

INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR WEB APACHE



SERVICIOS EN RED
JUAN CARLOS NAVIDAD GARCÍA

Índice:

1. ¿Qué es HTTP y HTTPS?:	3
2. ¿Qué es Webmin?:	3
3. ¿Cómo instalar Webmin?:	4
4. ¿Qué es Apache?:	6
5. ¿Cómo instalar Apache?:	6
6. Funcionamiento de Apache en Webmin:	6
7. ¿Qué son los módulos Apache?:	10
8. ¿Cómo funciona y se configura el módulo UserDir?:	10
9. Comprobación de funcionamiento:	14
10. ¿Qué son los ficheros .htaccess?:	15
11. ¿Cómo crear ficheros .htaccess?:	16
12. Bibliografía:	18



1. ¿Qué es HTTP y HTTPS?:

HTTP, de sus siglas en inglés: "**Hypertext Transfer Protocol**", es el nombre de un protocolo el cual nos permite realizar una **petición de datos y recursos**, como pueden ser documentos **HTML**. Es la base de cualquier intercambio de datos en la Web, y un protocolo de **estructura cliente-servidor**.

Básicamente se basa en sencillas operaciones de **solicitud/respuesta**. Un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado.

En cuanto al **servidor web** o **servidor HTTP**, es un software que forma parte del servidor y tiene como misión principal devolver información (páginas) cuando recibe peticiones por parte de los usuarios.

En otras palabras, es el software que permite que los usuarios que quieren ver una página web en su navegador puedan hacerlo.

Por otra parte, **HTTPS** (**HyperText Transfer Protocol Secure**, **protocolo seguro de transferencia de hipertexto**), es lo mismo pero la comunicación con el cliente está totalmente **cifrada**. Lo que hace este protocolo aún más seguro.

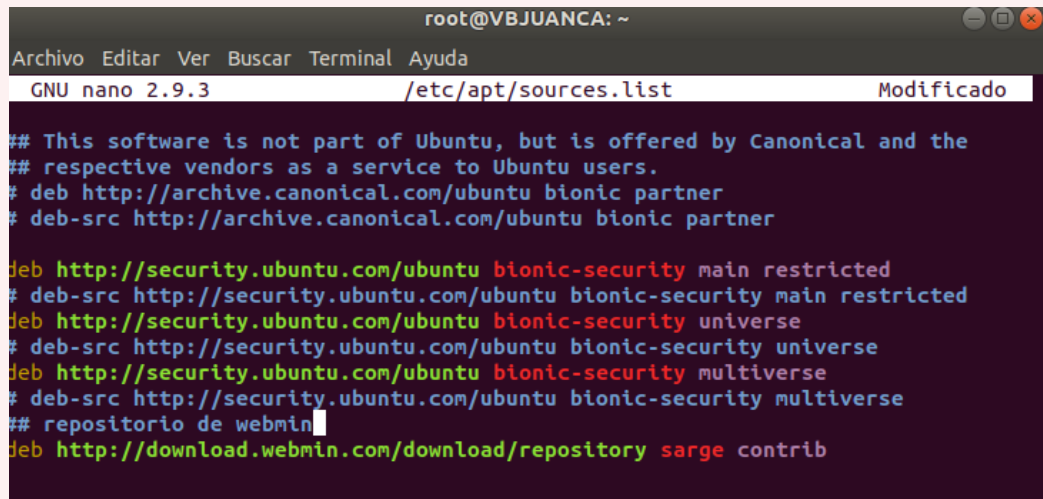
2. ¿Qué es Webmin?:

Para realizar esta práctica, vamos a utilizar un sistema Linux con **Apache2** mediante la interfaz gráfica de **Webmin**. Como ya he dicho en prácticas anteriores **Webmin** es una herramienta que nos proporciona una interfaz gráfica intuitiva y fácil de usar para **administrar el sistema, servidores**, etc.

