6. Servidor Web Apache.

Configuración con Webmin

IES Gonzalo Nazareno.

Instalación y mantenimiento de Servicios de Internet

María Jesús Gragera Fernández.

6.1. Introducción.

- **Apache** es el servidor web más utilizado en el mundo actualmente, es el encargado de servir las peticiones realizadas a través del protocolo **http**, de ahí que el demonio se denomine **httpd**.
- Gestiona las páginas de nuestro propio sitio Web.
- Es mantenido por **The Apache Software Foundation** que también se encarga de prestar soporte a otros proyectos como **PHP** (Personal Home Page).
- Su página oficial es www.apache.org.
- Podemos entrar en www.netcraft.com dedicada a estadísticas de uso de servidores, dominios, sistemas operativos, etc. Seguir el enlace Web server survey->What's that site running? Rellenamos el campo con el nombre de la máquina y usamos Examinar.
- Por defecto escucha en el puerto **80.**

- En función del tipo de páginas, **Apache** puede servir la información de varias formas.

  a) <u>Páginas estáticas:</u> es el modo básico y más primitivo,el cliente solicita una página a apache y éste le envía el código html, que será interpretado por el navegador del cliente. El cliente va pidiendo al servidor todos los elementos que contiene la página: imágenes, sonido, etc.

  Puede que con un servidor Linux de bajas prestaciones (puede ser un 486) consigamos estupendos resultados, si es esto lo que queremos.

  b) <u>Páginas dinámicas:</u> en ellas la información cambia constantemente, se mantienen actualizadas generándolas dinámicamente desde una base de datos. Para generar este tipo de contenido dinámico **Apache** dispone de varias herramientas.

- **Intérpretes incluidos en Apache:** están incluidos en el propio servidor, por lo que no es necesario lanzar un proceso externo para su ejecución. El más utilizado es PHP.
- **Servlets de Java:** programas escritos con Java que se ejecutan en el servidor.

- Para instalar Apache lo haremos con el siguiente comando

  # apt-get install apache2

- Directorios más importantes

/etc/apache2/: archivos de configuración. El fichero de configuración es apache2.conf

/var/www/: directorio raiz de documentación; donde debemos montar todas las páginas web incluida la de inicio. (index.htm, index.html, index.shtml)

/usr/share/doc: documentación

/usr/sbin: donde se encuentra el ejecutable.

Una vez hecha la instalación:

1. Iniciamos el servicio si no lo está:

/etc/init.d/apache2 start

/etc/init.d/apache2 restart

1. Comprobamos que funciona correctamente desde el navegador
   http://localhost

- Los archivos de configuración de apache2 se encuentran en la carpeta **/etc/apache2.**
  - El archivo principal de configuración es:

/etc/apache2/apache2.conf.

  - Antes de realizar cualquier cambio en este archivo, es conveniente realizar una copia de seguridad del mismo ya que si apache encuentra algún error en el archivo de configuración, no arrancará.

  - Se pueden configurar infinidad de parámetros. Aquí, para poner en marcha el servidor, editaremos el archivo apache2.conf y añadiremos únicamente el siguiente parámetro:

// Añadir en apache2.conf

ServerName www.gonzanaza.com

6.2. Configuración a través de Webmin.

- Accedemos a través de la categoría **Servidores**, puede dar error porque la ruta especificada para el ejecutable de Apache no sea la correcta. Este archivo se encuentra en **/usr/sbin/apache2**.

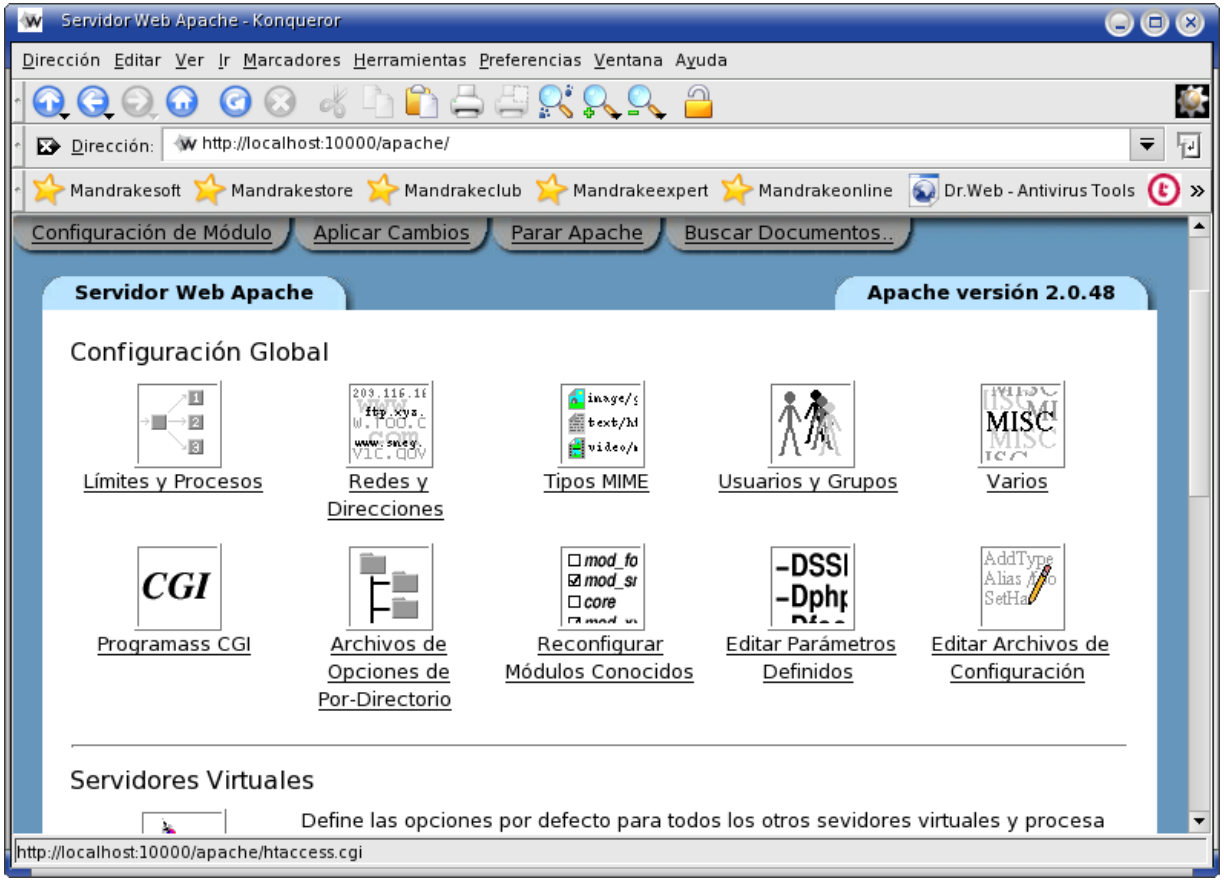
- Lo especificamos en el **“Módulo Servidor Web Apache”** en la casilla **“Trayectoria ejecutable httpd”** especificamos **/usr/sbin/apache2.**

