

# APACHE SERVER

GESTIÓN DE MÓDULOS. MOD\_USERDIR

Salvador Mira

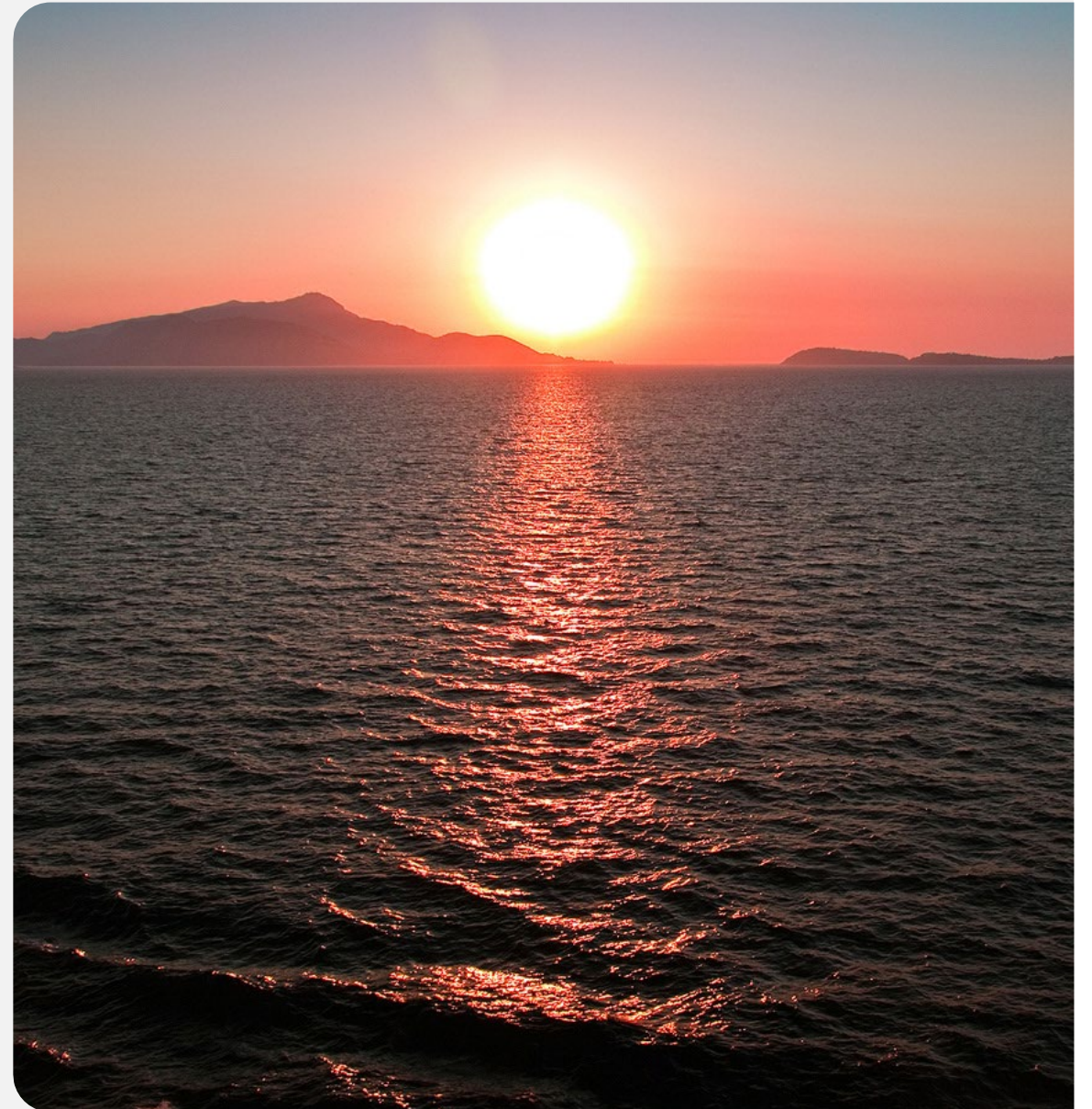
IES Pere Maria Orts

# MÓDULOS DE APACHE

Junto a su instalación, Apache incorpora gran cantidad de módulos que amplían su funcionalidad

