

MANUAL DE



APACHE

INSTALACIÓN Y FORTALECIMIENTO

GALINDO REYES DANIEL ADRIAN

EDITOR



INTRODUCCIÓN

Dentro de este manual se ofrece una guía completa para la instalación, configuración y gestión del servidor web Apache en sistemas operativos Ubuntu.

Apache es uno de los servidores web más utilizados a nivel mundial gracias a su robustez, flexibilidad y capacidad para adaptarse a una amplia variedad de entornos y proyectos. Este documento tiene como objetivo que el usuario aprenda a instalar y optimizar Apache desde cero, incluyendo la incorporación de módulos como `mod_security` y `mod_evasive`, la actualización de las dependencias del sistema, la configuración de sitios web únicos o múltiples mediante servidores virtuales, y la implementación de medidas de seguridad esenciales.

Además, este manual incluye instrucciones prácticas para verificar el correcto funcionamiento de Apache y su integración en entornos de prueba y producción.

Está diseñado tanto para principiantes como para administradores de sistemas con experiencia. Con esta guía, los usuarios podrán implementar Apache de manera eficiente, optimizar su rendimiento y garantizar la seguridad de los servidores y sitios web alojados en cada uno de los servidores virtuales configurados.

ÍNDICE

1

Requisitos previos para Apache

2

Actualización de paquetes del sistema

3

Instalación de Apache

4

Configuración básica de apache

5

Pruebas locales

6

Fortalecer Apache

- **Requisitos
previos
para
Apache**

REQUISITOS PREVIOS

Para instalar Apache, asegúrate de que el sistema cumpla con los siguientes requisitos mínimos:

- CPU: Procesador de 1 GHz o superior.
- RAM: Al menos 512 MB.
- Espacio en Disco: Mínimo 500 MB disponibles, más espacio si se alojan múltiples sitios.
- Distribución de Linux como sistema operativo, en este manual se utilizara Ubuntu. Sin embargo apache puede ser instalado en Windows, MacOS y otras distribuciones de linux como Fedora, Debian, CentOS, Red Hat, etc.
- Acceso con permisos de superusuario (root) en el Sistema operativo.
- Conexión a Internet: Necesaria para descargar los paquetes durante la instalación.

NOTA:

La configuración e instalación detallada de Ubuntu la podemos encontrar en el **Manual de Ubuntu: Instalación y funcionamiento (Galindo Reyes, 2024)**.

- **Actualización de paquetes del sistema**

ACTUALIZACIÓN DE PAQUETES DEL SISTEMA

Para comenzar con la instalación de Apache, verificaremos que las dependencias y paquetes dentro del sistema estén actualizados. Esto lo realizaremos de la siguiente forma en Ubuntu.

Una vez dentro del sistema operativo, utilizando un perfil con permisos de administración, nos dirigimos al menú de aplicaciones ubicado en la esquina inferior izquierda. Dentro del apartado de búsqueda, escribiremos "terminal" o "term" dependiendo del idioma instalado en el sistema operativo, como se muestra en las figuras 1.1 y 1.2.

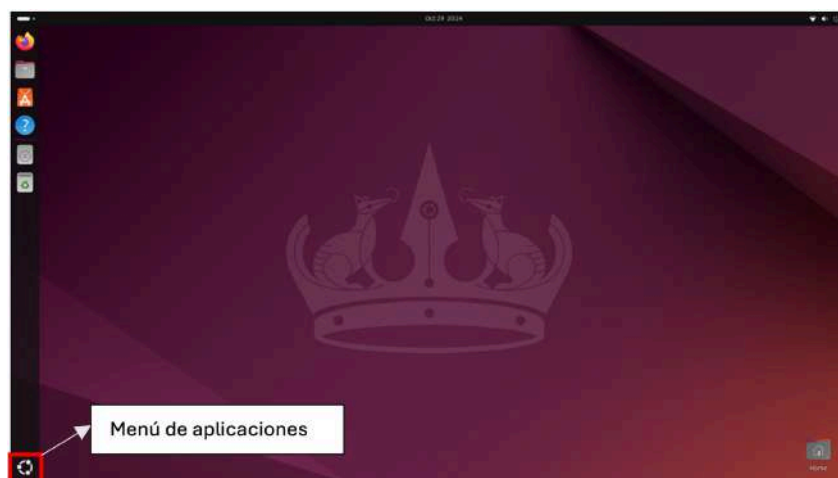


Figura 1.1. Escritorio de Ubuntu

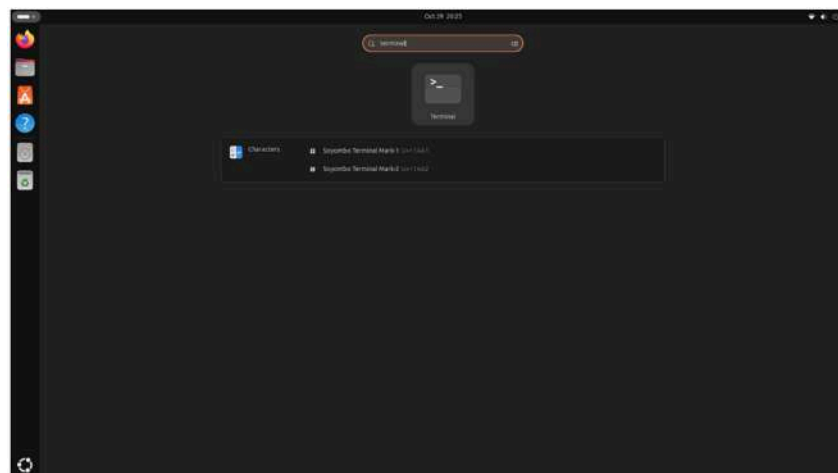
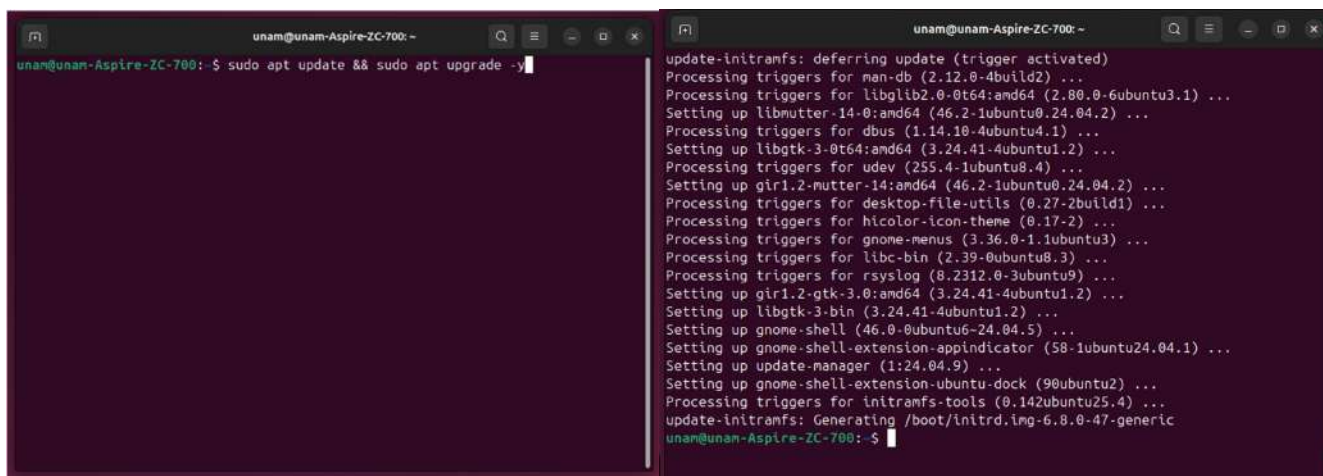


Figura 1.2. Ubicar terminal en menú de aplicaciones

Abrimos la terminal y escribiremos los comandos para actualizar los paquetes del sistema. Como primer comando, utilizaremos **sudo apt update**. Este comando actualiza la lista de paquetes disponibles en los repositorios configurados. A continuación, ejecutaremos el segundo comando: **sudo apt upgrade**, el cual instalará las versiones más recientes de los paquetes ya instalados en el sistema.

Podemos colocar ambos comandos en la misma línea de la terminal, como se muestra en la figura 1.3. Se agrega la bandera -y para aprobar los cambios automáticamente en caso de ser necesarios. Al utilizar la palabra sudo, será necesario ingresar la contraseña del sistema para confirmar la operación.



The image shows two terminal windows side-by-side. The left window displays the command `sudo apt update && sudo apt upgrade -y` being entered at the prompt. The right window shows the output of these commands, including messages about updating package lists and installing updates for various system components like `libmutter`, `libgtk`, and `gnome-shell`.

