HTTP Linux Apache (II)

ÍNDICE

1. Módulos

Como ya hemos visto, los módulos son una serie de programas u opciones complementarias que el servidor web Apache dispone para añadir funcionalidades adicionales mediante su utilización integrada con Apache.

Se organizan en dos directorios:

- `/etc/apache2/mods-available`. Contiene todos los módulos disponibles para ser utilizados, aunque previamente deben ser activados.
- `/etc/apache2/mods-enabled`. Contiene los módulos que han sido activados para ser utilizados por Apache. En realidad no son más que enlaces simbólicos a los ficheros de `mods-available`.

Los módulos se dividen a su vez en un fichero con la instrucción de carga (`*.load`) y otro con los comandos de configuración (`*.conf`).

1.1. Habilitación

Algunos módulos se activan en la instalación de Apache y otros se pueden activar más tarde según necesidad. Para activar un módulo utilizaremos el comando:

```
a2enmod nombre_de_modulo_a_activar
```

Nota: **a**ctivate **2** (to) **en**able **mod**ule, que podemos traducirlo como *activar para permitir módulo*.

El módulo `userdir` permite que los usuarios normales del sistema operativo puedan gestionar su propio sitio web en el servidor Apache, sin necesidad de tener los permisos especiales de `root`.

El módulo `userdir` permite que cada usuario cree una carpeta en su `home` y pueda dejar allí las página web que desea mostrar. El nombre por defecto de la carpeta en la que debe poner las páginas a mostrar por Apache es `public_html`.

Procederemos a activar el módulo `userdir`, por la que previamente comprobaremos que no está en activo, es decir, no existe su enlace simbólico en `/etc/apache2/mods-enabled`:

```
root@debian:/home/servicios# ls -l /etc/apache2/mods-enabled
total 0
lrwxrwxrwx 1 root root 28 dic 15 17:46 alias.conf -> ../mods-available/alias.conf
lrwxrwxrwx 1 root root 28 dic 15 17:46 alias.load -> ../mods-available/alias.load
lrwxrwxrwx 1 root root 33 dic 15 17:46 auth_basic.load -> ../mods-available/auth_basic.load
lrwxrwxrwx 1 root root 33 dic 15 17:46 authn_file.load -> ../mods-available/authn_file.load
lrwxrwxrwx 1 root root 36 dic 15 17:46 authz_default.load -> ../mods-available/authz_default.load
lrwxrwxrwx 1 root root 38 dic 15 17:46 authz_groupfile.load -> ../mods-available/authz_groupfile.load
lrwxrwxrwx 1 root root 33 dic 15 17:46 authz_host.load -> ../mods-available/authz_host.load
lrwxrwxrwx 1 root root 33 dic 15 17:46 authz_user.load -> ../mods-available/authz_user.load
lrwxrwxrwx 1 root root 32 dic 15 17:46 autoindex.conf -> ../mods-available/autoindex.conf
lrwxrwxrwx 1 root root 32 dic 15 17:46 autoindex.load -> ../mods-available/autoindex.load
lrwxrwxrwx 1 root root 27 dic 15 17:46 cgid.conf -> ../mods-available/cgid.conf
lrwxrwxrwx 1 root root 27 dic 15 17:46 cgid.load -> ../mods-available/cgid.load
lrwxrwxrwx 1 root root 30 dic 15 17:46 deflate.conf -> ../mods-available/deflate.conf
lrwxrwxrwx 1 root root 30 dic 15 17:46 deflate.load -> ../mods-available/deflate.load
lrwxrwxrwx 1 root root 26 dic 15 17:46 dir.conf -> ../mods-available/dir.conf
lrwxrwxrwx 1 root root 26 dic 15 17:46 dir.load -> ../mods-available/dir.load
lrwxrwxrwx 1 root root 26 dic 15 17:46 env.load -> ../mods-available/env.load
lrwxrwxrwx 1 root root 27 dic 15 17:46 mime.conf -> ../mods-available/mime.conf
lrwxrwxrwx 1 root root 27 dic 15 17:46 mime.load -> ../mods-available/mime.load
lrwxrwxrwx 1 root root 34 dic 15 17:46 negotiation.conf -> ../mods-available/negotiation.conf
lrwxrwxrwx 1 root root 34 dic 15 17:46 negotiation.load -> ../mods-available/negotiation.load
lrwxrwxrwx 1 root root 33 dic 15 17:46 reqtimeout.conf -> ../mods-available/reqtimeout.conf
lrwxrwxrwx 1 root root 33 dic 15 17:46 reqtimeout.load -> ../mods-available/reqtimeout.load
lrwxrwxrwx 1 root root 31 dic 15 17:46 setenvif.conf -> ../mods-available/setenvif.conf
lrwxrwxrwx 1 root root 31 dic 15 17:46 setenvif.load -> ../mods-available/setenvif.load
lrwxrwxrwx 1 root root 29 dic 15 17:46 status.conf -> ../mods-available/status.conf
lrwxrwxrwx 1 root root 29 dic 15 17:46 status.load -> ../mods-available/status.load
root@debian:/home/servicios#
```

