**PUESTA EN PRODUCCIÓN SEGURA**

**Práctica Apache**

Parte 4 del Proyecto

(19-04-2024)

Rubén de la Viuda Redondo

Ciberseguridad

Índice

[**1.** Partiendo del archivo VagrantFile que hay en la carpeta Archivos de la unidad arranca la máquina virtual con apache ya instalado (crear un directorio, colocar el archivo anterior dentro y ejecutar vagrant up): 3](#__RefHeading___Toc151_3309803758)

[**2.** Nos conectamos a la máquina virtual anterior (en una terminal): vagrant ssh: 3](#__RefHeading___Toc151_3309803758_Copia_1)

[**3.** Comprobamos que apache funciona: 4](#__RefHeading___Toc151_3309803758_Copia_2)

[**4.** Lista de comandos útiles: 5](#__RefHeading___Toc151_3309803758_Copia_3)

[**5.** Revisamos los archivos de configuración: 6](#__RefHeading___Toc151_3309803758_Copia_4)

[**6.** Servidor virtual por defecto: 6](#__RefHeading___Toc151_3309803758_Copia_5)

[**7.** Directivas: 8](#__RefHeading___Toc151_3309803758_Copia_6)

[**8.** Ficheros y directorios de prueba: 9](#__RefHeading___Toc151_3309803758_Copia_7)

[**9.** Opciones sobre directorios Directiva DirectoryIndex: 11](#__RefHeading___Toc151_3309803758_Copia_8)

[**10.** Opciones sobre directorios. Directivas Directory y Options Indexes: 12](#__RefHeading___Toc151_3309803758_Copia_9)

[**11.** Códigos del error. Directiva ErrorDocument: 13](#__RefHeading___Toc151_3309803758_Copia_a)

[**12.** Directorios Virtuales. Directiva Alias: 14](#__RefHeading___Toc151_3309803758_Copia_b)

[**13.** Redirecciones. Directiva Redirect: 15](#__RefHeading___Toc151_3309803758_Copia_c)

[**14.** Módulos: 15](#__RefHeading___Toc151_3309803758_Copia_d)

[**15.** Directivas Order, allow y deny. Directiva Require: 18](#__RefHeading___Toc151_3309803758_Copia_e)

[**16.** Autenticación y autorización Basic y Digest: 19](#__RefHeading___Toc151_3309803758_Copia_f)

[**17.** Fichero htaccess: 20](#__RefHeading___Toc151_3309803758_Copia_g)

[**18.** Módulos mod\_status y mod\_info: 20](#__RefHeading___Toc151_3309803758_Copia_h)

[**19.** Ficheros de registros (logs): 22](#__RefHeading___Toc151_3309803758_Copia_i)

[**20.** SSL en Apache: 23](#__RefHeading___Toc151_3309803758_Copia_j)

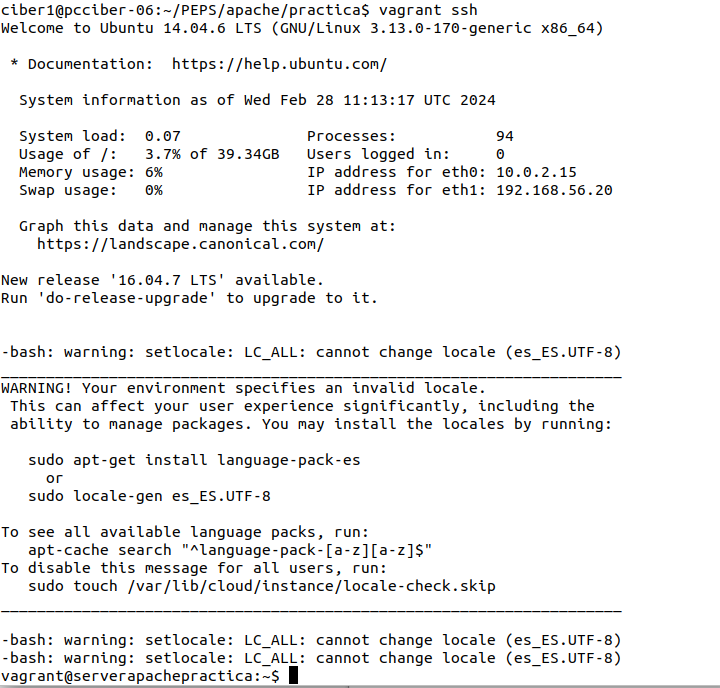
[**21.** Recomendaciones para securizar el servidor: 23](#__RefHeading___Toc151_3309803758_Copia_k)

# **1. Partiendo del archivo VagrantFile que hay en la carpeta Archivos de la unidad arranca la máquina virtual con apache ya instalado (crear un directorio, colocar el archivo anterior dentro y ejecutar vagrant up):**

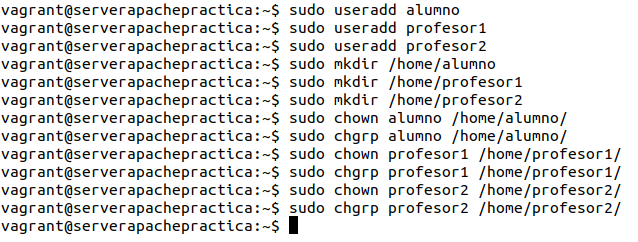
(No hace falta capturas ni información para este apartado)