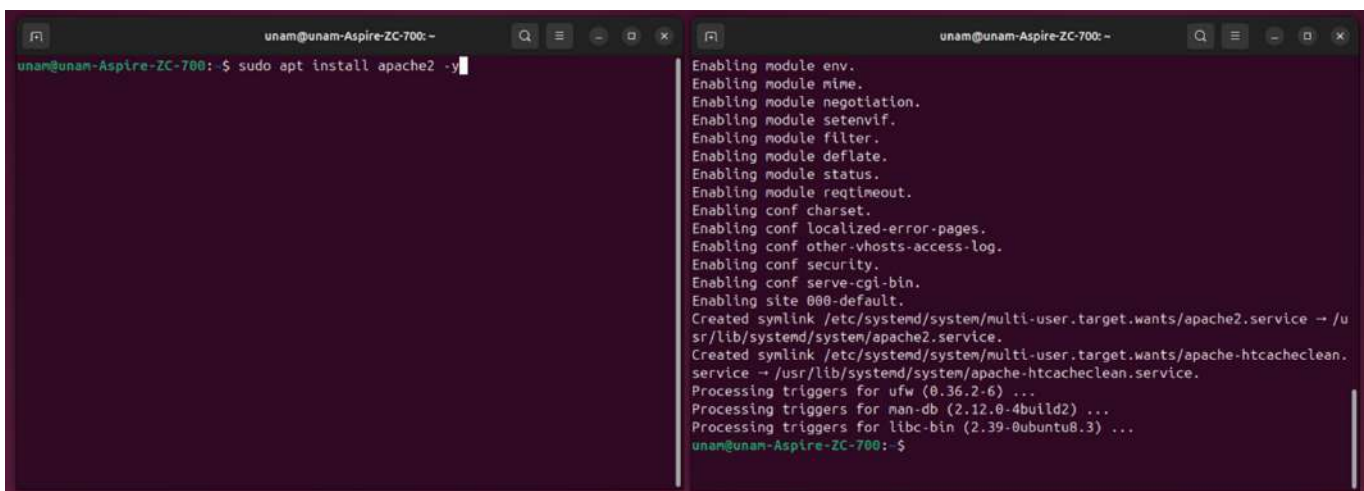
```
unam@unam-Aspire-ZC-700: ~  
unam@unam-Aspire-ZC-700: $ sudo apt update && sudo apt upgrade -y  
update-initramfs: deferring update (trigger activated)  
Processing triggers for man-db (2.12.0-4build2) ...  
Processing triggers for libgl1.0-0:amd64 (2.80.0-6ubuntu3.1) ...  
Setting up libmutter-14-0:amd64 (46.2-1ubuntu0.24.04.2) ...  
Processing triggers for dbus (1.14.10-4ubuntu4.1) ...  
Setting up libgtk-3-0:amd64 (3.24.41-4ubuntu1.2) ...  
Processing triggers for udev (255.4-1ubuntu8.4) ...  
Setting up gir1.2-mutter-14:amd64 (46.2-1ubuntu0.24.04.2) ...  
Processing triggers for desktop-file-utils (0.27-2build1) ...  
Processing triggers for hicolor-icon-theme (0.17-2) ...  
Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...  
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...  
Processing triggers for rsyslog (8.2312.0-3ubuntu9) ...  
Setting up gir1.2-gtk-3.0:amd64 (3.24.41-4ubuntu1.2) ...  
Setting up libgtk-3-bin (3.24.41-4ubuntu1.2) ...  
Setting up gnome-shell (46.0-0ubuntu6-24.04.5) ...  
Setting up gnome-shell-extension-appindicator (58-1ubuntu24.04.1) ...  
Setting up update-manager (1:24.04.9) ...  
Setting up gnome-shell-extension-ubuntu-dock (90ubuntu2) ...  
Processing triggers for initramfs-tools (0.142ubuntu25.4) ...  
update-initramfs: Generating /boot/initrd.img-6.8.0-47-generic  
unam@unam-Aspire-ZC-700: $
```

Figura 1.3. Comandos de actualización

- **Instalación y
descarga de
apache**

INSTALACIÓN Y DESCARGA DE APACHE

Una vez que las dependencias y paquetes del sistema operativo estén actualizados, procederemos con la descarga de Apache. En la terminal donde escribimos los comandos anteriores, o en una nueva, ingresaremos el siguiente comando **sudo apt install apache2 -y**. Este comando instalará todo lo necesario para Apache. Para este paso, es necesario contar con acceso a Internet. Una vez finalizada la instalación, tendremos Apache instalado en Ubuntu. La pantalla de finalización se muestra en la figura 1.4.

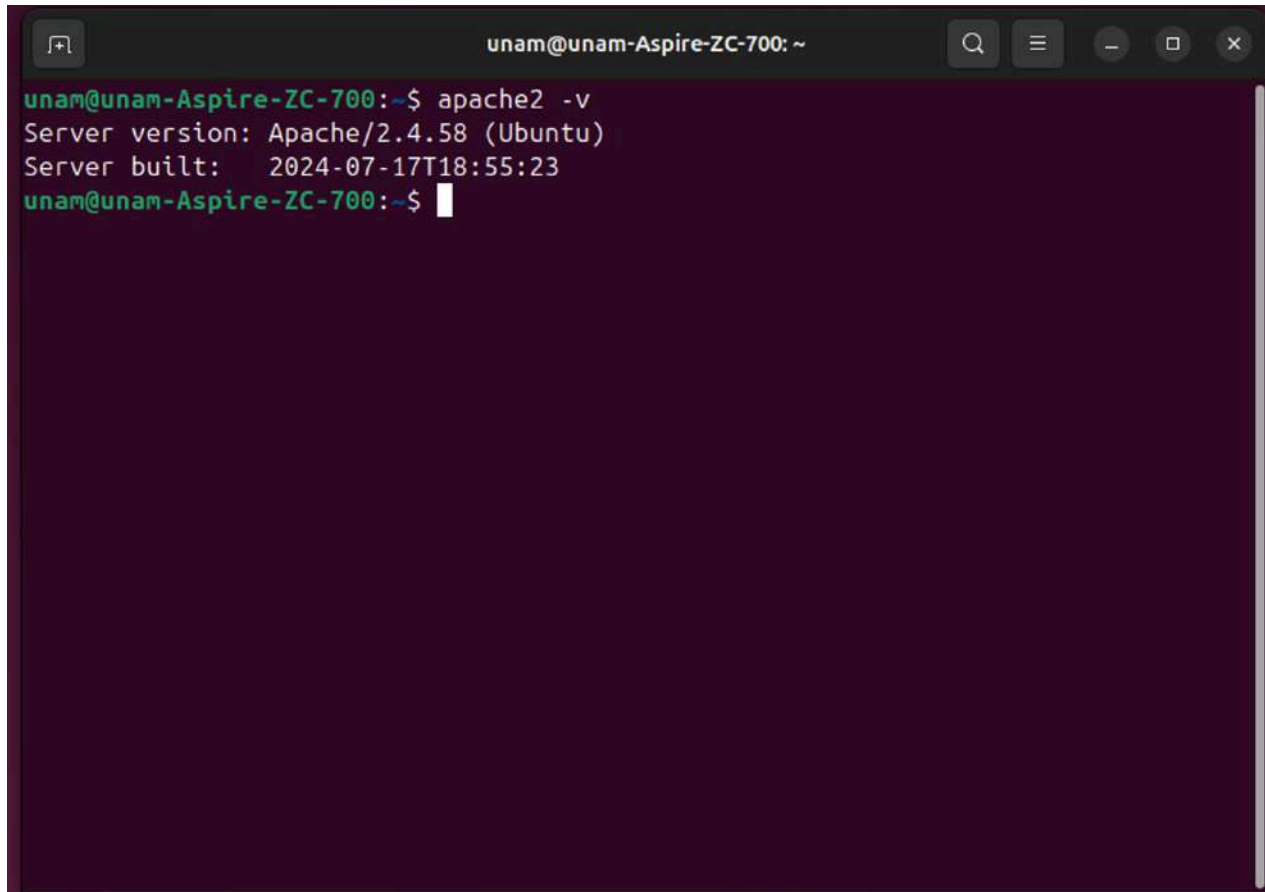


```
unam@unam-Aspire-ZC-700: ~  
unam@unam-Aspire-ZC-700:~$ sudo apt install apache2 -y  
Enabling module env.  
Enabling module mime.  
Enabling module negotiation.  
Enabling module setenvif.  
Enabling module filter.  
Enabling module deflate.  
Enabling module status.  
Enabling module reqtimeout.  
Enabling conf charset.  
Enabling conf localized-error-pages.  
Enabling conf other-vhosts-access-log.  
Enabling conf security.  
Enabling conf serve-cgi-bin.  
Enabling site 000-default.  
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /usr/lib/systemd/system/apache2.service.  
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /usr/lib/systemd/system/apache-htcacheclean.service.  
Processing triggers for ufw (0.36.2-6) ...  
Processing triggers for nan-db (2.12.0-4build2) ...  
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...  
unam@unam-Aspire-ZC-700:~$
```

Figura 1.4. Comandos de instalación y finalización

VERIFICACIÓN DE LA INSTALACIÓN Y ESTADO DE APACHE

Una vez instalado Apache podemos verificar la versión del mismo usando el comando **apache2 -v**. Esta nos devolverá versión de Apache instalada, como se ve en la figura 1.5.



```
unam@unam-Aspire-ZC-700: ~  
unam@unam-Aspire-ZC-700:~$ apache2 -v  
Server version: Apache/2.4.58 (Ubuntu)  
Server built: 2024-07-17T18:55:23  
unam@unam-Aspire-ZC-700:~$
```

