterano y popular del mundo. Como un chef experimentado, puede hacer de todo y es muy configurable. Lleva décadas funcionando en millones de sitios web.

Nginx:

Es el servidor web moderno y eficiente. Como un camarero rápido, puede atender muchas peticiones a la vez sin cansarse. Es perfecto para sitios con mucho tráfico.

Caddy:

Es el servidor web más nuevo y fácil de usar. Su superpoder es que configura **HTTPS** automáticamente, algo que con otros servidores requiere esfuerzo manual. Es como tener un mayordomo que lo hace todo por ti.

Certbot:

Es una herramienta que obtiene certificados **SSL/TLS** gratuitos de Let's Encrypt. Estos certificados son como el candado que ves en tu navegador cuando una página es segura (**HTTPS**).

¿Qué es HTTPS y por qué es importante?

HTTP es como enviar postales: cualquiera puede leer lo que pones.

HTTPS es como enviar cartas en sobres cerrados con sello de lacre: nadie puede leer el contenido ni falsificarlo.

Cuando ves el candado en tu navegador, significa que la comunicación está cifrada y es segura.

¿Por qué hacer esta práctica?

En el mundo real, las empresas no usan un solo servidor web. A veces tienen varios funcionando al mismo tiempo para diferentes propósitos:

- Uno para la página principal
- Otro para aplicaciones internas
- Otro como proxy para distribuir la carga

Además, **TODOS los sitios profesionales usan HTTPS**. Aprender a configurarlo es esencial para cualquier profesional de IT.

Al final de esta práctica comprenderás cómo funcionan los servidores web profesionales, cómo pueden coexistir en la misma máquina y cómo proteger las comunicaciones con **HTTPS**.

COMANDOS Y PROCEDIMIENTOS

PARTE 1: INSTALACIÓN Y CONFIGURACIÓN DE APACHE

1. Actualizar el sistema

Comando:

```
sudo apt update && sudo apt upgrade -y
```

```
root@Ubuntuprepu:/home/apolouser# sudo apt update && sudo apt upgrade -y
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:2 http://es.archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1,213 kB]
Get:5 http://es.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:6 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1,491 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [215 kB]
```

2. Instalar Apache2

Comando:

```
sudo apt install apache2 -y
```

```
root@Ubuntuprepu:/home/apolouser# sudo apt install apache2 -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:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3
  libaprutil1-ldap libaprutil1t64
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
```

Descripción: Instala el servidor web Apache en tu sistema.

3. Configurar Apache en puerto 8080

Comando:

```
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 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>

[ Wrote 13 lines ]
```

Descripción: Abre el archivo de configuración de puertos. Cambia Listen 80 por Listen 8080.

4. Modificar el VirtualHost

Comando:

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

```
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
```

Descripción: Cambia <VirtualHost *:80> por <VirtualHost *:8080>.

5. Instalar PHP

Comando:

```
sudo apt install php libapache2-mod-php -y
```

```
root@Ubuntuprepu:/home/apolouser# sudo apt install php libapache2-mod-php -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:
  libapache2-mod-php8.3 libsodium23 php-common php8.3 php8.3-cli php8.3-common
  php8.3-opcache php8.3-readline
Suggested packages:
  php-pear
```

Descripción: Instala PHP y su módulo para funcionar con Apache.