3. ¿Cómo instalar Webmin?:

- En primer lugar, debemos añadir el repositorio **Webmin** para poder instalar y actualizar **Webmin** fácilmente usando nuestro **administrador de paquetes /etc/apt/sources.list**. Esto se hace agregando el repositorio:

- deb http://download.webmin.com/download/repository sarge contrib**



```
root@VBJUANCA: ~
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3 /etc/apt/sources.list Modificado

## This software is not part of Ubuntu, but is offered by Canonical and the
## respective vendors as a service to Ubuntu users.
# deb http://archive.canonical.com/ubuntu bionic partner
# deb-src http://archive.canonical.com/ubuntu bionic partner

deb http://security.ubuntu.com/ubuntu bionic-security main restricted
# deb-src http://security.ubuntu.com/ubuntu bionic-security main restricted
deb http://security.ubuntu.com/ubuntu bionic-security universe
# deb-src http://security.ubuntu.com/ubuntu bionic-security universe
deb http://security.ubuntu.com/ubuntu bionic-security multiverse
# deb-src http://security.ubuntu.com/ubuntu bionic-security multiverse
## repositorio de webmin
deb http://download.webmin.com/download/repository sarge contrib
```

- A continuación, agrego la **clave PGP de Webmin** para que el sistema confíe en el nuevo repositorio:
- wget http://www.webmin.com/jcameron-key.asc**
- sudo apt-key add jcameron-key.asc**



```
root@VBJUANCA:~# wget http://www.webmin.com/jcameron-key.asc
--2021-10-16 12:47:38-- http://www.webmin.com/jcameron-key.asc
Resolviendo www.webmin.com (www.webmin.com)... 216.105.38.11
Conectando con www.webmin.com (www.webmin.com)[216.105.38.11]:80... conectado.
Petición HTTP enviada, esperando respuesta... 301 Moved Permanently
Ubicación: https://www.webmin.com/jcameron-key.asc [siguiente]
--2021-10-16 12:47:40-- https://www.webmin.com/jcameron-key.asc
Conectando con www.webmin.com (www.webmin.com)[216.105.38.11]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 1320 (1,3K) [text/plain]
Guardando como: "jcameron-key.asc"

jcameron-key.asc 100%[=====] 1,29K --.-KB/s en 0s

2021-10-16 12:47:41 (603 MB/s) - "jcameron-key.asc" guardado [1320/1320]

root@VBJUANCA:~# sudo apt-key add jcameron-key.asc
OK
root@VBJUANCA:~#
```

- Luego, actualizo la lista de paquetes para que incluya el repositorio **Webmin**:
- **sudo apt update**

```

root@VBJUANCA:~# sudo apt update
Obj:1 http://es.archive.ubuntu.com/ubuntu bionic InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu bionic-updates InRelease
Obj:3 http://security.ubuntu.com/ubuntu bionic-security InRelease
Obj:4 http://es.archive.ubuntu.com/ubuntu bionic-backports InRelease
Ign:5 http://download.webmin.com/download/repository sarge InRelease
Des:6 http://download.webmin.com/download/repository sarge Release [16,9 kB]
Des:7 http://download.webmin.com/download/repository sarge Release.gpg [173 B]
Des:8 http://download.webmin.com/download/repository sarge/contrib amd64 Package
s [1.387 B]
Des:9 http://download.webmin.com/download/repository sarge/contrib i386 Packages
[1.387 B]
Descargados 19,8 kB en 2s (9.481 B/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se pueden actualizar 308 paquetes. Ejecute «apt list --upgradable» para verlos.
root@VBJUANCA:~#

```

- Finalmente instalo **Webmin**:
- **sudo apt install webmin**

```

root@VBJUANCA: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@VBJUANCA:~# sudo apt install webmin
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
 fonts-liberation2 fonts-opensymbol gir1.2-gst-plugins-base-1.0
 gir1.2-gstreamer-1.0 gir1.2-gudev-1.0 gir1.2-udisks-2.0
 grilo-plugins-0.3-base gstreamer1.0-gtk3 libboost-date-time1.65.1
 libboost-filesystem1.65.1 libboost-iostreams1.65.1 libboost-locale1.65.1
 libcdr-0.1-1 libclucene-contribs1v5 libclucene-core1v5 libcmis-0.5-5v5
 libcolamd2 libdazzle-1.0-0 libe-book-0.1-1 libdataserverui-1.2-2 libeot0
 libepubgen-0.1-1 libetonyek-0.1-1 libevent-2.1-6 libexiv2-14
 libfreerdp-client2-2 libfreerdp2-2 libgic2 libgee-0.8-2 libgexiv2-2
 libgom-1.0-0 libgpgmepp6 libgpod-common libgpod4 liblangtag-common
 liblangtag1 liblirc-client0 liblua5.3-0 libmediaart-2.0-0 libmshpub-0.1-1
 libodfgen-0.1-1 libqqwing2v5 libraw16 librevenge-0.0-0 libsgutils2-2
 libssh-4 libsuitesparseconfig5 libvncclient1 libwinpr2-2 libxapian30
 libxmlsec1 libxmlsec1-nss lp-solve media-player-info python3-mako
 python3-markupsafe syslinux syslinux-common syslinux-legacy
 usb-creator-common
 Utilice «sudo apt autoremove» para eliminarlos.
 Se instalarán los siguientes paquetes adicionales:
 libauthen-pam-perl

