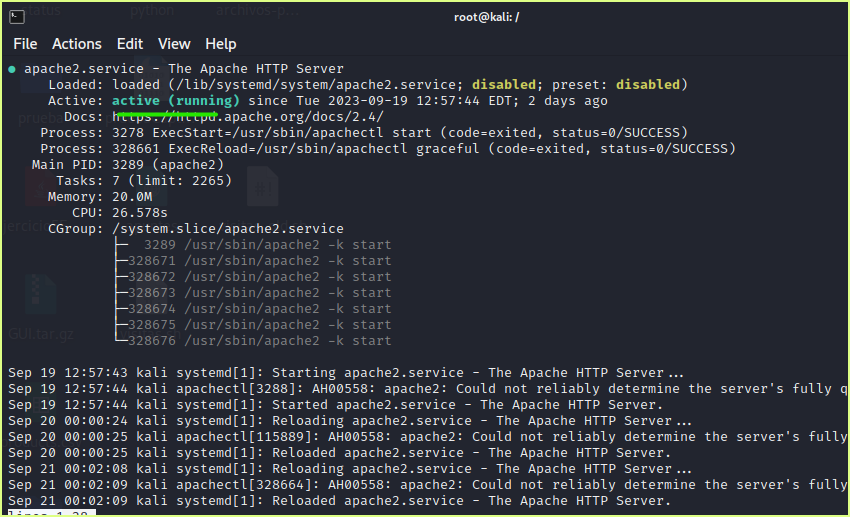


1. **Instalación del servidor web Apache2 en una máquina Linux (paquete apache2) y verificar que funciona.**

Verificamos que apache2 está ya instalado y comprobamos que funciona, con el comando

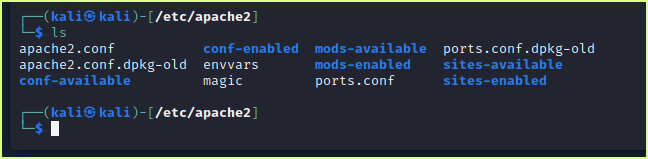
sudo service apache2 status.



1. **Explorar los ficheros de configuración del Apache recién instalado y hacer un resumen de los ficheros, su ubicación, estructuras de ficheros incluidos (Include) y directivas más importantes. Localiza qué módulos están activos por defecto.**

**2.1. FICHEROS**

Los ficheros de configuración apache se ubican en la siguiente ruta en un sistema Linux:



Estos son los ficheros más importantes a destacar:

|  |  |
| --- | --- |
| ***sites-available*** | *En esta carpeta se encuentran los ficheros de configuración disponibles.* |
| ***sites-enabled****:* | *Carpeta que contiene enlaces simbólicos a los ficheros de configuración disponibles y que están habilitados/activos.* |

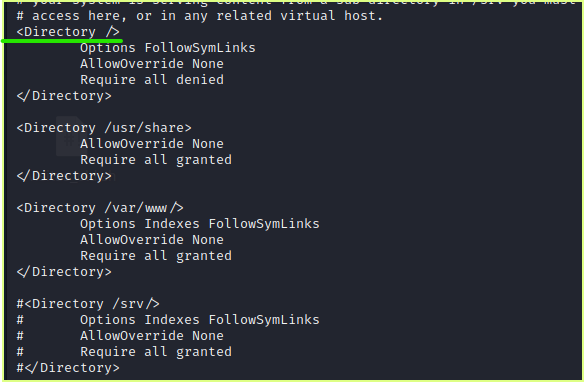
|  |  |
| --- | --- |
| ***mods-available*** | *En esta carpeta se encuentran los módulos de configuración disponibles en apache2.* |
| ***mods-enabled****:* | *Contiene enlaces simbólicos a los módulos que están habilitados.* |

|  |  |
| --- | --- |
| ***Apache2.conf*** | *Es el archivo de configuración general. No se debería tocar en lo posible.* |
| ***envvars*** | *Tiene variables de entorno que se pueden usar en la configuración.* |
| ***Ports.conf*** | *Es el archivo que permite configurar los puertos desde donde escuchar.* |
| ***magic*** | *Tiene Patrones para mod\_mime\_magic.* |

\*\*\* \*\*\* \*\*\*

**2.2. DIRECTIVAS**

Las directivas principales se encuentran en el fichero apache2.conf:



**Directiva AllowOverride:** Cuando esta directiva se establece en Ninguno, los archivos .htaccess se ignoran por completo. En este caso, el servidor ni siquiera intentará leer archivos .htaccess en el sistema de archivos.

**Directiva Require :** Prueba si un usuario autenticado está autorizado por un proveedor de autorización.

Cuando se estable all grantes Se permite el acceso incondicionalmente a todos.

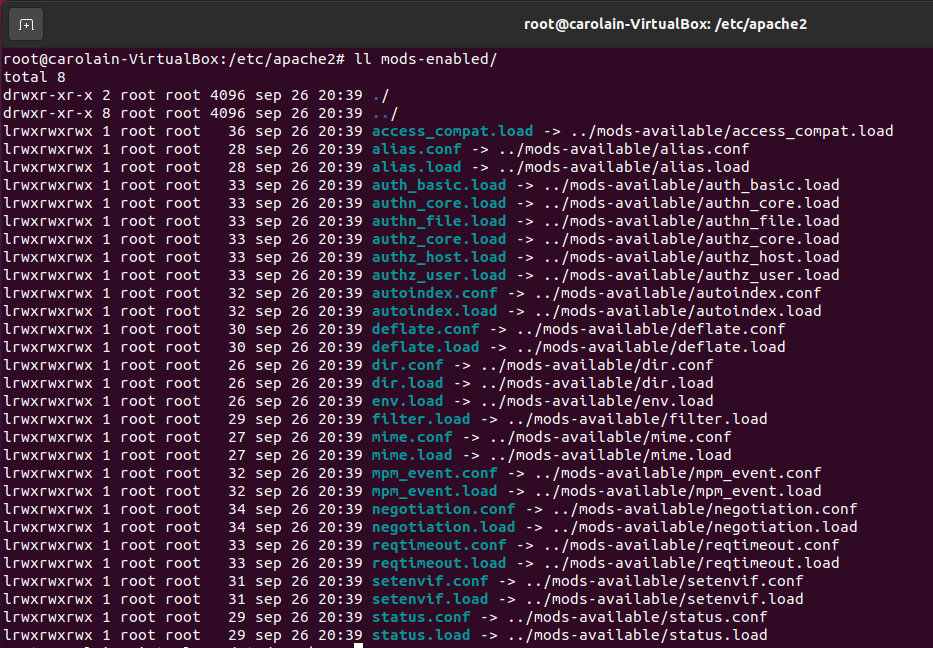
**Directiva Options:** Configura qué funciones están disponibles en un directorio en particular.

* FollowSymLinks --> El servidor seguirá enlaces simbólicos en este directorio.
* Indexes --> Si se solicita una URL que se asigna a un directorio y no hay DirectoryIndex (por ejemplo, index.html) en ese directorio, entonces mod\_autoindex devolverá una lista formateada del directorio.

**DirectoryIndex:** Especifica el fichero por defecto que buscará en cada directorio en caso de que no se especifique ninguno. Por defecto es index.html.

\*\*\* \*\*\* \*\*\*

**2.3. MÓDULOS ACTIVOS POR DEFECTO**

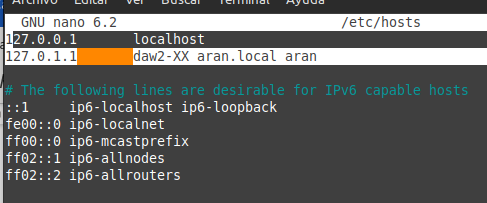


Entre ellos destacar:

* **mod\_alias:**  Habilita el reenvío http y la asignación de directorios.
* **mod\_auth\_basic:**  Autorización de usuario a través de archivos .txt, Palabra clave: .htpasswd
* **mod\_dir:** Permite la evaluación de URL con / final y la creación de índices de directorios.
* **mod\_mime:** Permite comportamientos distintos del servidor web basados en las terminaciones de los archivos.
* **mod\_alias:** Habilita el reenvío http y la asignación de directorios.

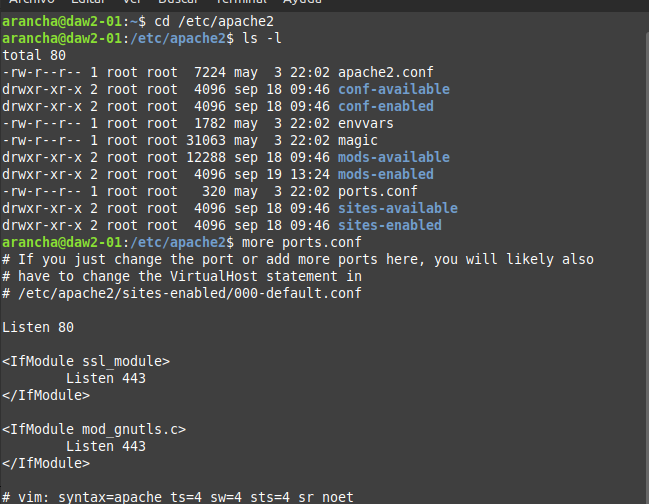
1. **Modifica el archivo del sistema /etc/hosts para tener varios nombres asociados localmente a una de tus ips, por ejemplo:**   
    **127.0.1.1 profesor servidor.daw2.net javi.local**

Lo hacemos con sudo nano /etc/hosts y añadimos más host/alias a mi ip:



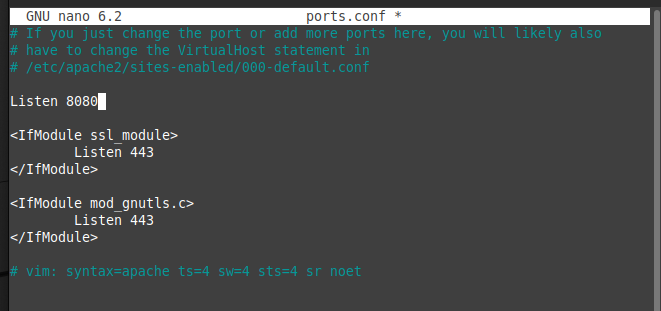
1. **Cambiar los siguientes valores de directivas:**

**4.1** **Puerto en el que escucha a 8080**

Primero desde la carpeta ports.conf, vemos que está usando el puerto 80 por defecto.  