# **2. Nos conectamos a la máquina virtual anterior (en una terminal): vagrant ssh:**

**Muestra un pantallazo de la consola anterior:**

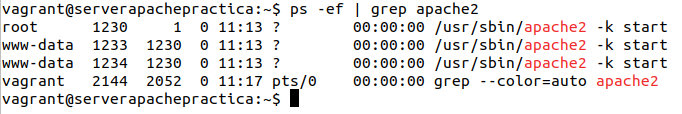


**Crea el usuario alumno, profesor, profesor1 y profesor2 . Indica la línea de comandos:**

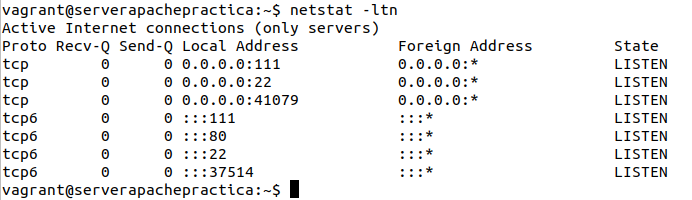


# **3. Comprobamos que apache funciona:**

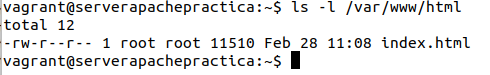
**Ejecutar el siguiente comando para ver el proceso principal de apache: ps -ef | grep apache2:**



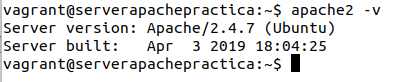
**Ejecutar el siguiente comando para ver los puertos en escucha: netstat -ltn:**



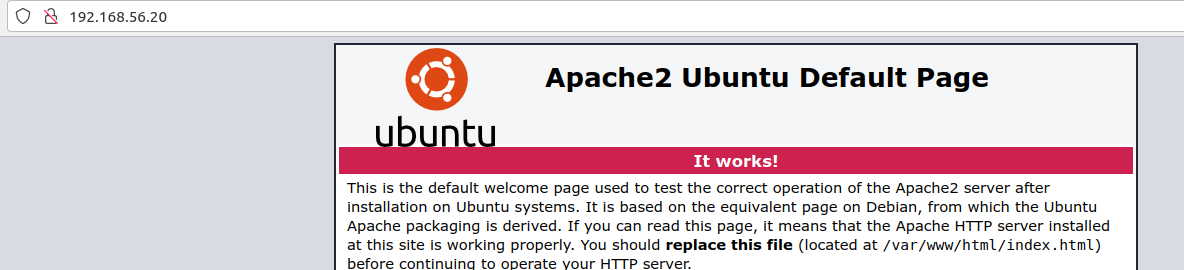
**Ejecutar el siguiente comando para ver el contenido del directorio donde se colocarán los archivos que se van a publicar: ls -l /var/www/html:**



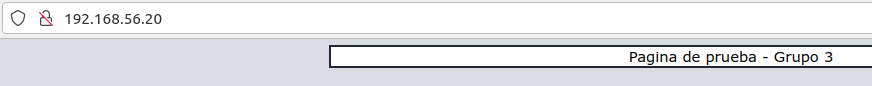
**Ejecutar el siguiente comando para ver con que versión está funcionando apache: apache2 -v:**



**Abrir un navegador y accedemos al servidor web anterior con la url: http://192.168.56.10:**



**Modifica el archivo /var/ww/html/index.html y comprueba en el navegador que se ven los cambios:**

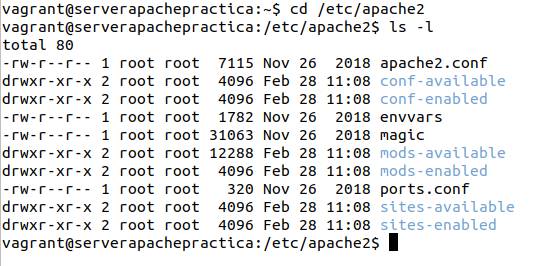


# **4. Lista de comandos útiles:**

(No hace falta capturas ni información para este apartado)

# **5. Revisamos los archivos de configuración:**

**Acceder al directorio /etc/apache2. Hacer un listado del directorio e indica los ficheros de configuración:**

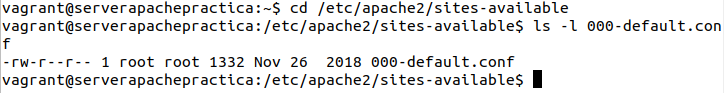


Los archivos de configuración son los que incluyen ‘conf’ en su nombre.

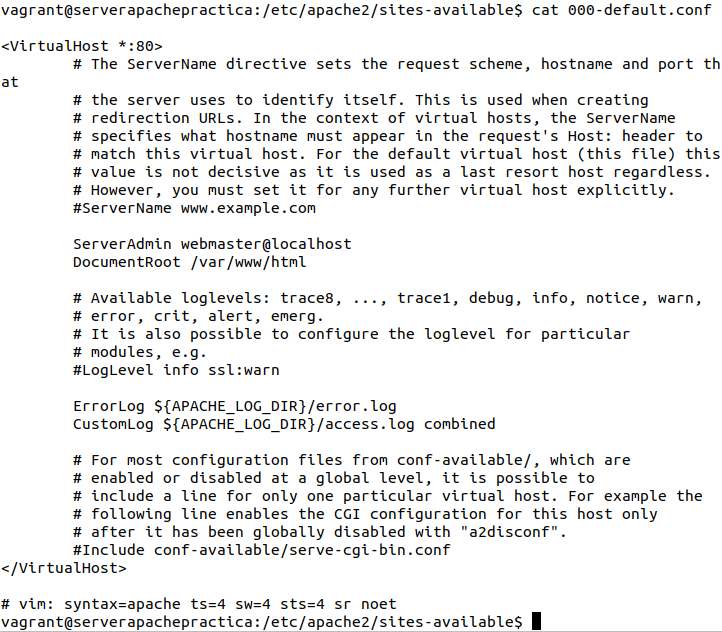
# **6. Servidor virtual por defecto:**

**Accede al directorio /etc/apache2/sites-available y comprueba que está creado el archivo 000-default. Este fichero contiene la configuración del servidor virtual por defecto. Lee brevemente su contenido. Explica que crees que se está estableciendo:**

Comprobamos que está creado:



Su contenido es el siguiente:



Está estableciendo un servidor virtual que escuche en el puerto 80 de todas sus direcciones IP. Establece el administrador del servidor virtual (webmaster de la propia máquina), cuál es la raíz de la página y cuáles son los directorios para almacenar los logs.