Figura 1.5. Version de apache

También podemos verificar el estado del servicio. Esto lo revisaremos con el comando **sudo systemctl status apache2**, esperando como resultado un mensaje como active (running), como se ve en la figura 1.6, para salir de la pantalla de estatus, presiona la combinación de teclas **Control + C**.

```
unam@unam-Aspire-ZC-700: ~  
unam@unam-Aspire-ZC-700:~$ systemctl status apache2  
● apache2.service - The Apache HTTP Server  
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: >  
   Active: active (running) since Tue 2024-10-29 20:45:07 CST; 1min 38s ago  
     Docs: https://httpd.apache.org/docs/2.4/  
  Main PID: 13582 (apache2)  
    Tasks: 55 (limit: 4506)  
   Memory: 5.6M (peak: 6.1M)  
      CPU: 180ms  
   CGroup: /system.slice/apache2.service  
           └─13582 /usr/sbin/apache2 -k start  
             └─13584 /usr/sbin/apache2 -k start  
               └─13586 /usr/sbin/apache2 -k start  
  
Oct 29 20:45:07 unam-Aspire-ZC-700 systemd[1]: Starting apache2.service - The A>  
Oct 29 20:45:07 unam-Aspire-ZC-700 apachectl[13581]: AH00558: apache2: Could no>  
Oct 29 20:45:07 unam-Aspire-ZC-700 systemd[1]: Started apache2.service - The Ap>  
lines 1-16/16 (END)
```

Figura 1.6. Estatus de apache2

Podemos cambiar el estado del servicio utilizando **sudo systemctl start apache2** para iniciarlo o **sudo systemctl stop apache2** para detenerlo. Si necesitamos reiniciar el servicio, usaremos **sudo systemctl restart apache2**. También es posible habilitarlo de forma automática al iniciar el sistema; para ello, podemos ejecutar **sudo systemctl enable apache2**. Los comandos se enlistan en la figura 1.7.

```
unam@unam-Aspire-ZC-700: ~  
unam@unam-Aspire-ZC-700:~$ systemctl restart apache2  
unam@unam-Aspire-ZC-700:~$ systemctl stop apache2  
unam@unam-Aspire-ZC-700:~$ systemctl start apache2  
unam@unam-Aspire-ZC-700:~$
```

Figura 1.7. Comandos de estado de apache

Podemos entrar a visualizar esto de una manera mas fácil entrando desde un navegador, en este caso Mozilla Firefox, a la dirección LocalHost, esto abrirá la pagina de apache si todo se encuentra funcionando correctamente, como se ve en la figura 1.8.

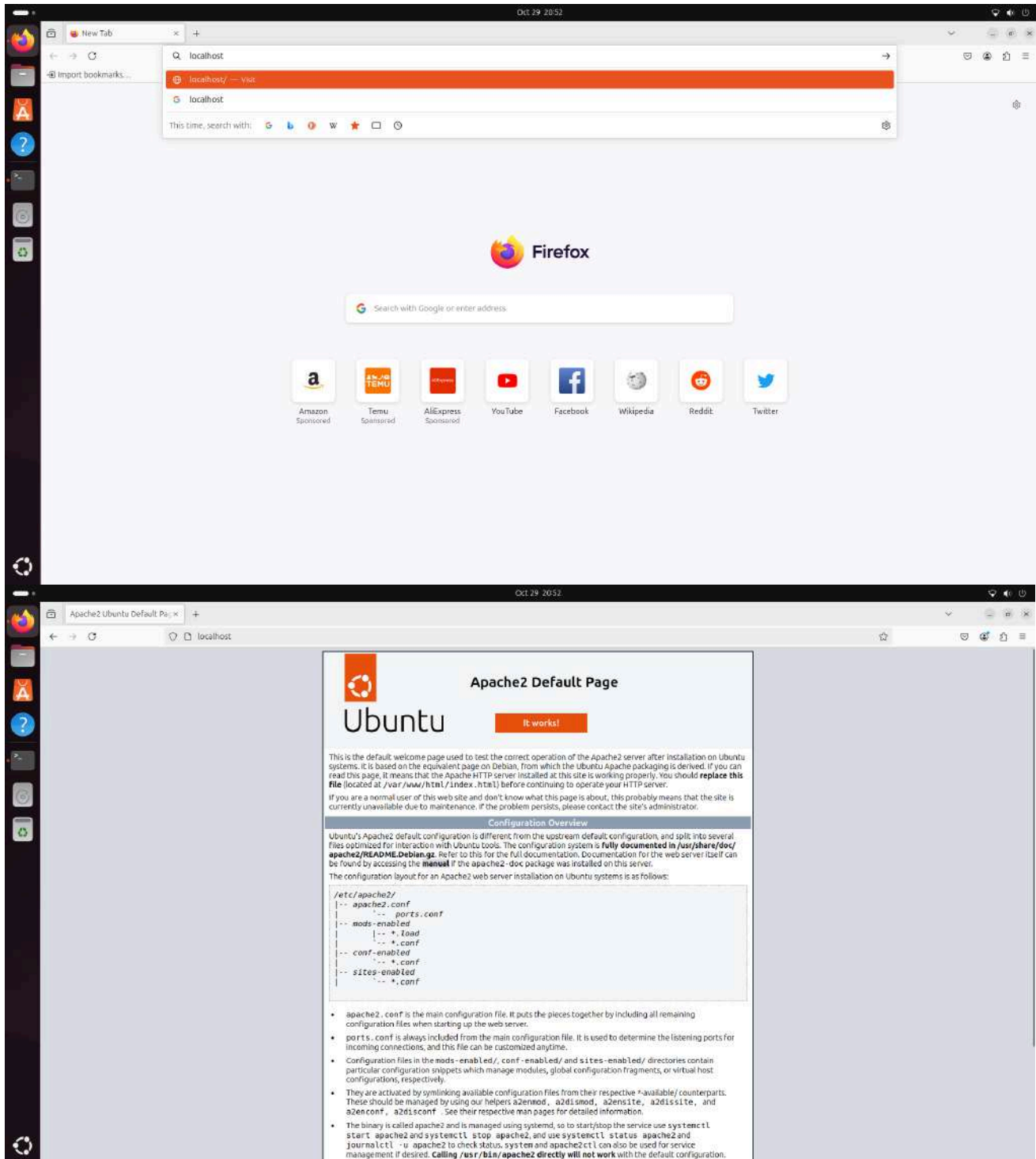


Figura 1.8. Pagina web de bienvenida

- **Configuración
basica de
apache**

CONFIGURACIÓN BÁSICA DE APACHE

La configuración inicial de Apache está orientada a alojar un solo sitio web en la ruta predeterminada **/var/www/html**. Esta ubicación puede variar según el sistema operativo utilizado. Dentro de esta carpeta se encuentra el archivo **index.html**, que sirve como la página inicial de Apache. Si solo se requiere alojar un sitio web, basta con colocar los archivos del sitio, incluido el archivo index, en esta carpeta. A esta configuración se le denomina servicio único.

Sin embargo, Apache permite una mayor flexibilidad mediante el uso de servidores virtuales (Virtual Hosts), que posibilitan alojar varios sitios web en el mismo servidor. Cada servidor virtual define su propio dominio o subdominio y puede tener una carpeta raíz distinta para los archivos del sitio. Esto es ideal para gestionar múltiples proyectos o clientes en un solo servidor, asignando configuraciones independientes a cada sitio web sin necesidad de varios servidores físicos o instancias adicionales.

Configuración de servidores virtuales:

Para comenzar, debemos crear las carpetas que contendrán los archivos de cada sitio. Esto se puede hacer desde el controlador de archivos accediendo a la ubicación **/var/www** y creando las diferentes páginas, o desde la terminal con los siguientes comandos, así como sus index:

```
sudo mkdir -p /var/www/<nombre de la carpeta>
sudo nano /var/www/<nombre de la carpeta>/index
```