- También debemos de especificar cual es el archivo de configuración, en **“Trayectoria a httpd.conf”** ponemos **/etc/apache2/apache2.conf** y en **“Archivo donde añadir servidores virtuales”** debemos poner /**etc/apache2/sites-available**.

Se presentan dos zonas diferenciadas:

.- Configuración Global.

.-Servidores Virtuales.

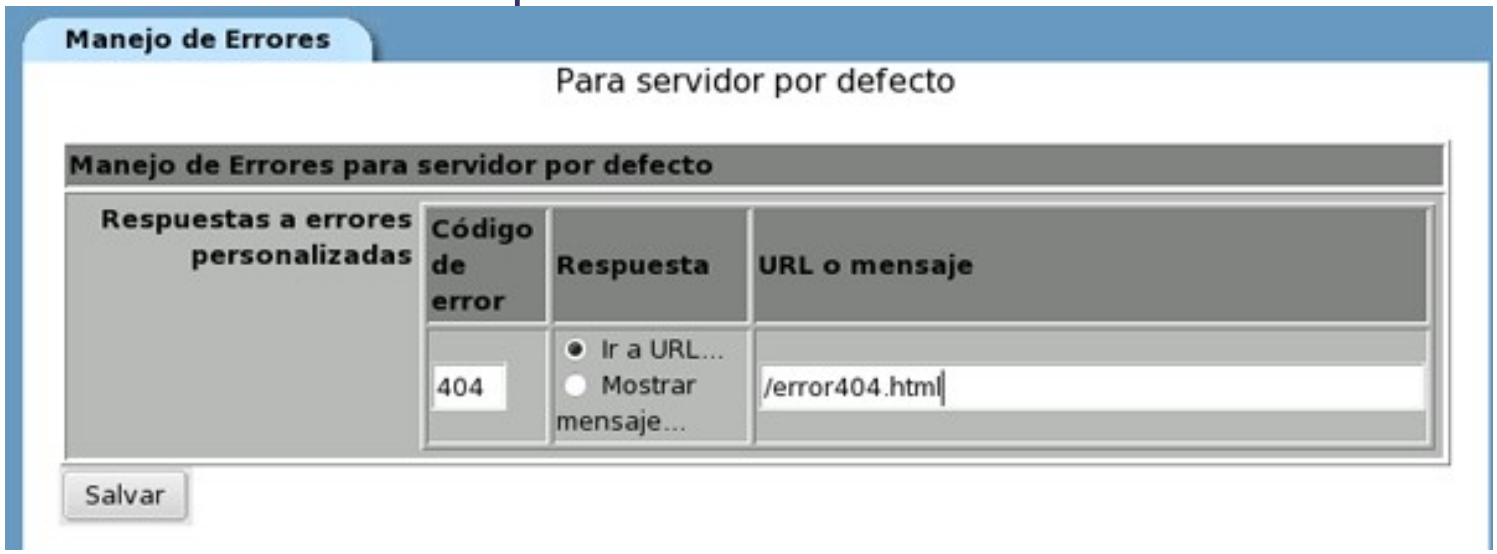


- Las opciones se pueden aplicar de forma global o a elementos concretos: servidores virtuales, localizaciones, directorios, ficheros.
- De forma que las opciones especificadas para un elemento afectarán a todos los que cuelgan de él en su jerarquía de elementos.
- Veremos las siguientes opciones de configuración:
  1. Control de errores.
  2. Opciones de directorio.
  3. Servidores Virtuales
  4. Control de acceso y autentificación

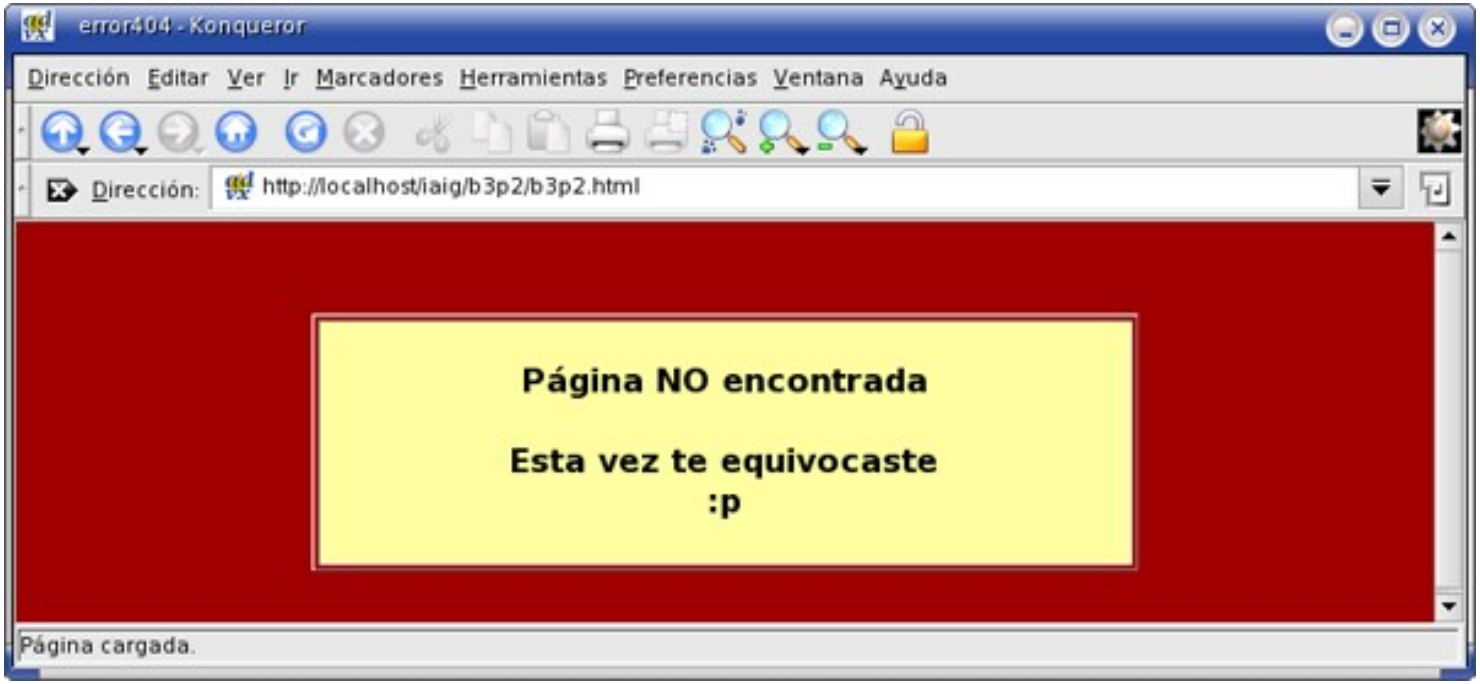
6.2.1.Manejo de Errores.

