“Apuntes M8”

# **Instalación de la MV:**

Para crear una máquina virtual de Ubuntu y obtener privilegios de superusuario (root), sigue estos pasos generales:

\*\*Paso 1: Instala un software de virtualización\*\*

Puedes usar herramientas como VirtualBox o VMware para crear tu máquina virtual. Aquí te mostraré los pasos usando VirtualBox, que es de uso gratuito y bastante popular.

1. Descarga e instala VirtualBox desde su [sitio web](https://www.virtualbox.org/).

\*\*Paso 2: Descarga la imagen de Ubuntu\*\*

2. Descarga una imagen ISO de Ubuntu desde su [sitio web oficial](https://ubuntu.com/download/desktop).

\*\*Paso 3: Crea una nueva máquina virtual en VirtualBox\*\*

3. Abre VirtualBox y haz clic en "Nuevo" para crear una nueva máquina virtual. Sigue el asistente y configura los detalles como el nombre, el tipo y la versión del sistema operativo (Ubuntu).

4. Asigna memoria RAM a tu máquina virtual. Se recomiendan al menos 2 GB para un funcionamiento fluido.

5. Crea un disco duro virtual o utiliza uno existente. Configura su tamaño según tus necesidades.

6. Selecciona la imagen ISO de Ubuntu que descargaste como el medio de instalación.

7. Completa el asistente y haz clic en "Iniciar" para iniciar la máquina virtual.

\*\*Paso 4: Instala Ubuntu en la máquina virtual\*\*

8. Sigue las instrucciones del instalador de Ubuntu. Cuando llegues a la pantalla de particionamiento, puedes elegir la opción de instalar Ubuntu junto con Fedora o personalizar la configuración de partición según tus necesidades.

9. Completa la instalación y reinicia la máquina virtual.

\*\*Paso 5: Obtén privilegios de superusuario (root)\*\*

10. Una vez que Ubuntu esté instalado y hayas iniciado sesión, abre una terminal.

11. Para obtener privilegios de superusuario, utiliza el siguiente comando y proporciona tu contraseña de usuario:

```bash

sudo su

```

También puedes usar `sudo` antes de comandos individuales para ejecutarlos con privilegios de superusuario.

Ahora deberías tener acceso de superusuario en tu máquina virtual de Ubuntu. Recuerda utilizar estos privilegios con precaución, ya que pueden afectar el sistema. ¡Buena suerte con tus ejercicios de Apache2!

# **Deshabilitar FireWall:**

Deshabilitar el firewall puede dejar tu sistema vulnerable a amenazas de seguridad, por lo que es una acción que se debe realizar con precaución y solo en entornos controlados. Si estás realizando ejercicios o pruebas locales y necesitas deshabilitar el firewall temporalmente, puedes hacerlo. Sin embargo, ten en cuenta que en un entorno de producción, es esencial tener configurado un firewall adecuado para proteger tu sistema.

Para deshabilitar el firewall en Ubuntu, puedes seguir estos pasos:

1. \*\*Verificar el estado del firewall:\*\*

Abre una terminal y ejecuta el siguiente comando para verificar si UFW (Uncomplicated Firewall) está instalado y en ejecución:

```bash

sudo ufw status

```

Si UFW está activo, deberías ver un mensaje que indica que está habilitado.

2. \*\*Deshabilitar UFW:\*\*

Para deshabilitar UFW, puedes utilizar el siguiente comando:

```bash

sudo ufw disable

```

Esto desactivará el firewall y detendrá sus reglas.

3. \*\*Verificar el estado nuevamente:\*\*

Para asegurarte de que el firewall esté desactivado, ejecuta el comando de estado nuevamente:

```bash

sudo ufw status

```

Deberías ver un mensaje que indica que UFW está inactivo.

Recuerda que, después de realizar tus pruebas, es recomendable volver a habilitar el firewall para garantizar la seguridad de tu sistema:

```bash

sudo ufw enable

```

Y nuevamente verifica su estado:

```bash

sudo ufw status

```

Ten en cuenta que deshabilitar el firewall puede exponer tu sistema a posibles amenazas, así que úsalo con precaución y solo en situaciones controladas.