Todas las configuraciones para los módulos disponibles se encuentran en **`/etc/apache2/mods-available`**

Sólo aquellos enlazados en **`/etc/apache2/mods-enabled`** estarán activos





# MÓDULOS DE APACHE

Para activar un módulo usaremos la utilidad **a2enmod** y para deshabilitarlo **a2dismod**

- Ej. **a2enmod rewrite** habilita el módulo **rewrite**
- Ej. **a2dismod rewrite** deshabilita el módulo **rewrite**

Tras la activación o desactivación de un módulo, debemos recargar la configuración



# MÓDULOS DE APACHE

Podemos conocer los módulos activos de Apache ejecutando el comando:  
**sudo apache2ctl -t -D DUMP\_MODULES**

```
administrador@debian-despliegue: /etc/apache2/mods-available
administrador@debian-despliegue:/etc/apache2/mods-available$ sudo apache2ctl -t
-D DUMP_MODULES
Loaded Modules:
core_module (static)
so_module (static)
watchdog_module (static)
http_module (static)
log_config_module (static)
logio_module (static)
version_module (static)
unixd_module (static)
access_compat_module (shared)
alias_module (shared)
auth_basic_module (shared)
authn_core_module (shared)
authn_file_module (shared)
authz_core_module (shared)
authz_host_module (shared)
authz_user_module (shared)
autoindex_module (shared)
deflate_module (shared)
dir_module (shared)
env_module (shared)
filter_module (shared)
```





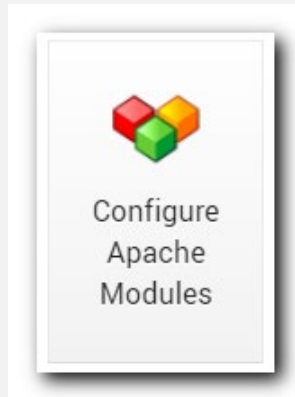
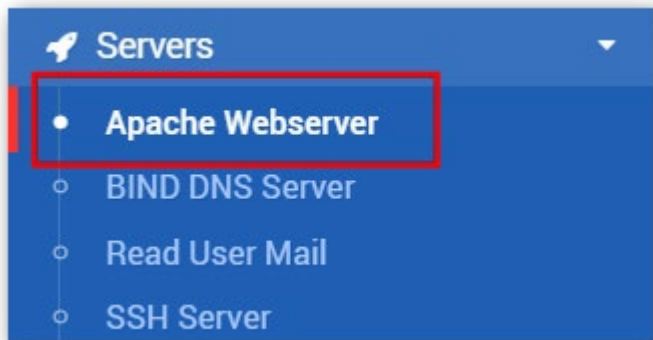
# MÓDULOS DE APACHE

## WEBMIN

Es posible gestionar los módulos de **Apache** desde **Webmin**

- Servers → Apache Webserver
- Pestaña **Global configuration** > **Configure Apache Modules**

**Webmin** nos mostrará los activos y nos permitirá activar o desactivar módulos disponibles



# MÓDULOS DE APACHE

## WEBMIN



### ☆ Configure Apache Modules

This page allows you to select which optional Apache modules are enabled, using the checkbox next to each module name. Be careful when disabling a module, as any existing directives that are specific to it will be no longer recognized.

