

# Módulos

## Introducción

- ▶ Ampliación del núcleo de Apache con módulos.
- ▶ Webs
  - <https://httpd.apache.org/docs/2.4/es/mod/>

### Funcionalidad Básica y Módulos de MultiProcesamiento

[core](#) Core Apache HTTP Server features that are always available

[mpm\\_common](#) A collection of directives that are implemented by more than one multi-processing module (MPM)

[beos](#) Este módulo de multiprocesamiento está optimizado para BeOS.

[event](#) An experimental variant of the standard [worker](#) MPM

[mpm\\_netware](#) Multi-Processing Module implementing an exclusively threaded web server optimized for Novell NetWare

[mpmt\\_os2](#) Hybrid multi-process, multi-threaded MPM for OS/2

[prefork](#) Implements a non-threaded, pre-forking web server

[mpm\\_winnt](#) This Multi-Processing Module is optimized for Windows NT.

[worker](#) Multi-Processing Module implementing a hybrid multi-threaded multi-process web server

### Otros Módulos

[A](#) | [C](#) | [D](#) | [E](#) | [F](#) | [H](#) | [I](#) | [L](#) | [M](#) | [N](#) | [P](#) | [R](#) | [S](#) | [U](#) | [V](#)

### Last 20 modules modified/added

[Apache Mobile Filter](#)  
[mod\\_vlimit](#)  
[mod\\_process\\_security](#)  
[mod\\_lalimit](#)  
[mod\\_rchecker](#)  
[mod\\_vlimitconn](#)  
[mod\\_xml2](#)  
[mod\\_ruid2](#)  
[DACS](#)  
[mod\\_badge](#)  
[mod\\_proxy\\_filter\\_xff](#)  
[Apache Rivet](#)  
[mod\\_limits](#)  
[mod\\_auth\\_useragent2](#)  
[mod\\_rangelimit](#)  
[mod\\_chxj](#)  
[mod\\_fortune](#)  
[mod\\_removepass](#)  
[mod\\_access\\_dnsbl](#)  
[mod\\_baik](#)

Total Module Count: 540

# Módulos

## Introducción

---

- ▶ Cada módulo
  - Funcionalidades.
  - Directivas para configurarlas.

### Apache Module `mod_alias`

Available Languages: [en](#) | [ja](#) | [ko](#) | [tr](#)

<b>Description:</b>	Provides for mapping different parts of the host filesystem in the document tree and for URL redirection
<b>Status:</b>	Base
<b>Module Identifier:</b>	alias_module
<b>Source File:</b>	mod_alias.c

#### Summary

The directives contained in this module allow for manipulation and control of URLs as requests arrive at the server. The [Alias](#) and [ScriptAlias](#) directives are used to map between URLs and filesystem paths. This allows for content which is not directly under the [DocumentRoot](#) served as part of the web document tree. The [ScriptAlias](#) directive has the additional effect of marking the target directory as containing only CGI scripts.

The [Redirect](#) directives are used to instruct clients to make a new request with a different URL. They are often used when a resource has moved to a new location.

[mod\\_alias](#) is designed to handle simple URL manipulation tasks. For more complicated tasks such as manipulating the query string, use the tools provided by [mod\\_rewrite](#).

#### Directives

[Alias](#)  
[AliasMatch](#)  
[Redirect](#)  
[RedirectMatch](#)  
[RedirectPermanent](#)  
[RedirectTemp](#)  
[ScriptAlias](#)  
[ScriptAliasMatch](#)

#### Topics

- [Order of Processing](#)

#### See also

- [mod\\_rewrite](#)
- [Mapping URLs to the filesystem](#)

# Módulos

## Introducción

---

### ► Tipos

- Módulos estáticos que se añaden cuando se compila *Apache*.
- Módulos que se cargan dinámicamente cuando se inicia el servidor.
  - Hay que compilar el servidor con la opción DSO (*Dynamic Shared Object*).
    - Ventajas
      - Servidor más flexible.
      - Más sencillo el prototipado y desarrollo de módulos.
    - Desventajas DSO
      - Servidor es más lento en el arranque.
      - Servidor más lento en funcionamiento.