- Cuando se solicita una página que no existe, o una página a la que tenemos prohibido el acceso, el servidor nos da un mensaje de error.
- Estos errores están codificados en apache, **404 para not found**, **403 para Forbidden**...

- Para cambiar la página de error que da por defecto (código 404) nos vamos a **Servidores Virtuales->Servidor por defecto->Manejo de Errores.**
- Configuramos el error 404 para página no encontrada creando una página web de error hecha a nuestra medida, la llamaremos error404.html y la colgaremos de **/var/www**, que es el directorio de documentación por defecto.



- Ahora al intentar ir a una página que no exista nos saldrá la creada por nosotros.



6.2.2. Opciones de Documento.

- **Servidores Virtuales->Servidor por defecto->Opciones de documento.**
- Se aplicaran a todos los documentos salvo que se especifiquen casos particulares con otras opciones.
- **Directorio raíz para documentos:** Es por defecto **/var/www**. Si no deseamos que sea ese, creamos otro directorio y ponemos la trayectoria en **Servidor por defecto->Opciones de directorio.**
- **Directorio www de usuario:** Es el nombre de un subdirectorio home de cada usuario del sistema. (public_html).
  - Este directorio no está creado lo creamos en el /home/usuario. Luego cualquiera desde otra máquina pude entrar siempre que tenga permiso de ejecución con http://nombre_maquina.domino/~usuario (~con ALTGR 4). En este directorio se cuelgan las páginas que el usuario quiere ofrecer públicamente.

- Vamos a crear índices de directorio para usuario.

Directory /home/*/public_html

- Si la dirección url solicitada acaba en un nombre de directorio donde no hay una página index.html podemos especificar que aparezca un índice con el contenido del directorio.
- En **opciones de documento** ponemos sí a generar índices de directorio.

Opciones de Documento - Konqueror

Dirección Editar Ver Ir Marcadores Herramientas Preferencias Ventana Ayuda

Dirección: https://localhost:10000/apache/edit_dir.cgi?virt=&idx=276&type=5

directorio

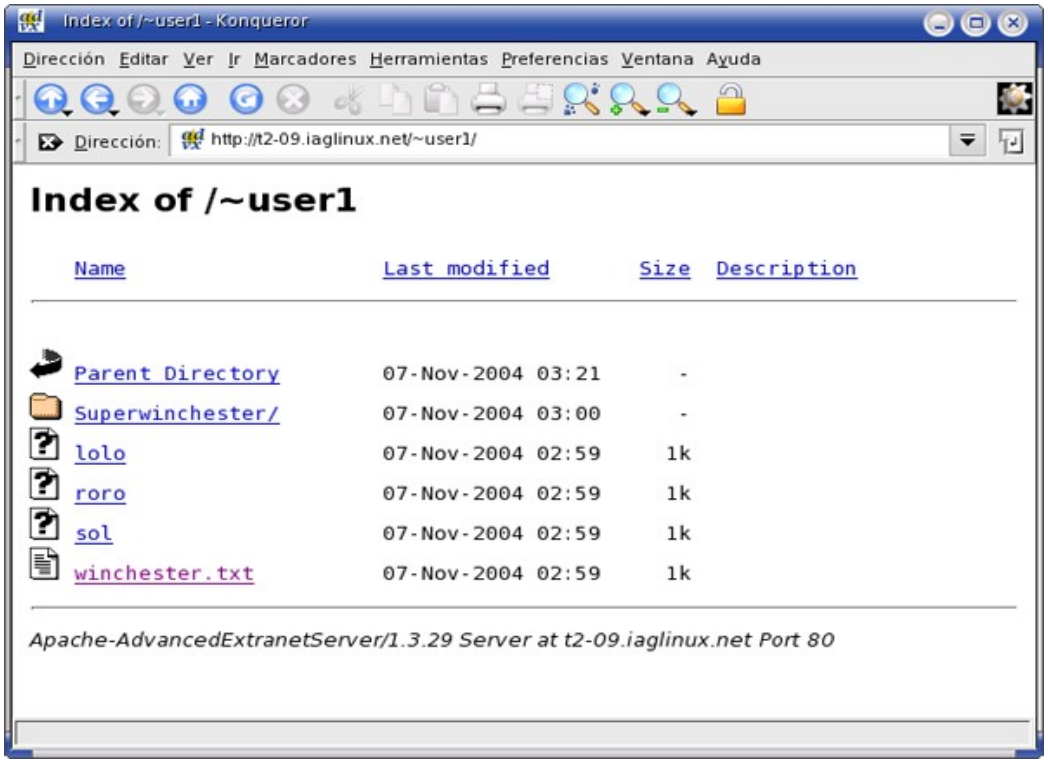
Opción	Configurar para el directorio	Combinar con el padre
Ejecutar programas CGI	○ Si ● No	○ Activar ○ Desactivar
Seguir enlaces simbólicos	● Si ○ No	○ Activar ○ Desactivar
Inclusiones y ejecuciones del lado del servidor	● Si ○ No	○ Activar ○ Desactivar
Inclusiones del lado del servidor	○ Si ● No	○ Activar ○ Desactivar
Generar índices de directorio	● Si ○ No	○ Activar ○ Desactivar
Generar multivistas	● Si ○ No	○ Activar ○ Desactivar
Seguir enlaces simbólicos si el propietario coincide	○ Si ● No	○ Activar ○ Desactivar

