

4. Administración de Apache

4.7. Módulos en *Linux*

Apache es un servidor modular. El núcleo del servidor tiene las funcionalidades básicas que se pueden ampliar añadiendo módulos adicionales. Cada módulo agrupa un conjunto de funcionalidades y directivas para configurarlas. Existen múltiples módulos que puedes consultar en <http://httpd.apache.org/docs/2.2/mod/> y <http://modules.apache.org/>.

Existen dos tipos de módulos:

- Módulos estáticos que se añaden cuando se compila *Apache*.
- Módulos que se cargan dinámicamente cuando se inicia el servidor.

De forma predeterminada, durante la compilación se incluye un juego básico de módulos en el servidor (estáticos). Si el servidor se compila con la opción (DSO. *Dynamic Shared Object*) se podrán cargar módulos dinámicos con la directiva *LoadModule*.

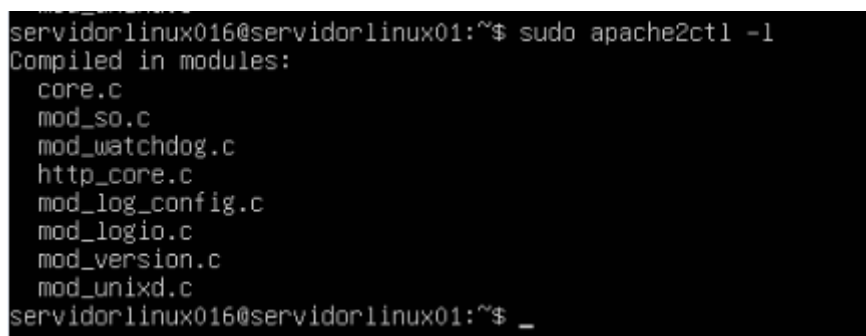
La directiva `<IfModule nombremodulo> </IfModule>` permite especificar directivas que se tendrán en cuenta si el módulo indicado está cargado.

Consulta los módulos estáticos y los módulos dinámicos cargados por defecto en la versión de *Apache* instalada en la máquina ServidorLinuxXX. Posteriormente, investiga la funcionalidad del módulo *userdir*, habilítalo y prueba su funcionalidad.

1. Módulos

- 1.1. Inicia una sesión en ServidorLinuxXX con un usuario con privilegios de administrador.
- 1.2. Comprueba los módulos estáticos que se han cargado al compilar el servidor ejecutando el siguiente comando. Figura 4.52:

```
sudo apache2ctl -l
```



```
servidorlinux016@servidorlinux01:~$ sudo apache2ctl -l
Compiled in modules:
  core.c
  mod_so.c
  mod_watchdog.c
  http_core.c
  mod_log_config.c
  mod_logio.c
  mod_version.c
  mod_unixd.c
servidorlinux016@servidorlinux01:~$ _
```

Figura 4.52: Módulos estáticos

- 1.3. Comprueba los módulos que se han cargado dinámicamente al arrancar el servidor consultando el directorio `/etc/apache2/mods-enabled`. Observa que los ficheros aparecen en este directorio son enlaces simbólicos a ficheros de `/etc/apache2/mods-available`.
- 1.4. Edita uno de los ficheros `.load` (por ejemplo `dir.load`) y observa cómo se utiliza la directiva *LoadModule*, Figura 4.53 para cargar el módulo. Comprueba cuál es la ruta donde está el código del módulo (archivo `.so`).

```
servidorlinux016@servidorlinux01:/etc/apache2/mods-enabled$ cat dir.load
LoadModule dir_module /usr/lib/apache2/modules/mod_dir.so
servidorlinux016@servidorlinux01:/etc/apache2/mods-enabled$ _
```

Figura 4.53: Fichero **dir.load**

- 1.5. Edita uno de los ficheros **.conf** (por ejemplo **dir.conf**) y observa cómo se añaden directivas dentro de una declaración `<IfModule nombremodulo>...</IfModule>`, *Figura 4.54* que se ejecutarán si se carga el módulo.

```
servidorlinux016@servidorlinux01:/etc/apache2/mods-enabled$ cat dir.load
LoadModule dir_module /usr/lib/apache2/modules/mod_dir.so
servidorlinux016@servidorlinux01:/etc/apache2/mods-enabled$ cat dir.conf
<IfModule mod_dir.c>
    DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
</IfModule>
```