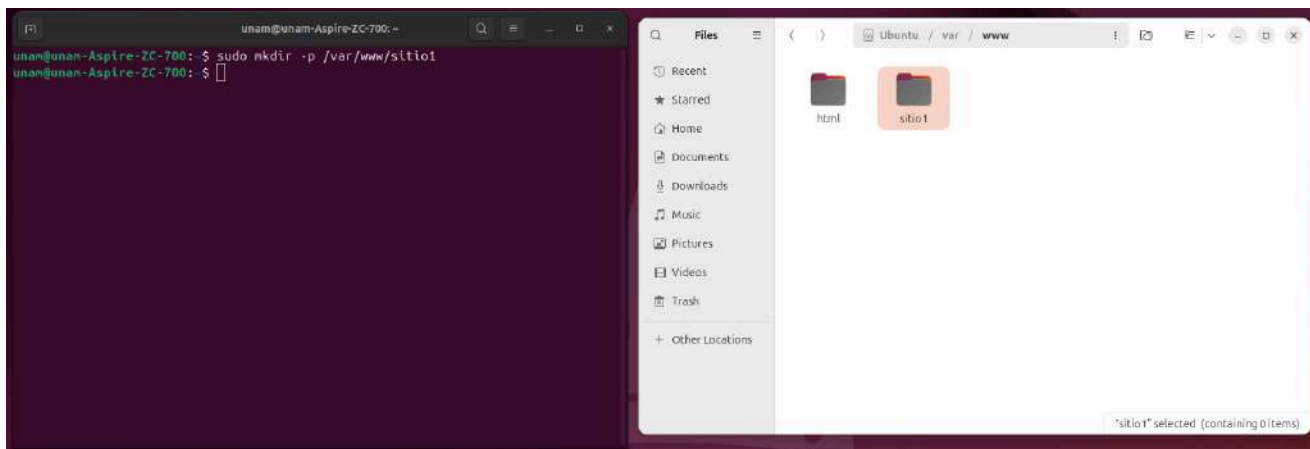



Figura 1.9. Creación de carpeta

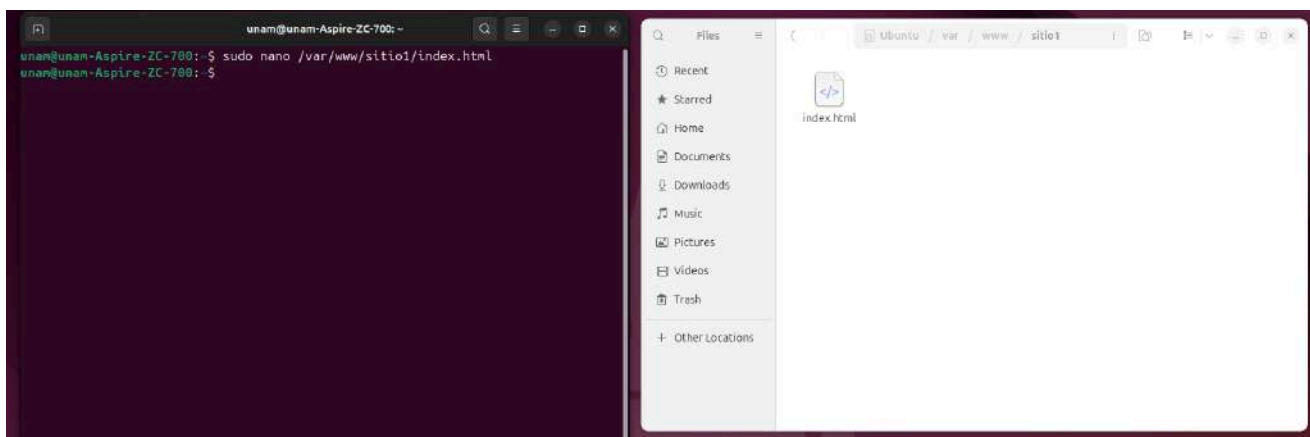


Figura 1.10. Creación de index

Una vez que tengamos los sitios divididos en carpetas, cada una con su **index.html** o **index.php**, debemos asignar los permisos necesarios. Esto lo haremos con los siguientes comandos:

```
sudo chown -R www-data:www-data /var/www/<nombre de la carpeta>
sudo chmod -R 755 /var/www/<nombre de la carpeta>
```


Explicación de los comandos

www-data:www-data asigna el usuario y grupo propietario.

En Apache, **www-data** es el usuario bajo el cual corre el servidor web. Esto le permite acceder, leer y servir los archivos de los sitios web sin problemas de permisos.

chmod -R 755 asigna permisos de Lectura, escritura y ejecución para el propietario

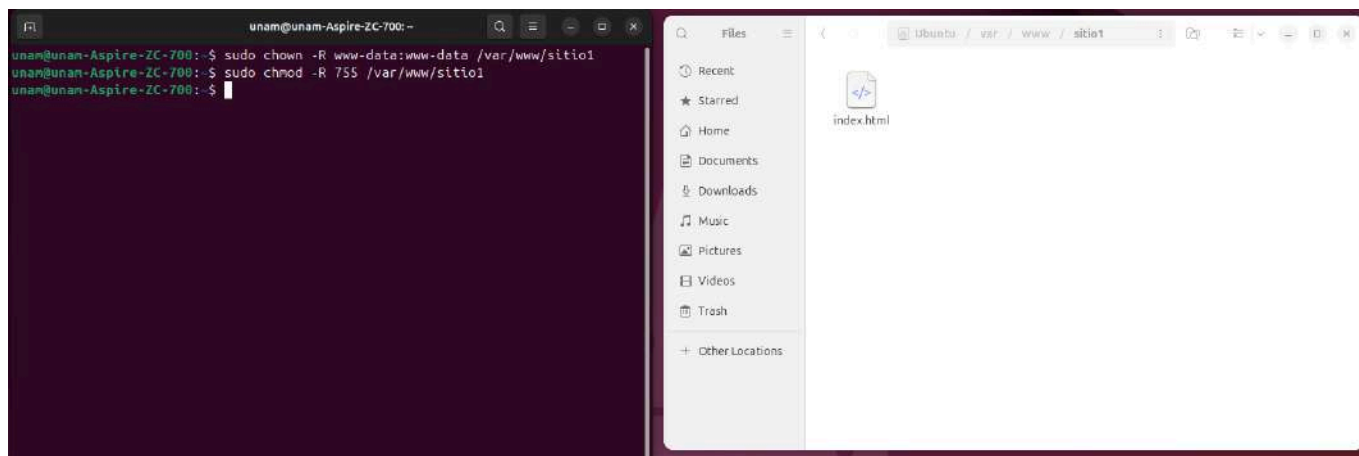


Figura 1.11. Asignación de permisos

Una vez que hemos configurado los archivos y dado los permisos necesarios, es momento de crear el servidor virtual. En Apache, este proceso es bastante sencillo, para cada sitio que se desee alojar, debemos crear un archivo **.conf** dentro de la carpeta **/etc/apache2/sites-available**. A continuación se muestra un ejemplo:

```
sudo nano /etc/apache2/sites-available/site1.conf
```

Dentro del archivo **.conf**, debemos definir las configuraciones básicas para que Apache sepa cómo manejar el sitio. Estas configuraciones incluyen:

- Nombre del servidor (**ServerName**) se debe poner la ip publica o dominio apuntando a la misma, en este caso se asume que site1.com apunta a la ip publica del servidor, o en su defecto a la ip local para pruebas, como se hará mas adelante.
- Ubicación del sitio (**DocumentRoot**).
- **Permisos de acceso** al directorio.

A continuación, se muestra un ejemplo de configuración para un sitio llamado site1, alojado en **/var/www/site1**, que recibirá peticiones en el puerto 80:

```
<VirtualHost *:80>
    ServerName site1.com
    DocumentRoot /var/www/site1

    <Directory /var/www/site1>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/site1_error.log
    CustomLog ${APACHE_LOG_DIR}/site1_access.log combined
</VirtualHost>
```

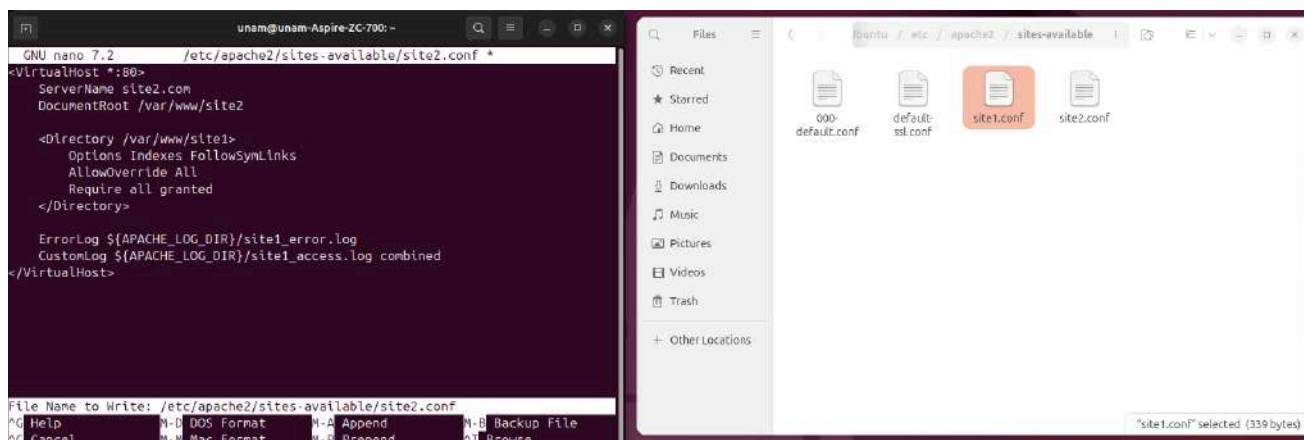


Figura 1.12. Archivo .conf de un virtual host

Algo importante al configurar los puertos de acceso para el tráfico web es que existen puertos estándar que se utilizan de forma predeterminada, si se busca manejar diferentes sitios con una misma ip publica se recomienda reedirigir el trafico a diferentes puertos dependiendo de el sitio al que se quiera acceder:

Puerto 80: Es el puerto por defecto para HTTP (HyperText Transfer Protocol), utilizado para tráfico web sin cifrado. Cuando un navegador accede a un sitio web mediante http:// , se conecta automáticamente al puerto 80, a menos que se especifique otro puerto.