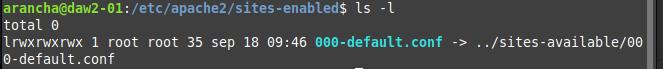
Y editamos el archivo con el comando nano, modificamos el número del puerto al 8080:





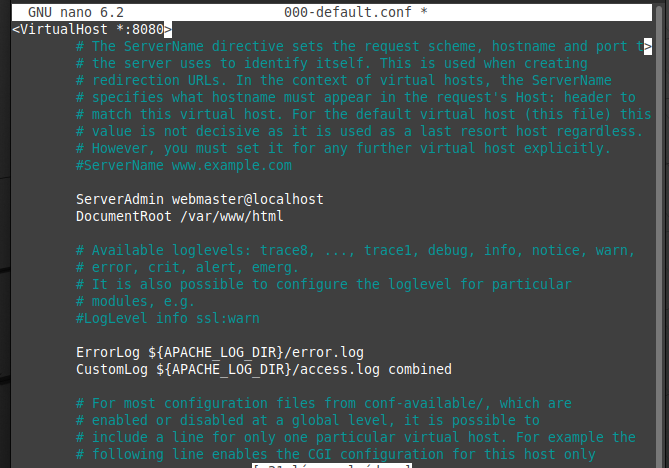
También es necesario modificar el host-virtual apache para iniciar el proceso de enlace.

Este se encuentra en la carpeta sites-enabled, y vemos que solo hay uno:

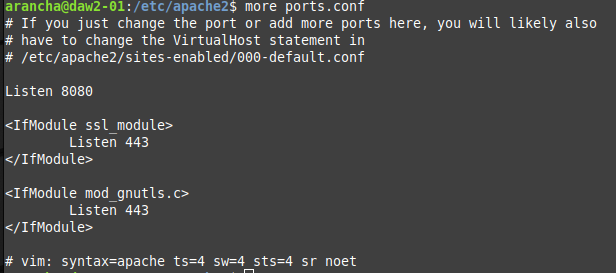


Y de nuevo usamos el comando nano con este archivo para modificar el puerto al 8080:

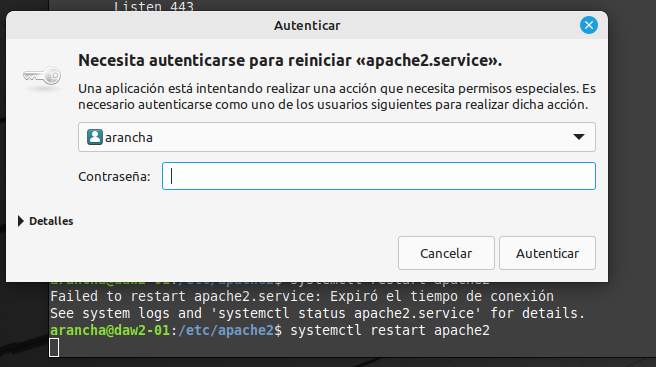




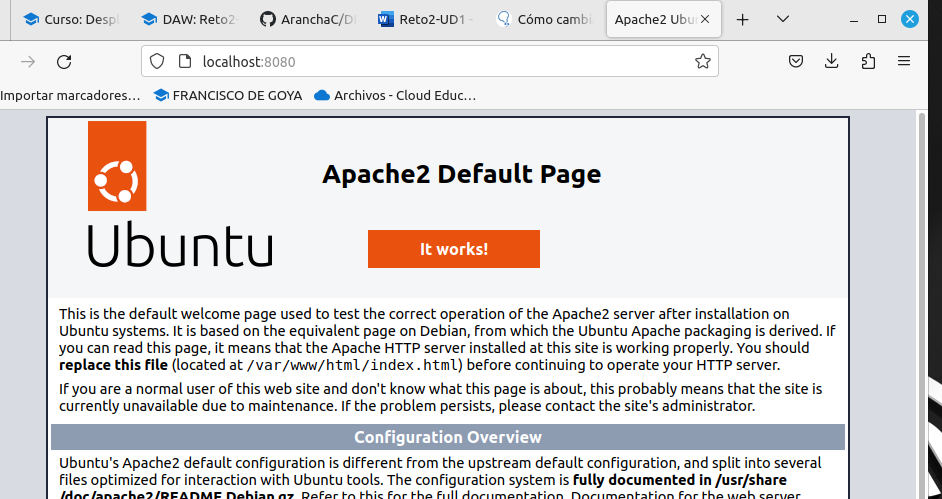
Comprobamos que el cambio se ha realizado correctamente abriendo la carpeta ports.conf:



Reiniciamos el servidor para aplicar los cambios, como siempre que modificamos algo:



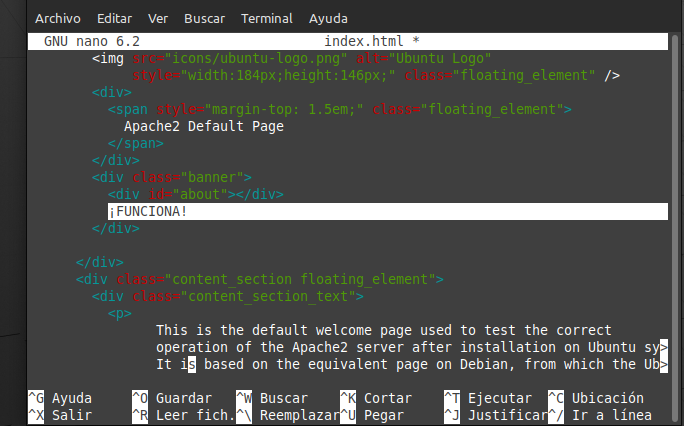
Y comprobar que la conexión es correcta, accederemos desde el navegador usando el nº de puerto:

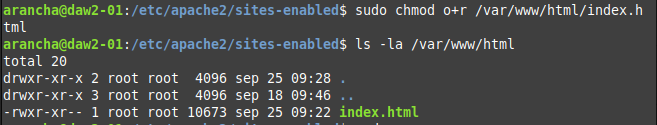


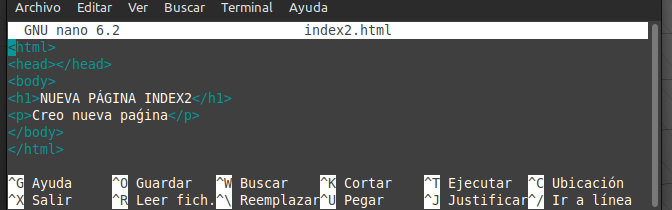
**4.2 Localiza la carpeta que está usando para servir páginas, personaliza el index.html y crea otra página más. Accede a ambas desde un navegador desde otra máquina.**

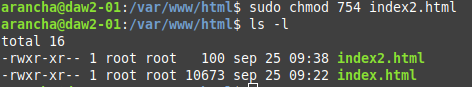
Localizo la carpeta y el archivo por defecto (index.html)  


Y con sudo nano y el nombre del html, lo abro para modificarlo:

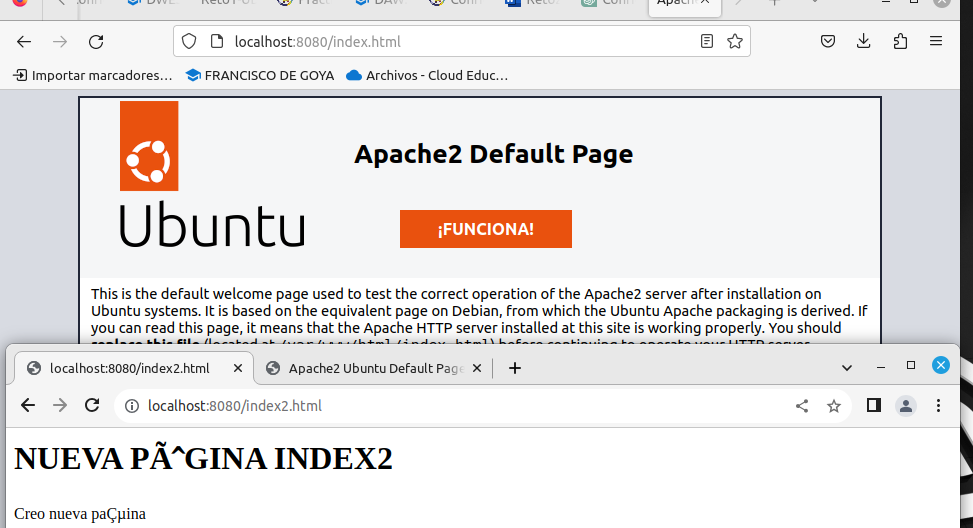


Compruebo permisos y le doy permisos de lectura y luego vuelvo a reiniciar el servidor con restart para aplicar cambios:  


Creo nueva página llamada index2.html con nano y le doy los permisos correspondientes:  


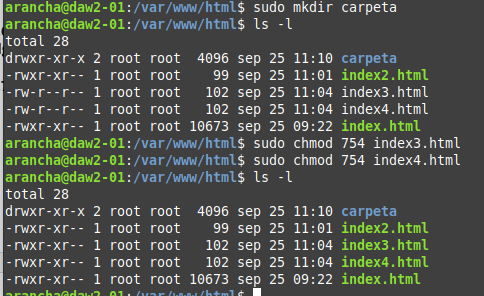


Y accedo a ambas páginas especificándolas en el navegador para comprobar la modificación de index.html y la visualización de la nueva página, también hay que especificar el nº de puerto que hemos cambiado anteriormente.

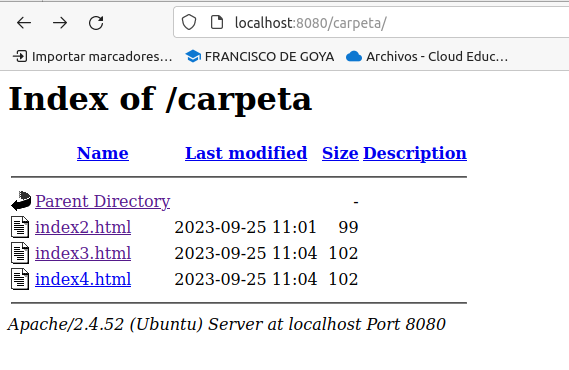


**4.3 Crea una carpeta en la raíz de los documentos servidos vía web (ej carpeta) mete varios archivos pero que ninguno sea index.html y comprueba el comportamiento al acceder a vía web (ej:** [**http://javi.local/carpeta/**](http://javi.local/carpeta/)**)**

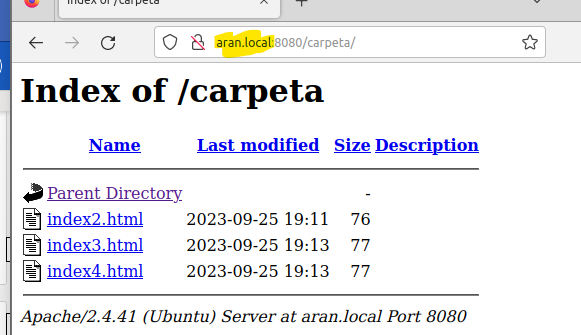
Antes creo dos páginas más llamadas index3.html e index4.html, para tener un mejor ejemplo (con sudo nano y el nombre, con un html sencillo igual que el index2.html, y les doy permisos).