**Accede a /etc/apache2/sites-enabled y comprueba si existe dicho fichero 000-default.conf. ¿Existe? ¿Que aparece?**

El directorio /etc/apache2 es el directorio raíz de configuración de Apache. Almacena toda la configuración relacionada con la aplicación.

**Explica que relación existe entre los ficheros y directorios mencionados (/etc/apache2, /etc/apache2/apache2.conf, /etc/apache2/sites-available y /etc/apache2/sites-enabled)**

El archivo /etc/apache2/apache2.conf es el archivo de configuración general de Apache. Cualquier configuración que se especifique en este archivo se aplicará a todo Apache.

El directorio /etc/apache2/sites-available es el directorio donde se almacenan todas las configuraciones individuales para cada uno de los sitios web que tengamos en el servidor.

El directorio /etc/apache2/sites-enabled es el directorio donde se especifica qué sitios web están activos y deben ser tenidos en cuenta. Sus archivos son simples enlaces a los archivos de configuración en /etc/apache2/sites-available de los sitios web activos.

# **7. Directivas:**

**Consulta la documentación de Apache y responde a las siguientes preguntas:**

**¿Se permiten conexiones persistentes (que todas las conexiones de un usuario se atienden en la misma conexión TCP)? ¿Qué directiva define este comportamiento?**

Sí, es la directiva: KeepAlive On

**¿Cuál es el fichero de errores? ¿Qué directiva lo define?**

Error.log, lo define ErrorLog

**Consulta el fichero /etc/apache2/ports.conf y comprueba cual es el puerto en el que escucha peticiones el servidor Apache. ¿En qué puerto escuchara también si se habilita el módulo ssl?**

Ahora mismo escucha el puerto 80, si activas el módulo ssl también escucha en el 443

**Consulta el fichero /etc/apache2/sites-available/000-default y observa las líneas contenidas entre las directivas <VirtualHost> y  </VirtualHost>**

**El valor de la directiva DocumentRoot es /var/www/html. ¿Para qué crees que sirve esta línea?**

Es la directiva que indica en que carpeta están los ficheros que mostrará apache.

**Observa la directiva Errorlog $APACHE\_LOG........ ¿Para qué crees que sirve esta línea?**

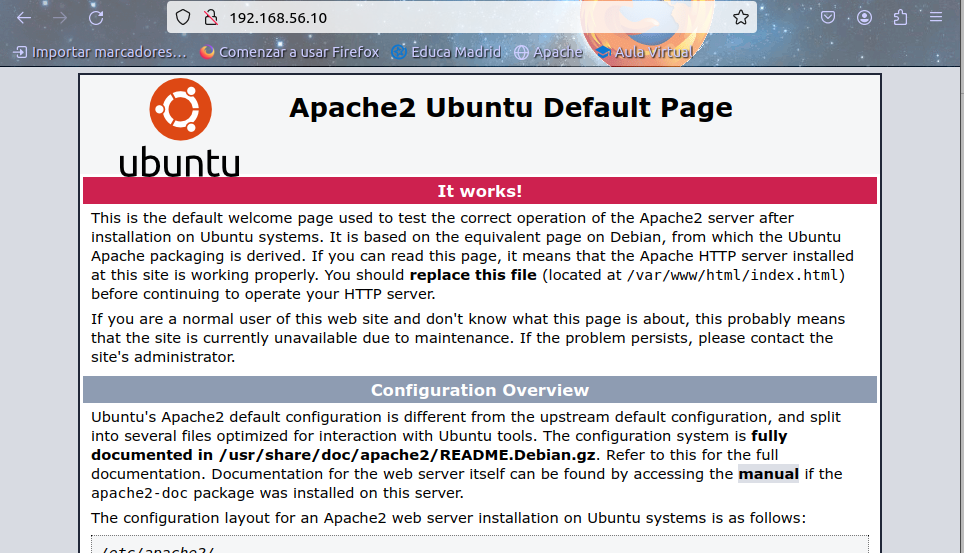
Sirve para indicar en qué log se guardan los errores producidos en apache (para los ofrecidos por 000-default)

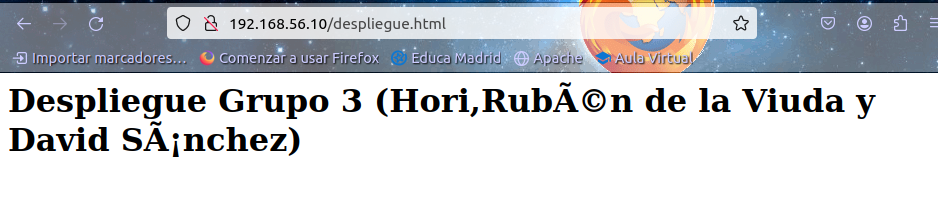
**Consulta el fichero /etc/apache2/apache2.conf. Aparecen varias secciones enmarcadas por las directivas <Directory> ...</Directory>. ¿Para qué crees que sirven estas líneas?**