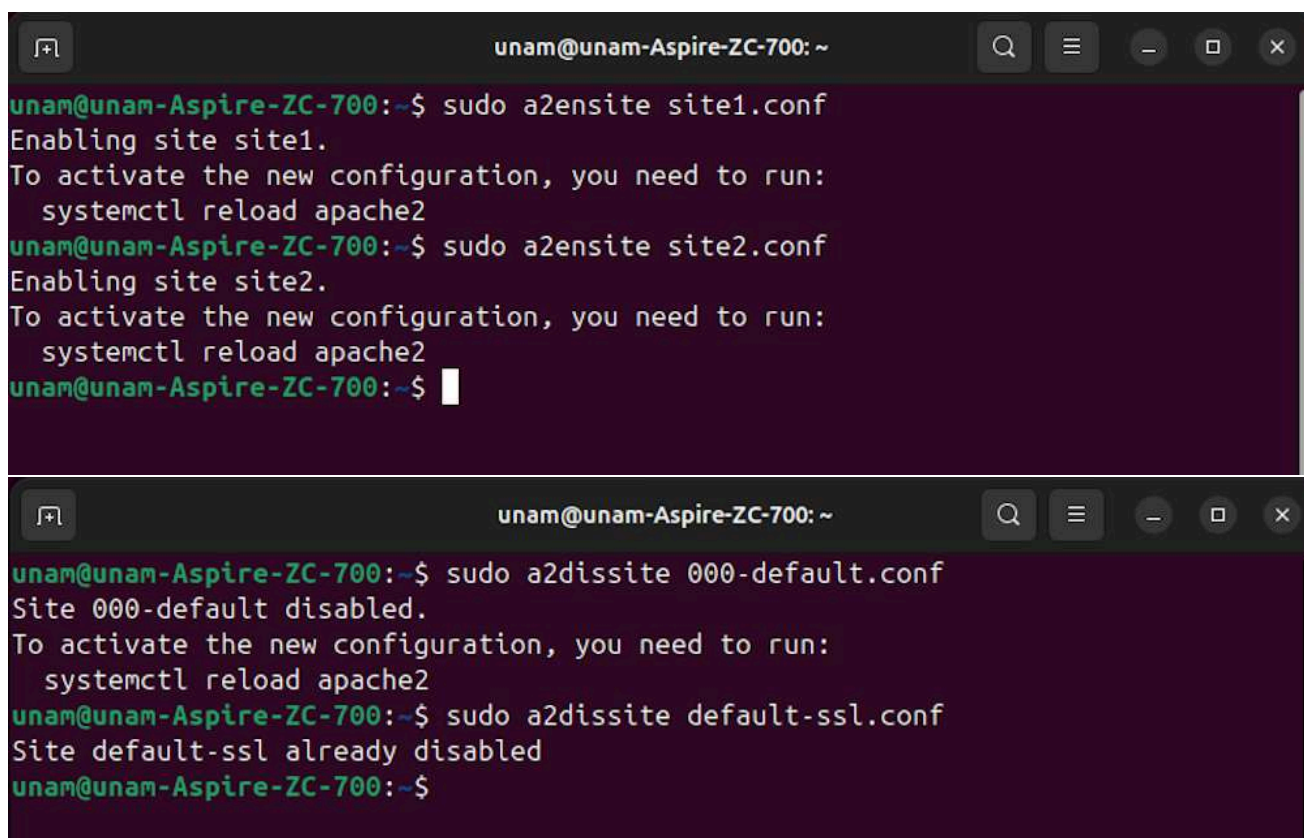
Puerto 443: Es el puerto predeterminado para conexiones HTTPS (HTTP Secure), utilizado para el tráfico web cifrado mediante SSL/TLS. Todas las conexiones seguras que comienzan con https:// utilizan este puerto, asegurando que los datos transmitidos entre el cliente y el servidor estén protegidos. Para habilitar esta conexión segura, es necesario implementar certificados SSL/TLS generados con herramientas como OpenSSL o adquiridos a través de certificadoras reconocidas.

NOTA:

La configuración detallada del uso del puerto 443 y los certificados SSL/TLS para una instalación específica de una página se encuentra en el *Manual de Nextcloud: Instalación y funcionamiento (Galindo Reyes, 2024, p. 31). Para más información, consulte el apartado "Sobre seguridad y rendimiento".*

Cada sitio que se agregue necesitará su propio archivo **.conf** en la carpeta **/etc/apache2/sites-available**, una vez creados y guardados los archivos **.conf** tendremos que reiniciar apache y actualizar los cambios, también se recomienda deshabilitar la pagina por defecto de apache:

```
sudo a2ensite site1.conf
sudo a2dissite 000-default.conf
sudo systemctl restart apache2
```



The image displays two terminal windows from a user named 'unam' on a system named 'unam-Aspire-ZC-700'. The top window shows the execution of 'sudo a2ensite site1.conf' and 'sudo a2ensite site2.conf', both of which output 'Enabling site' followed by instructions to run 'systemctl reload apache2'. The bottom window shows 'sudo a2dissite 000-default.conf' outputting 'Site 000-default disabled' and 'systemctl reload apache2', followed by 'sudo a2dissite default-ssl.conf' outputting 'Site default-ssl already disabled'.

```
unam@unam-Aspire-ZC-700: ~  
unam@unam-Aspire-ZC-700:~$ sudo a2ensite site1.conf  
Enabling site site1.  
To activate the new configuration, you need to run:  
    systemctl reload apache2  
unam@unam-Aspire-ZC-700:~$ sudo a2ensite site2.conf  
Enabling site site2.  
To activate the new configuration, you need to run:  
    systemctl reload apache2  
unam@unam-Aspire-ZC-700:~$  
  
unam@unam-Aspire-ZC-700: ~  
unam@unam-Aspire-ZC-700:~$ sudo a2dissite 000-default.conf  
Site 000-default disabled.  
To activate the new configuration, you need to run:  
    systemctl reload apache2  
unam@unam-Aspire-ZC-700:~$ sudo a2dissite default-ssl.conf  
Site default-ssl already disabled  
unam@unam-Aspire-ZC-700:~$
```

Figura 1.13. Habilitar/Deshabilitar servidores virtuales en apache

- **Pruebas
locales**

PRUEBAS LOCALES

Antes de abrir puertos y permitir tráfico desde redes externas, es esencial asegurarse de que la configuración del sitio web en Apache funcione correctamente de manera local.

Para ello, es necesario modificar el archivo hosts, ubicado en **/etc/hosts**. El objetivo es poder acceder a los sitios desde localhost, simulando que están en producción. Para lograr esto, debemos asociar una dirección IP local (por lo general, 127.0.0.1) a cada uno de los dominios de prueba.

Agregar entradas en /etc/hosts

Abre el archivo con permisos de administrador:

```
sudo nano /etc/hosts
```

Agrega las siguientes líneas para que todos los sitios apunten a 127.0.0.1:

```
127.0.0.1 site1.com  
127.0.0.1 site2.com
```

Cada vez que ingreses o en tu navegador, la solicitud será dirigida a tu máquina local.