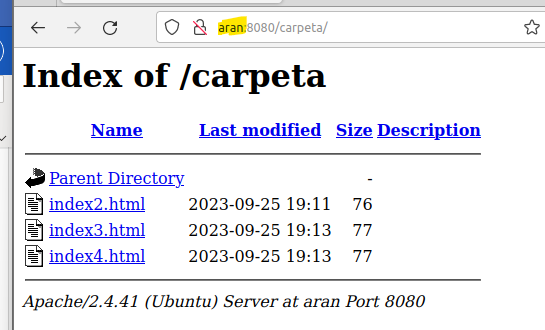
Y después creo la carpeta con mkdir, y muevo los archivos index dentro, excepto el index.html  




Y compruebo en la web que se visualiza un directorio con los archivos que hay en la carpeta, en lugar de un archivo en concreto, y podemos acceder a ellos dando un simple click.  


Hago la prueba también accediendo a través de los demás alias añadidos en host en el punto 3 (aran.local y aran):





**¿Te parece seguro el comportamiento por defecto? ¿Se puede cambiar?**

Sí, el comportamiento es normal ya que está activo en la configuración en Options Indexes, que establece que: Si se solicita una URL a la que se le asigna un directorio y no hay DirectoryIndex (por ejemplo, index.html en ese directorio), entonces mod\_autoindex devolverá una lista formateada del directorio.

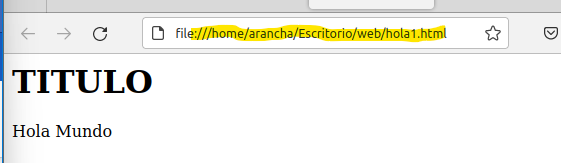
El **mod\_autoindex** está activo como se puede ver en el apartado anterior de módulos activos por defecto en apache2.

**4.4 Haz que, al acceder a una ruta determinada, ej:** [**http://javi.local/tmp/**](http://javi.local/tmp/) **sirva las páginas de otra carpeta distinta que no esté bajo DocumentRoot**

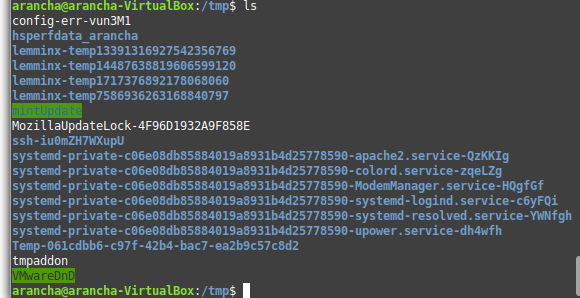
Lo he realizado con la opción de **directorio virtual usando la directiva Alias,** siguiendo el pdf del temario.

Primero, para luego verificar el resultado, voy a usar las páginas de la ruta /home/arancha/Escritorio/web, que solo tiene “hola1.html”, compruebo la ruta y, en el navegador, su contenido:



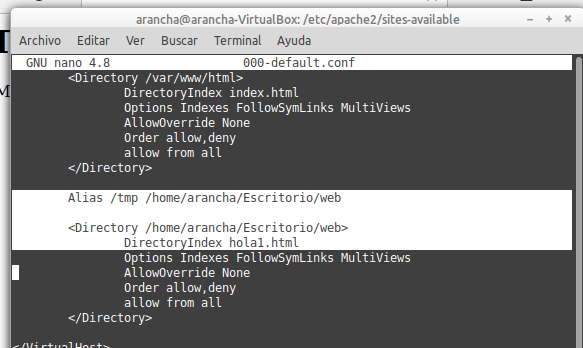


Y compruebo que en la carpeta /tmp, la que voy a usar para acceder desde el navegador, no tiene este archivo hola1.html:

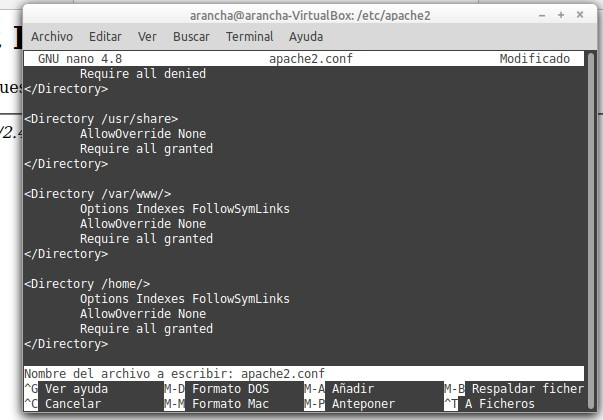


Ahora modifico en el archivo de configuración **000-default.conf** y añado un Alias y la directiva correspondiente.

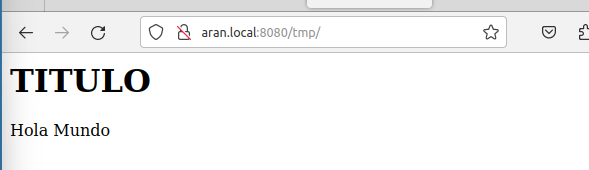
El alias se escribe con la url de la ruta a la que queremos acceder en el navegador, en este caso **/tmp**, seguido de la ruta donde se encuentran las páginas que queremos que sirva, **/home/arancha/Escritorio/web.** Esta ruta también se pone en la directiva, después de Directory, y el nombre de la página, hola1.html, dentro. Tal y como se ve en la imagen.



Pero también hay que modificar el archivo apache2.conf, para que el servidor tenga acceso al directorio home, que es donde se encuentran las páginas que vamos a lanzar al navegador. Por lo que hay que crear otra directiva en este archivo para indicar Require all granted en home, para darle acceso a todos. Ver imagen:



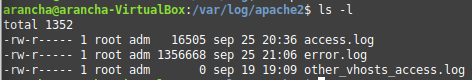
Reiniciamos servidor, y abrimos navegador indicando el nombre de usuario (aran.local, aran o localhost), puerto (8080, ya que lo cambiamos), y la carpeta a la que accedemos (tmp). Comprobamos resultado:



Se puede ver que ha funcionado, pues al acceder en el navegador a la ruta tmp, se muestra la página de la carpeta web de Escritorio.

**4.5 Tras varios accesos describe el contenido del archivo de registro access.log ¿Hay algo en el error.log?**

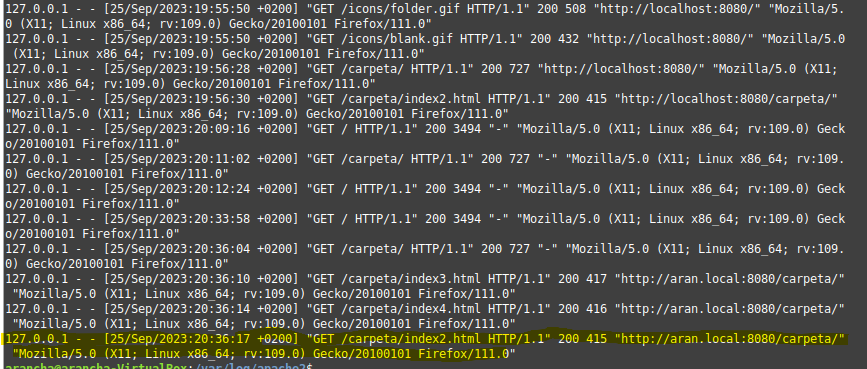
El archivo de registro se encuentra en la siguiente ruta, donde localizamos estos dos archivos:



**ACCESS.LOG**

Contiene los registros de cada solicitud de acceso al sitio web. Cada entrada de este archivo registra la información sobre una solicitud http, incluyendo la ip del cliente, la fecha y hora de acceso, la url solicitada, y más. Proporciona información muy valiosa sobre las interacciones de los usuarios con nuestro sitio web.

* Accedemos a **access.log** y visualizamos su contenido:



Describimos el último registro de la captura anterior:

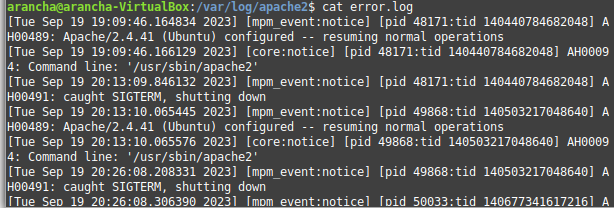
* **127.0.0.1:** La dirección IP del cliente que realizó la solicitud.
* **- -:** En este ejemplo, los guiones (-) representan espacios reservados para el nombre de usuario y la autenticación, pero están vacíos en este caso.
* **[25/Sep/2023:20:36:17 +0200]:** La fecha y hora en que se realizó la solicitud.
* **"GET /carpeta/index2.html HTTP/1.1":** El método HTTP utilizado (en este caso, "GET"), seguido de la URL solicitada ("/carpeta/index3.html") y la versión del protocolo HTTP ("HTTP/1.1").
* **200**: El código de respuesta HTTP que indica que la solicitud se completó con éxito. Un código de respuesta "200" significa "OK".
* **415**: El tamaño de la respuesta en bytes.
* **"http://aran.local:8080/carpeta/":** muestra la URL de la página desde la cual se originó la solicitud.
* **"Mozilla/5.0 (X11; Linux x86\_64; rv:109.0) Gecko/20100101 Firefox/111.0":**  información sobre el navegador o cliente que realizó la solicitud.

**ERROR.LOG**

Contiene registros detallados de los errores que ocurren mientras el servidor está operativo. Es muy valioso para diagnosticar y solucionar problemas en el servidor, por lo que sería útil revisarlo cuando tenemos algún problema.

Algunos de los errores que muestran son: códigos de error HTTP, errores de configuración que incluyen errores de sintaxis, errores de permisos, problemas de acceso, y más.