6 y 7. Reiniciar Apache y comprobar si está funcionando en el puerto 8080

Comando:

```
sudo systemctl restart apache2
```

```
root@Ubuntuprepu:/home/apolouser# sudo systemctl restart apache2
root@Ubuntuprepu:/home/apolouser# sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: >
   Active: active (running) since Fri 2025-10-10 07:11:57 UTC; 11s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 11320 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/S>
  Main PID: 11324 (apache2)
    Tasks: 6 (limit: 4602)
   Memory: 10.6M (peak: 10.8M)
      CPU: 34ms
   CGroup: /system.slice/apache2.service
           └─11324 /usr/sbin/apache2 -k start
             └─11326 /usr/sbin/apache2 -k start
               └─11327 /usr/sbin/apache2 -k start
                 └─11328 /usr/sbin/apache2 -k start
                   └─11329 /usr/sbin/apache2 -k start
                     └─11330 /usr/sbin/apache2 -k start

Oct 10 07:11:57 Ubuntuprepu systemd[1]: Starting apache2.service - The Apache H>
Oct 10 07:11:57 Ubuntuprepu apachectl[11323]: AH00558: apache2: Could not relia>
Oct 10 07:11:57 Ubuntuprepu systemd[1]: Started apache2.service - The Apache HT>
lines 1-20/20 (END)...skipping...
```

Descripción: Reinicia Apache para aplicar los cambios.

8. Crear archivo PHP de prueba

Comando:

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

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

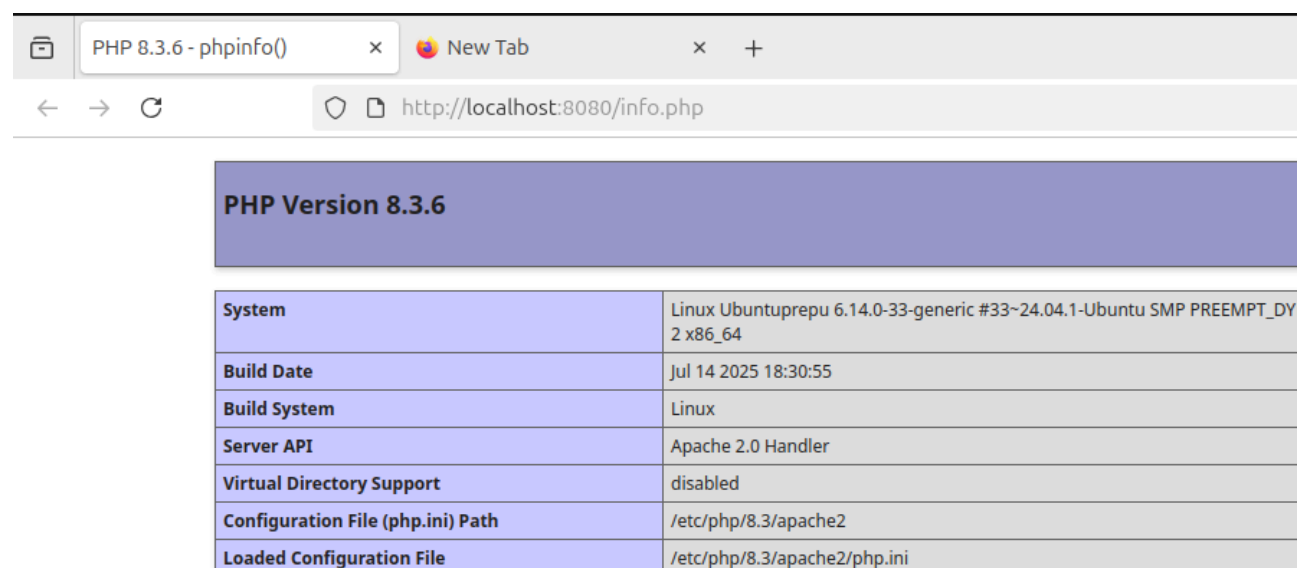
Descripción: Crea un archivo que muestra información del PHP instalado.

9. Probar Apache desde terminal

Comando:

```
curl http://localhost:8080/info.php
```

```
root@Ubuntuprepu:/home/apolouser# curl http://localhost:8080/info.php
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">
<html 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);}
```



PHP Version 8.3.6	
System	Linux Ubuntuprepu 6.14.0-33-generic #33~24.04.1-Ubuntu SMP PREEMPT_DYNAMIC 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

Descripción: Verifica que Apache sirve correctamente el contenido PHP.