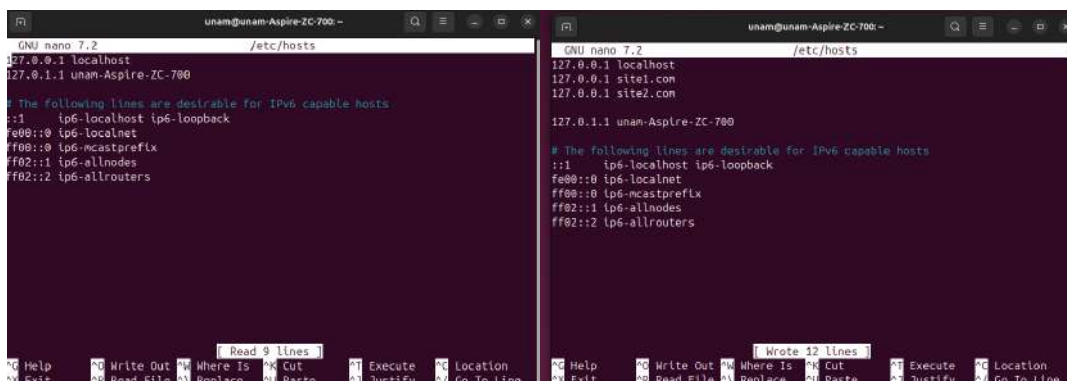


Figura 1.14. Modificaciones en el archivo hosts

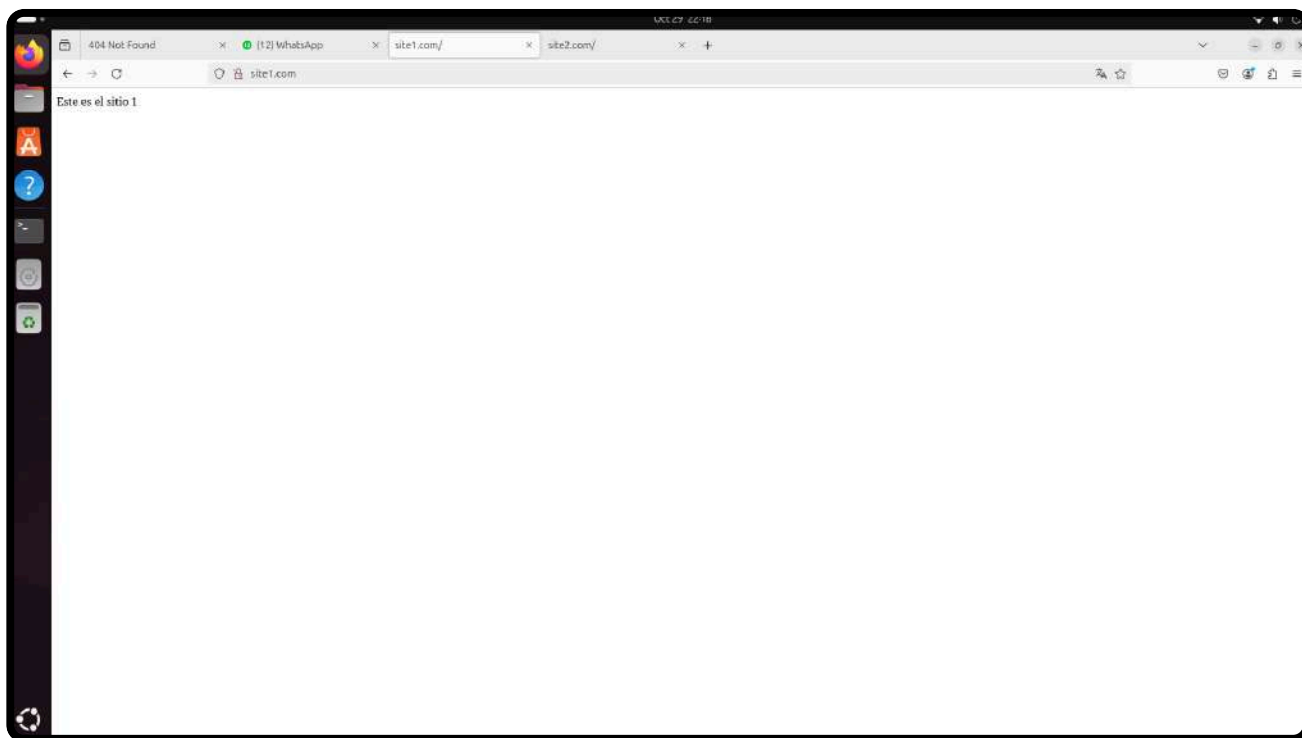


Figura 1.15. Pagina mostrada con la dirección site1.com

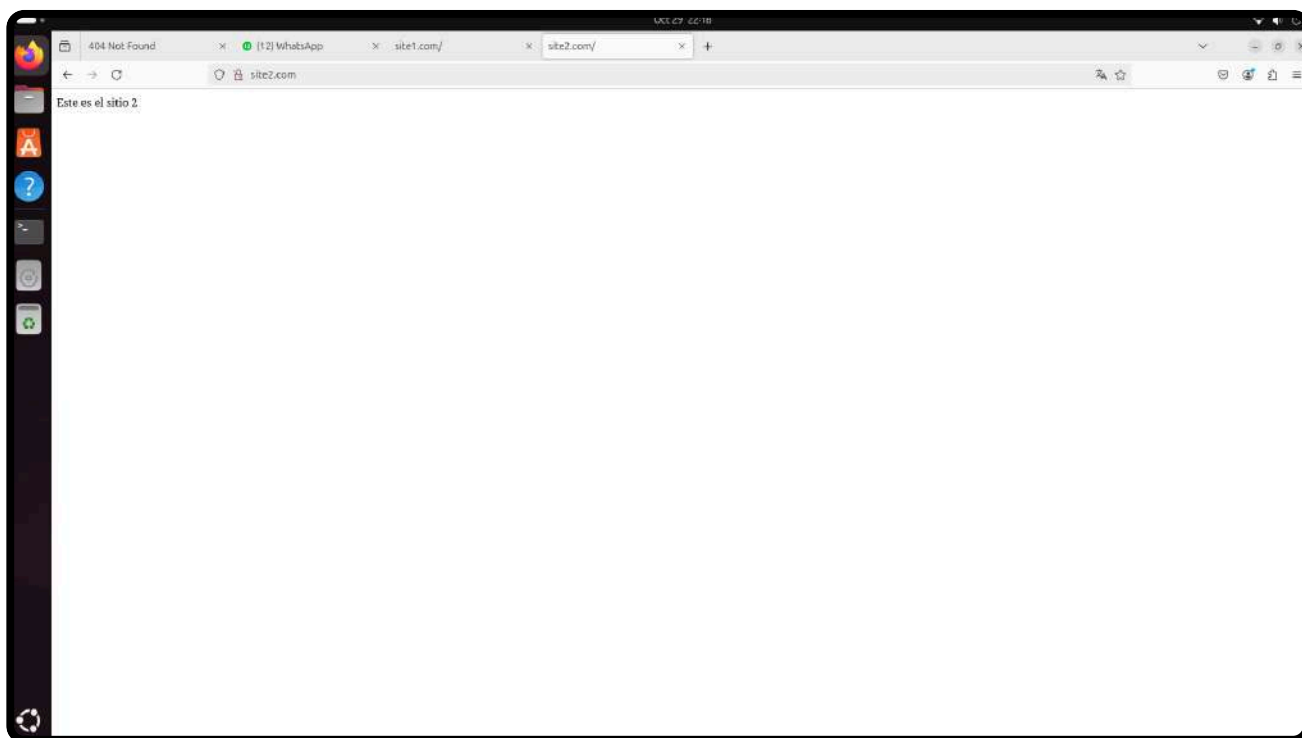


Figura 1.16. Pagina mostrada con la dirección site2.com

- **Fortalecer
Apache**

CONFIGURACION DE FIREWALL

Para que nuestro servidor con Apache pueda recibir tráfico y peticiones desde redes externas, es necesario realizar algunas modificaciones en las reglas de seguridad del firewall.

En Ubuntu, por defecto se utiliza UFW (Uncomplicated Firewall). Esta herramienta simplifica la gestión del firewall en sistemas operativos basados en Linux mediante comandos fáciles de usar.

Comandos básicos de UFW para Apache:

- Permitir tráfico HTTP (puerto 80) y HTTPS (puerto 443):

```
sudo ufw allow 80/tcp  
sudo ufw allow 443/tcp
```

- Habilitar UFW: Si aún no está habilitado, activa el firewall con:

```
sudo ufw enable
```

- Verificar el estado del firewall:

```
sudo ufw status
```

- UFW cuenta con un perfil predefinido que facilita la apertura de los puertos necesarios para que Apache funcione correctamente. Este perfil permite habilitar de una sola vez el tráfico para HTTP (puerto 80) y HTTPS (puerto 443).

```
sudo ufw allow 'Apache Full'
```

INSTALAR Y CONFIGURAR MOD_SECURITY

ModSecurity es un módulo de seguridad diseñado para servidores web, principalmente Apache, que funciona como un firewall de aplicaciones web (WAF, por sus siglas en inglés). Su principal objetivo es proteger las aplicaciones web contra una amplia gama de ataques y vulnerabilidades comunes, como **inyecciones SQL**, **cross-site scripting** (XSS), **ataques de fuerza bruta**, entre otros, que podrían comprometer la seguridad de un sitio web. Por esta razón, la instalación y configuración de este módulo es esencial para fortalecer la seguridad en Apache.

ModSecurity actúa interceptando, analizando y filtrando las solicitudes y respuestas HTTP que pasan a través del servidor web. Aplica reglas predefinidas (o personalizadas) para identificar y bloquear tráfico malicioso antes de que llegue a las aplicaciones. **Utiliza las reglas del OWASP Core Rule Set (CRS)**, un conjunto que aborda las vulnerabilidades más frecuentes, lo que lo convierte en una herramienta robusta para proteger aplicaciones y sitios web. Además, es posible personalizarlo agregando reglas adicionales para adaptarse a necesidades específicas y proteger contra amenazas más avanzadas.

Para instalar el modulo escribiremos la siguiente linea en la terminal:

```
sudo apt install libapache2-mod-security2 -y
```

```
ubuntu@ip-172-31-82-65:~$ sudo apt install libapache2-mod-security2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libapache2-mod-security2 is already the newest version (2.9.7-1build3).
0 upgraded, 0 newly installed, 0 to remove and 20 not upgraded.
ubuntu@ip-172-31-82-65:~$
```

Figura 1.17. mod-security2 instalado

Podemos verificar que mod_security se encuentra instalado de la manera correcta con el comando:

```
sudo a2enmod security2
```

```
[ubuntu@ip-172-31-82-65:~$ sudo a2enmod security2
Considering dependency unique_id for security2:
Module unique_id already enabled
Module security2 already enabled
ubuntu@ip-172-31-82-65:~$ █
```

Figura 1.18. mod-security2 enabled

ModSecurity incluye un conjunto de reglas básico llamado OWASP Core Rule Set (CRS), el cual protege contra ataques comunes como SQL injection y XSS. Habilítalo con el siguiente comando.

```
sudo cp /etc/modsecurity/modsecurity.conf-recommended
/etc/modsecurity/modsecurity.conf
```

```
[ubuntu@ip-172-31-82-65:/etc/modsecurity$ sudo cp /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf
[ubuntu@ip-172-31-82-65:/etc/modsecurity$ ls
crs  modsecurity.conf  modsecurity.conf-recommended  unicode.mapping
ubuntu@ip-172-31-82-65:/etc/modsecurity$ █
```

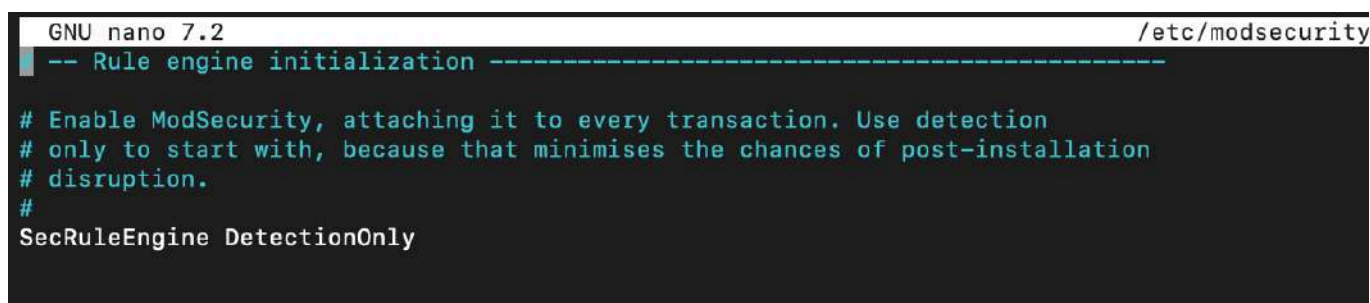
Figura 1.19. Activar configuración recomendada