# **Comandos:**

**apt-get update**: Actualizar los repositorios

**apt-get upgrade**: Actualizar los paquetes

**sudo apt-get install apache2**: Instalar apache

**http://<ip\_del\_server>** (Esto no es un comando, pero es la forma de visualizar el funcionamiento de la web, al terminar la instalación)

Para deshabilitar el delay que aparece al inicio de la máquina virtual, ejecute los siguientes pedidos:

**sudo systemctl disable systemd-networkd-wait-online.service**

**sudo systemctl mask systemd-networkd-wait-online.service**

Los principales archivos públicos del servidor web están en **/var/www/html**

**ls -la /var/www/html:** Para visualizar los principales archivos públicos del servidor web

Per instal·lar PHP a la darrera versió ho podem fer amb:

**sudo apt-get install libapache2-mod-ph**

**sudo service apache2 restart:** Reiniciar el servidor, luego de hacer cambios.

# 

# **Desactivar la generación automática de índices de directorios en Apache.**

Puedes utilizar un archivo `.htaccess` en el directorio específico. Pasos:

1. Crea un archivo llamado `.htaccess` si no existe en el directorio donde deseas desactivar los índices.

2. Abre el archivo `.htaccess` con un editor de texto y agrega la siguiente línea:

```

Options -Indexes

```

Esta línea le indica a Apache que desactive la generación automática de índices en ese directorio.

3. Guarda el archivo y ciérralo.

4. Asegúrate de que el archivo `.htaccess` esté ubicado en el directorio donde deseas desactivar los índices.

Con estos pasos, la opción de índices debería desactivarse en el directorio específico. Si Apache encuentra un archivo `index.html` en ese directorio, lo mostrará en lugar de la lista de archivos del directorio.

Ten en cuenta que la capacidad de usar archivos `.htaccess` para modificar la configuración del servidor debe estar habilitada en la configuración del servidor. Además, es una buena práctica revisar el registro de errores del servidor después de realizar cambios para asegurarte de que no haya problemas.

# **Habilitar la capacidad de usar archivos `.htaccess`**

Y permitir configuraciones específicas en directorios, debes asegurarte de que la directiva `AllowOverride` esté configurada correctamente en la configuración del servidor Apache. Esta directiva determina qué tipo de configuraciones pueden ser sobrescritas por archivos `.htaccess`. Aquí te dejo los pasos generales:

1. Edita el archivo de configuración de Apache:

La configuración principal de Apache generalmente se encuentra en archivos como `httpd.conf` o en archivos de configuración de sitios virtuales, como `000-default.conf` o `default-ssl.conf`. Puedes editar estos archivos con un editor de texto como `nano` o `vi`.

```bash

sudo nano /etc/apache2/httpd.conf

```

O en algunos sistemas:

```bash

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

```

2. \*\*Encuentra la sección del directorio:\*\*

Busca la sección correspondiente al directorio en el que deseas habilitar la capacidad de usar archivos `.htaccess`. Esto puede estar dentro de una etiqueta `<Directory>`.

3. \*\*Asegúrate de que `AllowOverride` esté configurado correctamente:\*\*

Asegúrate de que la directiva `AllowOverride` permita al menos la opción `FileInfo` o `All` para el directorio en cuestión. Esto permite la configuración de índices y otras opciones en archivos `.htaccess`. Aquí hay un ejemplo:

```apache

<Directory /ruta/a/tu/directorio>

Options Indexes FollowSymLinks

AllowOverride FileInfo

Require all granted

</Directory>

```

O para permitir todas las configuraciones posibles:

```apache

<Directory /ruta/a/tu/directorio>

Options Indexes FollowSymLinks

AllowOverride All

Require all granted

</Directory>

```

Asegúrate de cambiar `/ruta/a/tu/directorio` por la ruta real de tu directorio.

4. \*\*Reinicia Apache:\*\*

Después de realizar cambios en la configuración, reinicia Apache para que los cambios surtan efecto.

```bash

sudo service apache2 restart

```

Ahora, los archivos `.htaccess` en el directorio especificado deberían tener efecto en la configuración del servidor.

Recuerda que permitir todas las configuraciones a través de `.htaccess` puede tener implicaciones de seguridad, así que solo otorga los permisos necesarios para tus necesidades específicas.