PARTE 2: INSTALACIÓN Y CONFIGURACIÓN DE NGINX

1. Instalar Nginx

Comando:

```
sudo apt install nginx -y
```

```
root@Ubuntuprepu:/home/apolouser# sudo apt install nginx -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:
  nginx-common
Suggested packages:
  fcgiwrap nginx-doc
```

Descripción: Instala el servidor web Nginx en tu sistema.

2. Configurar Nginx en puerto 8081

Comando:

```
sudo nano /etc/nginx/sites-available/default
```

```
GNU nano 7.2 /etc/nginx/sites-available/default *
# 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;
```

Descripción: Abre la configuración por defecto. Cambia `listen 80` por `listen 8081`.

3. Crear página HTML personalizada

Comando:

```
echo "<h1>Servidor Nginx</h1><p>Funcionando en puerto 8081</p>" | sudo tee
/usr/share/nginx/html/index.html
```

```
root@Ubuntuprepu:/home/apolouser# sudo nano /etc/nginx/sites-available/default
root@Ubuntuprepu:/home/apolouser# 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>
```

Descripción: Crea una página HTML identificable para Nginx.

PARTE 3: INSTALACIÓN Y CONFIGURACIÓN DE CADDY

1. Instalar dependencias necesarias

Comando:

```
sudo apt install -y debian-keyring debian-archive-keyring apt-transport-https curl
```

```
root@Ubuntuprepu:/home/apolouser# sudo apt install -y debian-keyring debian-arch
ive-keyring apt-transport-https curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
curl is already the newest version (8.5.0-2ubuntu10.6).
The following package was automatically installed and is no longer required:
  libllvm19
Use 'sudo apt autoremove' to remove it.
```

Descripción: Instala herramientas necesarias para añadir repositorios externos.

2. Agregar repositorio de Caddy

Comando:

```
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
```

```
root@Ubuntuprepu:/home/apolouser# curl -1sLf 'https://dl.cloudsmith.io/public/ca
ddy/stable/gpg.key' | sudo gpg --dearmor -o /usr/share/keyrings/caddy-stable-arc
hive-keyring.gpg
File '/usr/share/keyrings/caddy-stable-archive-keyring.gpg' exists. Overwrite? (
y/N) y
root@Ubuntuprepu:/home/apolouser# curl -1sLf 'https://dl.cloudsmith.io/public/ca
ddy/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
```

Descripción: Añade el repositorio oficial de Caddy a tu sistema.

3. Actualizar e instalar Caddy

Comando:

```
sudo apt update && sudo apt install caddy -y
```

```
root@Ubuntuprepu:/home/apolouser# sudo apt update && sudo apt install caddy -y
Get:1 https://dl.cloudsmith.io/public/caddy/stable/deb/debian any-version InRele
ase [14.8 kB]
Hit:2 http://es.archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:4 http://es.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:5 http://es.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:6 https://dl.cloudsmith.io/public/caddy/stable/deb/debian any-version/main a
md64 Packages [4,329 B]
```

Descripción: Actualiza la lista de paquetes e instala Caddy.

4. Crear directorio para Caddy

Comando:

```
sudo mkdir -p /var/www/caddy
```

```
root@Ubutuprepu:/home/apolouser# sudo mkdir -p /var/www/caddy
```

Descripción: Crea un directorio específico para los archivos de Caddy.

5. Crear archivo Markdown de prueba

Comando:

```
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
```

```
GNU nano 7.2 prueba.md
echo "# Bienvenidoecho "#Bienvenido a Caddy" | sudo tee /var/www/caddy/README>
echo "" | sudo tee -a /var/www/caddy/README.md
echo "Este servidor está funcionando correctamente." | sudo tee -a /var/www/c>
echo "" | sudo tee -a /var/www/caddy/README.md
echo "## Características" | sudo tee -a /var/www/caddy/README.md echo "- Serv>
echo "- HTTPS automático" | sudo tee -a /var/www/caddy/README.md echo "- Fáci>
```

Descripción: Crea un archivo Markdown con contenido de ejemplo.