# Módulos

## Introducción

---

### ► Directivas

- **LoadModule**
  - Permite cargar módulos dinámicos.
- **<IfModule *nombre\_modulo*> ... </IfModule>**
  - Especificar directivas que se tendrán en cuenta si el módulo está cargado.

```
LoadModule dir_module /usr/lib/apache2/modules/mod_dir.so
```

```
<IfModule mod_dir.c>  
    DirectoryIndex index.html index.cgi index.pl index.php index.xhtml  
</IfModule>
```

# Módulos

## *Linux (Debian/Ubuntu)*

---

- ▶ Módulos disponibles
  - /usr/lib/apache2/modules

```
alumno@ServidorLinux01:/usr/lib/apache2/modules$ ls
httpd.exp      mod_cgi.so      mod_mime_magic.so
mod_actions.so mod_charset_lite.so mod_mime.so
mod_alias.so   mod_dav_fs.so   mod_negotiation.so
mod_asis.so    mod_dav_lock.so mod_proxy_ajp.so
mod_auth_basic.so mod_dav.so       mod_proxy_balancer.so
mod_auth_digest.so mod_dbd.so        mod_proxy_connect.so
mod_authn_alias.so mod_deflate.so    mod_proxy_ftp.so
mod_authn_anon.so mod_dir.so         mod_proxy_http.so
mod_authn_dbd.so mod_disk_cache.so mod_proxy_scgi.so
mod_authn_dbm.so mod_dumpio.so      mod_proxy.so
mod_authn_default.so mod_env.so         mod_reqtimeout.so
mod_authn_file.so mod_expires.so     mod_rewrite.so
mod_authnz_ldap.so mod_ext_filter.so  mod_setenvif.so
mod_authz_dbm.so mod_file_cache.so  mod_speling.so
mod_authz_default.so mod_filter.so       mod_ssl.so
mod_authz_groupfile.so mod_headers.so      mod_status.so
mod_authz_host.so mod_ident.so        mod_substitute.so
mod_authz_owner.so mod_imagemap.so     mod_suexec.so
mod_authz_user.so mod_include.so       mod_unique_id.so
mod_autoindex.so mod_info.so          mod_userdir.so
mod_cache.so    mod_ldap.so         mod_usertrack.so
mod_cern_meta.so mod_log_forensic.so mod_version.so
mod_cgid.so     mod_mem_cache.so    mod_whoalias.so
```

# Módulos

## *Linux (Debian/Ubuntu)*

---

- ▶ Directorios y ficheros de configuración (1)
  - `/etc/apache2/mods-available/`
    - Módulos disponibles.
    - Ficheros `.load`
      - Para cargar un módulo.
    - Ficheros `.conf`
      - Configuración básica para iniciar el módulo.

```
alumno@ServidorLinux01:/etc/apache2/mods-available$ ls
actions.conf      cern_meta.load   ident.load       proxy_http.load
actions.load      cgid.conf        imagemap.load    proxy.load
alias.conf        cgid.load        include.load     proxy_scgi.load
alias.load        cgi.load         info.conf        reqtimeout.conf
```

# Módulos

## *Linux (Debian/Ubuntu)*

---

- ▶ Directorios y ficheros de configuración (2)
  - `/etc/apache2/mods-enabled/`
    - Módulos habilitados.
    - Enlaces simbólicos a los ficheros de `mods-available`.
    - Módulos a cargar al iniciar Apache.

```
lrwxrwxrwx 1 root root 26 abr 30 14:23 dir.conf -> ../mods-available/dir.conf
lrwxrwxrwx 1 root root 26 abr 30 14:23 dir.load -> ../mods-available/dir.load
lrwxrwxrwx 1 root root 26 abr 30 14:23 env.load -> ../mods-available/env.load
lrwxrwxrwx 1 root root 27 abr 30 14:23 mime.conf -> ../mods-available/mime.conf
lrwxrwxrwx 1 root root 27 abr 30 14:23 mime.load -> ../mods-available/mime.load
```

# Módulos

## *Linux (Debian/Ubuntu)*

---

### ► Comandos

- Habilitar un módulo
  - **a2enmod *nombre\_modulo***
    - Crea un enlace simbólico en `mods_enabled`.
- Deshabilitar un módulo
  - **a2dismod *nombre\_modulo***
    - Borra el enlace simbólico de `mods_enabled`.