```

4. ¿Qué es Apache?:

El **Servidor Web Apache** es un software de servidor web **gratuito** y de **código abierto multiplataforma** con el cual se ejecutan el 46% de los sitios web de todo el mundo. Es mantenido y desarrollado por la **Apache Software Foundation**.

Les permite a los propietarios de sitios web servir contenido en la web, de ahí el nombre de **<servidor web>**. Es uno de los servidores web más antiguos y confiables, con la primera versión lanzada hace más de 20 años, en 1995.

Cuando alguien quiere visitar un sitio web, ingresa un nombre de dominio en la barra de direcciones de su navegador. Luego, el servidor web envía los archivos solicitados actuando como un repartidor virtual.

5. ¿Cómo instalar Apache?:

Instalar **Apache** es muy simple, así que cuando tengamos instalado **Webmin**, a no ser que ya lo tengamos instalado, introduciremos el siguiente comando en una terminal para poder instalar el paquete Apache:

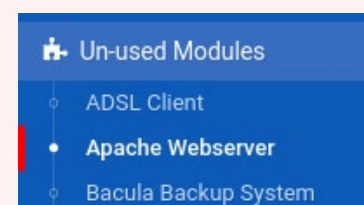
- **sudo apt-get install apache2**

```
jnaveroGZephyrus:~$ sudo apt-get install apache2
[sudo] contraseña para jnav:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  apache2-data apache2-utils
Paquetes sugeridos:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-data apache2-utils
```

6. Funcionamiento de Apache en Webmin:

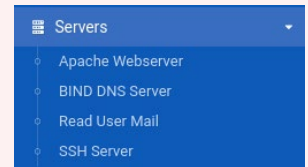
Una vez tengamos instalado el **servidor web Apache**, nos iremos a **Webmin**, recordando que se accede desde el navegador escribiendo **localhost:10000** o **<ip>:10000**.

Una vez dentro de **Webmin**, en el **panel de contenido de la parte izquierda**, nos iremos al apartado *Un-used Modules*, donde nos encontraremos con **Apache Webserver**.

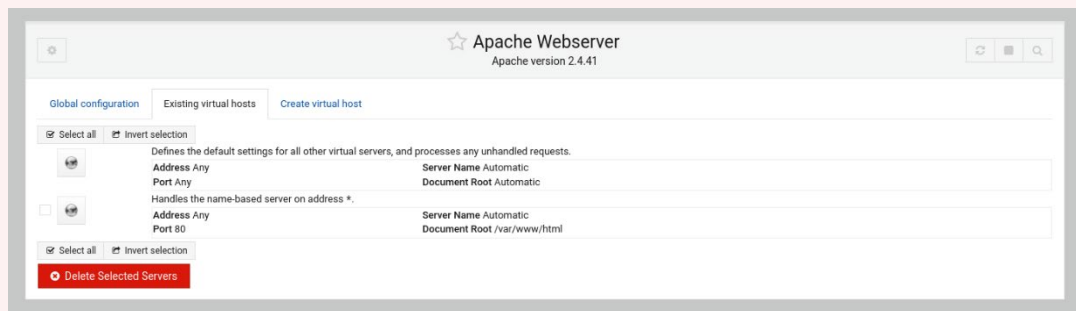


Podemos hacer que el **servidor Apache**, suba al apartado de servidores clicando en el botón *Refresh Modules*.

Después de pulsar el botón *Refresh Modules*, **Webmin** se recargará y ahora, nos saldrá **Apache Webserver** en el apartado **Servers** con todos nuestros demás servidores.