Edita el archivo **modsecurity.conf** creado para realizar algunas modificaciones:

```
sudo nano /etc/modsecurity/modsecurity.conf
```

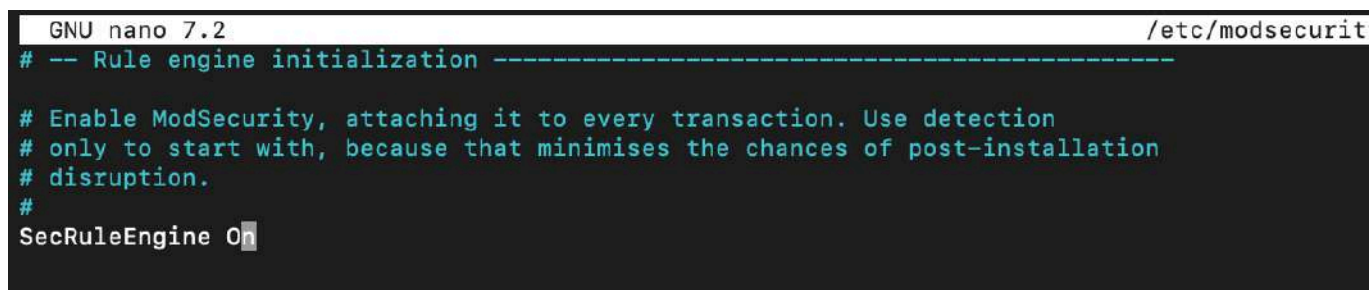
Cambiaremos la linea de SecRuleEngine como se ve a continuación:

```
SecRuleEngine DetectionOnly -> SecRuleEngine On
```

A screenshot of the GNU nano 7.2 text editor. The title bar shows 'GNU nano 7.2' on the left and '/etc/modsecurity' on the right. The main content area shows a configuration file with several lines of comments and one line of configuration: '# Enable ModSecurity, attaching it to every transaction. Use detection # only to start with, because that minimises the chances of post-installation # disruption.' followed by 'SecRuleEngine DetectionOnly'.

```
GNU nano 7.2 /etc/modsecurity
-- Rule engine initialization -----
# Enable ModSecurity, attaching it to every transaction. Use detection
# only to start with, because that minimises the chances of post-installation
# disruption.
#
SecRuleEngine DetectionOnly
```

Figura 1.20. SecRuleEngine DetectionOnly

A screenshot of the GNU nano 7.2 text editor, similar to the previous one. The title bar shows 'GNU nano 7.2' on the left and '/etc/modsecurity' on the right. The main content area shows the same configuration file, but the line 'SecRuleEngine On' is now present, with the cursor positioned at the end of the word 'On'.

```
GNU nano 7.2 /etc/modsecurity
-- Rule engine initialization -----
# Enable ModSecurity, attaching it to every transaction. Use detection
# only to start with, because that minimises the chances of post-installation
# disruption.
#
SecRuleEngine On
```

Figura 1.21. SecRuleEngine On

Para terminar la instalación reiniciamos el servicio de apache con **sudo systemctl daemon-reload** y posterior **sudo systemctl restart apache2**.

A terminal screenshot showing the execution of two systemctl commands. The prompt is '[ubuntu@ip-172-31-82-65:~\$]'. The first command is 'sudo systemctl daemon-reload' and the second is 'sudo systemctl restart apache2'. The output shows the commands being executed successfully.

```
[ubuntu@ip-172-31-82-65:~$] sudo systemctl daemon-reload
[ubuntu@ip-172-31-82-65:~$] sudo systemctl restart apache2
[ubuntu@ip-172-31-82-65:~$]
```

Figura 1.22. Aplicar cambios en apache

Algunas reglas predeterminadas en **ModSecurity** pueden generar conflictos con servicios previamente configurados. Por ello, comunidades como **SpiderLabs** y otras realizan ajustes en las reglas estándar para solucionar problemas de compatibilidad. Si deseas descargar e implementar estas reglas personalizadas, puedes acceder al repositorio oficial en <https://github.com/SpiderLabs/owasp-modsecurity-crs> para obtener las últimas actualizaciones y configuraciones optimizadas. Posteriormente, es necesario acceder a la carpeta "rules" y agregar los archivos proporcionados por estas comunidades dentro de sus propias carpetas rules. Generalmente, estos archivos se encuentran en la ruta **/usr/share/modsecurity-crs/rules** de la siguiente forma:

```
sudo cp <carpetaDescargada/rules> /usr/share/modsecurity-crs/rules
```

Además, se recomienda revisar los siguientes archivos de configuración, las reglas seguirán este orden de configuración y prioridad:

- **modsecurity.conf**: ubicado en /etc/modsecurity
- **crs-setup.conf**: ubicado en /etc/modsecurity/crs
- **owasp-crs.load**: El archivo que carga las rutas de las reglas, ubicado en /usr/share/modsecurity-crs.

Esto permitirá asegurar que las configuraciones sean coherentes y compatibles con los servicios en uso, no olvides reiniciar apache con **sudo systemctl restart apache2**.

Para ver estos logs, podemos usar el siguiente comando:

```
sudo tail -f /var/log/apache2/modsec_audit.log
```


Por ejemplo, para implementar el servicio de **Nextcloud** con **ModSecurity**, una vez descargadas las reglas en la carpeta rules predeterminada, podemos asegurar su funcionamiento simplemente cambiando un valor dentro del archivo **crs-setup.conf**:

```
# -- [[ Application Specific Rule Exclusions ]] -----
#
# Some well-known applications may undertake actions that appear to be
# malicious. This includes actions such as allowing HTML or Javascript within
# parameters. In such cases the CRS aims to prevent false positives by allowing
# administrators to enable prebuilt, application specific exclusions on an
# application by application basis.
# These application specific exclusions are distinct from the rules that would
# be placed in the REQUEST-900-EXCLUSION-RULES-BEFORE-CRS configuration file as
# they are prebuilt for specific applications. The 'REQUEST-900' file is
# designed for users to add their own custom exclusions. Note, using these
# application specific exclusions may loosen restrictions of the CRS,
# especially if used with an application they weren't designed for. As a result
# they should be applied with care.
# To use this functionality you must specify a supported application. To do so
# uncomment rule 900130. In addition to uncommenting the rule you will need to
# specify which application(s) you'd like to enable exclusions for. Only a
# (very) limited set of applications are currently supported, please use the
# filenames prefixed with 'REQUEST-903' to guide you in your selection.
# Such filenames use the following convention:
# REQUEST-903.9XXX-{APPNAME}-EXCLUSIONS-RULES.conf
#
# It is recommended if you run multiple web applications on your site to limit
# the effects of the exclusion to only the path where the excluded webapp
# resides using a rule similar to the following example:
# SecRule REQUEST_URI "@beginsWith /wordpress/" setvar:tx.crs_exclusions_wordpress=1

#
# Modify and uncomment this rule to select which application:
#
SecAction \
  "id:900130,\
  phase:1,\
  nolog,\
  pass,\
  t:none,\
  setvar:tx.crs_exclusions_nextcloud=1"
# setvar:tx.crs_exclusions_wordpress=1,\
# setvar:tx.crs_exclusions_xenforo=1"
# setvar:tx.crs_exclusions_cpanel=1,\
# setvar:tx.crs_exclusions_drupal=1,\
# setvar:tx.crs_exclusions_dokuwiki=1,\
```

Figura 1.23. *crs-setup.conf* aplicando restricciones específicas para servicio

INSTALAR Y CONFIGURAR MOD_EVASIVE

Mod_evasive es un módulo de seguridad diseñado para servidores web, principalmente Apache, que actúa como una protección efectiva contra ataques de denegación de servicio (DoS), ataques de fuerza bruta y escaneos malintencionados en los sitios web. Este módulo es especialmente útil para mitigar intentos de saturar el servidor con solicitudes repetitivas o para prevenir accesos no autorizados que intentan explotar vulnerabilidades mediante ataques automatizados.

El funcionamiento de **Mod_evasive** se basa en la detección y bloqueo de patrones de solicitudes sospechosas. Por ejemplo, si una misma IP realiza demasiadas peticiones en un corto periodo de tiempo, **Mod_evasive** identifica esta actividad como maliciosa y bloquea temporalmente dicha IP. Además, este módulo puede enviar notificaciones al administrador del sistema o a un script externo para reaccionar de forma inmediata frente a posibles amenazas.

Gracias a su simplicidad y eficiencia, **Mod_evasive** es una herramienta esencial para proteger sitios web contra amenazas que podrían comprometer la disponibilidad del servidor. Este módulo no solo mejora la seguridad general del sistema, sino que también ayuda a mantener la estabilidad y el rendimiento del servidor al prevenir sobrecargas derivadas de ataques de tipo DoS.