Cuando activamos el módulo, se nos indica que debemos reiniciar el servicio para que se tome la nueva configuración, aunque ya aparecerá su enlace simbólico:

```
root@debian:/home/servicios# a2enmod userdir
Enabling module userdir.
Run '/etc/init.d/apache2 restart' to activate new configuration!
root@debian:/home/servicios# ls -l /etc/apache2/mods-enabled
total 0
lrwxrwxrwx 1 root root 28 dic 15 17:46 alias.conf -> ../mods-available/alias.conf
lrwxrwxrwx 1 root root 28 dic 15 17:46 alias.load -> ../mods-available/alias.load
lrwxrwxrwx 1 root root 33 dic 15 17:46 auth_basic.load -> ../mods-available/auth_basic.load
lrwxrwxrwx 1 root root 33 dic 15 17:46 authn_file.load -> ../mods-available/authn_file.load
lrwxrwxrwx 1 root root 36 dic 15 17:46 authz_default.load -> ../mods-available/authz_default.load
lrwxrwxrwx 1 root root 38 dic 15 17:46 authz_groupfile.load -> ../mods-available/authz_groupfile.load
lrwxrwxrwx 1 root root 33 dic 15 17:46 authz_host.load -> ../mods-available/authz_host.load
lrwxrwxrwx 1 root root 33 dic 15 17:46 authz_user.load -> ../mods-available/authz_user.load
lrwxrwxrwx 1 root root 32 dic 15 17:46 autoindex.conf -> ../mods-available/autoindex.conf
lrwxrwxrwx 1 root root 32 dic 15 17:46 autoindex.load -> ../mods-available/autoindex.load
lrwxrwxrwx 1 root root 27 dic 15 17:46 cgid.conf -> ../mods-available/cgid.conf
lrwxrwxrwx 1 root root 27 dic 15 17:46 cgid.load -> ../mods-available/cgid.load
lrwxrwxrwx 1 root root 30 dic 15 17:46 deflate.conf -> ../mods-available/deflate.conf
lrwxrwxrwx 1 root root 30 dic 15 17:46 deflate.load -> ../mods-available/deflate.load
lrwxrwxrwx 1 root root 26 dic 15 17:46 dir.conf -> ../mods-available/dir.conf
lrwxrwxrwx 1 root root 26 dic 15 17:46 dir.load -> ../mods-available/dir.load
lrwxrwxrwx 1 root root 26 dic 15 17:46 env.load -> ../mods-available/env.load
lrwxrwxrwx 1 root root 27 dic 15 17:46 mime.conf -> ../mods-available/mime.conf
lrwxrwxrwx 1 root root 27 dic 15 17:46 mime.load -> ../mods-available/mime.load
lrwxrwxrwx 1 root root 34 dic 15 17:46 negotiation.conf -> ../mods-available/negotiation.conf
lrwxrwxrwx 1 root root 34 dic 15 17:46 negotiation.load -> ../mods-available/negotiation.load
lrwxrwxrwx 1 root root 33 dic 15 17:46 reqtimeout.conf -> ../mods-available/reqtimeout.conf
lrwxrwxrwx 1 root root 33 dic 15 17:46 reqtimeout.load -> ../mods-available/reqtimeout.load
lrwxrwxrwx 1 root root 31 dic 15 17:46 setenvif.conf -> ../mods-available/setenvif.conf
lrwxrwxrwx 1 root root 31 dic 15 17:46 setenvif.load -> ../mods-available/setenvif.load
lrwxrwxrwx 1 root root 29 dic 15 17:46 status.conf -> ../mods-available/status.conf
lrwxrwxrwx 1 root root 29 dic 15 17:46 status.load -> ../mods-available/status.load
lrwxrwxrwx 1 root root 30 ene 12 17:42 userdir.conf -> ../mods-available/userdir.conf
lrwxrwxrwx 1 root root 30 ene 12 17:42 userdir.load -> ../mods-available/userdir.load
root@debian:/home/servicios#
```

Para afianzar los conocimientos, vamos a ver un caso práctico. Tras reiniciar el servicio crearemos un nuevo usuario en el sistema para probar la funcionalidad del módulo `userdir`.

```
useradd -m usu1
passwd usu1 → Clave_00
```

Una vez creado el usuario accedemos con él al servidor para crear en su /`home` la carpeta `public_html` y poner allí las páginas web deseadas.

```
$ pwd
/home/usu1
$ ls -l
total 0
$ mkdir public_html
$ cd public_html
$ nano index.html
$ ls -l
total 4
-rw-r--r-- 1 usu1 usu1 68 ene 12 18:21 index.html
$ cat index.html
<H1>PAGINA INICIO USU1</H1>
<p>usu1
<p>usu1
<p>usu1
<p>usu1
<p>usu1
$
```

Desde el cliente web accedemos al servidor Apache utilizando como dirección el valor: `FQDN/~nombre_usuario`

Nota. Para obtener el carácter ~ debemos pulsar conjuntamente [*Alt Gr* + *4*].