6. Crear imagen de prueba (*cuidado wsl hay que hacer ajustes*)

Comando:

```
curl -o /tmp/test-image.jpg "https://www.python.org/static/apple-touch icon-144x144-precomposed.png"
```

```
sudo mv /tmp/test-image.jpg /var/www/caddy/test.jpg
```

```
root@Ubutuprepu:/home/apolouser# curl -o /tmp/test-image.jpg "https://www.pyt
hon.org/static/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  Curren
t
                                 Dload  Upload  Total  Spent    Left  Speed
100  146  100  146    0    0   891      0  --:--:-- --:--:-- --:--:--   895
```

Descripción: Descarga una imagen de prueba para verificar que Caddy sirve archivos estáticos.

7. Crear Caddyfile personalizado

Comando:

```
sudo nano /etc/caddy/Caddyfile
```

```
GNU nano 7.2 /etc/caddy/Caddyfile

: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

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

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

Descripción: Abre el archivo de configuración de Caddy.

Escribe el siguiente contenido:

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

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

8. Reiniciar Caddy

Comando:

```
sudo systemctl restart caddy
```

```
root@Ubuntuprepu:/home/apolouser# sudo systemctl restart caddy
```

Descripción: Reinicia Caddy para aplicar la nueva configuración.

9. Verificar estado de Caddy

Comando:

```
sudo systemctl status caddy
```

```
root@Ubuntuprepu:/home/apolouser# sudo 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 08:08:06 UTC; 7s ago
     Docs: https://caddyserver.com/docs/
  Main PID: 17574 (caddy)
    Tasks: 10 (limit: 4602)
   Memory: 12.0M (peak: 12.2M)
      CPU: 87ms
   CGroup: /system.slice/caddy.service
           └─17574 /usr/bin/caddy run --environ --config /etc/caddy/Caddyfile
```

Descripción: Comprueba que Caddy está funcionando en el puerto 8082.

10. Probar Caddy desde terminal

Comando:

`curl http://localhost:8082/`

```
root@Ubutuprepu:/home/apolouser# curl http://localhost:8082/

<!DOCTYPE html>
<html>
  <head>
    <title>/</title>
    <link rel="canonical" href="/" />
    <meta charset="utf-8">
    <meta name="color-scheme" content="light dark">
    <meta name="viewport" content="width=device-width, initial-sca
le=1.0">
    <style nonce="d38cf698-5b0b-4cc5-ae88-5bdc68a58536">
    * { padding: 0; margin: 0; box-sizing: border-box; }
```

Descripción: Lista los archivos disponibles en el servidor Caddy.

11. Probar archivo Markdown

Comando:

`curl http://localhost:8082/README.md`

Descripción: Verifica que Caddy sirve correctamente archivos Markdown.

PARTE 4: CONFIGURACIÓN DE HTTPS CON CERTBOT EN APACHE

1. Instalar Certbot y el plugin de Apache

Comando:

```
sudo apt install certbot python3-certbot-apache -y
```

```
root@Ubuntuprepu:/home/apolouser# sudo apt install certbot python3-certbot-apache -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Descripción: Instala Certbot y su integración con Apache para gestionar certificados SSL.

2. Verificar dominio o usar localhost

Nota: Para obtener certificados reales de Let's Encrypt necesitas un dominio público. Para esta práctica usaremos certificados autofirmados.