Para instalar **Mod_evasive** en un servidor Apache, utilizaremos el siguiente comando en la terminal y realizaremos la instalación correspondiente a nuestras necesidades:

```
sudo apt-get install libapache2-mod-evasive
```

Esto nos llevará a una pantalla de configuración e instalación para habilitar el envío de correos con los logs generados por el módulo. En este caso, seleccionaremos la opción "local only", ya que no se contempla la creación de un servidor Postfix. Sin embargo, si ya se cuenta con uno configurado, se puede integrar y personalizar desde esta misma pantalla, dentro de esta opción nos pedirá un nombre de dominio para los correos, se recomienda dejar por defecto.

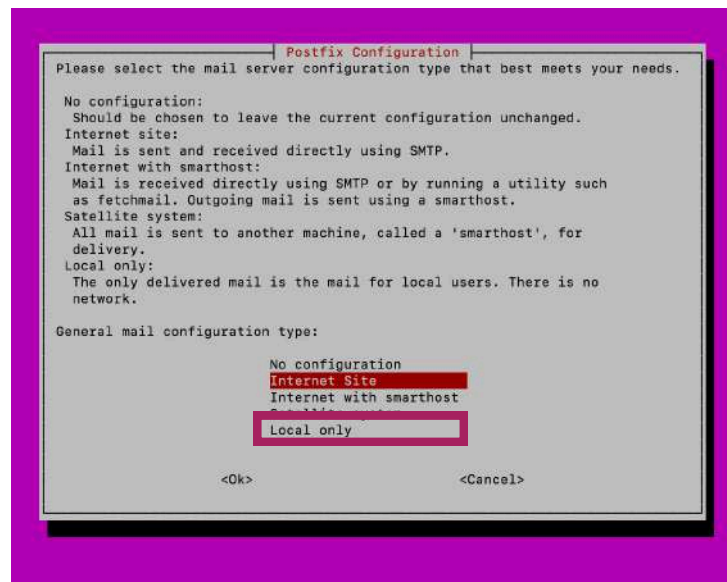
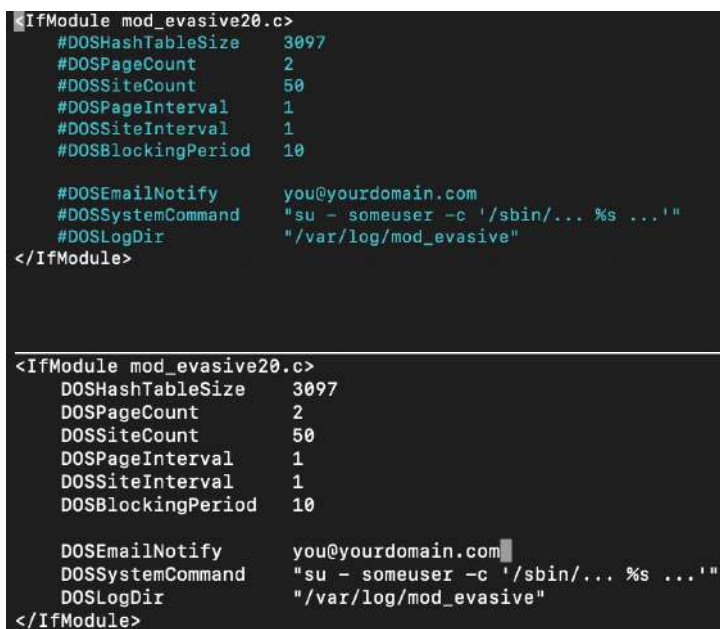


Figura 1.24. Configuración de Postfix

Para habilitar las protecciones de ModEvasive, será necesario activar las reglas que este módulo proporciona. Esto se logra editando su archivo de configuración principal, llamado `evasive.conf`. Puedes acceder y modificar este archivo utilizando el siguiente comando:

```
sudo nano /etc/apache2/mods-available/evasive.conf
```


Por defecto, el archivo **evasive.conf** tiene sus reglas comentadas, lo que significa que están inactivas. Para habilitarlas, simplemente debemos quitar los caracteres de comentario (#) al inicio de cada línea con configuración válida. Así, el archivo `evasive.conf` debería verse como se muestra a continuación:



```
<IfModule mod_evasive20.c>
#DOSHHashTableSize 3097
#DOSPageCount 2
#DOSSiteCount 50
#DOSPageInterval 1
#DOSSiteInterval 1
#DOSBlockingPeriod 10

#DOSEmailNotify you@yourdomain.com
#DOSSystemCommand "su - someuser -c '/sbin/... %s ...'"
#DOSLogDir "/var/log/mod_evasive"
</IfModule>

<IfModule mod_evasive20.c>
DOSHashTableSize 3097
DOSPageCount 2
DOSSiteCount 50
DOSPageInterval 1
DOSSiteInterval 1
DOSBlockingPeriod 10

DOSEmailNotify you@yourdomain.com
DOSSystemCommand "su - someuser -c '/sbin/... %s ...'"
DOSLogDir "/var/log/mod_evasive"
</IfModule>
```

Figura 1.25. evasive.conf sin comentar

Por defecto, el archivo `evasive.conf` almacena los registros en la carpeta `/var/log/mod_evasive`. Sin embargo, esta carpeta no se crea automáticamente durante la instalación del módulo, por lo que es necesario crearla manualmente y asignarle los permisos adecuados para que el usuario que ejecuta Apache pueda acceder y escribir en ella. Esto se puede lograr utilizando los siguientes comandos:

```
sudo mkdir -p /var/log/mod_evasive
```

```
sudo chown -R www-data:www-data /var/log/mod_evasive
```

```
sudo chmod -R 755 /var/log/mod_evasive
```

Parámetros principales de protección:

- **DOSHashTableSize:** Define el tamaño de la tabla hash utilizada para rastrear las solicitudes. Un tamaño más grande permite un monitoreo más preciso de tráfico, pero consume más memoria.
- **DOSPageCount:** Establece el número máximo de solicitudes permitidas a una misma página dentro del intervalo definido por **DOSPageInterval**. Superar este límite activa una acción de bloqueo.
- **DOSSiteCount:** Define el número máximo de solicitudes permitidas desde una dirección IP al sitio completo durante el intervalo especificado en **DOSSiteInterval**.
- **DOSPageInterval** y **DOSSiteInterval:** Intervalos de tiempo en segundos que determinan cómo se mide el tráfico para páginas individuales (**DOSPageInterval**) y el sitio completo (**DOSSiteInterval**).
- **DOSBlockingPeriod:** Duración del bloqueo (en segundos) impuesto a una dirección IP que exceda los límites establecidos.

Estos parámetros, si no son configurados adecuadamente, pueden bloquear solicitudes legítimas que no necesariamente constituyen un ataque. Por este motivo, es fundamental ajustar cada uno de ellos según las necesidades específicas de cada servidor y el tipo de tráfico que maneja.

Se recomienda realizar pruebas y monitorear el comportamiento del sistema después de aplicar los cambios para garantizar que la configuración no afecte negativamente a los usuarios legítimos mientras se mantiene la protección contra posibles ataques.

Este manual ofrece una guía detallada y práctica para la instalación, configuración y gestión del servidor web Apache en sistemas Ubuntu. Desde los pasos básicos de instalación hasta aspectos más avanzados como la implementación de servidores virtuales, configuración de módulos adicionales y ajustes de seguridad mediante firewall y sistemas de prevención de intrusiones, cada sección está diseñada para garantizar un funcionamiento eficiente y seguro del servidor.

Con los procedimientos establecidos en este manual, el servidor queda completamente configurado para recibir tráfico de manera segura desde Internet. Para futuros proyectos, será suficiente con seguir los pasos indicados para la creación de nuevos servidores virtuales y realizar los ajustes necesarios en las reglas de seguridad de UFW (Uncomplicated Firewall) y WAF (Web Application Firewall), adaptándolos a las necesidades específicas de cada caso. Este enfoque asegura la escalabilidad y flexibilidad del servidor Apache frente a nuevas implementaciones.

Bibliografía:

- Apache HTTP Server Project. (s. f.). Apache HTTP Server: The Number One HTTP Server On The Internet. Fundación Apache Software. Recuperado el 30 de octubre de 2024, de <https://httpd.apache.org>
- Canonical Ltd. (s. f.). Ubuntu: The Leading Linux Operating System for Cloud, IoT, and Computers. Ubuntu. Recuperado el 30 de octubre de 2024, de <https://ubuntu.com>
- SpiderLabs. (s. f.). OWASP ModSecurity Core Rule Set. GitHub. Recuperado el 25 de noviembre de 2024, de <https://github.com/SpiderLabs/owasp-modsecurity-crs>
- Santidediego. (s. f.). Instalación y configuración de mod-evasive. GitHub. Recuperado el 25 de noviembre de 2024, de <https://github.com/santidediego/swap1415/blob/master/Asegurar%20un%20servidor%20Apache/Instalaci3n%20y%20configuraci3n%20de%20mod-evasive.md>
- ModSecurity Project. (s. f.). ModSecurity: Open Source Web Application Firewall. ModSecurity. Recuperado el 25 de noviembre de 2024, de <https://modsecurity.org>
- Galindo Reyes, D. A. (Ed.). (s.f.). Manual de Nextcloud: Instalación y funcionamiento. Instituto de Ciencias Aplicadas y Tecnología (ICAT), UNAM.
- Galindo Reyes, D. A. (Ed.). (s.f.). Manual de Ubuntu: Instalación y funcionamiento. Instituto de Ciencias Aplicadas y Tecnología (ICAT), UNAM.