El archivo de opciones puede pasar por alto... ○ Por defecto ● Seleccionado debajo...

- ☒ Opciones de autentificación
- ☒ Tipos MIME y codificaciones
- ☒ Indexaciones y archivos de índice

Generar resumen MD5 ○ Si ○ No ● Por defecto

- Al generar índices de directorios al ir al directorio en concreto y no haber una página index.html aparecerán los directorios y ficheros de ese sitio.



6.2.3. Servidores Virtuales.

1. Permite alojar varios dominios en una sola máquina.
2. Permite crear subdominios.
3. Se pueden basar en IP o en nombre.
4. Un host virtual hereda toda la configuración del sitio principal, a no ser que se redefina.

6.2.3.1. Host virtuales basados en nombre.

- Nos permite albergar varios sitios Web en nuestro servidor **Apache** con distinto nombre pero con la misma IP.
- Vamos a crear dos dominios virtuales basados en nombre.
- Cada dominio debería de estar dado de alta en el **servidor DNS**, y luego en **Apache** creamos dos servidores virtuales con directorio raíz diferente, páginas que sirven distintas, restricciones de acceso distintas etc.

- **Apache** se encargará de dirigir las peticiones que llegan a la misma IP a uno u otro servidor virtual, puesto que las distingue por el nombre.

- Al editar el fichero encontramos dos configuraciones de ejemplo, una basada en IP y otra en nombre, pero las dos aparecen comentadas, por lo que podemos tomarlas como referencia para hacer las nuestras en este mismo fichero para tenerlas identificadas.

- Veamos ahora un **ejemplo** de una configuración de dos hosts virtuales basados en nombre, haremos que nuestra máquina responda a peticiones de dos dominios distintos sobre la misma IP:

```
NameVirtualHost 192.168.1.X
<VirtualHost 192.168.1.X>
ServerName servidor1.miservidor.org
DocumentRoot /var/www/servidor1
</VirtualHost>
<VirtualHost 192.168.1.X>
ServerName servidor2.miservidor.org
DocumentRoot /var/www/servidor2
</VirtualHost>
```

- La directiva **NameVirtualHost** especifica la dirección IP sobre la que se van a montar los host virtuales,
- Se utilizan también contenedores **<VirtualHost></VirtualHost>,** todo lo que se incluya en ellos sólo afectará al host virtual.
- **ServerName** especifica el dominio y **DocumentRoot** el directorio raíz del host virtual, evidentemente se puede utilizar dentro del host virtual cualquier otra directiva de Apache.

Práctica: Crear un par de dominios virtuales www-1.gonzanaza.com y www-2.gonzanaz.com con la misma IP.

- Para crearlos es necesario primero cambiar los DNS.

#vi /etc/bind9/gonzanaza.com.hosts

- Agregamos www-1.gonzanaza.com correspondiente a 192.168.1.X. y www-2.gonzanaza.com con la misma ip.
- Serán servidores virtuales basados en nombres.

- Reiniciamos el DNS y comprobamos que funciona.
- Comprobamos con # host www-1.sint.net y nos responde con la dirección IP 192.168.1.X
- A continuación los creamos en **Apache** mediante **Webmin** *(se puede hacer también mediante el fichero de configuración /etc/sites-available/default).*
- Servidor Web Apache->Crear un nuevo Servidor Virtual.

Dirección: 192.168.1.X

Puerto: por defecto

Raíz para documentos: /var/www/www-1

Nombre del servidor: www-1.gonzanaza.com

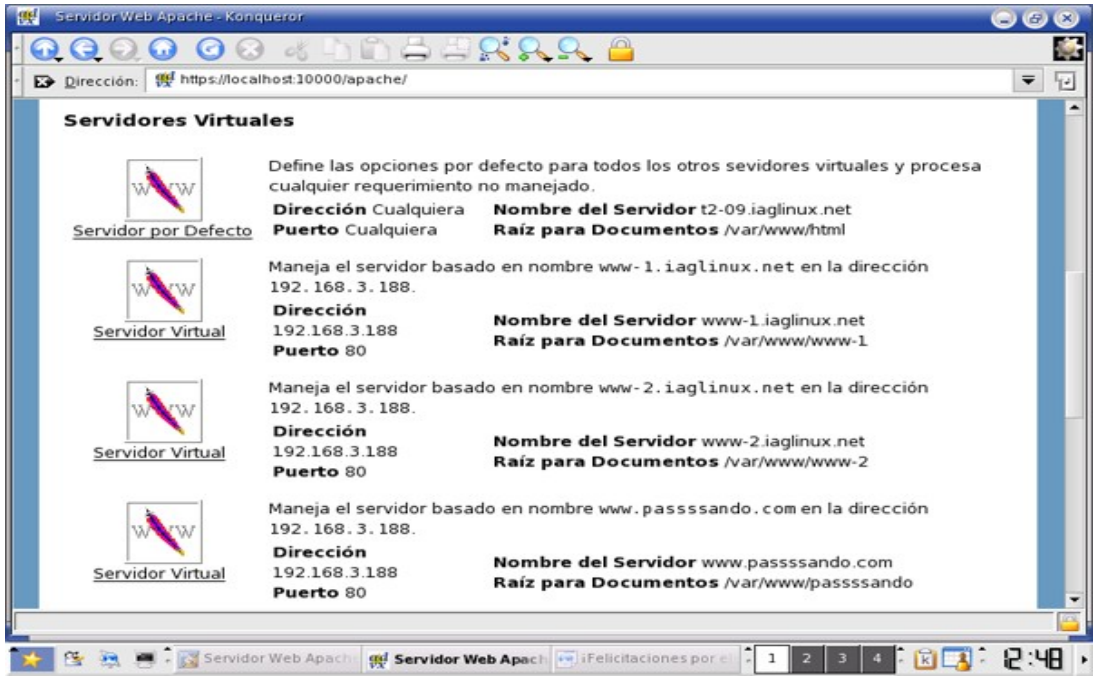
Importante:

El servidor por defecto pasa a ser el primer servidor virtual definido, así:

http://www.gonzanaza.com irá a http://www-1.gonzanaza.com,

Sólo puedo acceder al servidor por defecto original poniendo http://localhost

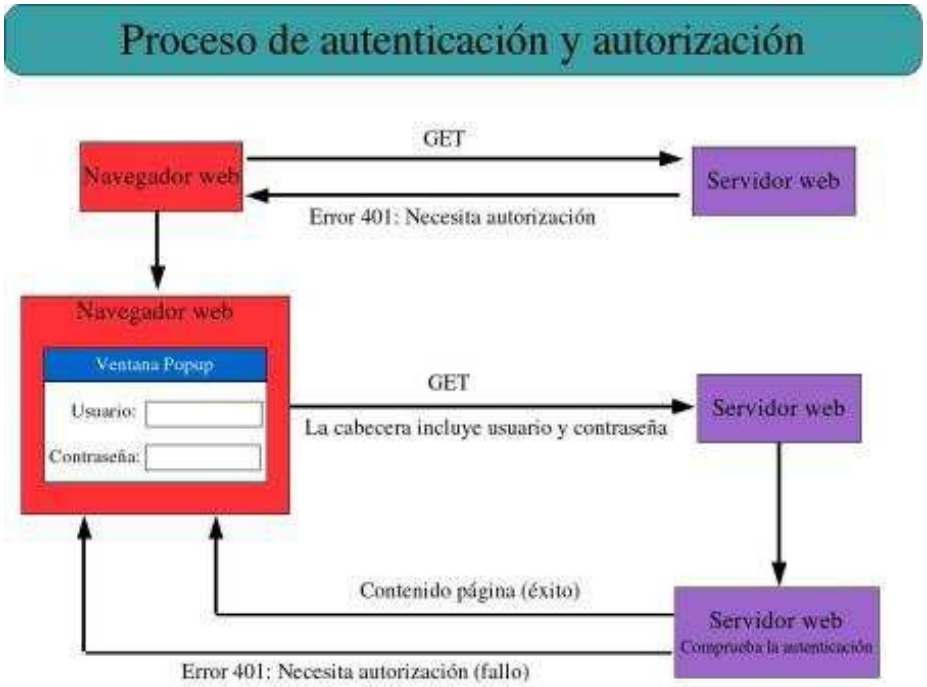
- Apache no envía peticiones al servidor por defecto cuando existe uno virtual, es decir, si necesitamos un servidor virtual, todos tendrán que ser virtuales para su correcto funcionamiento.
- Para la misma IP podremos crear tantos servidores virtuales como queramos.



6.2.4. Control de Acceso y Autentificación.

- Podemos conseguir que para acceder a determinados recursos un cliente tenga que autenticarse ante el servidor.
- La información contenida en esa zona sólo deberá ser vista por el usuario o grupo de usuarios que establezcamos.
- **Apache** tiene varios módulos que permiten realizar la autenticación de usuarios, el módulo por defecto es mod_auth. Mod_auth almacena las credenciales en ficheros de texto plano, si bien las contraseñas se guardan encriptadas.
- El uso de este módulo está recomendado para sistemas que no superan los cien usuarios.
- El proceso de autenticación es simple. El navegador solicita la página, el servidor mediante un código 401 solicita el identificador y la contraseña y el cliente envía su nombre y contraseñas.

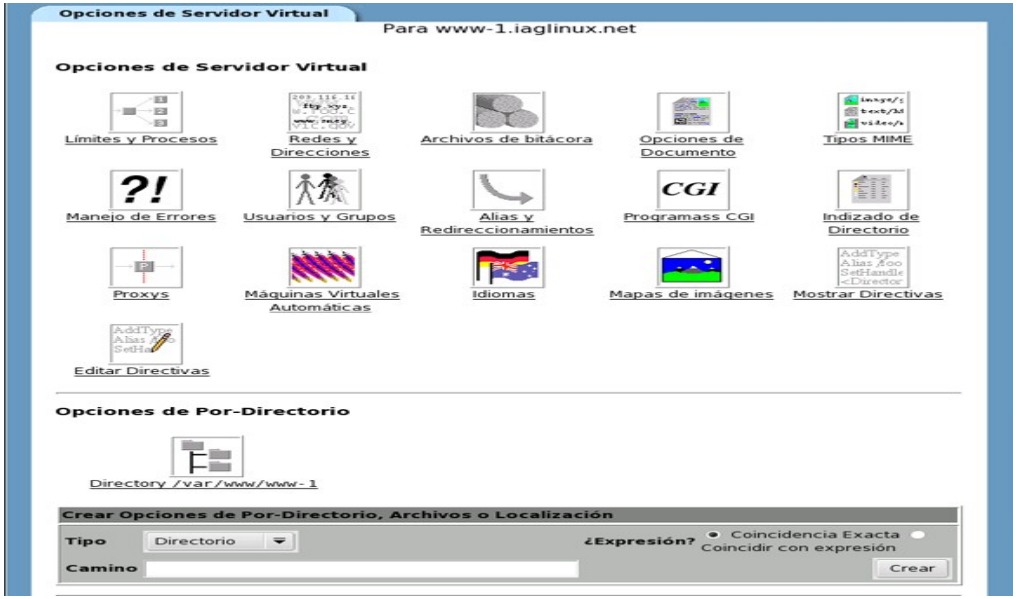
- A continuación **Apache** comprueba su archivo de nombres y contraseñas cifradas para ver si el cliente tiene derecho a acceder, en caso afirmativo envía el documento solicitado y en caso contrario vuelve a mandar un código 401 solicitando de nuevo la autenticación y la contraseña.