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
```

```
root@Ubuntuprepu:/home/apolouser# 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]: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) []:Apolo
Email Address []:kumogi.2007@gmail.com
```

Descripción: Crea un certificado autofirmado para practicar HTTPS localmente. Completa los campos solicitados (*puedes usar valores por defecto*).

3. Habilitar módulo SSL en Apache

Comando:

```
sudo a2enmod ssl
```

```
root@Ubuntuprepu:/home/apolouser# 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@Ubuntuprepu:/home/apolouser# systemctl restart apache2
root@Ubuntuprepu:/home/apolouser# sudo a2enmod ssl
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
```

Descripción: Activa el módulo SSL necesario para HTTPS en Apache.

4. Crear configuración SSL para Apache

Comando:

```
sudo nano /etc/apache2/sites-available/default-ssl.conf
```

Descripción: Edita el archivo y asegúrate de que incluye estas líneas dentro de

```
<VirtualHost *:443>:
```

```
SSLEngine on
```

```
SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
```

```
SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
```

5. Cambiar puerto SSL

Comando:

```
sudo nano /etc/apache2/ports.conf
```

Descripción: Añade la línea `Listen 8443` para que Apache escuche HTTPS en `puerto 8443`.

6. Modificar VirtualHost SSL

Comando:

```
sudo nano /etc/apache2/sites-available/default-ssl.conf
```

Descripción: Cambia `<VirtualHost *:443>` por `<VirtualHost *:8443>`.

7. Habilitar sitio SSL

Comando:

```
sudo a2ensite default-ssl.conf
```

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

Descripción: Activa la configuración SSL en Apache.

8. Reiniciar Apache

Comando:

```
sudo systemctl restart apache2
```

```
root@Ubuntuprepu:/home/apolouser# sudo systemctl restart apache2
```

Descripción: Aplica todos los cambios de configuración SSL.

9. Verificar HTTPS

Comando:

```
curl -i -k https://localhost:8443
```

Descripción: Prueba la conexión HTTPS (el flag -k ignora el aviso del certificado autofirmado).

PARTE 5: VERIFICACIÓN FINAL DE LOS TRES SERVIDORES

1. Verificar que todos los servicios están activos

Comando:

```
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 Fri 2025-10-10 08:25:20 UTC; 1min 52s ago

● 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 07:24:20 UTC; 1h 2min ago

● caddy.service - Caddy
   Loaded: loaded (/usr/lib/systemd/system/caddy.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-10-10 08:08:06 UTC; 19min ago
```

Descripción: Muestra el estado de los tres servidores simultáneamente.

2. Verificar puertos en uso

Comando:

```
sudo netstat -tulpn | grep -E '8080|8081|8082|8443'
```

```
root@Ubutuprepu:/home/apolouser# sudo netstat -tulpn | grep -E '8080|8081|8082|8443'
tcp        0      0 0.0.0.0:8081        0.0.0.0:*        LISTEN    13090/nginx:
  master
tcp6       0      0 :::8443            :::*             LISTEN    22093/apache
  2
tcp6       0      0 :::8082            :::*             LISTEN    17574/caddy
tcp6       0      0 :::8081            :::*             LISTEN    13090/nginx:
  master
```

Descripción: Lista los puertos donde están escuchando los servidores.

3. Probar todos los servidores

Comandos:

```
curl http://localhost:8080
```

```
curl http://localhost:8081
```

```
curl http://localhost:8082
```

```
curl -k https://localhost:8443
```

Todos los servidores se conectan

Descripción: Verifica que cada servidor responde correctamente en su puerto asignado.

CRITERIOS DE EVALUACIÓN

1. Funcionamiento Correcto (70%)

- Apache sirve correctamente PHP en puerto 8080 (15%).
- Apache sirve correctamente HTTPS en puerto 8443 con certificado (20%).
- Nginx sirve correctamente HTML en puerto 8081 (15%).
- Caddy sirve correctamente archivos Markdown e imágenes en puerto 8082 (20%).

2. Documentación (30%)

- Capturas claras de todos los comandos ejecutados (10%).
- Capturas de archivos de configuración modificados (10%).
- Capturas de verificación de los cuatro servicios funcionando simultáneamente (10%).

Solución de problemas comunes:

- Si un puerto ya está en uso:

```
sudo lsof -i :PUERTO
```
- Si un servicio no arranca:

```
sudo journalctl -u SERVICIO -n 50
```
- Si Apache/Nginx fallan: revisa los logs en:

```
/var/log/apache2/ o /var/log/nginx/
```