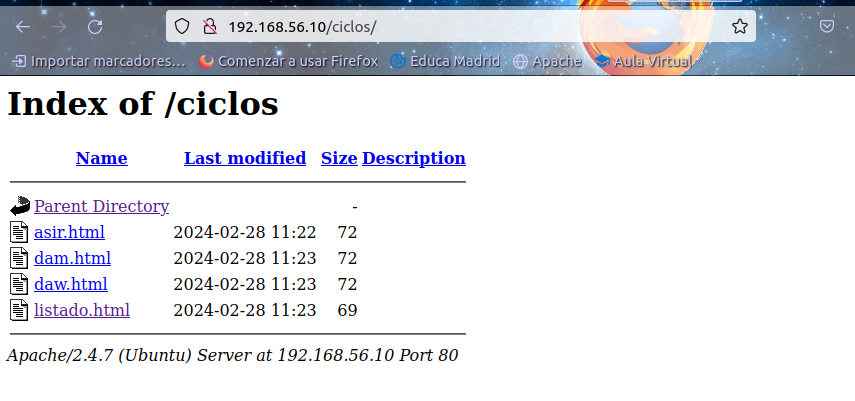
Son las directivas que marcan el funcionamiento y los permisos de los ficheros y directorios ofrecidos por 000-default

# **8. Ficheros y directorios de prueba:**

**Muestra pantallazos de cada web:**

****

****

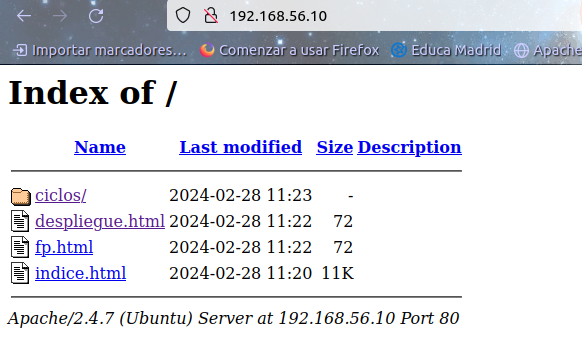
****

****

# **9. Opciones sobre directorios Directiva DirectoryIndex:**

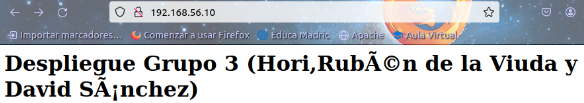
**Desde Cliente abre un navegador y establece una conexión a http://192.168.56.10**

**Renombra el fichero /var/www/html/index.html a /var/www/html/indice.html y vuelve a establecer una conexión a http://192.168.56.10 ¿Cuál es el resultado? ¿Qué se muestra en pantalla?¿Por qué?**



Muestra el directorio ya que apache está configurado para mostrar el contenido de “index.html” por defecto al conectarte a la ip, pero al no existir, te muestra todos los archivos.

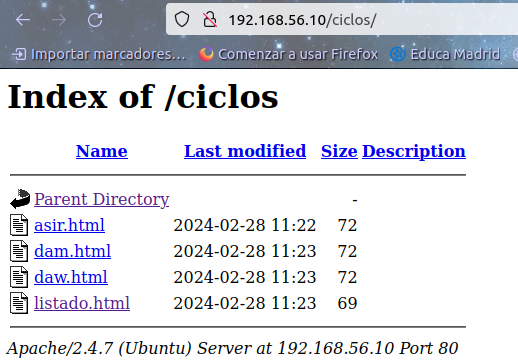
**Edita el archivo /etc/apache2/sites-available/000-default.conf y crea la sección<Directory /var/www/html> ... </Directory>. Añade la directiva DirectoryIndex con el valor despliegue.html. Reinicia el servidor para que los cambios tengan efecto. Vuelve a establecer una conexión con http://192.168.56.10 ¿Que aparece en pantalla? ¿Por qué?**



Aparece el contenido de despliegue, ya que la directiva usada indica que fichero usar como el mostrado por defecto al intentar ver el contenido de /var/www/html (La carpeta por defecto de apache2)

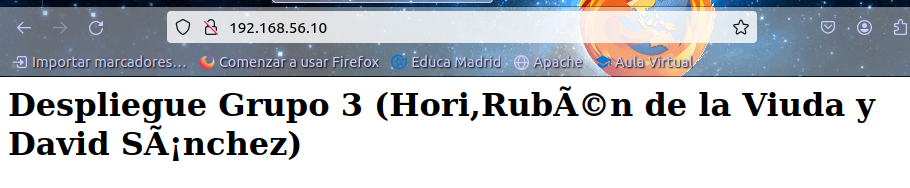
# **10. Opciones sobre directorios. Directivas Directory y Options Indexes:**

**Desde Cliente abre un navegador y establece una conexión a http://192.168.56.10/ciclos . ¿Que aparece en nuestro navegador? ¿Por qué?**

****

Aparece el sistema de archivos del directorio, ya que no hay configuraciones especificas para el directorio ciclos.