Figura 4.54: Fichero **dir.conf**

- 1.6. Consulta el directorio **/usr/lib/apache2/modules/** y observa los módulos disponibles para cargar.
- 1.7. Ejecuta el siguiente comando para mostrar los paquetes disponibles en los repositorios de *Ubuntu* que permiten instalar módulos adicionales en *Apache*, *Figura 4.55*

```
sudo apt-cache search libapache2-mod
```

```
libapache2-mod-musicindex - Browse, stream, download and search through MP3/Ogg/FLAC files
libapache2-mod-netcgi-apache - OCaml application-level Internet libraries - netcgi2 Apache2 connecto
r
libapache2-mod-parser3 - Parser 3, HTML-embedded scripting language (Apache2 module)
libapache2-mod-passenger - Rails and Rack support for Apache2
libapache2-mod-proxy-msrpc - Apache module for Outlook Anywhere support in reverse proxy setups
libapache2-mod-python - Python-embedding module for Apache 2
libapache2-mod-python-doc - Python-embedding module for Apache 2 - documentation
libapache2-mod-qos - quality of service module for the apache2
libapache2-mod-r-base - server-side R integration with Apache 2
libapache2-mod-rivet - Server-side Tcl programming system combining ease of use and power
libapache2-mod-rivet-doc - Documentation for Rivet, a server-side Tcl programming system
libapache2-mod-rpaf - module for Apache2 which takes the last IP from the 'X-Forwarded-For' header
libapache2-mod-ruid2 - suexec module for Apache 2
libapache2-mod-ruwsgi - uwsgi module for Apache2 (mod_Ruwsgi)
libapache2-mod-ruwsgi-dbg - debugging symbols for Apache2 mod_Ruwsgi
libapache2-mod-security2 - Tighten web applications security for Apache
libapache2-mod-shib - Federated web single sign-on system (Apache module)
libapache2-mod-svn - Apache Subversion server modules for Apache httpd
libapache2-mod-upload-progress - upload progress support for the Apache web server
libapache2-mod-uwsgi - uwsgi module for Apache2 (mod_uwsgi)
libapache2-mod-uwsgi-dbg - debugging symbols for Apache2 mod_uwsgi
libapache2-mod-vhost-ldap - Apache 2 module for Virtual Hosting from LDAP
libapache2-mod-watchcat - Process monitoring Apache module
libapache2-mod-xforward - Apache module implements redirection based on X-Forward response header
libapache2-mod-xsendfile - Serve large static files efficiently from web applications
liblinux-pid-perl - wrapper around the getpid() and getppid() C functions
libocamlnet-ocaml-dev - OCaml application-level Internet libraries - core development libraries
mapcache-cgi - tile caching server - CGI binary
mono-apache-server4 - ASP.NET 4.5 backend for mod_mono Apache module
mono-fastcgi-server4 - ASP.NET 4.0 backend for FastCGI web servers
mono-fpm-server - ASP.NET backend for FastCGI Process Manager
php7.4-fpm - server-side, HTML-embedded scripting language (FPM-CGI binary)
radicale - simple calendar and addressbook server - daemon
shibboleth-sp-utils - Federated web single sign-on system (daemon and utilities)
tclws - Tcl Web Services
servidorlinux016@servidorlinux01:/usr/lib/apache2/modules$ _
```

Figura 4.55: Módulos disponibles en los repositorios de *Ubuntu*

2. Directorios personales de usuarios (módulo *userdir*)

- 2.1. Comprueba, consultando el directorio **/etc/apache2/mods-enabled** que el módulo *userdir* no está habilitado.
- 2.2. Habilita el módulo ejecutando el siguiente comando:

```
sudo a2enmod userdir
```

El comando *a2dismod* permite deshabilitar un módulo (borrar los enlaces simbólicos del directorio *mods-enabled*).

- 2.3. Verifica que dentro del directorio **/etc/apache2/mods-enabled** se han creado enlaces simbólicos del módulo *userdir* (archivos **.conf** y **.load**) hacia **/etc/apache2/mod_ables**, Figura 4.56.

```
lrwxrwxrwx 1 root root 30 jul  4 19:22 userdir.conf -> ../mods-available/userdir.conf
lrwxrwxrwx 1 root root 30 jul  4 19:22 userdir.load -> ../mods-available/userdir.load
servidorlinux016@servidorlinux01:/etc/apache2/mods-enabled$
```

Figura 4.56: Módulo *userdir* habilitado

- 2.4. Reinicia el servidor para que los cambios tengan efecto.
- 2.5. Consulta el fichero **/etc/apache2/mods-enabled/userdir.conf**. Observa que está habilitado el uso de directorios personales para todos los usuarios excepto para el usuario **root** y que **public_html** es el nombre del subdirectorio que pueden crear los usuarios en su directorio **home** para poner sus páginas personales.
- 2.6. Inicia una sesión en **ServidorLinuxXX** con el usuario **alumno**.
- 2.7. Crea en directorio **/home/alumno/public_html**. Dentro del directorio crea un fichero denominado **personal.html** que contenga el texto "PAGINA PERSONAL DEL USUARIO ALUMNO".
- 2.8. Desde **DesarrolloW7XX** accede a http://172.16.10.XX6/~alumno/personal_html y verifica que se sirve la página.

Nota: Usando el módulo *userdir* cada usuario puede crear sus páginas en */home/directorio-personal/public_html* y acceder a ellas. (No es necesario que el Administrador del Servidor Web realice ninguna operación y tampoco es necesario reiniciar el servidor cuando un usuario cambie o añada una página web). Figura 4.57

Nota: Si queremos que las páginas que creen los usuarios las guarden en un directorio llamado *miweb* en lugar de *public_html* debemos modificar el fichero **/etc/apache2/mods-available/userdir.conf** en las líneas siguientes:

```
...      UserDir      miweb
...
<Directory /home/*/miweb>
...
</Directory>
```

- 2.9. Deshabilita el módulo ejecutando el siguiente comando:

```
sudo a2dismod userdir
```

- 2.10. Reinicia el servidor para que los cambios tengan efecto.

```
sudo service apache2 restart
```

Nota: Usa Alt+126 para sacar ~

```
servidorlinux016@servidorlinux01:/etc/apache2/mods-enabled$ cat 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>
```

Figura 4.57: Archivo *userdir.conf*