# **Añadir usuarios a un grupo:**

En sistemas Unix o Linux, puedes asignar varios propietarios a un grupo utilizando la herramienta `chown`. Sin embargo, es importante destacar que un archivo o directorio puede tener solo un propietario principal. El grupo, por otro lado, puede tener varios miembros.

Aquí hay un enfoque paso a paso:

1. \*\*Crea un grupo:\*\*

Si aún no tienes un grupo creado para los propietarios que deseas agregar, puedes hacerlo con el siguiente comando:

```bash

sudo addgroup mi\_grupo

```

Reemplaza "mi\_grupo" con el nombre que desees para el grupo.

2. \*\*Agrega usuarios al grupo:\*\*

Puedes agregar usuarios al grupo recién creado utilizando el siguiente comando:

```bash

sudo usermod -aG mi\_grupo usuario1

sudo usermod -aG mi\_grupo usuario2

```

Reemplaza "usuario1" y "usuario2" con los nombres de los usuarios que deseas agregar al grupo. El argumento `-aG` agrega el usuario al grupo sin eliminarlo de otros grupos.

3. \*\*Asigna el grupo a un directorio o archivo:\*\*

Luego, puedes asignar el grupo recién creado a un directorio o archivo utilizando `chown`. Por ejemplo:

```bash

sudo chown :mi\_grupo /ruta/al/directorio

```

Este comando asigna el grupo "mi\_grupo" al directorio. Todos los usuarios que son miembros de "mi\_grupo" tendrán acceso y permisos sobre ese directorio.

4. \*\*Asegúrate de que los permisos sean adecuados:\*\*

Además de asignar el grupo, asegúrate de que los permisos del archivo o directorio permitan la lectura, escritura y ejecución según tus necesidades específicas.

```bash

sudo chmod 770 /ruta/al/directorio

```

En este ejemplo, se otorgan permisos completos al propietario y al grupo, y se niegan todos los permisos a otros.

Recuerda que estos comandos pueden variar según la distribución de Linux que estés utilizando. Además, ten en cuenta las implicaciones de seguridad al otorgar permisos y asegúrate de seguir las mejores prácticas de seguridad para tu sistema.

# **Eliminar usuarios:**

Para eliminar un usuario en sistemas basados en Linux, puedes seguir estos pasos. Ten en cuenta que debes tener privilegios de administrador (puedes usar el comando `sudo`).

Supongamos que deseas eliminar el usuario "usuario\_a\_eliminar". Sigue estos pasos:

1. \*\*Elimina el usuario:\*\*

```bash

sudo deluser usuario\_a\_eliminar

```

2. \*\*Elimina el directorio de inicio (opcional):\*\*

Si deseas eliminar el directorio de inicio del usuario, puedes usar el siguiente comando:

```bash

sudo rm -r /home/usuario\_a\_eliminar

```

Ten en cuenta que esta acción eliminará todos los archivos y carpetas en el directorio de inicio del usuario.

3. \*\*Elimina la carpeta del correo del usuario (opcional):\*\*

Si el usuario tiene un correo asociado, puedes eliminar la carpeta de correo (por lo general, ubicada en `/var/mail`):

```bash

sudo rm /var/mail/usuario\_a\_eliminar

```

4. \*\*Elimina las entradas del usuario en los archivos de sistema:\*\*

Puedes eliminar las entradas del usuario en algunos archivos de sistema, como `/etc/passwd` y `/etc/shadow`. Ten en cuenta que realizar cambios directos en estos archivos puede ser riesgoso, así que ten precaución.

```bash

sudo nano /etc/passwd

sudo nano /etc/shadow

```

Busca y elimina las líneas que contienen el nombre del usuario.

5. \*\*Elimina las entradas del grupo del usuario (opcional):\*\*

Si el usuario tenía su propio grupo, puedes eliminarlo con el siguiente comando:

```bash

sudo delgroup usuario\_a\_eliminar

```

Recuerda que al realizar estas acciones, estás eliminando permanentemente al usuario y, si eliges eliminar el directorio de inicio, todos los archivos y carpetas dentro de ese directorio. Asegúrate de tener una copia de seguridad o de estar seguro de que no necesitas la información almacenada en ese usuario antes de proceder.