Aquí se puede ver su contenido:





1. **Crea al menos dos sitios virtuales por nombre, usando nombres definidos en fichero hosts (en /etc) para una sola IP, cada uno deberá devolver una página raíz distinta. Para conseguir conectar con ellos, en la máquina cliente desde la que uses el navegador, también tendrás que establecer vía fichero hosts o equivalente los nombres creados.**

**Modifica:**

**5.1** **Nombre de cada servidor virtual con vuestros nombres (creados en hosts).**

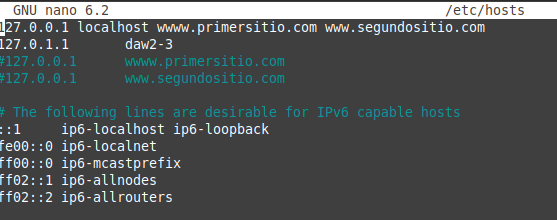
El nombre de los dos sitios es:

* [www.primersitio.com](http://www.primersitio.com)
* [www.segundositio.com](http://www.segundositio.com)

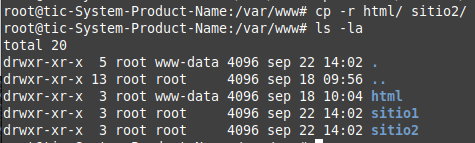
**5.2** **Ficheros de log de acceso por separado.**

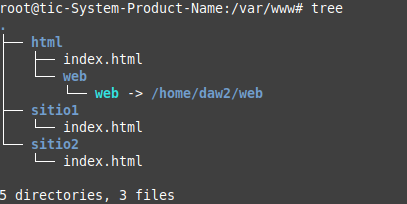
* Ficheros del primer sitio: error\_sitio1.log y access\_sitio1.log
* Ficheros del segundo sitio: error\_sitio2.log y access\_sitio2.log

Lo primero que haremos es poner los nombres de los servidores virtuales en /etc/hosts:

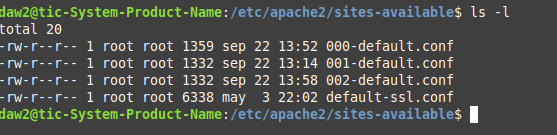


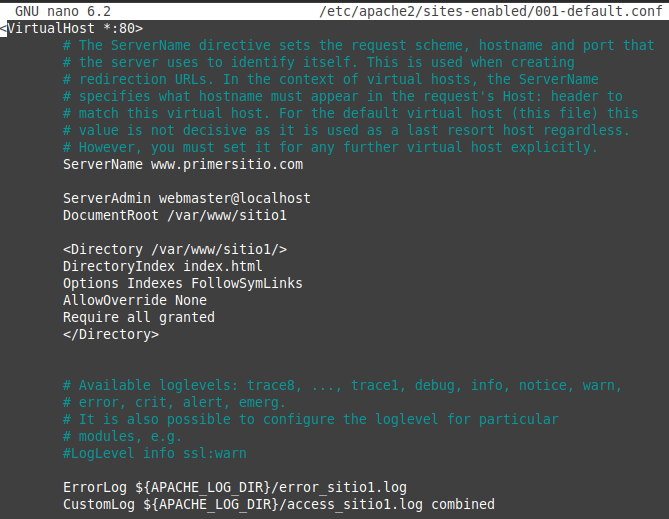
Luego, crearemos las carpetas raíz para los dos sitios, con su respectivo index.html:

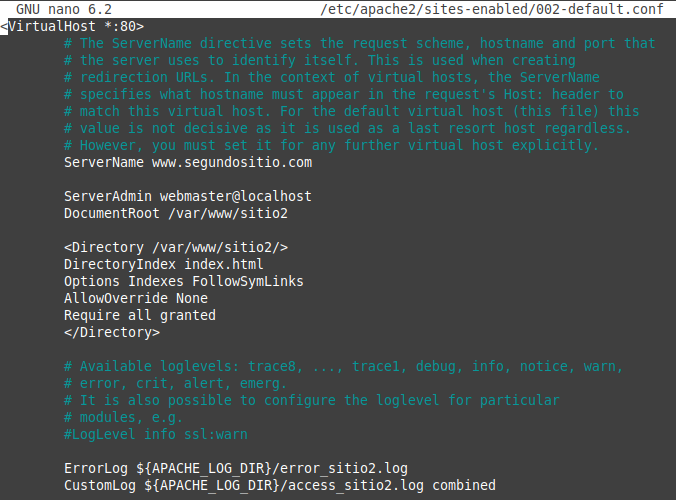




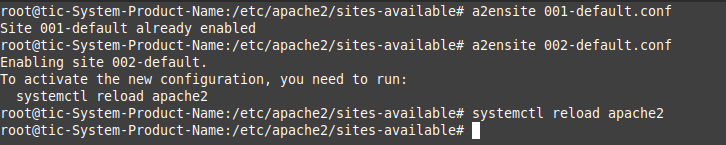
Creamos el fichero de configuración de ambos sitios:

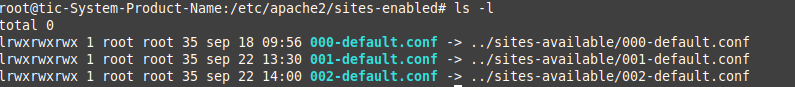


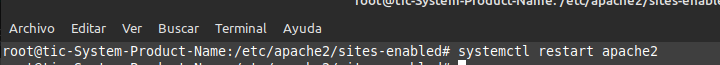
001-default.conf para www.primersitio.com:

002-default.conf para [www.segundositio.com](http://www.segundositio.com):

Habilitamos cada sitio con el comando: # a2ensite 001-default.conf y luego recargamos el servicio apache2 con #systemctl reload apache2 o reiniciarlo con #systemctl restart apache2:



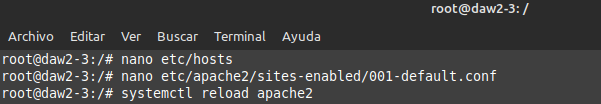




Resultado:



Como [www.primersitio.com](http://www.primersitio.com) da problemas al resolver el DNS. He modificado el nombre del sitio por [www.sitio1.com](http://www.sitio1.com) lo cual me ha dado buen resultado:



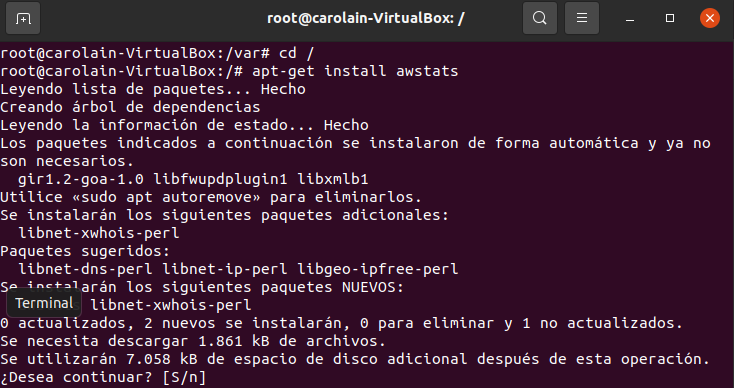




**5.2.1 Opcional: localiza software para generar gráficas a partir de los ficheros de registro de accesos (logs)**

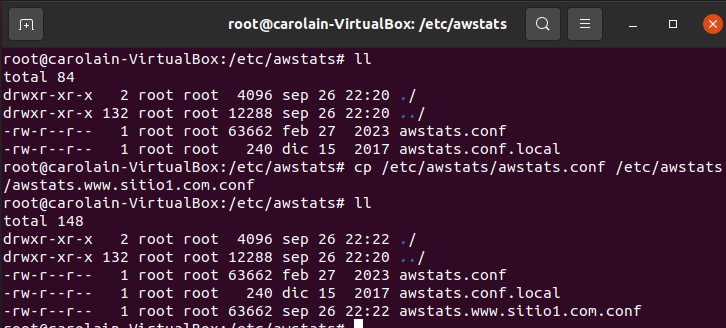
**Awstats**

Instalamos awstats con: #apt-get install awstats:



Creamos una copia de awstats.conf por dominio:

# cp /etc/awstats/awstats.conf /etc/awstats/awstats.www.sitio.com.conf:



Modificamos el fichero:

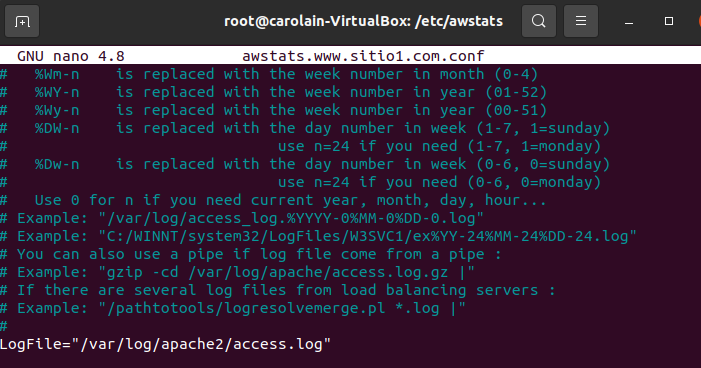
nano /etc/awstats/awstats.www.sitio1.com.conf

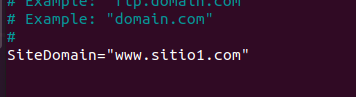
Hacemos los siguientes cambios:

LogFile="/var/log/apache2/access.log"

SiteDomain="yourdomain.ext"

HostAliases="localhost 127.0.0.1 yourdomain.ext"

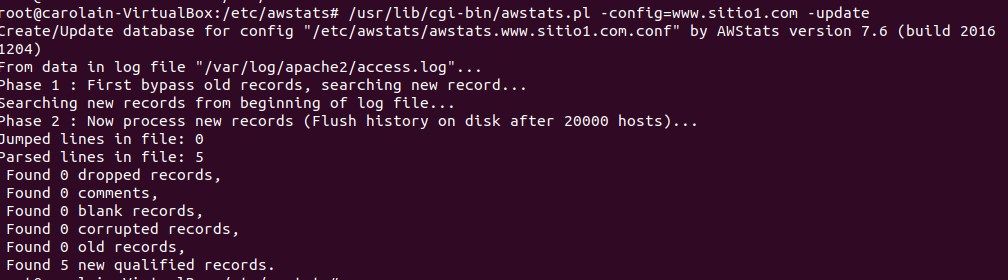




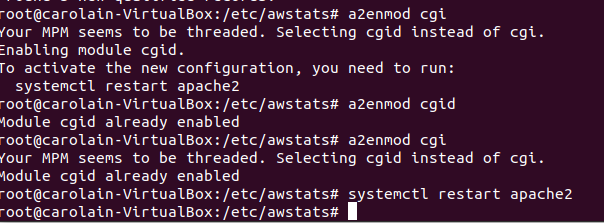


Generar el estado inicial para AWStats basado en var/log/apache2/access\_sitio1.log:

/usr/lib/cgi-bin/awstats.pl -config=www.sitio1.com -update

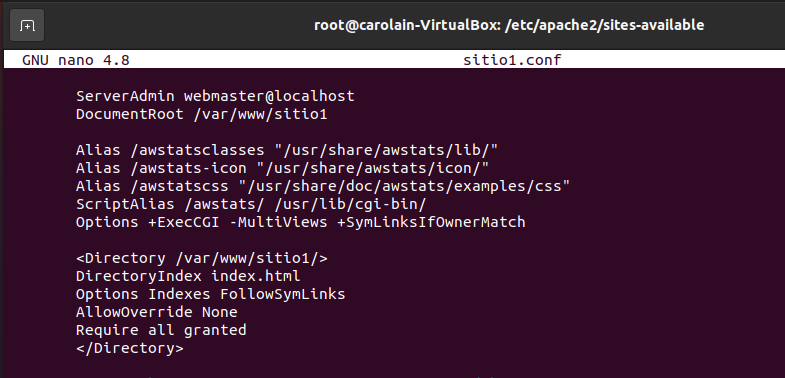


Activar el mod\_cgi : #a2enmod cgi

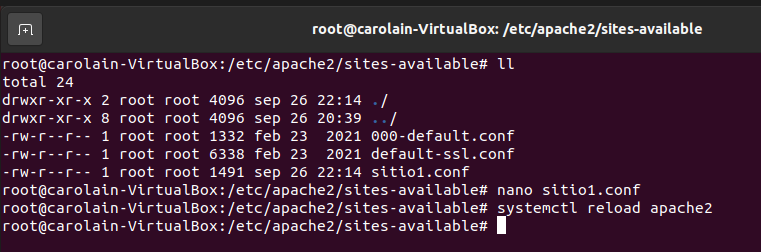


En /etc/apache2/sites-available/sitio1.conf añadimos las siguientes líneas:

*Alias /awstatsclasses "/usr/share/awstats/lib/"*  
 *Alias /awstats-icon "/usr/share/awstats/icon/"*  
 *Alias /awstatscss "/usr/share/doc/awstats/examples/css"*  
 *ScriptAlias /awstats/ /usr/lib/cgi-bin/*  
 *Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch*



Recargamos apache #systemctl reload apache2:



Las estadísticas para www.sitio1.com ahora deberían estar disponibles en:

*http://www.sitio1.com/awstats/awstats.pl*

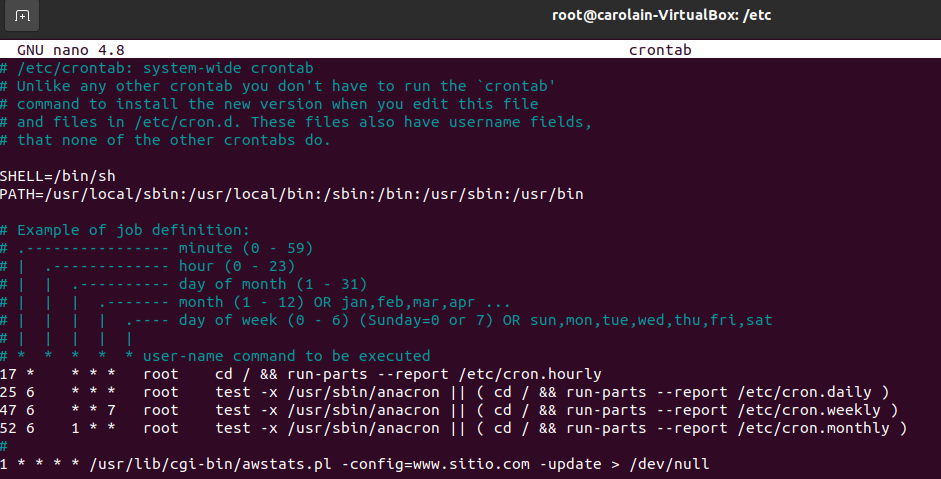
y:

por si tiene varios archivos de configuración. Tenga en cuenta que estas estadísticas son públicas a menos que las proteja.

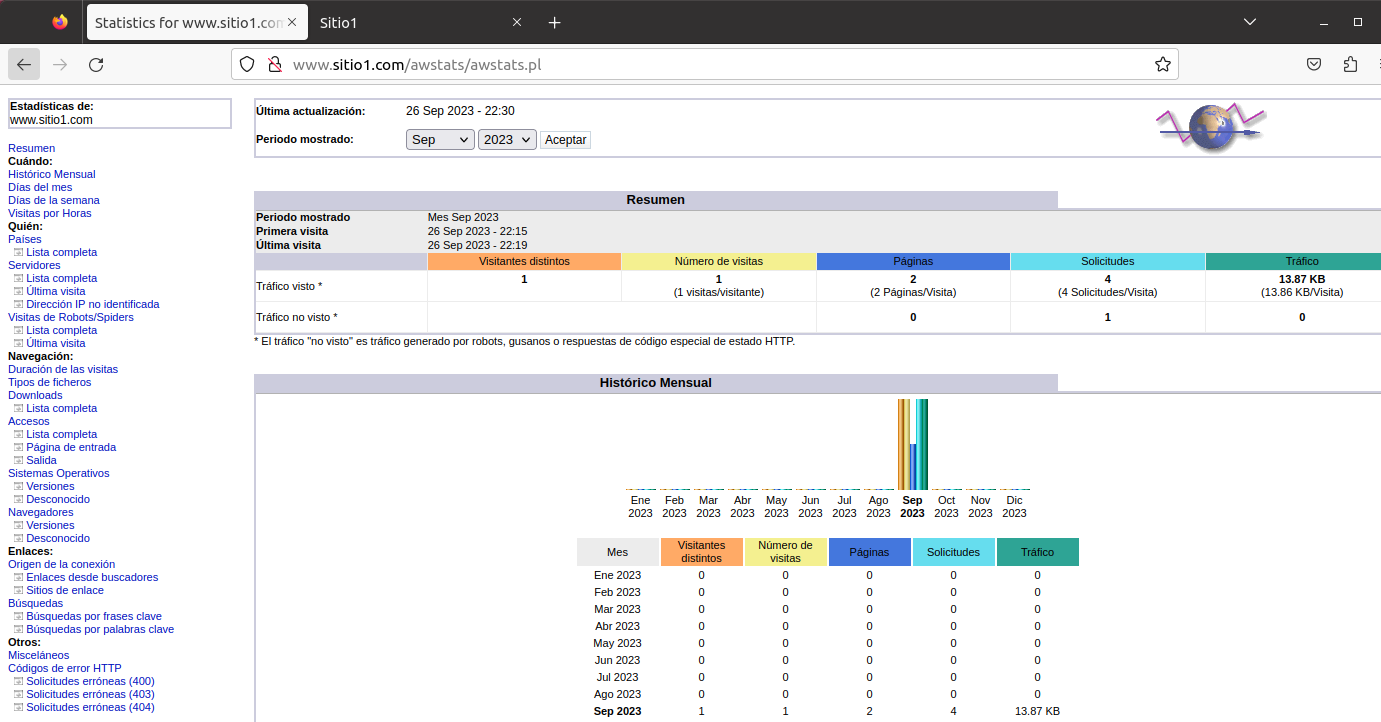
*http://www.sitio1.com/awstats/awstats.pl?config=tudominio.ext*

***Configuración de crontab***

*Agregue el siguiente código a /etc/crontab para cada dominio:*  
*0 \*/3 \* \* \* /usr/lib/cgi-bin/awstats.pl -config=www.sitio.com -update > /dev/null*



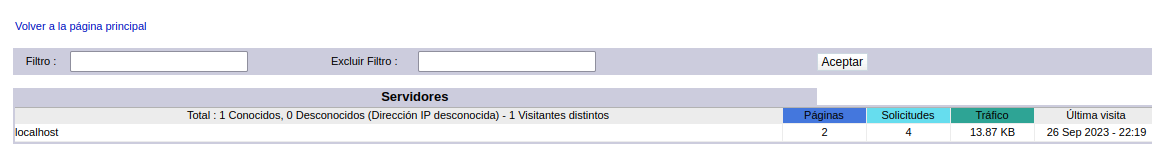
**Resultado:**



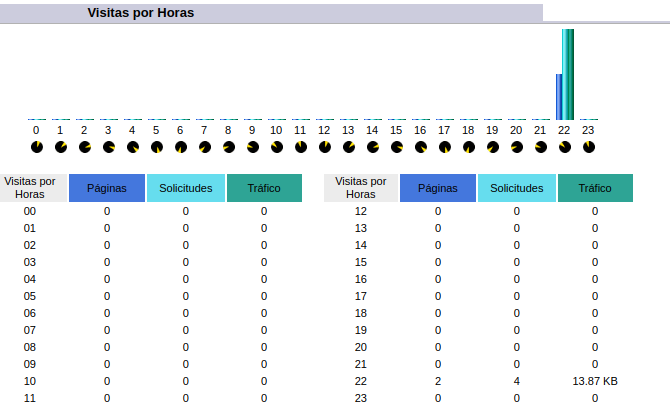
Se puede ver en la imagen el número de visitas que fue 1 las solicitudes 4 y el tráfico de datos.