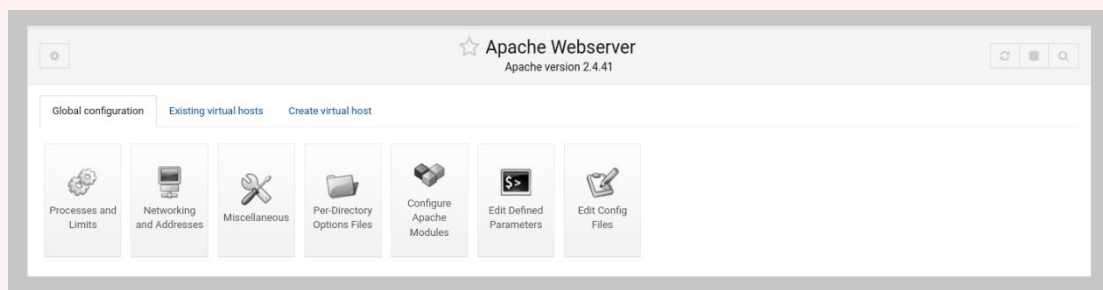


Si accedemos a **Apache Webserver**, la pantalla principal será la de **Hosts virtuales existentes**:



Esta pantalla corresponde a la de los **Hosts virtuales activos o existentes**. Un **host virtual**, o **VirtualHost**, en **Apache** nos permite mantener múltiples nombres de host en nuestro servidor y ubicar en cada uno de ellos una o varias páginas HTML.

La primera pantalla es la de **Configuración local**:



Podemos configurar aquí:

- Límites y procesos.
- Redes y Direcciones.
- Tipos MIME
- Creación de ficheros .htaccess
- Configuración de módulos.
- Edición de parámetros.
- Edición de los ficheros de configuración.

La última pantalla correspondería a la **creación de hosts virtuales**.

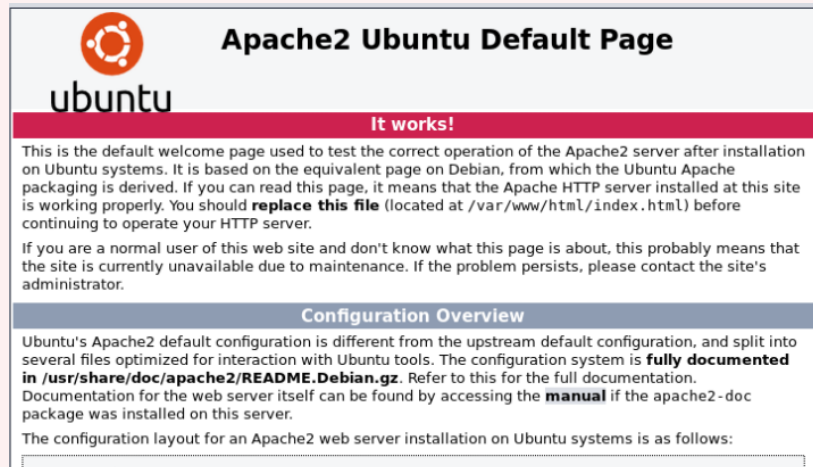
En esta pantalla podremos crear y medio configurar un **nuevo host virtual**, lo cual ya he explicado anteriormente. Tendrá todos los parámetros a configurar que se pueden observar en la captura.

A parte de **Webmin**, como se hace en otros servidores, también se pueden configurar manualmente, el archivo de configuración principal de apache es **/etc/apache2/apache2.conf** y tendría la siguiente forma:

```
GNU nano 4.8 /etc/apache2/apache2.conf
# This is the main Apache server configuration file. It contains the
# configuration directives that give the server its instructions.
# See http://httpd.apache.org/docs/2.4/ for detailed information about
# the directives and /usr/share/doc/apache2/README.Debian about Debian specific
# hints.
#
# Summary of how the Apache 2 configuration works in Debian:
# The Apache 2 web server configuration in Debian is quite different to
# upstream's suggested way to configure the web server. This is because Debian's
# default Apache2 installation attempts to make adding and removing modules,
# virtual hosts, and extra configuration directives as flexible as possible, in
# order to make automating the changes and administering the server as easy as
# possible.
#
# It is split into several files forming the configuration hierarchy outlined
# below, all located in the /etc/apache2/ directory:
#
# /etc/apache2/
# |-- apache2.conf
# |   |-- ports.conf
# |   |-- mods-enabled
# |   |   |-- *.load
# |   |   |-- *.conf
# |   |-- conf-enabled
# |   |   |-- *.conf
# |   |-- sites-enabled
# |       |-- *.conf
```


Por otro lado, si lo que queremos es configurar la página web principal de **Apache**, nos debemos de ir al directorio `/var/www/`, allí se encontrará el **index.html**, la página principal de **Apache**. Esta la podremos modificar por cualquier otra, incluso hacer uso de **javascript**, **hojas de estilos**, básicamente páginas web completas.