- Hay que reiniciar Apache2 al habilitar /deshabilitar módulos.

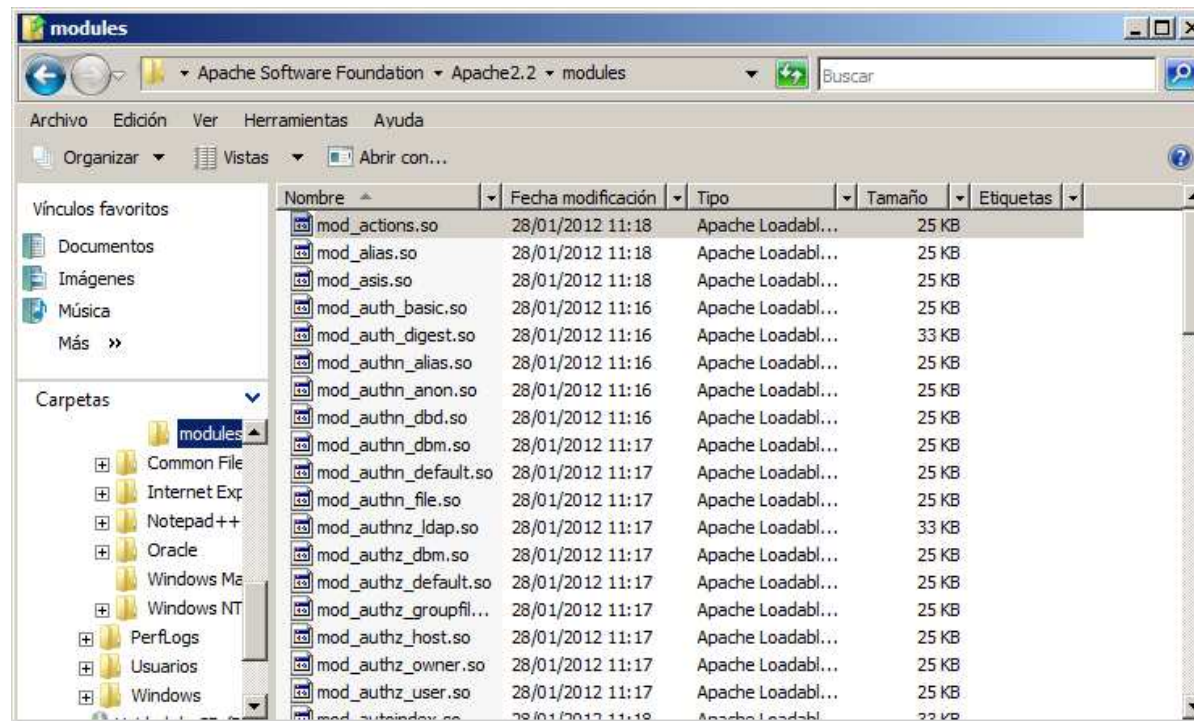


# Módulos

## Windows

### ► Módulos disponibles

- C:\Program Files\Apache Software Foundation\Apache2.2\modules

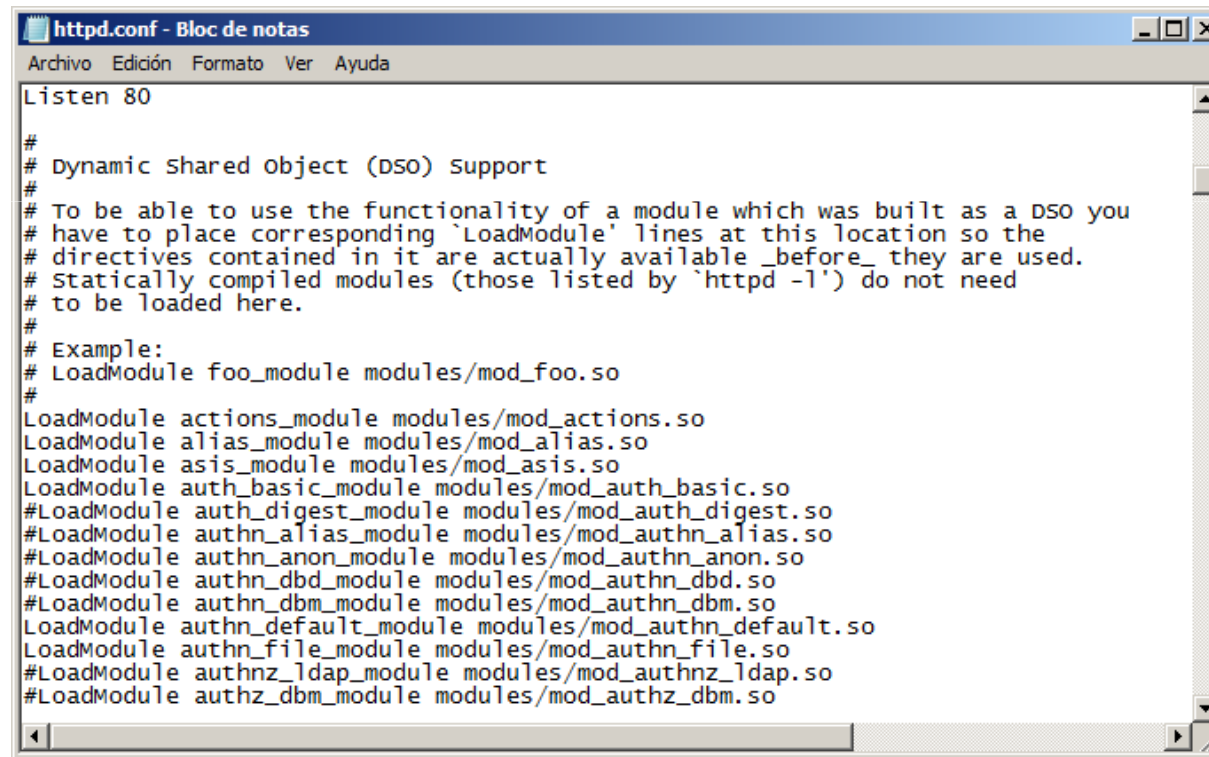


# Módulos

## *Windows*

---

- ▶ Habilitar/deshabilitar módulos.
  - httpd.conf



```
httpd.conf - Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
Listen 80

#
# Dynamic shared object (DSO) support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding 'LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by 'httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
LoadModule actions_module modules/mod_actions.so
LoadModule alias_module modules/mod_alias.so
LoadModule asis_module modules/mod_asis.so
LoadModule auth_basic_module modules/mod_auth_basic.so
#LoadModule auth_digest_module modules/mod_auth_digest.so
#LoadModule authn_alias_module modules/mod_authn_alias.so
#LoadModule authn_anon_module modules/mod_authn_anon.so
#LoadModule authn_dbd_module modules/mod_authn_dbd.so
#LoadModule authn_dbm_module modules/mod_authn_dbm.so
LoadModule authn_default_module modules/mod_authn_default.so
LoadModule authn_file_module modules/mod_authn_file.so
#LoadModule authnz_ldap_module modules/mod_authnz_ldap.so
#LoadModule authz_dbm_module modules/mod_authz_dbm.so
```

# Práctica

---

- ▶ **Práctica 4.7**
  - Módulos en *Linux*.

```
LoadModule alias_module /usr/lib/apache2/modules/mod_alias.so
```

```
<IfModule alias_module>
#
# Aliases: Add here as many aliases as you need (with no limit). The format is
# Alias fakename realname
#
# Note that if you include a trailing / on fakename then the server will
# require it to be present in the URL. So "/icons" isn't aliased in this
# example, only "/icons/". If the fakename is slash-terminated, then the
# realname must also be slash terminated, and if the fakename omits the
# trailing slash, the realname must also omit it.
#
# We include the /icons/ alias for FancyIndexed directory listings. If
# you do not use FancyIndexing, you may comment this out.
#
Alias /icons/ "/usr/share/apache2/icons/"

<Directory "/usr/share/apache2/icons">
    Options FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
</IfModule>
```