1.2. Deshabilitación

En caso de no querer seguir utilizando un módulo deberemos deshabilitarlo como medida de seguridad y reiniciar el servicio para tomar los nuevos cambios. Para ello ejecutaremos el comando:

```
a2dismod nombre_de_modulo_a_desactivar
```

Nota: **a**ctivate **2** (to) **dis**able **mod**ule, que podemos traducirlo como *activar para deshabilitar módulo*.

```
root@debian:/home/servicios# a2dismod userdir
Module userdir disabled.
Run '/etc/init.d/apache2 restart' to activate new configuration!
root@debian:/home/servicios# service apache2 restart
Restarting web server: apache2apache2: Could not reliably determine the server's fully qualified domain nam
e, using 127.0.1.1 for ServerName
 ... waiting apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.
1 for ServerName
.
root@debian:/home/servicios#
```

Al volver a intentar conectar con el cliente se producirá el error 404, es decir, página solicitada por el cliente no encontrada por el servidor.

1.3. Configuración

Ya hemos visto que mediante el fichero `*.conf` del módulo que sea, podemos proceder a la configuración del mismo. De esta manera, siguiendo con el ejemplo de `userdir`, las opciones por defecto habilitadas, para este módulo, se pueden configurar mediante su fichero de configuración `/etc/apache2/mods-enabled/userdir.conf` (que es un *enlace simbólico* a `/etc/apache2/mods-available/userdir.conf`). Como ejemplo podemos acceder a él y observar la definición del nombre de la carpeta a utilizar y que el usuario `root` está deshabilitado para dicho módulo.

```
 GNU nano 2.2.4   Fichero: /etc/apache2/mods-enabled/userdir.conf

<IfModule mod_userdir.c>
        UserDir public_html
        UserDir disabled root
```

El fichero de código binario utilizado por `userdir` es `/usr/lib/apache2/modules/mod_userdir.so` y lo podemos comprobar en su fichero de carga `/etc/apache2/mods-enabled/userdir.load` (enlace simbólico a `/etc/apache2/mods-available/userdir.load`).

```
 GNU nano 2.2.4   Fichero: /etc/apache2/mods-enabled/userdir.load

LoadModule userdir_module /usr/lib/apache2/modules/mod_userdir.so
```

2. Autenticación

Ya sabemos que la **autenticación** es el proceso por el que se verifica que un usuario es quien dice ser. El servidor web Apache presenta dos formas básicas de autenticación:

a. **`Basic`**. Es un método de autenticación para usuarios específicos de Apache, no es necesario que existan en el sistema operativo, siendo el primer método de autenticación implementado por Apache. La autenticación `basic` viene por defecto activa mediante el módulo `auth_basic`, que si observamos existe su carga, aunque no su configuración. Se debe tener en cuenta que la clave de autenticación enviada por el cliente desde su navegador, por defecto será en texto plano, aunque se puede configurar su acceso por HTTPS para hacerlo totalmente seguro.

```
root@debian:/home/servicios# ls /etc/apache2/mods-enabled
alias.conf            autoindex.conf  env.load           setenvif.load
alias.load            autoindex.load  mime.conf          status.conf
auth_basic.load       cgid.conf       mime.load          status.load
authn_file.load       cgid.load       negotiation.conf   userdir.conf
authz_default.load    deflate.conf    negotiation.load   userdir.load
authz_groupfile.load  deflate.load    reqtimeout.conf
authz_host.load       dir.conf        reqtimeout.load
authz_user.load       dir.load        setenvif.conf
root@debian:/home/servicios#
```

b. **`Digest`**. Igual que el caso anterior, permite autenticación para usuarios específicos de Apache sin necesidad que existan en el sistema operativo. Es un método posterior a `basic` e intentó solucionar los problemas de inseguridad de este, ya que la contraseña se envía encriptada por el cliente en el momento de la conexión, pero el cifrado es débil e inseguro. Para utilizar la autenticación `digest` debe ser activado el módulo que la soporta, es decir, `auth_digest`, que, como en el caso anterior, existe el módulo para su carga, pero no el de su configuración. También, como en el caso de `basic`, se puede configurar su acceso por HTTPS para que sea totalmente seguro.

```
root@debian:/home/servicios# ls -l /etc/apache2/mods-available/auth_*
-rw-r--r-- 1 root root 72 mar 22  2011 /etc/apache2/mods-available/auth_basic.load
-rw-r--r-- 1 root root 74 mar 22  2011 /etc/apache2/mods-available/auth_digest.load
root@debian:/home/servicios#
```

2.1. Autenticación básica. Ejemplo

En el servidor web Apache sobre Linux Debian deberás crear la carpeta `/var/www/basic` para que únicamente pueda acceder a ella los usuarios *basic1* y *basic2* utilizando autenticación `basic`.