Hace una gráfica por mes, días y horas. En la parte izquierda te muestra las diferentes opciones que tiene para analizar las estadísticas realizadas por Awstats a tu dominio. En este caso [www.sitio1.com](http://www.sitio1.com)

EJ La última visita:



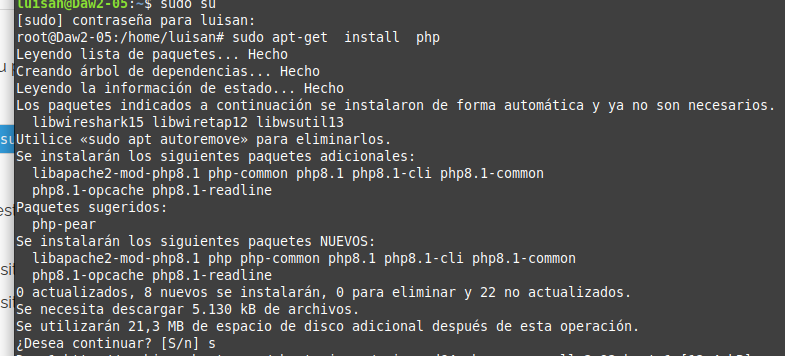
Histórico por horas:



1. **Instala la conexión de Apache2 con PHP (libapache2-mod-php) y configura el sistema para que las páginas con extensión .php se sirvan a través del intérprete de PHP.**

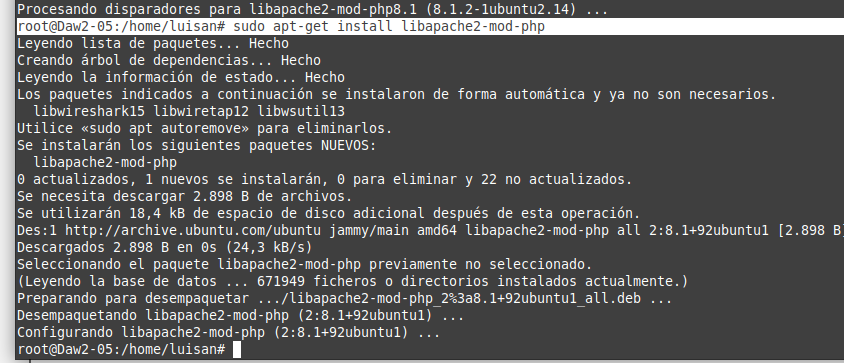
Instalación de php:

**sudo apt-get install php**



Necesitaremos también el módulo PHP para Apache para que conecte nuestro servidor con nuestro motor PHP. Lo instalamos con el siguiente comando.

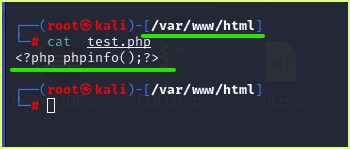
**sudo apt-get install libapache2-mod-php**

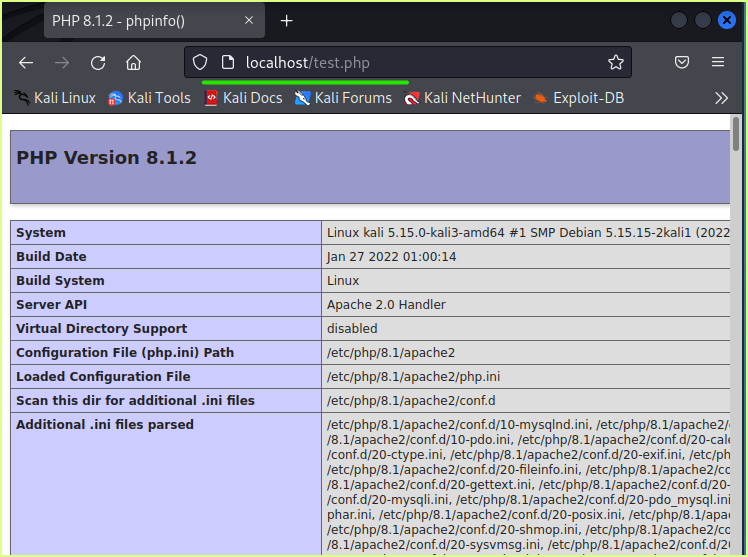


**Comprueba si funcionan un par de páginas con este código:**

**6.1** **<?php phpinfo(); ?> (para mostrar la información de la instalación)**

Creamos un fichero php con la siguiente línea <**?php phpinfo(); ?>** dentro del fichero test.php n la ruta /var/www/html/



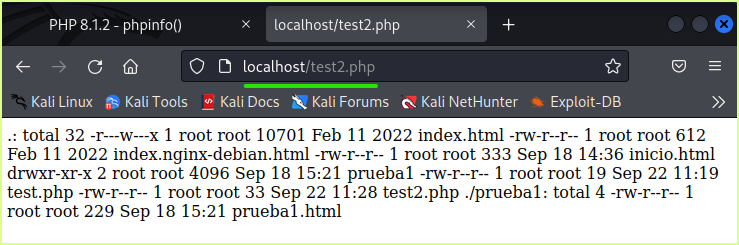
Si llamamos desde localhost al fichero test.php , debes visualizar la siguiente pantalla.

**6.2** **<?php passthru ["ls -lR"] ?> (con paréntesis en vez de corchetes, para crear más carga listando todos los archivos del sistema)**

Creamos otro fichero php en la misma ruta para visualizar el resultado.



Ejecución de del fichero:



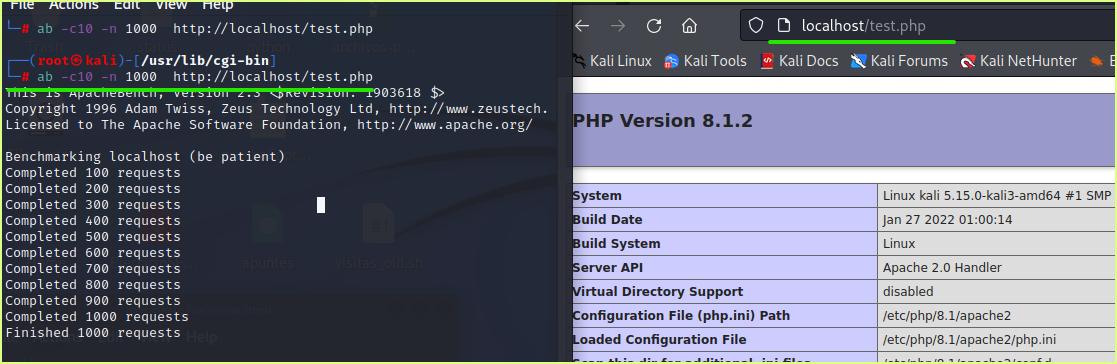
El código dentro del php muestra un listado de los ficheros y directorios de forma descendente e incluye información de permisos.

El comando passthru() un usuario normal no debería poder ejecutar dicho comando.

1. **Con el comando ab de apache, vamos a realizar peticiones masivas y analizar los tiempos de respuesta. Ejemplo de uso:**

**7.1** **ab c 10 n 1000** [**http://tumaquina/ruta**](http://tumaquina/ruta) **(lanzará 1000 peticiones de 10 en 10 a la dirección dada)**

Lanzamos el siguiente comando *ab -c10 -n 1000 http://localhost/test.php* desde una terminal, que lanza peticiones a la dirección y visualizamos el resultado.



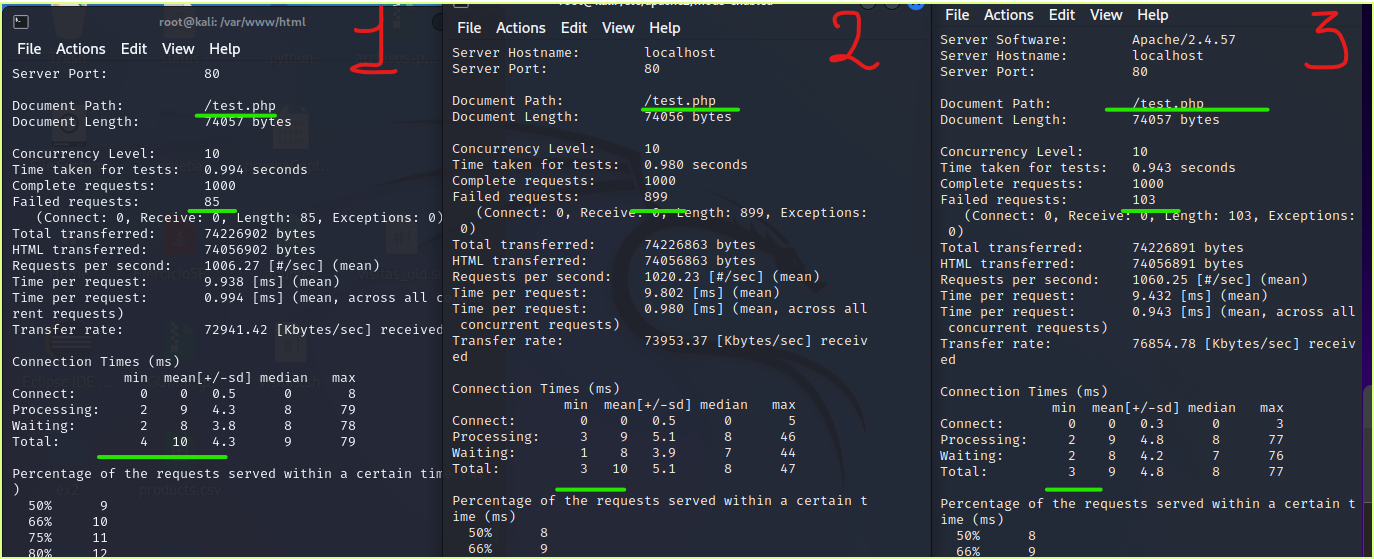


**7.2** **lanza ese mismo comando desde varios terminales y observa si van cambiando los tiempos de acceso. Hazlo sobre una página estática y luego con las páginas php, observa los cambios en el tiempo de respuesta.**

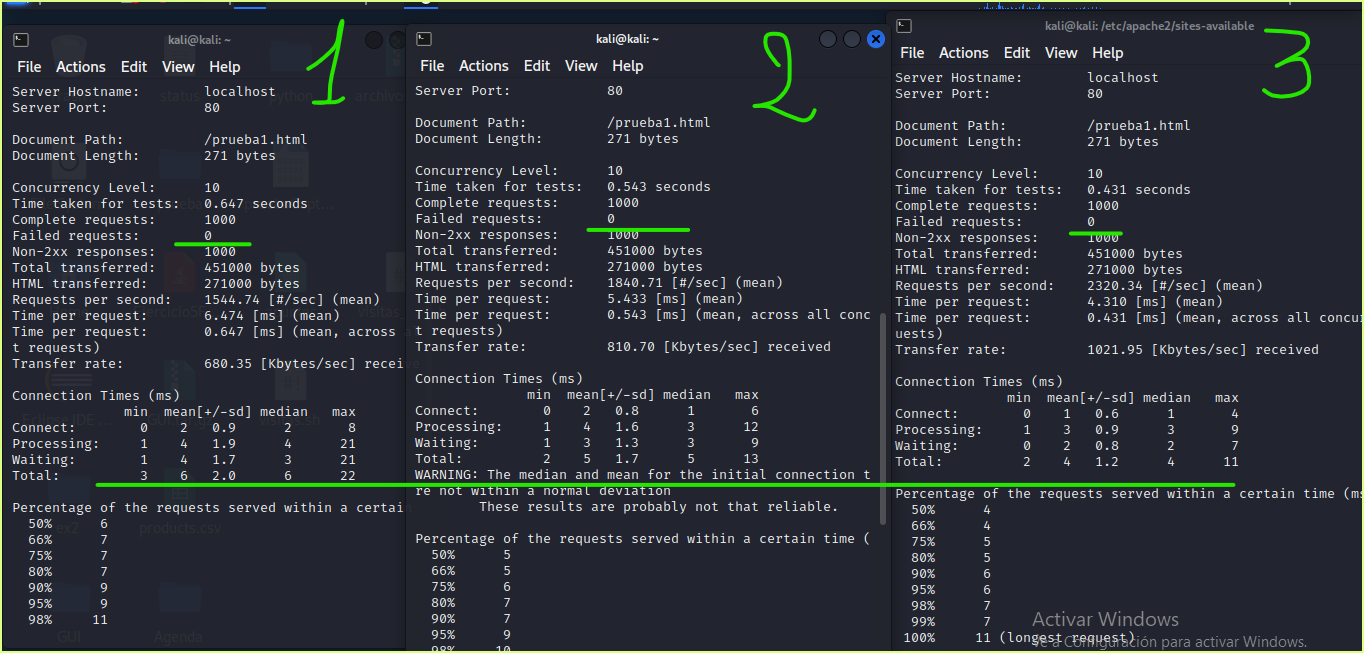
Lanzamos el comando desde 3 terminales distintas, uno detrás de otro a la página test.php y visualizamos que hay diferencias en las solicitudes rechazadas o fallidas.