# Práctica

---

## ► Práctica 4.8

### ◦ Módulos en *Windows*.

```
LoadModule userdir_module modules/mod_userdir.so

# Settings for user home directories
#
# Required module: mod_userdir
#
# UserDir: The name of the directory that is appended onto a user's home
# directory if a ~user request is received. Note that you must also set
# the default access control for these directories, as in the example below.
#
UserDir "My Documents/My website"

#
# Control access to UserDir directories. The following is an example
# for a site where these directories are restricted to read-only.
#
<Directory "C:/Users/*/My Documents/My website">
    AllowOverride FileInfo AuthConfig Limit Indexes
    Options Multiviews Indexes SymLinksIfOwnerMatch IncludesNoExec
    <Limit GET POST OPTIONS>
        order allow,deny
        Allow from all
    </Limit>
    <LimitExcept GET POST OPTIONS>
        order deny,allow
        Deny from all
    </LimitExcept>
</Directory>
```

# Control de acceso

---

- ▶ Control de acceso a recursos: ficheros, directorios, URLs, ...
  - Control de acceso por host (IP/nombre\_dominio)
    - Módulo [mod\\_authz\\_host](#).
  - Control de acceso por variables de entorno
    - Módulo [mod\\_authz\\_host](#).
    - Módulo [mod\\_setenvif](#).
  - Control de acceso usando el módulo rewrite
    - Módulo [mod\\_rewrite](#).
- ▶ Web
  - <http://httpd.apache.org/docs/2.2/es/howto/access.html>

# Control de acceso

---

## ► Control de acceso (IP/nombre\_dominio) (1)

```
<Directory /var/www/profesor>  
    Options Indexes FollowSymLinks MultiViews  
    AllowOverride None  
    Order allow,deny  
    allow from 127.0.0.1  
    allow from 192.168.1.16  
</Directory>
```

- Order Deny,Allow | Allow,Deny
- Allow from
- Deny from

# Control de acceso

---

## ► Control de acceso (IP/nombre\_dominio) (2)

- Order Deny, Allow
  - **El acceso está permitido por defecto.** Las directivas Deny se evalúan antes que las directivas Allow. Cualquier cliente que “no case” con una directiva Deny a Allow tendrá permitido el acceso. Si el cliente “casa” al mismo tiempo en una directiva Allow y otra Deny, tendrá permitido el acceso por que las directivas Allow se evalúan las últimas.
- Order Allow, Deny
  - **El acceso está denegado por defecto.** Las directivas Allow se evalúan antes que las directivas Deny. Cualquier cliente que “no case” con una directiva Deny a Allow tendrá denegado el acceso. Si el cliente “casa” al mismo tiempo en una directiva Allow y otra Deny, tendrá denegado el acceso por que las directivas Deny se evalúan las últimas.

# Control de acceso

---

## ► Control de acceso (IP/nombre\_dominio) (3)

### ◦ Ejemplos (1)

Order Deny,Allow	1) Acceso permitido por defecto
Deny from all	2) Todos los hosts so denegados
Allow from daw.org	3) Se permite el acceso a los hosts de dominio *.daw.org

**Resultado: Solo los host de \*.daw.org son permitidos.**

Order Allow, Deny	1) Acceso denegado por defecto
Allow from daw.org	2) Se permite el acceso a los hosts de dominio *.daw.org
Deny from bbdd.daw.org	3) Se deniega el acceso a los hosts de dominio *.bbdd.daw.org

**Resultado: Los hosts de \*.daw.org son permitidos execepto los de \*.bbdd.daw.org.**

**¿Qué ocurre si se cambia el orden a Order Deny,Allow ?**



# Control de acceso

---

- ▶ Control de acceso (IP/nombre\_dominio) (4)
  - Ejemplos (2)

Order Allow,Deny  
Allow from 200.200.100.0/34  
Deny from www.daw.org

¿Resultado?