Creamos la nueva carpeta y añadimos en ella páginas web.

```
mkdir /var/www/basic
nano /var/www/basic/index.html
```

Para usar la autenticación `basic` hay que crear un fichero accesible por Apache (dentro de la carpeta `/etc/apache2/`), pero fuera del acceso de los clientes web (no crearlo dentro de `/var/www/`). En dicho fichero se guardarán los usuarios y sus contraseñas de acceso.

Para crear este fichero y almacenar en él los usuarios con sus claves, se utilizará el comando **`htpasswd`**. Para crear el fichero, añadir el usuario *basic1* y su contraseña ejecutaríamos el comando:

```
htpasswd -c /etc/apache2/usuarios_basic basic1
```

La opción **-c** sólo se utiliza para la creación del fichero, es decir, para añadir posteriormente usuarios con el fichero ya creado no sería necesario. Si por error incluimos nuevamente la opción -c el fichero existente se borrará y se creará uno nuevo. Llegados a este punto, debemos añadir el segundo usuario:

```
htpasswd /etc/apache2/usuarios_basic basic2
```

Si accedemos al contenido del fichero podemos apreciar los usuarios allí almacenados y sus claves encriptadas.

Nota. Se debe tener en cuenta que cuando el cliente envía su clave desde el navegador, por defecto se enviará en texto plano.

```
root@debian:/home/servicios# mkdir /var/www/basic
root@debian:/home/servicios# nano /var/www/basic/index.html
root@debian:/home/servicios# htpasswd -c /etc/apache2/usuarios_basic usu1
New password:
Re-type new password:
Adding password for user usu1
root@debian:/home/servicios# htpasswd -c /etc/apache2/usuarios_basic basic1
New password:
Re-type new password:
Adding password for user basic1
root@debian:/home/servicios# htpasswd /etc/apache2/usuarios_basic basic2
New password:
Re-type new password:
Adding password for user basic2
root@debian:/home/servicios# cat /etc/apache2/usuarios_basic
basic1:oNT5GOmXDwqPk
basic2:86S7ZOVNq3wiI
```

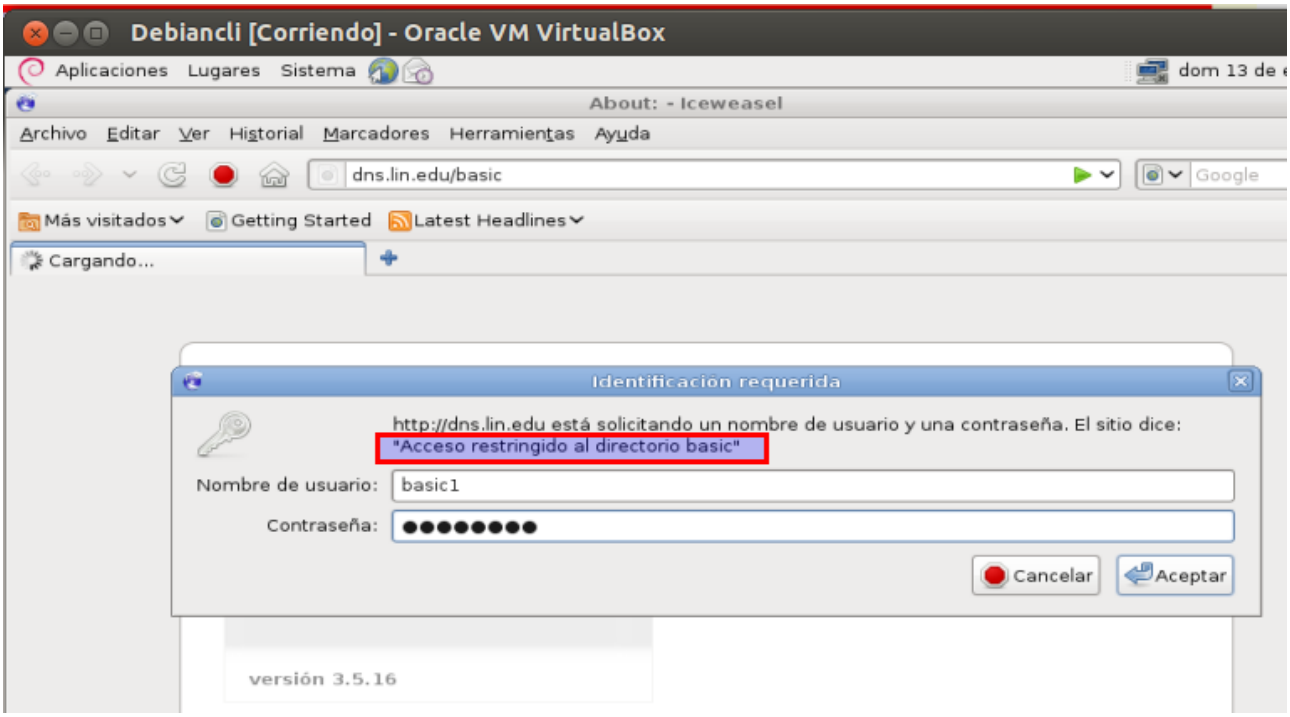
Ahora accederemos al fichero de configuración global de Apache (`/etc/apache2/sites-enabled/000-default`) para añadir los comandos o directivas relativas a la autenticación `basic` en el directorio `/var/www/basic`:

```
 GNU nano 2.2.4   Fichero: /etc/apache2/sites-enabled/000-default

                DirectoryIndex index.html
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>

        <Directory /var/www/basic/>
                AuthType Basic
                AuthName "Acceso restringido al directorio basic"
                AuthUserFile /etc/apache2/usuarios_basic
                Require user basic1 basic2
        </Directory>
```