Por otro lado, podemos apreciar que en la primera ejecución cambia en los tiempos respecto a la primera ejecución en la que solo se lanzó desde un terminal.

Prueba páginas php:



Prueba páginas estáticas, lanzamos comando:



Visualizamos que los tiempos no son los mismos, pero no con mucha diferencias, pero lo que si hay diferencias, es que no hay peticiones fallidas.

1. **Opcional: instala phpmyadmin( previo install de mysql-server) y verifica cómo se ha integrado con Apache2 y si puedes acceder y visualizar las bases de datos. Si te pide contraseña de mysql para enlazarlos se encuentra en el fichero debian.conf (carpeta mysql en etc)**

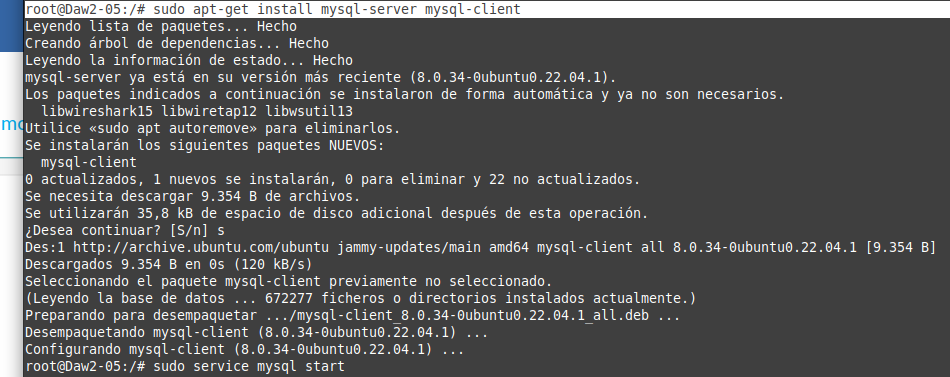
Para instalar phpmyadmin revisamos si cumplimos los requisitos:

* Un servidor web de cualquier tipo
* PHP 7.2.5 o más reciente
* Instalar Mysql y que este activo

Instalamos MySQL y levantamos MySQL:

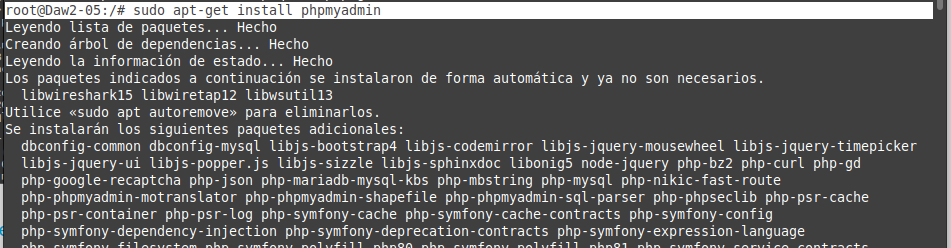
*sudo apt-get install mysql-server*

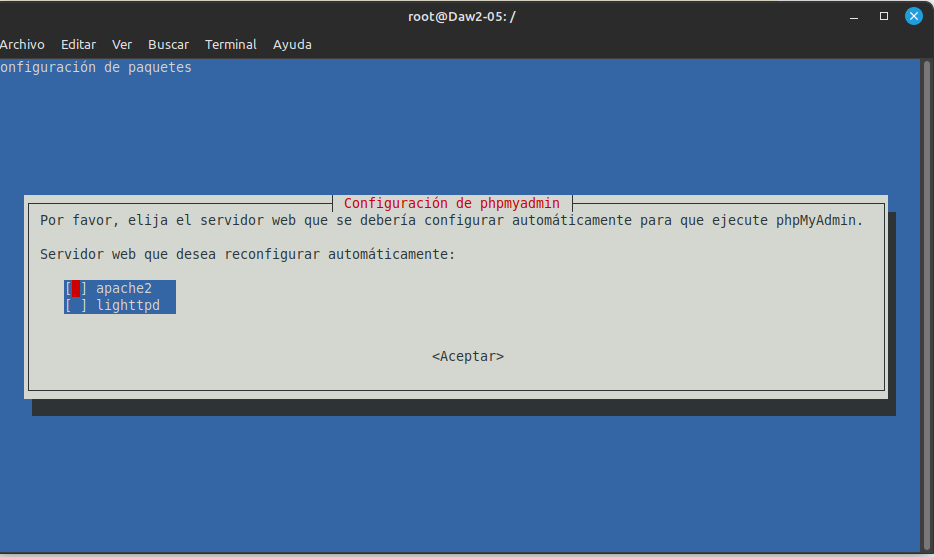
*sudo service mysql start*



Una vez instalado MySQL, instalamos phpmyadmin:

sudo apt-get install phpmyadmin



Elegimos el servidor apache:

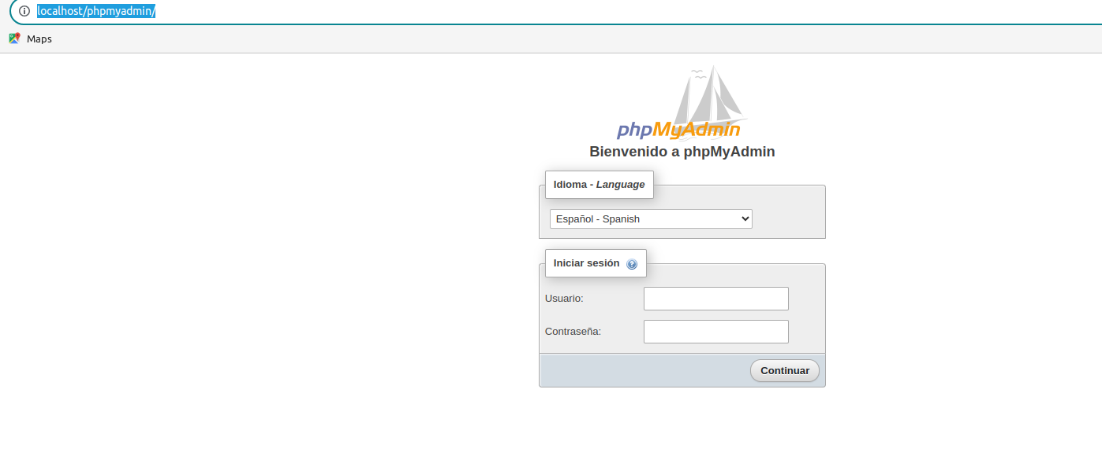
Pulsamos OK:

:

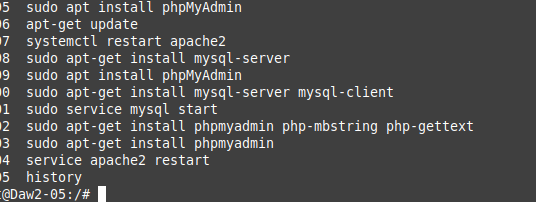
Ponemos una contraseña:



Si llamamos la url *http://localhost/phpmyadmin/*, veremos la siguiente imagen



Comandos utilizados:



Si ponemos contraseñas fáciles, google te recordará que los cambies:



1. **Opcional para optar al 10:**

**12.1 Realizar y documentar los apartados anteriores en una máquina Ubuntu.**

*Hemos realizado todos los ejercicios desde Ubunto.*

**12.2** **Explora más módulos de Apache, por ejemplo userdir, info, status, alias, dir, autoindex … Crea un contador de visitas, bien con PHP o con un script (shell, perl, python) mediante mod\_cgi**

**userdir**

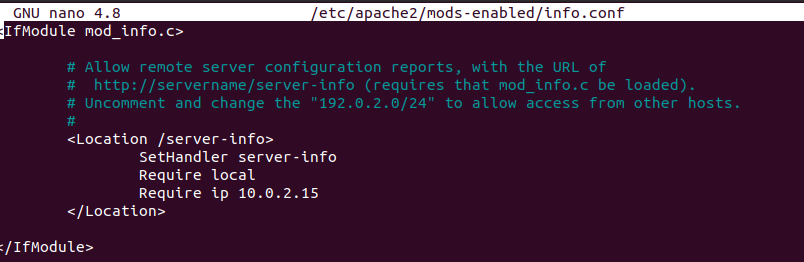
El módulo userdir se utiliza para crear páginas webs personales por cada usuario con la creación de la carpeta public\_html (donde se aloja sus páginas webs) y la ruta especifica es: http://localhost/~nombredeusuario. Ejercicio hecho en la práctica anterior que se entregó de entorno servidor.

**info**

El módulo mod\_info sirve para la información de configuración de nuestro servidor Apache de forma remota, a través de una página HTML que accedemos desde el navegador.