Order Allow,Deny  
Allow from 192.168.0.0/16  
Deny from all

¿Resultado?

# Práctica

---

## ► Práctica 4.9

- Control de acceso por IP y nombre de dominio.

```
<Directory /var/www/profesor>  
    Options Indexes FollowSymLinks MultiViews  
    AllowOverride None  
    Order allow,deny  
    allow from 127.0.0.1  
    allow from 192.168.1.16  
</Directory>
```

# Autenticación y autorización

---

## ▶ Autenticación

- Proceso para verificar que alguien es realmente quien dice ser.

## ▶ Autorización

- Proceso por el que se permite a alguien hacer o acceder a algo que quiere.

# Autenticación y autorización

---

- ▶ Tipos de autenticación

- *Basic*

- Módulo `mod_auth_basic`.

- *Digest*

- Módulo `mod_auth_digest`.

- ▶ *Formularios HTTL.*

- ▶ *Certificados digitales*

- ▶ Módulo `mod_ssl`.

- ▶ Web

- <http://httpd.apache.org/docs/2.2/es/howto/auth.html>

# Autenticación y autorización

---

## ► Proveedores de autenticación

- Módulos que ofrecen la posibilidad de acceder a credenciales (usuarios, contraseñas, certificados, ...) usados en la autenticación en:
  - Ficheros de texto.
  - Bases de datos.
  - Servidores de directorios (LDAP).
  - ...

## ► Web

- <http://httpd.apache.org/docs/2.2/es/howto/auth.html>

# Autenticación y autorización

---

## ► Módulos de autorización

- Módulos que permite realizar el proceso de autorización sobre:
  - Ficheros de texto.
  - Bases de datos.
  - Servidores de directorios (LDAP).
  - ...

## ► Web

- <http://httpd.apache.org/docs/2.2/es/howto/auth.html>
- <http://httpd.apache.org/docs/2.2/es/mod/core.html#require>

# Autenticación y autorización

---

## ► Autenticación *Basic* (1)

- [mod\\_auth\\_basic](#)
- La contraseña es enviada por el cliente en texto plano.
- Autenticación y autorización sobre fichero de texto (htpasswd).
  - [mod\\_authn\\_file](#)
  - [mod\\_authz\\_user](#)

# Autenticación y autorización

---

## ► Autenticación *Basic* (2)

- 1) Crear fichero con usuarios/contraseñas
  - **htpasswd**
    - <http://httpd.apache.org/docs/2.2/es/programs/htpasswd.html>

```
# La primera vez que se invoca el comando se
# utiliza a opción -c para crear el fichero
htpasswd -c /etc/apache2/passwd profesor1

# Añade un nuevo usuario al fichero
htpasswd /etc/apache2/passwd profesor2

# Borrar un nuevo usuario al fichero
htpasswd -D /etc/apache2/passwd profesor1
```



# Autenticación y autorización

---

## ► Autenticación *Basic* (3)

- 2) Definir directivas

```
<Directory /var/www/profesor>  
    Options Indexes FollowSymLinks MultiViews  
    AllowOverride None  
    Order allow,deny  
    allow from 127.0.0.1  
    allow from 192.168.1.16  
    AuthType Basic  
    AuthName "Acceso restringido"  
    AuthUserFile /etc/apache2/passwd  
    Require user profesor1 profesor2  
</Directory>
```

- AuthType, AuthName, AuthUserFile,  
AuthGroupFile, Require, ...