# **Para que cada usuario pueda tener su sitio web y que no interfiera al de los demás usuarios:**

Si estás configurando un servidor web donde cada usuario puede tener su propio sitio web sin interferir con los demás, es posible que desees utilizar la funcionalidad de usuarios virtuales o configurar entornos de usuario separados. A continuación, te proporciono una guía básica utilizando Apache y el módulo `mod\_userdir`, que te permite asignar un directorio personal para cada usuario.

1. \*\*Asegúrate de que `mod\_userdir` esté habilitado:\*\*

En la mayoría de las instalaciones de Apache, `mod\_userdir` está disponible pero puede no estar habilitado por defecto. Puedes habilitarlo ejecutando:

```bash

sudo a2enmod userdir

```

Luego, reinicia Apache:

```bash

sudo service apache2 restart

```

2. \*\*Configura el directorio principal de cada usuario:\*\*

Crea un directorio principal para cada usuario en algún lugar de tu sistema. Por ejemplo:

```bash

sudo mkdir /var/www/user1

sudo mkdir /var/www/user2

```

3. \*\*Asigna permisos a los directorios de usuario:\*\*

Asegúrate de que los usuarios de Linux tengan acceso al directorio de su sitio web personal:

```bash

sudo chown -R user1:user1 /var/www/user1

sudo chown -R user2:user2 /var/www/user2

```

4. \*\*Crea un archivo `.html` para cada usuario:\*\*

Dentro de cada directorio de usuario, puedes crear un archivo `index.html` predeterminado o permitir que cada usuario coloque su propio contenido.

5. \*\*Accede a los sitios personales:\*\*

Una vez que hayas configurado todo, los sitios web personales de los usuarios estarán disponibles en la siguiente URL:

```

http://tu-dirección-ip/~nombre-de-usuario

```

Por ejemplo:

```

http://tu-dirección-ip/~user1

http://tu-dirección-ip/~user2

```

Los usuarios pueden cargar sus archivos en sus respectivos directorios personales y acceder a sus sitios web utilizando estas URL personalizadas.

Este enfoque proporciona a cada usuario su propio espacio separado y permite que gestionan sus sitios web de forma independiente. Ten en cuenta que esto puede requerir ajustes adicionales dependiendo de tu configuración específica y del sistema operativo que estés utilizando.

# **Práctica 1:**

## **Act 1:**

Entendido, el ejercicio implica realizar varias acciones en un servidor web Apache en un entorno Linux. Aquí tienes una guía paso a paso para realizar esas tareas:

### 1. Poner un mensaje de bienvenida personalizado como usuario root:

```bash

sudo nano /var/www/html/index.html

```

Añade un mensaje de bienvenida personalizado al archivo HTML. Guarda los cambios.

### 2. Modificar el archivo sin actuar como root:

Primero, crea el grupo "web":

```bash

sudo addgroup web

```

Luego, otorga permisos adecuados a la carpeta `/var/www/html`:

```bash

sudo chown -R root:web /var/www/html

sudo chmod -R 775 /var/www/html

```

Esto permite al usuario root y a los miembros del grupo "web" leer, escribir y ejecutar en el directorio.

Añade al usuario principal al grupo "web":

```bash

sudo usermod -aG web tu\_usuario

```

### 3. Sal de la sesión y vuelve a entrar:

```bash

exit

```

Vuelve a entrar y asegúrate de que el usuario ahora es miembro del grupo "web".

### 4. Crea una nueva página HTML:

```bash

nano /var/www/html/nueva\_pagina.html

```

Añade algún texto a la nueva página HTML y guárdala.

### 5. Visualiza la página con el navegador:

Abre tu navegador y accede a la nueva página en la siguiente URL:

```

http://tu-direccion-ip/nueva\_pagina.html

```

Asegúrate de que puedes ver el contenido de la nueva página.

### Informe:

En tu informe, proporciona una explicación detallada de las comandos utilizados y justifica los permisos asignados a la carpeta `/var/www/html`. Puedes explicar que se asignaron permisos 775 para permitir lectura, escritura y ejecución al propietario y al grupo, pero solo permisos de lectura y ejecución para otros usuarios. Además, menciona que el propietario es root y el grupo es "web", permitiendo que los usuarios del grupo "web" realicen cambios en el directorio.