Observa que para /var/www/html/ciclos no se ha definido la opción Indexes . Reinicia el servidor y desde Cliente abre un navegador y establece una conexión a http://192.168.56.10/ . ¿Que aparece en pantalla? ¿Por qué? Explica cómo funciona esta herencia:



Esto muestra despliegue.html, ya que coge la herencia de /var/www/html y tiene como default a despliegur.html

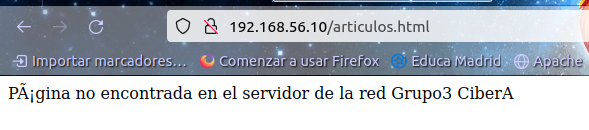
**Desde Cliente abre un navegador y establece una conexión a http://192.168.56.10/ciclos . ¿Que aparece en pantalla? ¿Por qué? Explica cómo funciona esta herencia:**



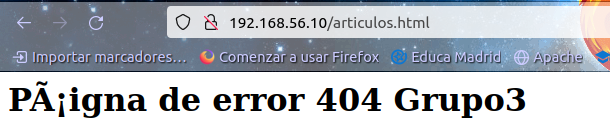
En este caso está forbiden, ya que coge los parámetros de /var/www/html/ciclos y estos prohíben el acceso a un directorio para ver los archivos de dentro, aunque si permite ver los propios archivos si los pones en la url.

# **11. Códigos del error. Directiva ErrorDocument:**

**Configura el servidor virtual por defecto para que cuando retorne el código de error 404 (pagina no encontrada) envíe el texto “Página no encontrada en el servidor de la red NombreDeTuServidor ". Reinicia el servidor para que los cambios tengan efecto. Desde Cliente abre un navegador y establece una conexión a http://192.168.56.10/articulos.html. ¿Qué aparece en pantalla?:**

****

**Configura el servidor virtual por defecto para que cuando retorne el código de error 404 (página no encontrada) envíe la página 404.html almacenada en el directorio raíz del servidor (Crea el fichero /var/www/html/404.html con el contenido que quieras). Comprueba que funciona correctamente. Muestra pantallazo de la configuración:**

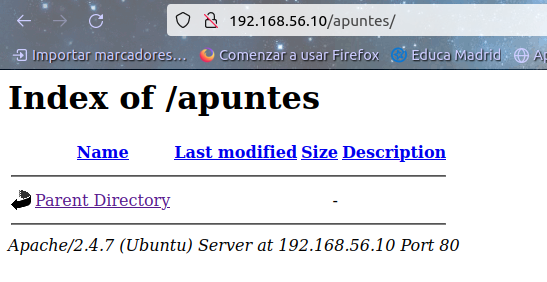
****

****

# **12. Directorios Virtuales. Directiva Alias:**

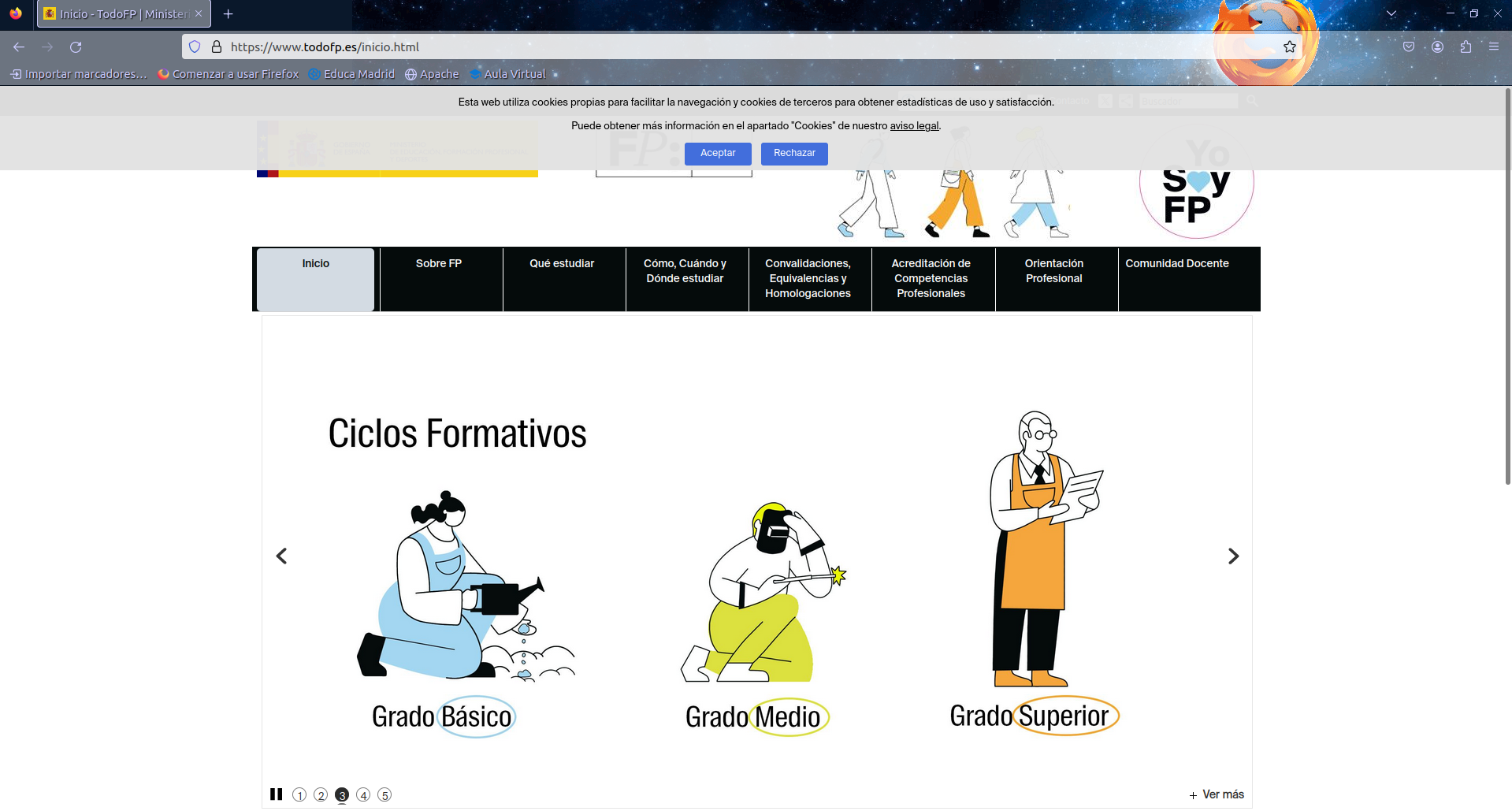
**Desde Cliente abre un navegador y establece una conexión a http://192.168.56.10/apuntes**