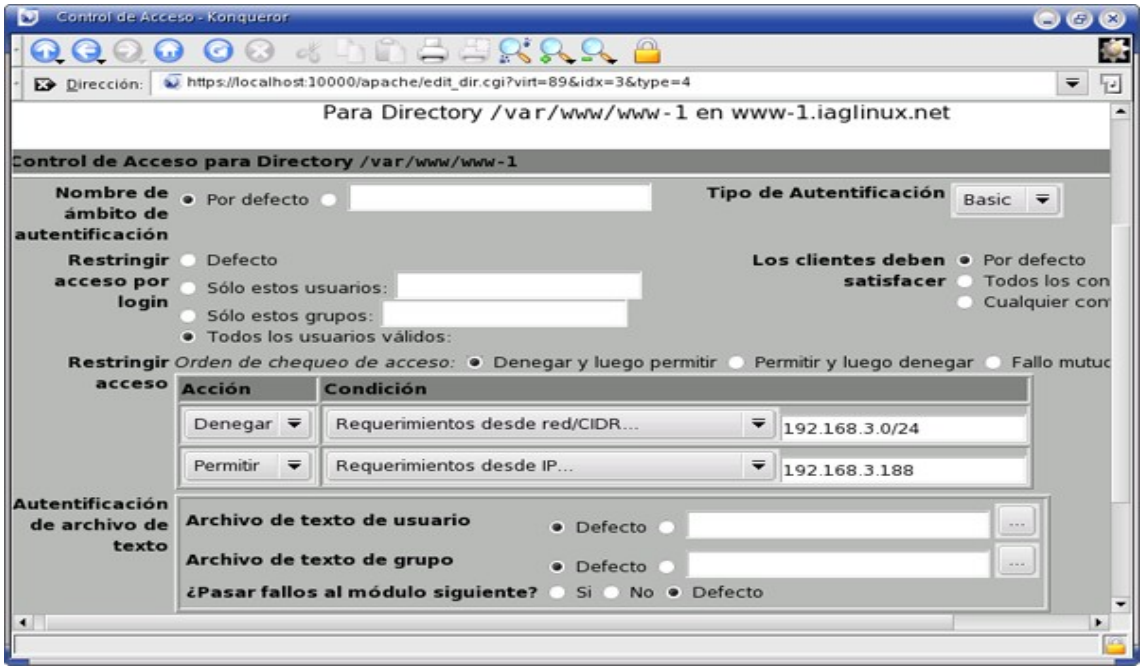
- El control de acceso y autentificación podremos aplicarlos a directorios, localizaciones y archivos, no así de forma global o a servidores. Se pueden establecer mediante host (IP) o mediante usuario y clave.

6.2.4.1. Control de Acceso por host

- Para controlar los accesos al servidor web mediante direcciones de máquinas con webmin, establecemos los parámetros en:
- Servidores virtuales->En nuestro caso servidor virtual www-1.gonzanaza.com (o el sitio a controlar)
- En Opciones por Directorio Directory /var/www/www-1



- **Orden de chequeo de acceso:** se puede establecer denegando primero y luego permitiendo, o viceversa. La evaluación de los permisos no se realizará de forma jerárquica sino de forma global.



- Las restricciones las hace mediante:
  - **Acción:** Denegar o Permitir
  - **Condición:**
    - Requerimientos desde máquina (Nombre de dominio de máquina.
    - Requerimientos desde IP (ip de una máquina).
    - Requerimientos desde IP parcial(ip incompleta. ej: 127.0.0)
    - Requerimientos desde red/máscara de red.
    - Requerimientos desde red/CIDR (ej: 192.168.242.0/24)

- El fichero de configuración quedará de la siguiente forma

/etc/httpd/conf/httpd.conf - KEdit

Archivo Editar Ir Herramientas Preferencias Ayuda

```
NameVirtualHost 192.168.3.188

<VirtualHost 192.168.3.188>
DocumentRoot "/var/www/www-1"
ServerName www-1.iaglinux.net
<Directory "/var/www/www-1">
allow from 192.168.3.188
allow from 192.168.3.193
Options +Indexes +Includes
deny from 192.168.3.0/255.255.255.0
order deny,allow
</Directory>
</VirtualHost>
```

INS Línea: 249 Col: 1

- Mediante allow y deny from se especifican los hosts para permitirles o dengarles acceso.
- Mediante order se especifica el orden en que se evalúan los permisos:
- **Order allow, deny:** procesa las reglas allow y luego las reglas deny. En caso de conflicto, las reglas deny anulan a las allow. Se deniega el acceso a los hosts que no se contemplen en ninguna regla.
- **Order deny, allow:** se procesan primero las reglas deny y luego las allow. En caso de conflicto, las allow anulan a las deny. Se permite el acceso a los hosts que no se contemplen en ninguna regla.