## **Act 2:**

Para completar el Ejercicio 2, sigue estos pasos:

### 1. Cambia al directorio `/var/www/html`:

```bash

cd /var/www/html

```

### 2. Crea un archivo "prova.php" con el contenido dado:

```bash

nano prova.php

```

Agrega el siguiente contenido al archivo y guarda los cambios:

```php

<html>

<body>

<?php

phpinfo();

?>

</body>

</html>

```

### 3. Visualiza el archivo con el navegador:

Abre tu navegador y accede al siguiente URL:

```

http://tu-direccion-ip/prova.php

```

Asegúrate de que puedes ver la información de PHP en la página.

### 4. Captura de pantalla:

Toma una captura de pantalla que muestre la información de PHP y el escritorio. Asegúrate de que la captura de pantalla sea clara y no recortada.

### Informe:

En tu informe, describe brevemente los pasos que seguiste para crear y visualizar el archivo "prova.php". Incluye la captura de pantalla y asegúrate de que sea visible el escritorio sin recortes para confirmar que todo el contenido está siendo mostrado.

# **Práctica 2:**

## **Act 1:**

Entendido. Para cambiar el nombre de `index.html` a `index2.html` y observar el efecto en la visualización de la carpeta con Apache, sigue estos pasos:

### 1. Cambia al directorio `/var/www/html`:

```bash

cd /var/www/html

```

### 2. Cambia el nombre de `index.html` a `index2.html`:

```bash

mv index.html index2.html

```

### 3. Visualiza el directorio en el navegador:

Accede al directorio en tu navegador:

```

http://tu-direccion-ip/

```

### Informe:

En tu informe, describe el efecto de cambiar el nombre de `index.html` a `index2.html`. Observa si ahora se muestra automáticamente el contenido del directorio en lugar de cargar una página predeterminada. Incluye una captura de pantalla que muestre cómo se ve la visualización del directorio en el navegador después de realizar este cambio. Explica cualquier observación sobre la visualización y cómo cambió con el nuevo nombre del archivo.

## **Act 2:**

Entendido, parece que el ejercicio implica desactivar la generación de índices (listas de archivos) para toda la web y luego activarla solo para ciertos directorios específicos utilizando un archivo `.htaccess`. Aquí tienes una guía paso a paso:

### Desactivar índices para toda la web:

1. \*\*Accede al archivo de configuración del sitio:\*\*

Abre el archivo de configuración del sitio virtual. Por lo general, se encuentra en la carpeta `/etc/apache2/sites-available/`.

```bash

sudo nano /etc/apache2/sites-available/tu-sitio.conf

```

Reemplaza "tu-sitio.conf" con el nombre real del archivo de configuración de tu sitio.

2. \*\*Busca la sección `<Directory>`:\*\*

Dentro de la sección `<Directory>` correspondiente al directorio raíz de tu sitio, agrega o modifica la línea que contiene `Options` para desactivar los índices.

```apache

<Directory /ruta/a/tu/directorio>

Options -Indexes

# ...otras configuraciones...

</Directory>

```

Guarda los cambios y sale del editor.

3. \*\*Reinicia Apache:\*\*

```bash

sudo service apache2 restart

```

### Desactivar índices para un directorio específico usando `.htaccess`:

1. \*\*Crea un archivo `.htaccess` en el directorio "fotos":\*\*

```bash

cd /ruta/a/tu/directorio/fotos

nano .htaccess

```

Agrega la siguiente línea para desactivar los índices solo para este directorio:

```apache

Options -Indexes

```

Guarda los cambios y sale del editor.

2. \*\*Asegúrate de que `AllowOverride` esté configurado correctamente:\*\*

Verifica que el archivo de configuración del sitio permita el uso de archivos `.htaccess`. En el archivo de configuración del sitio, dentro de la sección `<Directory>`, debe haber una línea similar a la siguiente:

```apache

AllowOverride All

```

Si la línea no está presente o está configurada como `None`, cámbiala y guarda los cambios.

3. \*\*Reinicia Apache:\*\*

```bash

sudo service apache2 restart

```

### Resultado Final:

- Los índices están desactivados para toda la web.