La **página principal de Apache** sería así:



En mi caso, he modificado el código del `/var/www/index.html` por este:



La página se vería así:



7. ¿Qué son los módulos Apache?:

Un **módulo** es una parte independiente de un programa. La mayor parte de la funcionalidad de **Apache** está contenida en **módulos** que pueden incluirse o excluirse.

Entonces, el **servidor HTTP Apache** es un programa **modular** en el que el administrador puede elegir qué funcionalidades se incluyen mediante la selección de un **conjunto de módulos**.

Los módulos se guardan en la configuración de **Apache** en **dos directorios**:

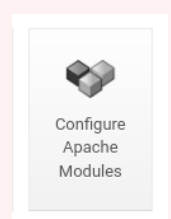
- */etc/apache2/mods-available/*: Directorio que contiene los módulos disponibles en la instalación actual.
- */etc/apache2/mods-enabled/*: Directorio que incluye mediante enlaces simbólicos al directorio anterior, los módulos que se van a cargar en memoria la próxima vez que se inicie Apache.

8. ¿Cómo funciona y se configura el módulo UserDir?:

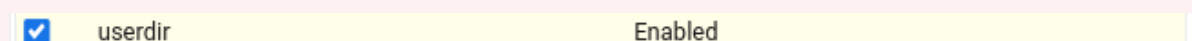
En sistemas con **múltiples usuarios**, cada usuario puede tener un sitio web en su directorio **home** usando el módulo **UserDir**.

Los visitantes de una URL "*http://<IP o dominio>/<nombre de usuario>/*" recibirán el contenido del directorio **/home/<usuario>/public_html** del usuario.

Para activar el módulo **UserDir** nos iremos al **servidor web Apache** en **Webmin**, nos iremos al apartado de **configuración global** y **configuración de los módulos Apache**:

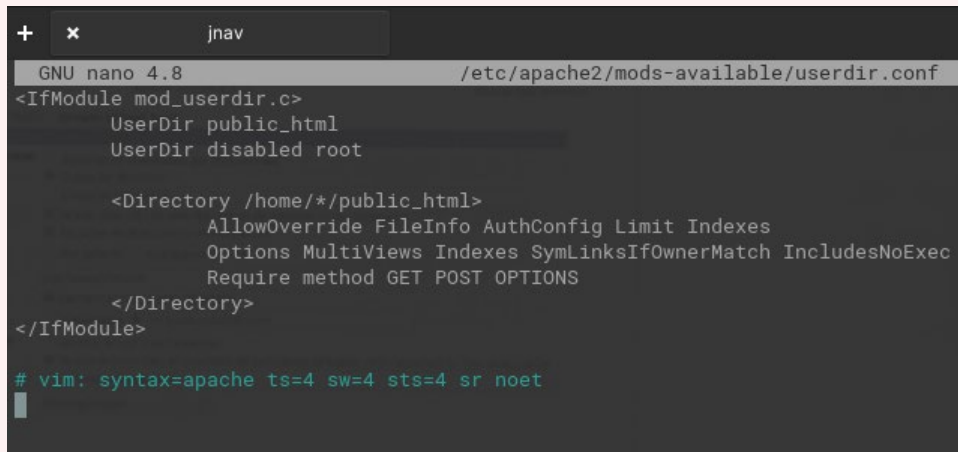


Nos encontraremos con una gran lista de **módulos**, en específico buscaremos el módulo **UserDir** y clicaremos sobre su **checkbox** para activarlo:



Una vez activado, el módulo **UserDir** puede ser configurado desde la **interfaz gráfica de Webmin** o editando directamente su **archivo de configuración**.