6.2.4.2. Control de Acceso por usuario/clave

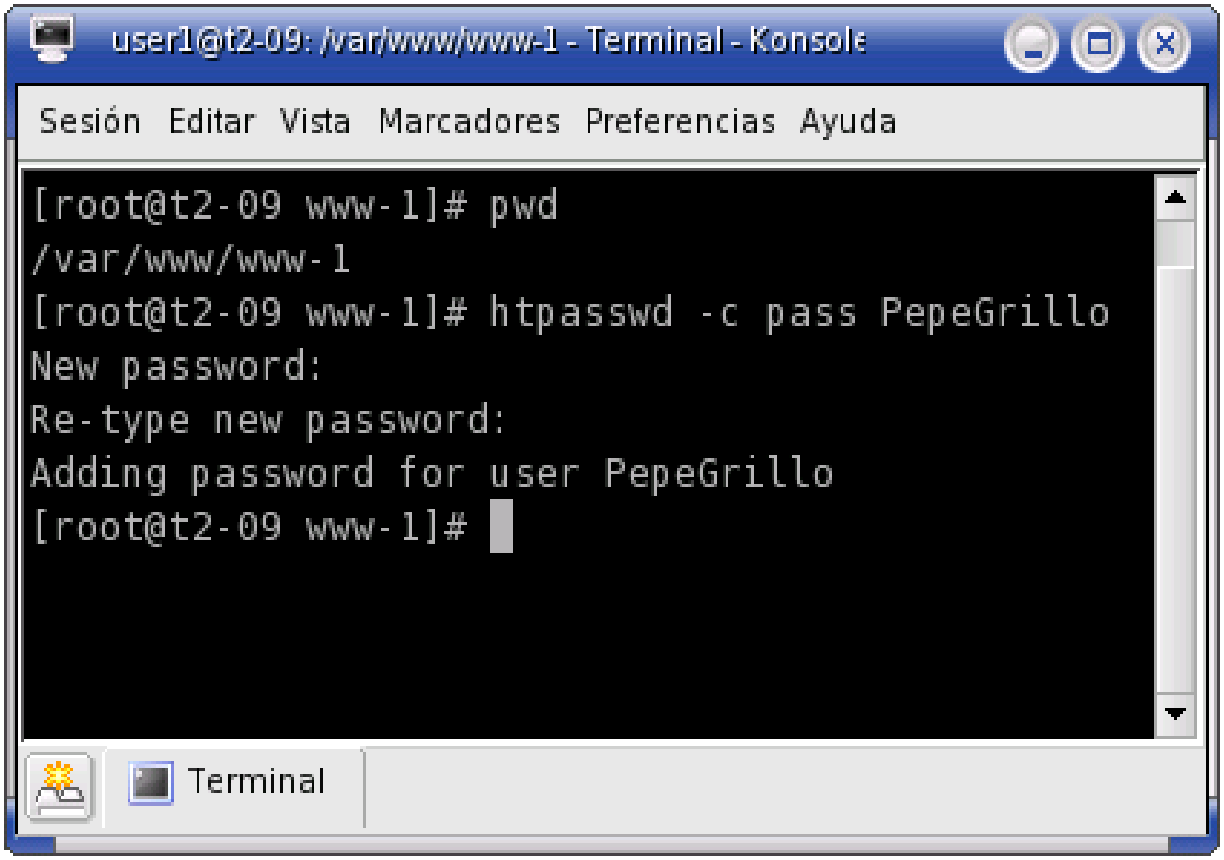
- Se pide autentificación mediante nombre de usuario y clave. Estos usuarios son propios de apache. También se pueden crear grupos. Las directivas que se utilizan son las siguientes:
- **AuthName texto:** es el texto que aparece cuando se pide la autentificación.
- **AuthType Basic/Digest:** formato de los datos que intercambian cliente y servidor durante la autorización. Se utiliza Basic porque algunos navegadores no soportan Digest.
- **require:** especifica a quién se permite el acceso (usuario, grupos, todos)
  - require user <usuario1><usuario2>
  - require group <grupo1><grupo2>

- **AuthUserFile:** especifica la dirección del archivo que guardará los usuarios.
- **AuthGroupFile:** especifica la dirección del archivo que guardará los grupos.
- El fichero de usuarios guarda el nombre del usuario y separada por dos puntos, su contraseña encriptada.
- El fichero de grupos guarda el nombre del grupo y separados por dos puntos, los nombres de los usuarios apache que pertenecen al grupo separados entre ellos por espacios.
- Ejemplo:

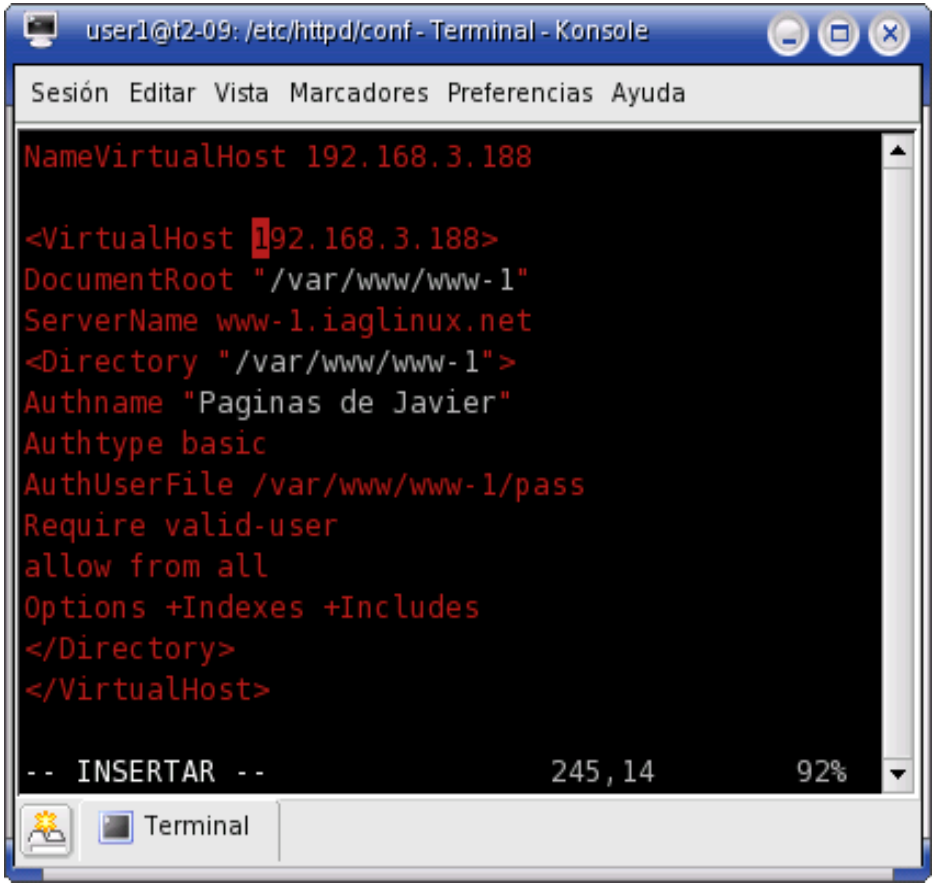
```
<Directory /var/www/html/privado>
AuthType Basic
AuthName "Pagina de Cervantes"
AuthUserFile /var/www/passwd/.htpasswd
require user cervantes
</Directory>
```