**Muestra un pantallazo:**



# **13. Redirecciones. Directiva Redirect:**

**Desde Cliente abre un navegador y establece una conexión a http://192.168.56.10/fp. Muestra un pantallazo:**

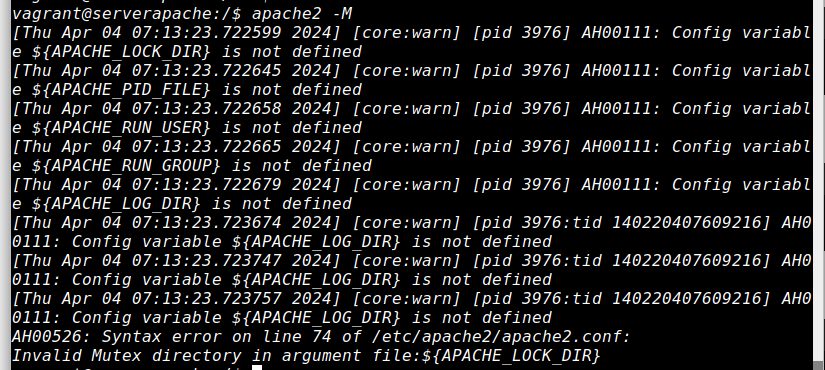
****

# **14. Módulos:**

**Consulta de Módulos cargados**

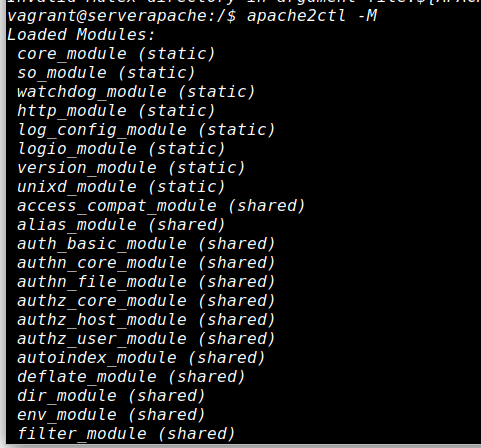
**Comprueba los módulos estáticos que se han cargado al compilar el servidor. ¿Qué comando utilizamos? ¿Cuáles son los módulos?**

apache2 -M

****

**Comprueba los módulos dinámicos que se han cargado al arrancar el servidor. ¿Qué comando utilizamos? ¿Que obtenemos? ¿Cuáles son los módulos?:**

apache2ctl -M | Obtenemos una lista de los módulos dinámicos:



**Edita uno de los ficheros .load (por ejemplo dir.load ) y observa cómo se utiliza la directiva LoadModule para cargar el módulo. Comprueba cual es la ruta donde está el código del módulo (archivo .so):**

/usr/lib/apache2/modules/

**Consulta el directorio /usr/lib/apache2/modules/ y observa los módulos disponibles para cargar. Elige un par de módulos y busca en http://httpd.apache.org/docs/2.4/es/mod/ su funcionalidad:**

El módulo mod\_unique\_id.so sirve para generar un identificador único para cada solicitud que recibe el servidor.

El módulo mod\_session.so proporciona funcionalidades para la gestión de sesiones de usuario

**Ejecuta el comando “sudo apt-cache search libapache2-mod” para mostrar los paquetes disponibles en los repositorios de Ubuntu que permiten instalar módulos adicionales en Apache. Elige un paquete y busca documentación acerca de dicho paquete:**

El paquete libapache2-mod-webauth es un módulo de apache que proporciona integración con el sistema de autenticación WebAuth

**Módulo userdir**

**Habilita el módulo. (Indica el comando necesario):**

sudo a2enmod userdir

**Verifica que dentro del directorio /etc/apache2/mods-enabled se han creado enlaces simbólicos del módulo userdir hacia /etc/apache2/mod-availables. ¿Como se llaman los ficheros .conf y .load creados? Reinicia el servidor para que los cambios surtan efecto:**



**Consulta el fichero /etc/apache2/mod enabled/userdir.conf . ¿Qué significado tiene el contenido de dicho fichero?**

El contenido del fichero permite el manejo de acceso a los directorios personales de los usuarios a través de userdir.