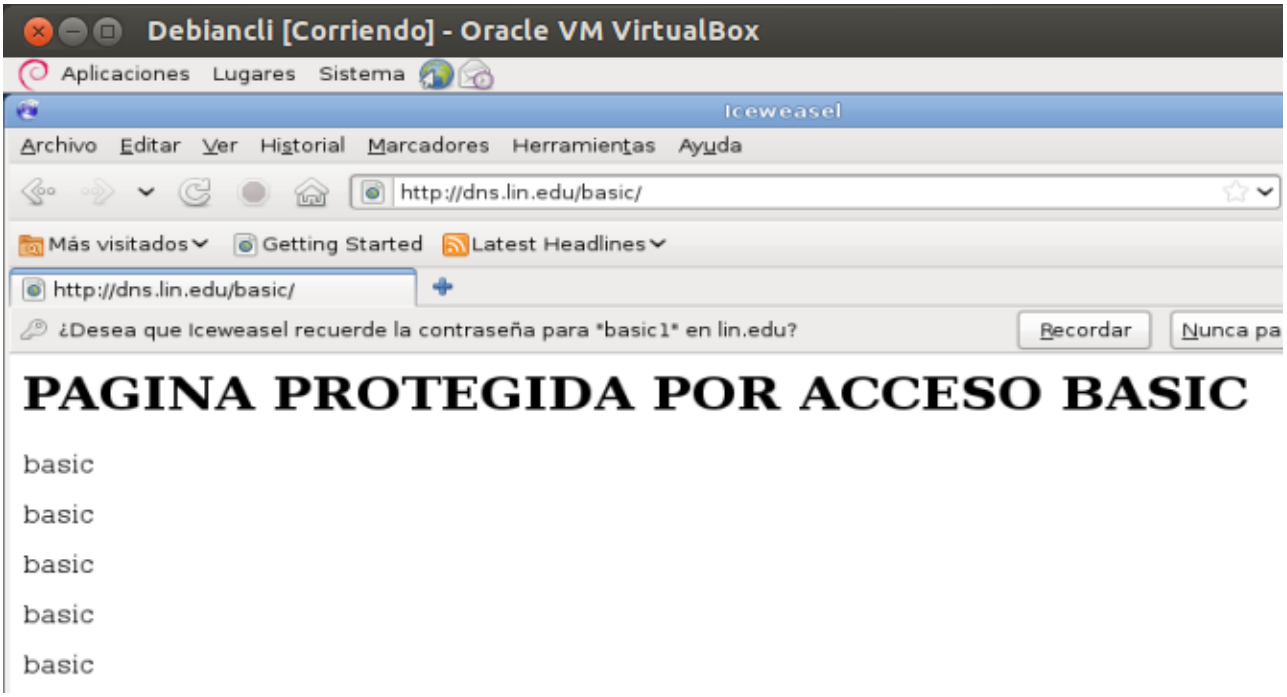
donde:

- `AuthType`. Identifica el tipo de autenticación.
- `AuthName`. Mensaje a mostrar en la ventana emergente para entrada de usuario y clave.
- `AuthUserFile`. Fichero contenedor de los usuarios permitidos y sus claves encriptadas.
- `Require user`. Usuarios a los que se les permite el acceso.

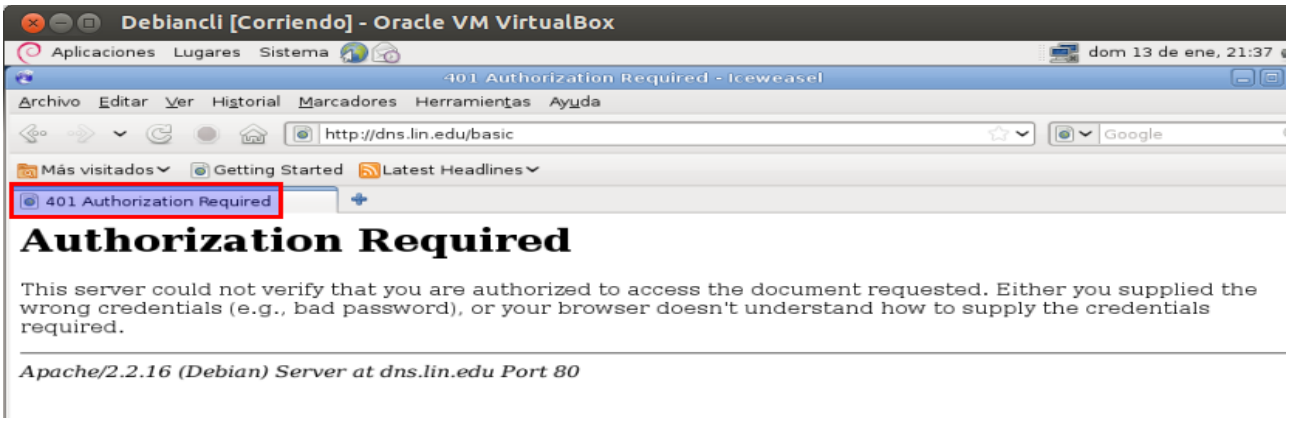
Tras reiniciar el servicio web para tomar los últimos cambios, al acceder a la carpeta desde el cliente se nos solicitará la autenticación:



Tras introducir usuario y clave adecuados pulsaremos sobre *Aceptar* para acceder al contenido.



En caso de pulsar sobre *Cancelar* nos retornará el error 401 indicando que se requiere autenticación:



2.2. Autenticación digest. Ejemplo

En el servidor web Apache sobre Linux Debian deberás crear la carpeta `/var/www/digest` para que solamente pueda acceder a ella los usuarios *digest1* y *digest2* utilizando autenticación `digest`. Lo primero que debemos hacer es activar el módulo `auth_digest` y reiniciar el servicio:

```
root@debian:/etc/apache2/mods-available# a2enmod auth_digest
Enabling module auth_digest.
Run '/etc/init.d/apache2 restart' to activate new configuration!
root@debian:/etc/apache2/mods-available# service apache2 restart
Restarting web server: apache2apache2: Could not reliably determine the server's
 fully qualified domain name, using 127.0.1.1 for ServerName
 ... waiting apache2: Could not reliably determine the server's fully qualified
domain name, using 127.0.1.1 for ServerName
.
root@debian:/etc/apache2/mods-available#
```

Creamos la nueva carpeta y añadimos en ella páginas web.

```
root@debian:/var/www/digest# ls -l
total 4
-rw-r--r-- 1 root root 88 ene 16 15:46 index.html
root@debian:/var/www/digest# nano index.html
root@debian:/var/www/digest# cat index.html
<H1>PAGINA PROTEGIDA POR ACCESO DIGEST</H1>
<p>digest
<p>digest
<p>digest
<p>digest
<p>digest
root@debian:/var/www/digest#
```

Para usar la autenticación `digest` hay que crear un fichero accesible por Apache (dentro de la carpeta `/etc/apache2/`), pero fuera del acceso de los clientes web (no crearlo dentro de `/var/www/`). En dicho fichero se guardarán los usuarios y sus contraseñas de acceso.

Para crear este fichero y almacenar en él los usuarios con sus claves, se utilizará el comando **`htdigest`**, parecido al caso anterior, pero en el que se deberá introducir información sobre el dominio donde crearemos el control de autenticación `digest`:

```
root@debian:/var/www# htdigest -c /etc/apache2/usuarios_digest lin.edu digest1
Adding password for digest1 in realm lin.edu.
New password:
Re-type new password:
root@debian:/var/www# htdigest /etc/apache2/usuarios_digest lin.edu digest2
Adding user digest2 in realm lin.edu
New password:
Re-type new password:
root@debian:/var/www#
```

Si accedemos al contenido del fichero podemos apreciar los usuarios allí almacenados, el dominio y sus claves encriptadas.

```
root@debian:/etc/apache2# cat usuarios_digest
digest1:lin.edu:593cd875ae7de5a0fa2f09449a6936f6
digest2:lin.edu:63f50e960fd49ef5cd91a90dec8d5915
root@debian:/etc/apache2#
```

Ahora accederemos al fichero de configuración global de Apache (`/etc/apache2/sites-enabled/000-default`) para añadir los comandos o directivas relativas a la autenticación `digest` en el directorio `/var/www/digest`:

```
 GNU nano 2.2.4      Fichero: /etc/apache2/sites-enabled/000-default

                AuthUserFile /etc/apache2/usuarios_basic
                Require user basic1 basic2
        </Directory>

        <Directory /var/www/digest/>
                AuthType Digest
                AuthName "lin.edu"
                AuthUserFile /etc/apache2/usuarios_digest
                Require user digest1 digest2
        </Directory>

        <Directory /var/www/otrodir/>
                Options FollowSymLinks MultiViews
        </Directory>
```