Desde **Webmin** se haría directamente configurando el **host virtual**, pero desde el **archivo de configuración**, se haría editando el fichero */etc/apache2/mods-available/userdir.conf*.



```
GNU nano 4.8 /etc/apache2/mods-available/userdir.conf
<IfModule mod_userdir.c>
    UserDir public_html
    UserDir disabled root

    <Directory /home/*/public_html>
        AllowOverride FileInfo AuthConfig Limit Indexes
        Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
        Require method GET POST OPTIONS
    </Directory>
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

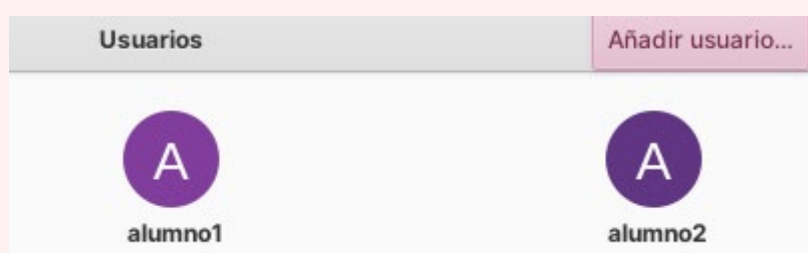
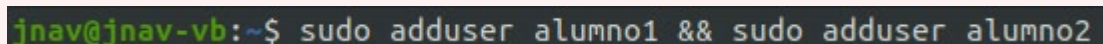
En este fichero lo más importante, es que podemos editar donde queremos que se encuentre el directorio donde el usuario va a alojar su página web, por defecto es el directorio */home/usuario/public_html*.

Si queremos cambiarlo, deberemos modificar la **segunda** y **quinta línea** indicando la nueva ruta.

Para poner en uso el **módulo UserDir**, primero debemos **crear los usuarios** en los que queremos albergar una página web si es que no los tenemos creados ya.

Entonces, para esta práctica, crearemos dos usuarios, **alumno1** y **alumno2**. Los podemos crear rápidamente con el comando:

- **sudo adduser alumno1 && sudo adduser alumno2**



Cuando hayamos creado los dos usuarios de prueba, crearemos el directorio *public_html/* en el home de cada usuario.

Para crearlos de manera rápida, podemos usar el siguiente comando:

- `sudo mkdir /home/alumno1/public_html /home/alumno2/public_html`

```
jnav@jnav-vb:~$ sudo mkdir /home/alumno1/public_html /home/alumno2/public_html
jnav@jnav-vb:~$ ls /home/alumno1/ /home/alumno2/
/home/alumno1/:
public_html

/home/alumno2/:
public_html
jnav@jnav-vb:~$
```

Una vez creados los directorios *public_html/*, vamos a añadirles una **página HTML** a cada uno llamada **index.html**. En mi caso he utilizado la plantilla utilizada para el sustituir la **página principal de Apache**, pero si queremos crearla de cero, se puede crear con el siguiente comando:

- `sudo nano /home/alumno1/public_html/index.html`

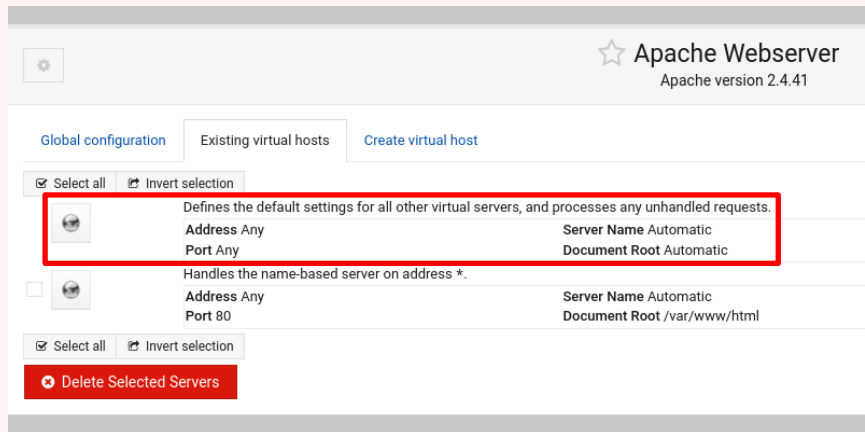
Se nos abrirá el editor de texto **nano** y ahí introduciremos el **código HTML**, mi código para cada uno de los usuarios es el siguiente:

```
GNU nano 4.8 /home/alumno1/public_html/index.html
<!DOCTYPE html>
<html>
<head>
  <title>Alumno 1</title>
  <meta charset="utf-8" />
</head>
<body style="background-color: #FFF1ED;">
  <p style="color:#CD5C5C; font-size:50px; text-align: center; font-family: arial; font-weight:bold;">Bienvenido a la página principal del:<br>
  Servidor Web Apache2 del alumno1 de Juan Carlos Navidad.<br>
  rog232.aulaSER232.com.<br></p>
  
</body>
</html>
```

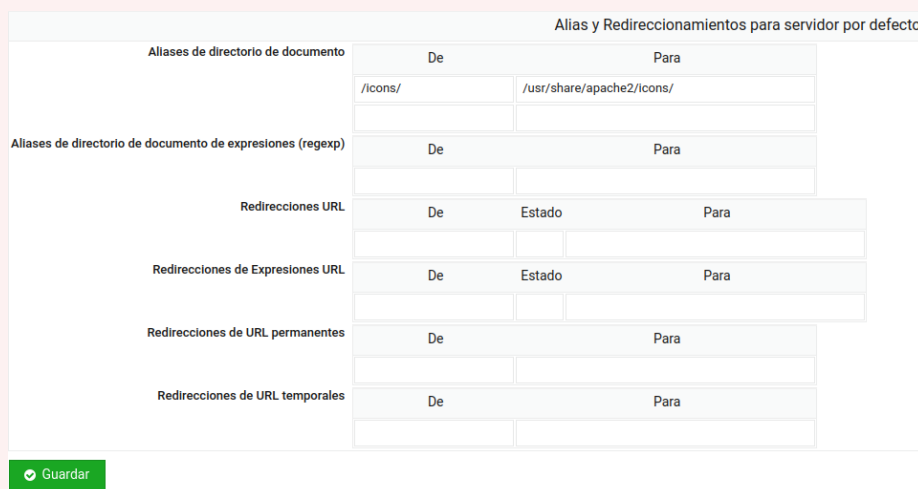
```
GNU nano 4.8 /home/alumno2/public_html/index.html
<!DOCTYPE html>
<html>
<head>
  <title>Alumno 2</title>
  <meta charset="utf-8" />
</head>
<body style="background-color: #FFF1ED;">
  <p style="color:#CD5C5C; font-size:50px; text-align: center; font-family: arial; font-weight:bold;">Bienvenido a la página principal del:<br>
  Servidor Web Apache2 del Alumno2 de Juan Carlos Navidad.<br>
  rog232.aulaSER232.com.<br></p>
  
</body>
</html>
```

Después de tener creadas los **index.html** o, mejor dicho, las páginas que se van a mostrar en cada uno de los usuarios, debemos configurar el **host virtual** para que muestre nuestras páginas al introducir nuestro **servidor** o **IP** en el navegador.

Para eso, nos iremos a **Apache Webserver** dentro de **Webmin**, en el apartado **Existing virtual hosts**, clicaremos sobre el **primer host virtual** que aparece:



Nos aparecerá la siguiente pantalla:



De esta pantalla solamente utilizaremos el **primer apartado, Alias de directorio de documento**. En el cuál borraremos la **primera entrada: /icons/ /usr/share/apache2/icons/**.

Una vez eliminada, escribiremos lo siguiente:

Alias de directorio de documento	
De	Para
/alumno1/	/home/alumno1/public_html/
/alumno1	/home/alumno1/public_html/
/alumno2/	/home/alumno2/public_html/
/alumno2	/home/alumno2/public_html/

Introduciremos los usuarios por duplicado, para que cuando introduzcamos la URL, acepte tanto **ip/alumnoX** como **ip/alumnoX/**, a continuación, se podrá ver en las capturas.

Una vez rellenados estos campos, le daremos a *Guardar* y a *Aplicar cambios*.

9. Comprobación de funcionamiento:

Después de haber realizado todos los pasos anteriores de manera correcta, debería de funcionar a la perfección, así que, para ponerlo a prueba, comprobaremos que funciona, introduciendo las **URLs** desde el navegador:



Incluso si tenemos activo el **servidor DNS** y nuestra **dirección IP** tiene asignada un **dominio**, podemos introducir el **dominio más /usuario**:



No se observa, pero si yo introduzco tanto *192.168.103.232/alumno1* como si introduzco *192.168.103.232/alumno1/*, me carga la página igualmente.

10.¿Qué son los ficheros .htaccess?:

El archivo **.htaccess** (**acceso de hipertexto**) es un archivo oculto que se utiliza para configurar funciones adicionales para sitios web alojados en el **servidor web Apache**. Permiten a los usuarios que no tienen permisos para modificar la configuración, de esa manera poder ejercer algún control sobre el comportamiento de su parte del servidor.

Entre lo que pueden hacer, puede **reescribir la URL**, **proteger directorios con contraseña**, **habilitar la protección de enlaces directos**, **no permitir el acceso a direcciones IP específicas**, **cambiar la zona horaria de tu sitio web** o **alterar la página de índice predeterminada**, y mucho más.

Se debe situar en el directorio al que tiene que afectar, junto a todos sus subdirectorios, y las modificaciones introducidas que no requieren reiniciar el servidor web.

11. ¿Cómo crear ficheros .htaccess?:

Hay dos maneras para crearlos, podemos **crearlos manualmente** o mediante la **interfaz gráfica de Webmin**.

Para crearlo manualmente, abre tu editor de texto favorito y ahora sólo necesitas saber que código agregar, pondré algún que otro código que puede ser útil:

- **Bloquear IPs**