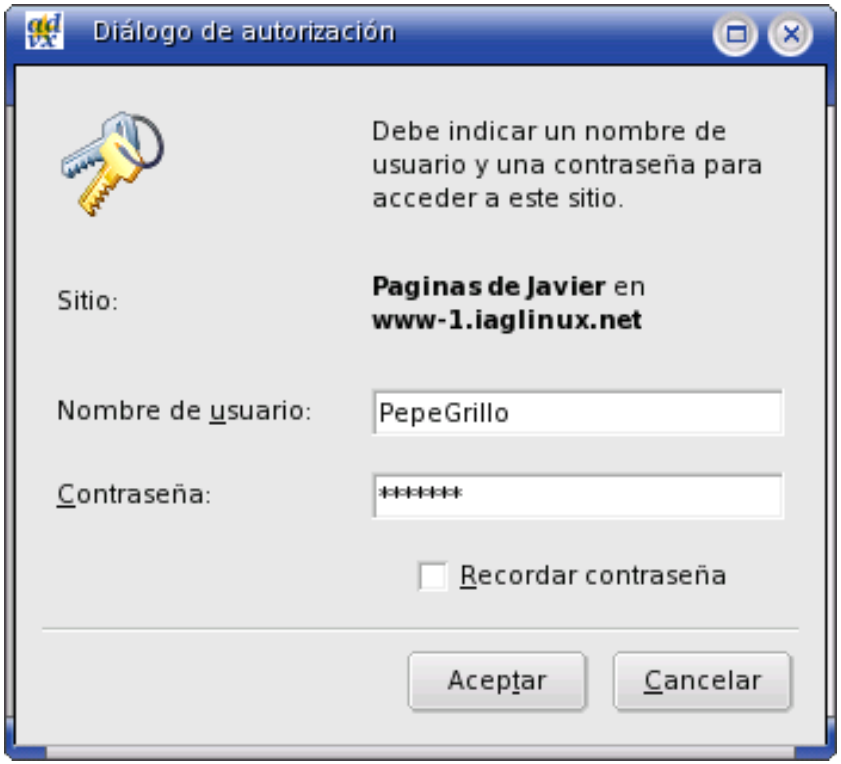
- Veamos lo anterior con un ejemplo, primero lo haremos con el fichero de configuración y después mediante webmin.
- **1. Fichero de configuración**
- Crearemos el archivo donde tienen que estar los usuarios y las contraseñas, el comando es htpasswd, así pondremos htpasswd -c (created) <ubicación archivo> <usuario a añadir>

  Pedirá inmediatamente la contraseña para ese usuario.
- Si quisiera agregar otro usuario no sería necesario poner -c.

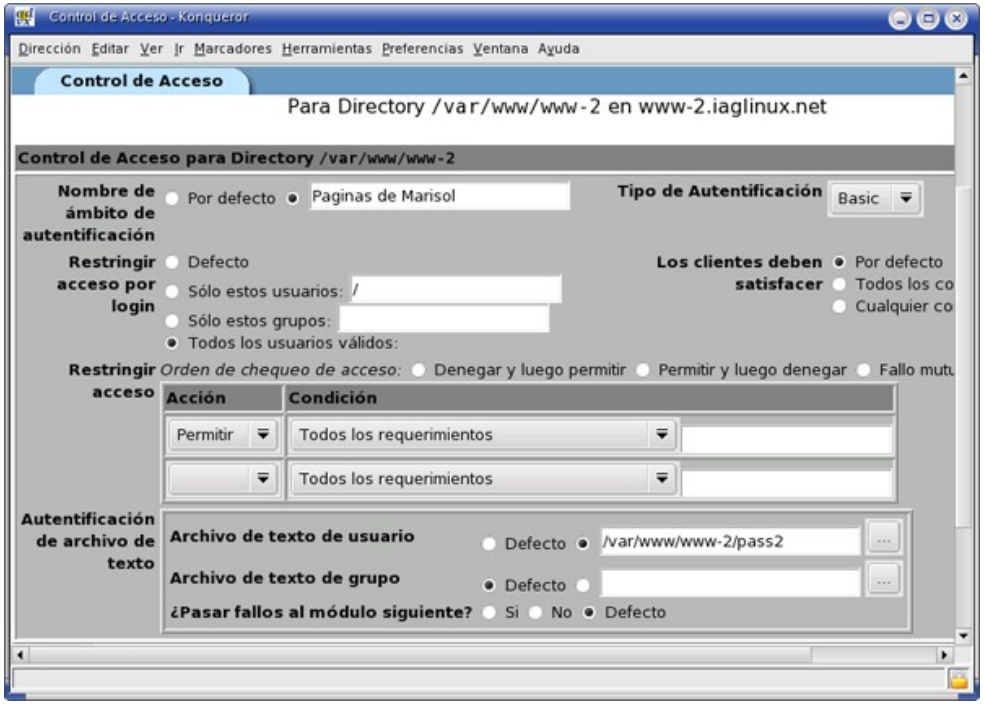

user1@t2-09: /var/www/www-1 - Terminal - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda
[root@t2-09 www-1]# pwd
/var/www/www-1
[root@t2-09 www-1]# htpasswd -c pass PepeGrillo
New password:
Re-type new password:
Adding password for user PepeGrillo
[root@t2-09 www-1]#
Terminal

- Y la siguiente imagen mostraría un ejemplo de configuración para nuestro servidor virtual www-1.



- Reiniciamos el servicio, y vamos a www-1.gonzanaza.com y vemos que nos pide la contraseña y el nombre de usuario.
- 

- **2. Webmin.**
- Haremos los mismos pasos ahora con Webmin para controlar el acceso por usuario y clave al servidor virtual www-2, para ello vamos a Directory /var/www/www-2, y escogemos Control de Acceso.



6.3. Práctica.

- Explicación y configuración de un servidor Apache a través de webmin.
- Crear una página de error 404 y 403 para el servidor Apache.
- Crear una página de inicio para el servidor global y para un usuario.
- Crear un par de dominios virtuales profesores.gonzanaza.com y alumnos.gonzanaza.com con la misma IP