Module	Current state	Module	Current state
<input checked="" type="checkbox"/> access_compat	Enabled	<input type="checkbox"/> lbmethod_bytraffic	Disabled
<input type="checkbox"/> actions	Disabled	<input type="checkbox"/> lbmethod_heartbeat	Disabled
<input checked="" type="checkbox"/> alias	Enabled	<input type="checkbox"/> ldap	Disabled
<input type="checkbox"/> allowmethods	Disabled	<input type="checkbox"/> log_debug	Disabled
<input type="checkbox"/> asis	Disabled	<input type="checkbox"/> log_forensic	Disabled
<input checked="" type="checkbox"/> auth_basic	Enabled	<input type="checkbox"/> lua	Disabled
<input type="checkbox"/> auth_digest	Disabled	<input type="checkbox"/> macro	Disabled
<input type="checkbox"/> auth_form	Disabled	<input type="checkbox"/> md	Disabled
<input type="checkbox"/> authn_anon	Disabled	<input checked="" type="checkbox"/> mime	Enabled
<input checked="" type="checkbox"/> authn_core	Enabled	<input type="checkbox"/> mime_magic	Disabled
<input type="checkbox"/> authn_dbd	Disabled	<input checked="" type="checkbox"/> mpm_event	Enabled
<input type="checkbox"/> authn_dbm	Disabled	<input type="checkbox"/> mpm_prefork	Disabled
<input checked="" type="checkbox"/> authn_file	Enabled	<input type="checkbox"/> mpm_worker	Disabled
<input type="checkbox"/> authn_socache	Disabled	<input checked="" type="checkbox"/> negotiation	Enabled
<input type="checkbox"/> authnz_fcgi	Disabled	<input type="checkbox"/> proxy	Disabled
<input type="checkbox"/> authnz_ldap	Disabled	<input type="checkbox"/> proxy_ajp	Disabled
<input checked="" type="checkbox"/> authz_core	Enabled	<input type="checkbox"/> proxy_balancer	Disabled
<input type="checkbox"/> authz_dbd	Disabled	<input type="checkbox"/> proxy_connect	Disabled
<input type="checkbox"/> authz_dbm	Disabled	<input type="checkbox"/> proxy_express	Disabled
<input type="checkbox"/> authz_groupfile	Disabled	<input type="checkbox"/> proxy_fcgi	Disabled

# MÓDULOS DE APACHE

EJEMPLO: MOD\_USERDIR

Vamos a habilitar un **módulo** para que los **usuarios de Linux** puedan publicar sus páginas web desde su directorio personal

Los **requisitos** son:

- Habilitar el módulo **userdir**
- En la configuración estándar de este módulo, el Apache espera encontrar un directorio **public\_html** en la carpeta personal de cada usuario
- Este directorio debe tener permisos **755**



# MÓDULOS DE APACHE

EJEMPLO: MOD\_USERDIR

- **Paso 1:** Activar el módulo userdir con `sudo a2enmod userdir` (o bien desde Webmin)
- **Paso 2:** Reiniciar Apache con `sudo service apache2 restart`
- **Paso 3:** Crear el directorio **public\_html** en la carpeta de usuario y darle permisos **755**
- **Paso 4:** Crear una página de prueba llamada **index.html** dentro de **public\_html**
- **Paso 5:** Comprobar la página accediendo a la dirección **`http://servidor/~usuario`**





# MÓDULOS DE APACHE

EJEMPLO: MOD\_USERDIR



```
administrador@debian-despliegue: ~/public_html
administrador@debian-despliegue:~/public_html$ pwd
/home/administrador/public_html
administrador@debian-despliegue:~/public_html$ cat index.html
<html>
<head><meta charset="UTF-8"/></head>
<body>
<h1>Bienvenido a la página del administrador</h1>
</body>
</html>
administrador@debian-despliegue:~/public_html$ |
```

# MÓDULOS DE APACHE

EJEMPLO: MOD\_USERDIR

Podemos mejorar más aún esto, haciendo que cada vez que se cree un usuario en el sistema se le cree el directorio personal **public\_html** más una página **index.html** de prueba

Para ello hay que dirigirse a **/etc/skel** y crear allí el directorio **public\_html** con la página **index.html** en su interior

Todo el contenido de **/etc/skel** se copia en el perfil de usuario en su creación

Probar a crear un usuario desde la línea de comandos mediante **sudo adduser *nombreusuario*** y ver los resultados