- Los índices están desactivados para el directorio "fotos".

### Capturas de pantalla y Documentación:

Toma capturas de pantalla que muestren la configuración del sitio y los archivos `.htaccess`, y documenta cada paso realizado en tu informe. Explica las razones detrás de cada cambio y cómo afecta la seguridad y la visibilidad de los archivos en la web.

## **Act 3:**

Entendido, para activar el módulo "userdir" y realizar el ejercicio, sigue estos pasos:

### 1. Activa el módulo "userdir":

```bash

sudo a2enmod userdir

```

### 2. Crea la carpeta del usuario principal:

Voy a utilizar "funky" como ejemplo de usuario principal. Puedes cambiarlo según sea necesario.

```bash

mkdir /home/funky/public\_html

```

### 3. Agrega una página HTML al directorio del usuario:

```bash

echo "<html><body><h1>Benvingut a la meva pàgina!</h1></body></html>" > /home/funky/public\_html/index.html

```

### 4. Reinicia Apache:

```bash

sudo service apache2 restart

```

### 5. Visualiza la página en el navegador:

Accede a la página del usuario principal en tu navegador:

```

http://tu-direccion-ip/~funky

```

### 6. Captura de pantalla:

Toma una captura de pantalla que muestre la página HTML creada y cómo se ve en el navegador.

### Informe:

En tu informe, describe cada paso realizado, explicando cómo activaste el módulo "userdir", creaste la carpeta del usuario principal, agregaste una página HTML y mostraste la página en el navegador. Incluye la captura de pantalla en tu informe para ilustrar el resultado.

## **Act 4:**

Entendido, para activar PHP en los directorios de usuarios (userdirs) en Apache, sigue estos pasos:

### 1. Encuentra el artículo que explique cómo activar PHP en userdirs:

Puedes buscar información específica sobre cómo activar PHP en userdirs. Aquí proporcionaré una guía básica basada en el archivo de configuración de PHP para Apache.

### 2. Abre el archivo de configuración de PHP para la versión 7.0:

```bash

sudo nano /etc/apache2/mods-available/php7.0.conf

```

### 3. Busca y modifica las líneas necesarias:

Dentro del archivo, busca líneas que contengan `php\_admin\_flag engine` y `php\_admin\_value`.

Asegúrate de que estas líneas estén configuradas de la siguiente manera:

```apache

<IfModule mod\_php7.c>

<FilesMatch "\.php$">

SetHandler application/x-httpd-php

</FilesMatch>

<FilesMatch "\.phps$">

SetHandler application/x-httpd-php-source

</FilesMatch>

# To active user's PHP in their public\_html directories

php\_admin\_flag engine On

</IfModule>

```

### 4. Reinicia Apache:

```bash

sudo service apache2 restart

```

### 5. Comprueba PHP en userdirs:

Abre el navegador y accede a la página de prueba PHP en userdirs:

```

http://tu-direccion-ip/~funky/prova.php

```

### 6. Captura de pantalla:

Toma una captura de pantalla que muestre la página de prueba PHP y cómo se ve en el navegador.

### 7. Añade el artículo a la webgrafía del informe:

Incluye el enlace al artículo que utilizaste para realizar la configuración en la sección de webgrafía de tu informe.

Recuerda que los nombres de archivos y las ubicaciones pueden variar según la configuración específica de tu sistema. Asegúrate de adaptar los comandos y configuraciones según tu entorno.

## **Act 5:**

Entendido, para configurar dos Virtual Hosts en Apache que apunten a las carpetas de dos usuarios diferentes, sigue estos pasos:

### 1. Crea los archivos de configuración para los Virtual Hosts:

```bash

sudo nano /etc/apache2/sites-available/zeus.local.conf

```

Añade la siguiente configuración para el Virtual Host de `www.zeus.local`:

```apache

<VirtualHost \*:80>

ServerAdmin webmaster@zeus.local

ServerName www.zeus.local

DocumentRoot /home/zeus/public\_html

<Directory /home/zeus/public\_html>

Options Indexes FollowSymLinks

AllowOverride All

Require all granted

</Directory>

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>

```

Repite el proceso para el segundo dominio:

```bash

sudo nano /etc/apache2/sites-available/mercuri.local.conf