# Autenticación y autorización

---

## ► Autenticación *Digest* (1)

- [mod\\_auth\\_digest](#)
- La contraseña se envía cifrada (¡¡ **cifrado débil, no es seguro !!**) por el cliente.
- Autenticación y autorización sobre fichero de texto (htdigest).
  - [mod\\_auth\\_digest](#)
  - [mod\\_authz\\_user](#)

# Autenticación y autorización

---

## ► Autenticación *Digest* (2)

- 1) Crear fichero con usuarios/contraseñas asociados a un dominio (*realm*).
  - **htdigest**
    - <http://httpd.apache.org/docs/2.2/es/programs/htdigest.html>

```
# La primera vez que se invoca el comando se
# utiliza a opción -c para crear el fichero
htdigest -c /etc/apache2/passwd informatica admin1

# Añade un nuevo usuario al fichero
Htdigest /etc/apache2/passwd informatica admin2

# Borrar un nuevo usuario al fichero
htdigest -D /etc/apache2/passwd informatica admin1
```

# Autenticación y autorización

---

## ► Autenticación *Digest* (3)

- 2) Definir directivas

```
<Directory /var/www/departamento>  
    Options Indexes FollowSymLinks MultiViews  
    AllowOverride None  
    AuthType Digest  
    AuthName "informatica"  
    AuthDigestProvider file  
    AuthUserFile /etc/apache2/digest  
    Require user admin1 admin2  
</Directory>
```

- AuthType, AuthName, AuthDigestProvider, AuthUserFile, AuthGroupFile, Require, ...

# Práctica

---

## ► Práctica 4.10

- Autenticación y autorización *Basic* y *Digest*

```
<Directory /var/www/profesor>
  Options Indexes FollowSymLinks MultiViews
  AllowOverride None
  Order allow,deny
  allow from 127.0.0.1
  allow from 192.168.1.16
  AuthType Basic
  AuthName "Acceso restringido"
  AuthUserFile /etc/apache2/passwd
  Require user profesor1 profesor2
</Directory>
```

```
<Directory /var/www/departamento>
  Options Indexes FollowSymLinks MultiViews
  AllowOverride None
  AuthType Digest
  AuthName "informatica"
  AuthDigestProvider file
  AuthUserFile /etc/apache2/digest
  Require user admin1 admin2
</Directory>
```

# Ficheros .htaccess

---

- ▶ Ficheros que permiten la configuración personalizada de directorios.
  - Fichero de configuración de *Apache*

```
Alias /blog /home/profesor/blog  
<Directory /home/profesor/blog>  
    AllowOverride All  
</Directory>
```

- Fichero .htaccess dentro de un directorio

```
Options Indexes  
Order allow,deny  
allow from 192.168.1.16  
AuthType Digest  
AuthName "informatica"  
AuthUserFile /home/profesor/.htdigest  
Require user blog
```

# Ficheros .htaccess

---

- ▶ Cada vez que se produce una petición:
  - El servidor busca en la ruta del recurso que ha solicitado el cliente un fichero con el nombre .htaccess.
  - Aplica sobre el directorio las directivas definidas.
- ▶ En la configuración del servidor hay que permitir el uso de estos ficheros.
- ▶ Web
  - <http://httpd.apache.org/docs/2.2/es/howto/htaccess.html>

# Ficheros .htaccess

---

- ▶ Definida la siguiente directiva a nivel del servidor principal para que los ficheros que empiecen con .ht no sea visibles por los clientes.

- Windows

```
<FilesMatch "^\.ht">  
    Order allow,deny  
    Deny from all  
    Satisfy All  
</FilesMatch>
```

- Linux

```
<Files ~ "^\.ht">  
    Order allow,deny  
    Deny from all  
    Satisfy all  
</Files>
```



# Ficheros .htaccess

---

- ▶ No se deben usar a menos que no se tenga acceso al archivo de configuración del servidor (Ej.: Servidor de *hosting*)
  - Eficiencia.
  - Seguridad.
- ▶ El nombre .htaccess se puede cambiar con la directiva `AccessFileName`.

# Práctica

---

- ▶ **Práctica 4.1 1**
  - Ficheros .htaccess.

```
Alias /blog /home/profesor/blog
<Directory /home/profesor/blog>
    AllowOverride All
</Directory>
```

```
Options Indexes
Order allow,deny
allow from 192.168.1.16
AuthType Digest
AuthName "informatica"
AuthUserFile /home/profesor/.htdigest
Require user blog
```