```
Order Deny, Allow
```

```
Deny from ip1
```

```
Deny from ip2
```

- **Bloquear acceso a archivos del servidor**

```
Order deny,allow
```

```
Deny from all
```

```
<Files ~ ".(xml|css|js|jpeg|png|gif|pdf|zip|rar)$">
```

```
Allow from all
```

```
</Files>
```

- **Redirecciones**

```
RewriteEngine on
```

```
RewriteCond %{HTTP_HOST} !^$
```

```
RewriteCond %{HTTP_HOST} !^www\. [NC]
```

```
RewriteCond %{HTTPS}s ^on(s)|
```

```
RewriteRule ^ http%1://www.%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
```


Para crearlo desde la **interfaz gráfica de Webmin** debemos irnos dentro del **Servidor Web Apache** al apartado **Configuración Global**, y dentro de este apartado, clicar sobre *Archivos de Opciones de Por-Directorio*.



Dentro de este apartado, en la **primera línea** debemos indicar la ruta donde se va a hacer efectivo el **archivo .htaccess**.

La **segunda línea** nos da la opción de poder **importar un archivo .htaccess** creado manualmente, si le clicamos en el botón, los buscará **automáticamente**, también podemos **introducir la ruta directamente**.

Se pueden especificar en un archivo (normalmente llamado .htaccess) opciones adicionales de por-directorio dentro de cada directorio, en otro archivo de opciones.

☒ Automáticamente
 ☐ Desde el directorio
 /

Si por ejemplo creamos o importamos un archivo **.htaccess**, este a continuación nos aparecerá más arriba dentro del mismo apartado:

Archivos de opciones existentes

/home/server.htaccess

☒ Automáticamente
 ☐ Desde el directorio
 /

Si clicamos sobre el fichero **.htaccess**, podremos configurarlo con una gran variedad de opciones:

☆ Archivo de Opciones de Por-Directorio
/home/server.htaccess

Límites y Procesos	Control de Acceso	Opciones de Documento	Tipos MIME	Manejo de Errores	Alias y Redireccionamientos	Programas CGI	Indicador de Directorio	Filtros	Idiomas	Ver Directivas	Editar Directivas
--------------------	-------------------	-----------------------	------------	-------------------	-----------------------------	---------------	-------------------------	---------	---------	----------------	-------------------

Crear Opciones de Por-Directorio

¿Expresión? * Coincidencia exacta ☐ Coincidir con expresión ☐

Caminio

12. Bibliografía:

1. ¿Qué es HTTP y HTTPS?:

<https://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/http.html>

<https://www.webempresa.com/hosting/que-es-servidor-web.html>

<https://developer.mozilla.org/es/docs/Web/HTTP/Overview>

2. ¿Qué es Apache?:

<https://www.hostinger.es/tutoriales/que-es-apache/>

3. ¿Qué son los módulos Apache?:

<http://luismido.wikidot.com/modulos-en-apache-2>

<https://plataforma.josedomingo.org/pledin/cursos/apache24/curso/u20/>

4. ¿Qué son los ficheros .htaccess?:

Libro

<https://www.hostinger.es/tutoriales/que-es-el-archivo-htaccess>

5. ¿Cómo crear ficheros .htaccess?:

<https://www.hostinger.es/tutoriales/que-es-el-archivo-htaccess>