**Indica los pasos a seguir para que aparezca en pantalla el contenido de un fichero denominado personal.html (con el contenido que quieras) cuando desde Cliente se accede a http://192.168.56.10/~alumno. Asegurate que el usuario alumno esta creado en la máquina.**

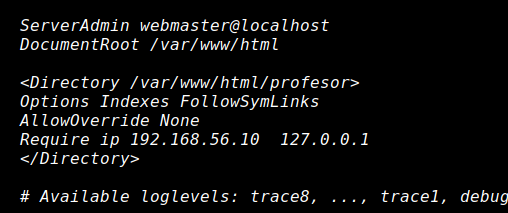
Una vez está creado el usuario, creamos un directorio de inicio del usuario como por ejemplo public\_html donde meteremos la página personal.html y le daremos los permisos correspondientes para que no de problemas, comprobamos que userdir está habilitado y si está todo bien con la URL http://192.168.56.10/~alumno/personal.html podremos visualizar la página

**Deshabilita el módulo. Indica el comando a utilizar:**

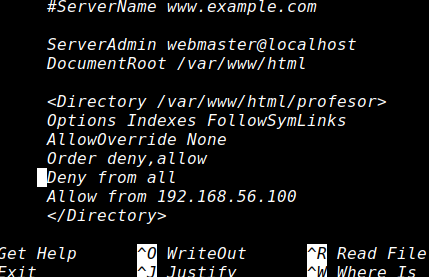
sudo a2dismod userdir

# **15. Directivas Order, allow y deny. Directiva Require:**

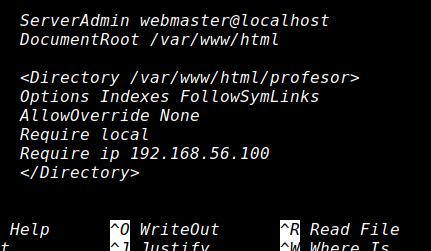
**En la máquina crea el directorio /var/www/html/profesor y crea en este directorio un fichero html con el contenido que quieras. Configura Apache para que solo se pueda acceder desde el equipo local y desde la máquina Cliente (utilizando su dirección IP). Muestra pantallazo:**

****

**Indica y prueba dos soluciones distintas, utilizando las directivas Order, Allow y Deny. Comprueba que se puede acceder a http://192.168.56.10/profesor/ desde Cliente pero no desde otras máquinas de compañeros. Muestra pantallazo:**

****

**Vuelve a realizar los dos puntos anteriores pero esta vez utilizando la directiva Require en lugar de Order, Allow y Deny. Muestra pantallazo:**

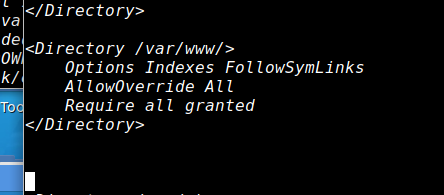
****

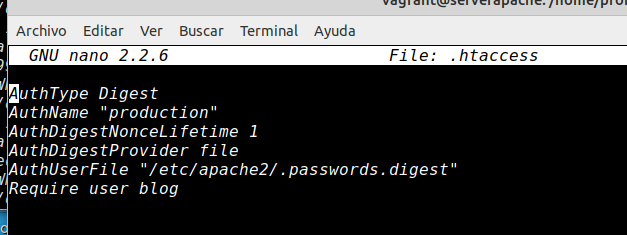
# **16. Autenticación y autorización Basic y Digest:**

(No hace falta capturas ni información para este apartado)

# **17. Fichero htaccess:**

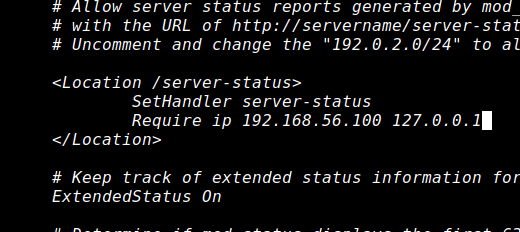
**Muestra pantallazo:**





# **18. Módulos mod\_status y mod\_info:**

**Edita el fichero de configuración del módulo /etc/apache2/mods-enabled/status.conf y habilita el acceso a /server-status desde Cliente. Muestra pantallazo:**