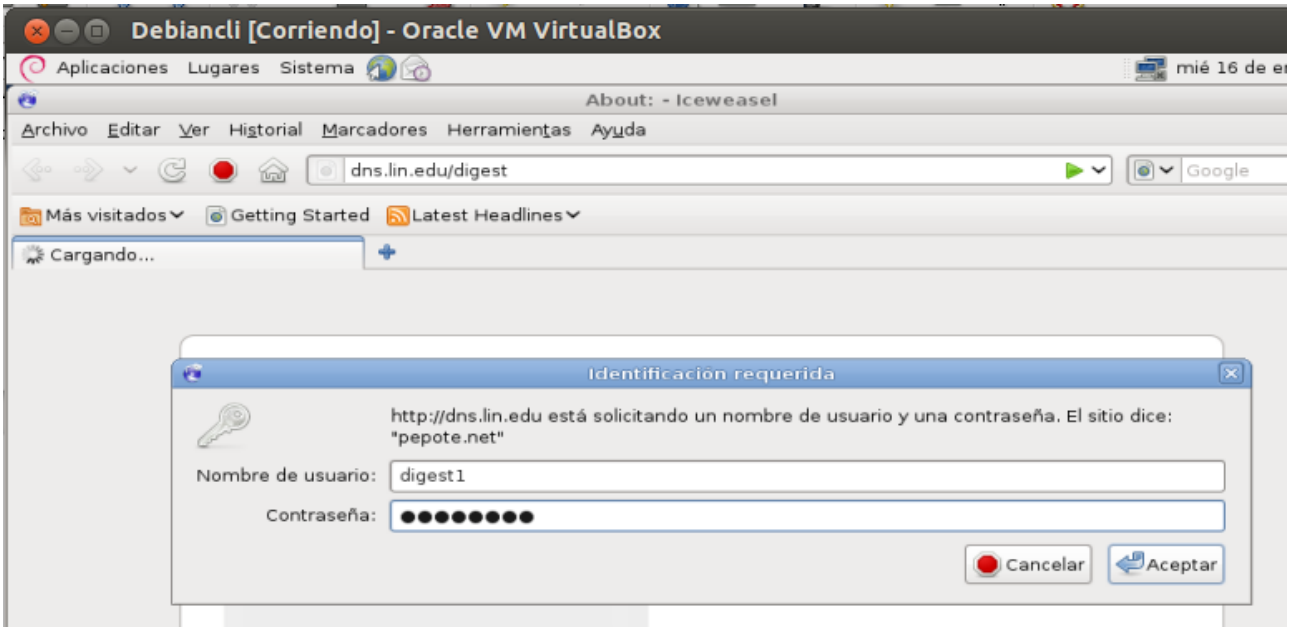
donde:

- `AuthType`. Identifica el tipo de autenticación.
- `AuthName`. Nombre de dominio, que se mostrará en la ventana emergente.
- `AuthUserFile`. Fichero contenedor de los usuarios permitidos, dominio y sus claves encriptadas.
- `Require user`. Usuarios a los que se les permite el acceso.

Reiniciamos el servicio web Apache para que se tomen los últimos cambios:

```
root@debian:/var/www/digest# service apache2 restart
Restarting web server: apache2apache2: Could not reliably determine the server's
 fully qualified domain name, using 127.0.1.1 for ServerName
 ... waiting apache2: Could not reliably determine the server's fully qualified
domain name, using 127.0.1.1 for ServerName
.
root@debian:/var/www/digest#
```

Al acceder a la carpeta desde el cliente se nos solicitará la autenticación:



Al introducir usuario y clave correcta nos permitirá el acceso a la página solicitada:

3. Sitios web virtuales (host virtuales) por nombre de dominio

Cuando queremos tener varios sitios web en un único servidor web, donde cada uno de ellos responda a un FQDN distinto, no tenemos más remedio que configurar sitios virtuales. Lo primero que debemos tener claro es que necesitamos tener configurado un servicio DNS que resuelva de forma correcta los nombres de dominio.

Utilizaremos un ejemplo para ir contando y explicando la configuración de distintos sitios virtuales.

Crearemos un nuevo sitio virtual que se corresponderá con un subdominio del utilizado hasta ahora, como por ejemplo: *www.sub.lin.edu*, almacenando sus páginas a mostrar en `/var/www/subdominio`, utilizando como página por defecto `inicio.html` y ficheros de error y accesos personalizados.

Crearemos otro nuevo sitio virtual que se corresponderá con otro dominio distinto al utilizado hasta ahora, como por ejemplo: *www.pepote.net*, almacenando sus páginas a mostrar en `/var/www/pepote`, utilizando como página de defecto `primera.html` y ficheros de error y accesos personalizado.

1) Configuración DNS:

```
  GNU nano 2.2.4       Fichero: /etc/bind/named.conf.local          Modif

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "lin.edu"{
   type master;
   file "/var/lib/bind/lin.edu.maestro";
};

zone "sub.lin.edu"{
   type master;
   file "/var/lib/bind/sub.lin.edu.maestro";
};

zone "pepote.net"{
   type master;
   file "/var/lib/bind/pepote.net.maestro";
};
```

```
  GNU nano 2.2.4          Fichero: sub.lin.edu.maestro

@ IN SOA www correo (1 2 3 4 5)
  IN NS www
www IN A 172.26.25.100
```

```
  GNU nano 2.2.4          Fichero: pepote.net.maestro

@ IN SOA www correo (1 2 3 4 5)
  IN NS www
www IN A 172.26.25.100
```