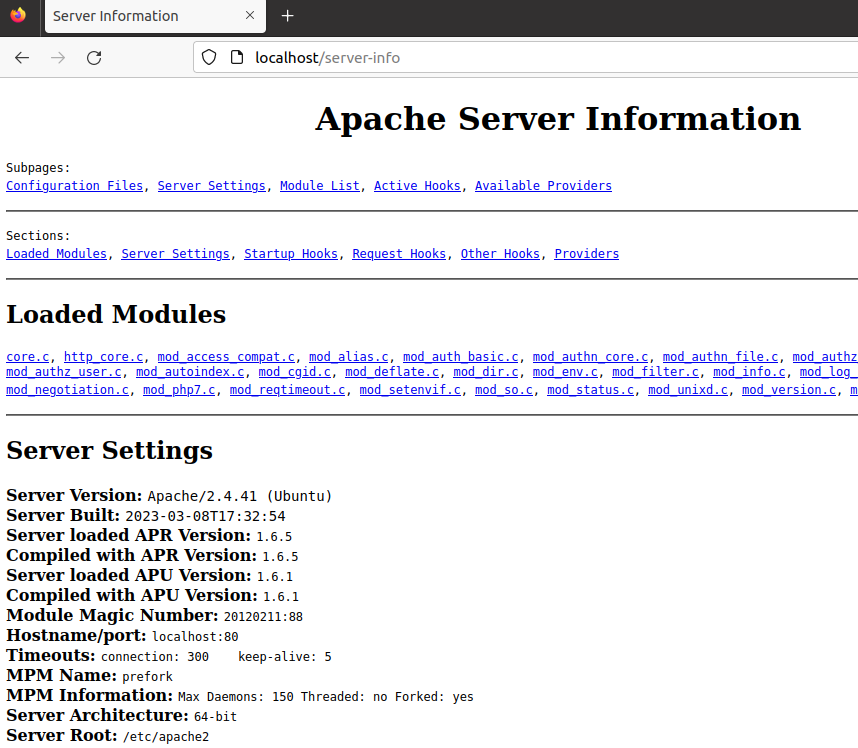
Comandos usados para su instalación:

* # a2enmod info
* #systemctl restart apache2
* #nano /etc/apache2/mods-enabled/info.conf (10.0.2.15 es mi la ip de mi máquina local)



systemctl restart apache2

Resultado:



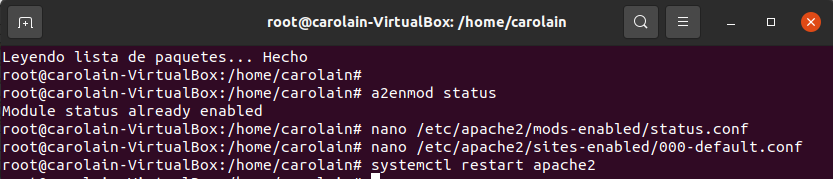
**status**

El módulo mod\_status provee información de la actividad y rendimiento del servidor.

Los detalles que se dan son:

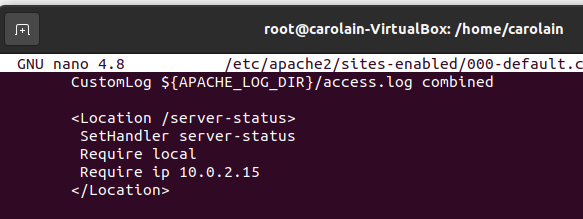
* El estado de cada worker, el número de peticiones que ese worker ha realizado y el número total de bytes servido por el worker (\*)
* Un número total de accesos y recuento de bytes servidos.
* La hora a la que el servidor ha sido arrancado/reiniciado y el tiempo que se ha estado ejecutando
* Medias indicando el número de peticiones por segundo, el número de bytes servido por segundo y la media de bytes por petición.
* El porcentaje actual de CPU usado.
* Los hosts actuales y peticiones que se están procesando en el momento.

Comandos usados para su instalación:

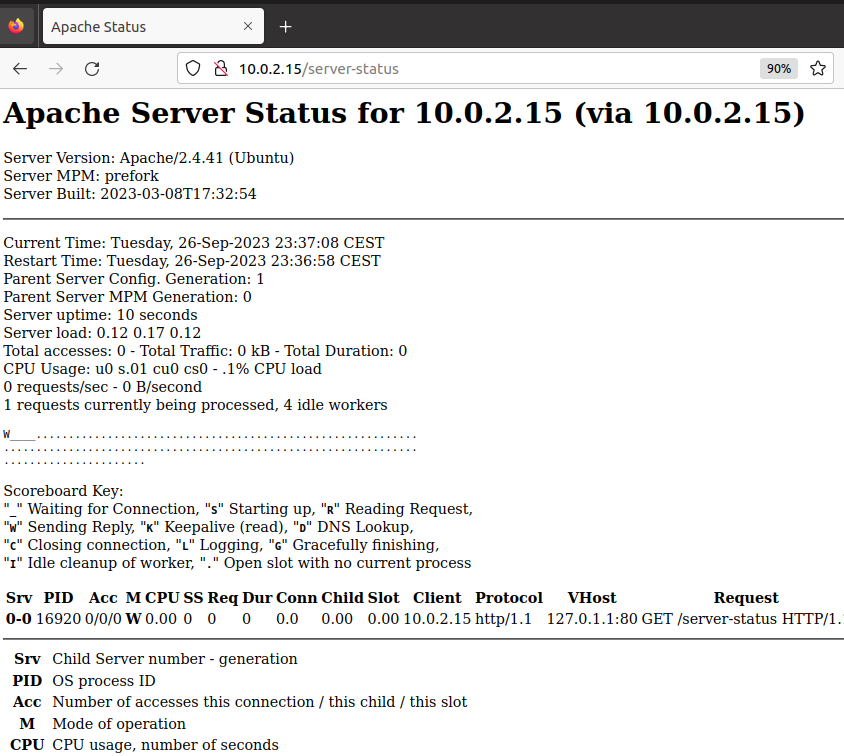


Modificación en 000-default.conf

Añadí las siguientes líneas: (10.0.2.15 es mi la ip de mi máquina local)



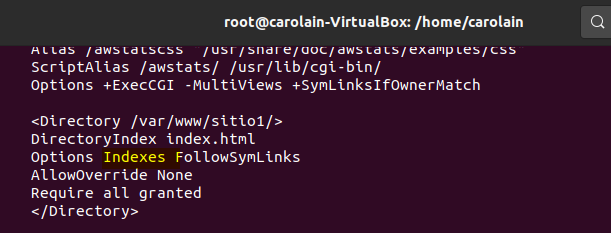
Resultado:



https://javiergarciaescobedo.es/despliegue-de-aplicaciones-web/81-administracion-de-

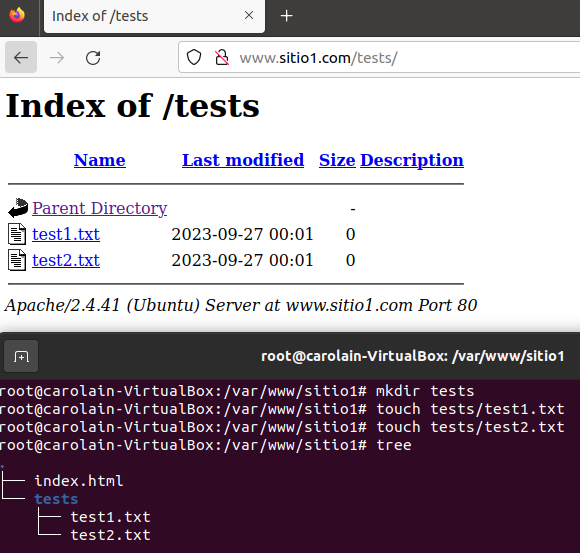
**autoindex**

El módulo mod\_autoindex Genera índices de directorios automáticamente. Por defecto esta ya activo y se habilita en la directiva Options al poner “Indexes”.



Vamos a hacer una prueba:

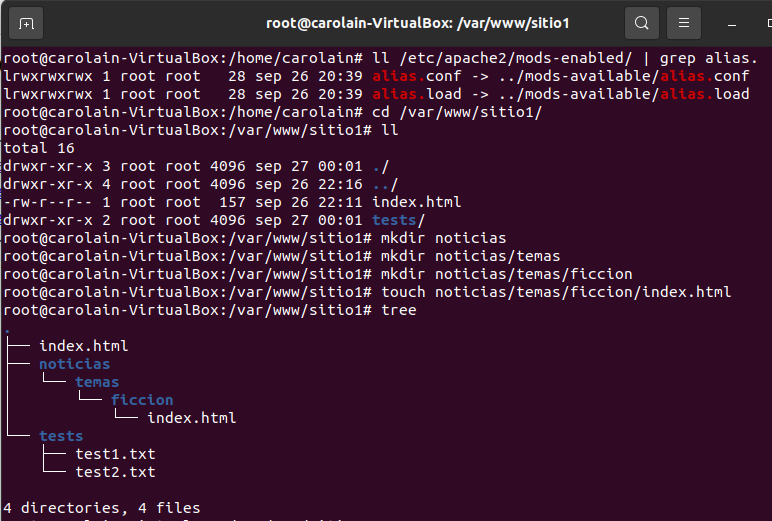
He creado una carpeta llamada test y dentro dos ficheros. Buscando la carpeta en el sitio web me genera un directorio automáticamente.



**alias**

El módulo mod\_alias facilita el mapeo a diferentes partes del sistema de ficheros del host en el árbol de documentos y la redirección de URLs.

He comprobado que está el módulo de alias activo y he creado una ruta de carpetas en [www.sitio1.com](http://www.sitio1.com)



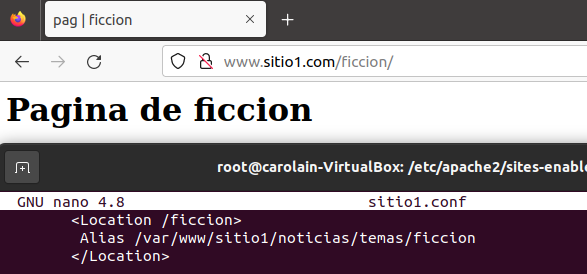
En el sitio de configuración de sitio1.conf añado las siguientes líneas.

* Alias /ficcion /var/www/sitio1/noticias/temas/ficcion
* <Directory var/www/sitio1/noticias/temas/ficcion>
* Require all granted
* </Directory>

Reiniciamos el servicio: #systemctl restart apache2 y el resultado es el siguiente:



Con el uso de Location quedaría así:



**Módulo dir**

El módulo mod\_dir Proporciona redirecciones de "barra diagonal" y entrega de archivos de índice de directorio.

Un archivo escrito por el usuario, normalmente llamado index.html. La directiva DirectoryIndex establece el nombre de este archivo. Esto está controlado por mod\_dir.

De lo contrario, un listado generado por el servidor. Esto lo proporciona mod\_autoindex.