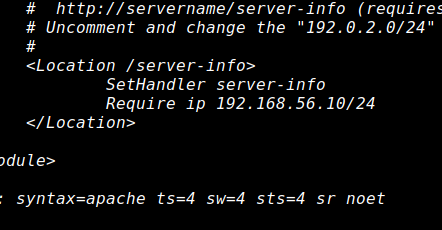


**Desde Cliente conéctate a http://192.168.56.10/server-status. Muestra pantallazo:**

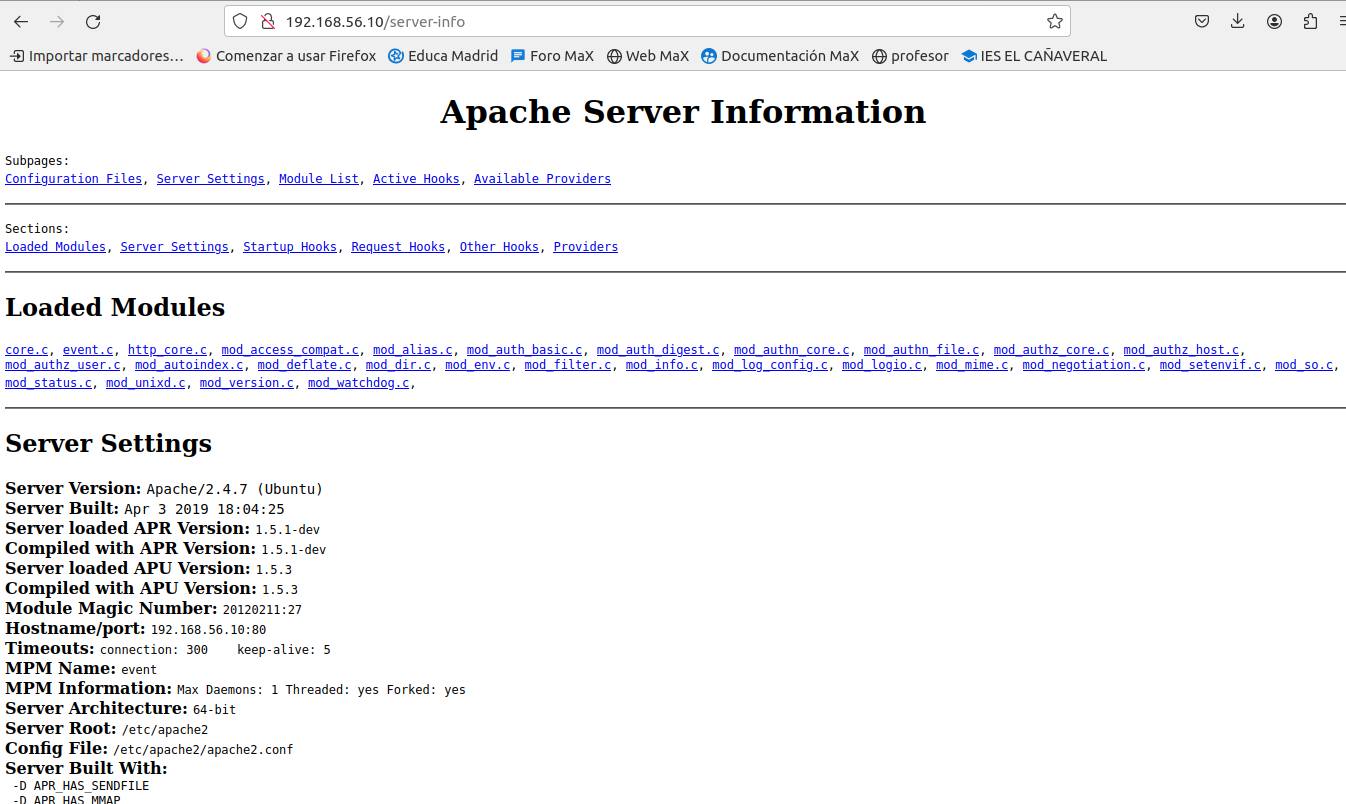


Nota: No se porque no me deja acceder si esta bien configurado

**Edita el fichero de configuración del módulo /etc/apache2/mods-enabled/info.conf y habilita el acceso a /server-info desde Cliente. Muestra pantallazo:**

****

**Desde Cliente conéctate a http://192.168.56.10/server-info. Muestra pantallazo:**

****

# **19. Ficheros de registros (logs):**

**¿Cuál es el fichero de logs de errores (directiva ErrorLog) y cuál es su nivel de prioridad (directiva LogLevel)?:**

#LogLevel info ssl:warn

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

**Observa cual es el fichero de logs de accesos (directiva CustomLog) y cuál es su formato (como no especifica ningún formato con LogFormat se usa el predefinido para el servidor principal en el fichero /etc/apache2/apache2.conf):**

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

# **20. SSL en Apache:**

(No hace falta capturas ni información para este apartado)

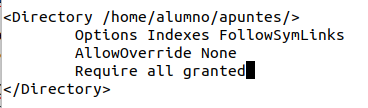
# **21. Recomendaciones para securizar el servidor:**

**Permitir sólo determinados métodos de solicitud HTTP (LimitExcept en cada archivo site)**

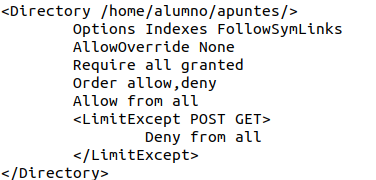
Por defecto, Apache acepta multitud de métodos de solicitud dentro del protocolo HTTP. Por ejemplo, suele aceptar protocolos como OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, CONNECT.

Sin embargo, en la gran mayoría de las veces solo nos interesan los métodos POST y GET; mientras que el resto no los vamos a utilizar y mantenerlos activos puede romper la seguridad de nuestra aplicación web.

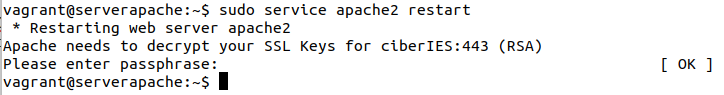
Para permitir únicamente los métodos POST y GET modificamos con nano el archivo /etc/apache2/apache.conf y nos movemos hasta el directorio que queramos modificar:



Una vez aquí, añadimos lo siguiente a su interior:



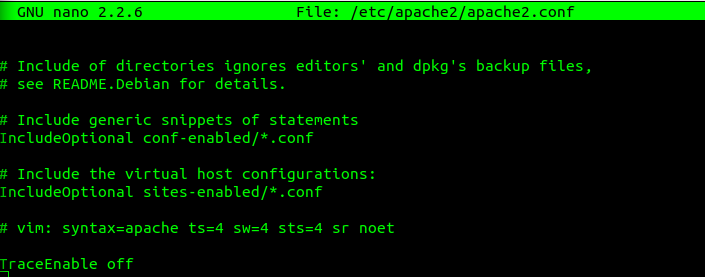
Guardamos y reiniciamos Apache:



Ya tenemos configurados los métodos de HTTP permitidos para dicho directorio. Ahora tendremos que hacer lo mismo para cada uno de los directorios que queramos proteger de esta forma.

**Deshabilitar la solicitud HTTP de seguimiento (TraceEnable en apache2.conf)**

La solicitud TraceEnable permite preguntar a apache2 sobre las acciones y el camino que realiza para responder la petición, su uso debería ser solo para debug y nunca en producción, ya que permite descubrir vulnerabilidades a los atacantes, para deshabilitarlo:



Una vez escrito “TraceEnable off” en el archivo de configuración global de apache, reiniciamos el servicio, y ya tenemos TraceEnable desactivado.