```

Añade la siguiente configuración para el Virtual Host de `www.mercuri.local`:

```apache

<VirtualHost \*:80>

ServerAdmin webmaster@mercuri.local

ServerName www.mercuri.local

DocumentRoot /home/mercuri/public\_html

<Directory /home/mercuri/public\_html>

Options Indexes FollowSymLinks

AllowOverride All

Require all granted

</Directory>

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>

```

### 2. Crea enlaces simbólicos a los Virtual Hosts habilitados:

```bash

sudo a2ensite zeus.local.conf

sudo a2ensite mercuri.local.conf

```

### 3. Reinicia Apache:

```bash

sudo service apache2 restart

```

### 4. Añade las entradas correspondientes en el archivo hosts:

Abre el archivo `/etc/hosts` con un editor de texto:

```bash

sudo nano /etc/hosts

```

Añade las siguientes líneas:

```

127.0.0.1 www.zeus.local

127.0.0.1 www.mercuri.local

```

### 5. Comprueba en el navegador:

Accede a `www.zeus.local` y `www.mercuri.local` en tu navegador y asegúrate de que cada uno apunta a la carpeta correspondiente del usuario.

### 6. Captura de pantalla:

Toma una captura de pantalla que muestre el resultado en el navegador con ambos dominios.

### Informe:

En tu informe, describe cada paso realizado, incluyendo la configuración de los Virtual Hosts, la creación de enlaces simbólicos, la modificación del archivo hosts y cómo se ve el resultado en el navegador. Asegúrate de incluir la captura de pantalla en tu informe para ilustrar el funcionamiento de los dos Virtual Hosts.

## **Act Final:**

Para realizar este ejercicio, primero debes realizar algunas tareas previas:

### 1. Crear los usuarios y las carpetas web:

```bash

sudo adduser zeus

sudo adduser mercuri

sudo adduser enric

```

### 2. Poner una página HTML de bienvenida en cada directorio de usuario:

```bash

echo "<html><body><h1>Benvingut a www.zeus.local!</h1></body></html>" > /home/zeus/public\_html/index.html

echo "<html><body><h1>Benvingut a www.mercuri.local!</h1></body></html>" > /home/mercuri/public\_html/index.html

echo "<html><body><h1>Benvingut a www.elteunom.local!</h1></body></html>" > /home/enric/public\_html/index.html

```

### 3. Configurar el archivo hosts en el CLIENTE y en el servidor:

Añade las siguientes líneas en el archivo `/etc/hosts` tanto en el CLIENTE como en el servidor:

```

127.0.0.1 www.zeus.local

127.0.0.1 www.mercuri.local

127.0.0.1 www.elteunom.local

```

### 4. Comprueba la conexión a los dominios desde el navegador:

Asegúrate de que puedes acceder a `www.zeus.local`, `www.mercuri.local` y `www.elteunom.local` desde el navegador en el CLIENTE.

### 5. Configurar los Virtual Hosts de Apache:

Crea los archivos de configuración para los Virtual Hosts:

```bash

sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/mercuri.conf

sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/enric.conf

```

Edita `mercuri.conf`:

```bash

sudo nano /etc/apache2/sites-available/mercuri.conf

```

Añade o modifica las líneas adecuadas según la configuración deseada.

Edita `enric.conf`:

```bash

sudo nano /etc/apache2/sites-available/enric.conf

```

Añade o modifica las líneas adecuadas según la configuración deseada.

### 6. Activa los Virtual Hosts y recarga Apache:

```bash

sudo a2ensite mercuri.conf

sudo a2ensite enric.conf

sudo service apache2 reload

```

### 7. Comprueba los resultados:

Asegúrate de que puedes acceder a los nuevos dominios configurados en los Virtual Hosts (`www.elteunom.local`, `www.practica.enric.local`, `www.aula.enric.local`) desde el navegador en el CLIENTE.

### 8. Colgar páginas en PHP:

Coloca una página PHP en cada directorio correspondiente y comprueba que se muestra correctamente en el navegador:

```bash

echo "<?php phpinfo(); ?>" > /home/enric/public\_html/index.php

```

Recuerda ajustar y adaptar cada paso según tus necesidades y configuración específica.