2) Reiniciamos el servicio DNS y comprobamos su funcionamiento.

```
root@debian:/var/lib/bind# service bind9 restart
Stopping domain name service...: bind9 waiting for pid 821 to die.
Starting domain name service...: bind9.
root@debian:/var/lib/bind# ping www.sub.lin.edu
PING www.sub.lin.edu (172.26.25.100) 56(84) bytes of data.
^C64 bytes from 172.26.25.100: icmp_req=1 ttl=64 time=0.201 ms

--- www.sub.lin.edu ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.201/0.201/0.201/0.000 ms
root@debian:/var/lib/bind# ping www.pepote.net
PING www.pepote.net (172.26.25.100) 56(84) bytes of data.
^C64 bytes from 172.26.25.100: icmp_req=1 ttl=64 time=0.055 ms

--- www.pepote.net ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.055/0.055/0.055/0.000 ms
root@debian:/var/lib/bind#
```

3) Creamos las carpetas contenedoras de los nuevos sitios virtuales y les añadimos las páginas a mostrar:

```
su
cd /var/www
mkdir subdominio
cd subdominio
nano inicio.html
cd ..
mkdir pepote
cd pepote
nano primera.html
```

4) Creamos los dos ficheros de configuración de los nuevos sitios en `/etc/apache2/sites-available`:

```
 GNU nano 2.2.4                Fichero: hv_sub_lin_edu

<VirtualHost *:80>
        ServerName www.sub.lin.edu
        DocumentRoot /var/www/subdominio

        <Directory /var/www/subdominio>
                DirectoryIndex inicio.html
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/sub_lin_edu_error.log

        LogLevel warn

        CustomLog ${APACHE_LOG_DIR}/sub_lin_edu_access.log combined

</VirtualHost>
```

```
 GNU nano 2.2.4     Fichero: /etc/apache2/sites-available/hv_pepote_net

<VirtualHost *:80>
        ServerName www.pepote.net
        DocumentRoot /var/www/pepote

        <Directory /var/www/pepote>
                DirectoryIndex primera.html
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/pepote_net_error.log

        LogLevel warn

        CustomLog ${APACHE_LOG_DIR}/pepote_net_access.log combined

</VirtualHost>
```

5) De forma parecida a como trabajamos con los módulos, deberemos activar cada uno de los dos nuevos sitios, para ello utilizamos el comando `a2ensite` seguido del nombre del sitio a activar:

```
root@debian:/etc/apache2/sites-available# a2ensite hv_sub_lin_edu
Enabling site hv_sub_lin_edu.
Run '/etc/init.d/apache2 reload' to activate new configuration!
root@debian:/etc/apache2/sites-available# a2ensite hv_pepote_net
Enabling site hv_pepote_net.
Run '/etc/init.d/apache2 reload' to activate new configuration!
root@debian:/etc/apache2/sites-available#
```

Nota. Recordemos que, si quisiéramos desactivar un sitio ejecutaríamos: `a2dissite fichero_sitio`

6) Comprobamos la creación del enlace simbólico en `/etc/apache2/sites-enabled` y reiniciamos el servicio web para que se apliquen los cambios:

```
root@debian:/etc/apache2/sites-enabled# ls -l
total 0
lrwxrwxrwx 1 root root 26 dic 15 17:46 000-default -> ../sites-available/default
lrwxrwxrwx 1 root root 32 ene 15 10:17 hv_pepote_net -> ../sites-available/hv_pepote_net
lrwxrwxrwx 1 root root 33 ene 15 10:17 hv_sub_lin_edu -> ../sites-available/hv_sub_lin_edu
root@debian:/etc/apache2/sites-enabled# service apache2 restart
Restarting web server: apache2apache2: Could not reliably determine the server's fully quali
fied domain name, using 127.0.1.1 for ServerName
 ... waiting apache2: Could not reliably determine the server's fully qualified domain name,
 using 127.0.1.1 for ServerName
.
root@debian:/etc/apache2/sites-enabled#
```

7) Probamos el funcionamiento de los sitios desde un cliente:

8) Para probar el funcionamiento de los ficheros de accesos y errores mostraremos el contenido de uno ellos:

```
root@debian:/var/log/apache2# cat sub_lin_edu_access.log
172.26.25.101 - - [15/Jan/2013:10:20:41 +0100] "GET / HTTP/1.1" 200 391 "-" "Mozilla/5.0 (X1
1; U; Linux i686; es-ES; rv:1.9.1.16) Gecko/20110323 Iceweasel/3.5.16 (like Firefox/3.5.16)"
172.26.25.101 - - [15/Jan/2013:10:20:44 +0100] "GET /favicon.ico HTTP/1.1" 404 505 "-" "Mozi
lla/5.0 (X11; U; Linux i686; es-ES; rv:1.9.1.16) Gecko/20110323 Iceweasel/3.5.16 (like Firef
ox/3.5.16)"
root@debian:/var/log/apache2#
```

9) Podemos comprobar que sigue funcionando el sitio web por defecto: