

ACTIVITAT AVALUABLE AC3**Mòdul:** MP08- Desplegament d'aplicacions web**Professor:** Marc Callejón**Data límit d'entrega:** 14/10/2025 23:59**Mètode d'entrega:** Per mitjà del Clickedu de l'assignatura. Les activitats entregades més enllà de la data límit només podran obtenir una nota de 5.**Tasques a realitzar:****Apache: servidores virtuales (por nombre de dominio)**

Tarea 1. Por defecto, un servidor Apache puede servir las páginas webs ubicadas en ciertos directorios del sistema de ficheros que lo aloja.

- a) Analiza las directivas *Directory* del fichero de configuración principal de Apache: ¿qué directorios están accesibles para ser servidos al cliente?

```
# access here, or in any related virtual host.
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Require all denied
</Directory>

<Directory /usr/share>
    AllowOverride None
    Require all granted
</Directory>

<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

Estas son las accesibles por el usuario.

- b) La directiva *DocumentRoot* (que no se especifica en el fichero de configuración principal) establece cuál es el directorio desde el que se sirven páginas web por defecto. Con lo que sabes hasta ahora, puedes deducir cuál es el *DocumentRoot* del servidor?

Contiene los archivos accesibles del servidor.

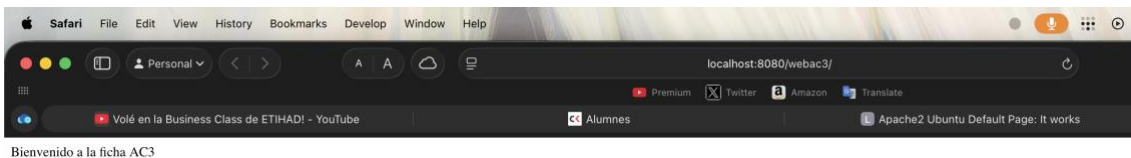
- c) Añade, en el directorio que haga de *DocumentRoot*, una nueva carpeta llamada *webac3*, que incluya un *index.html* con el contenido “Bienvenido a la ficha AC3”.



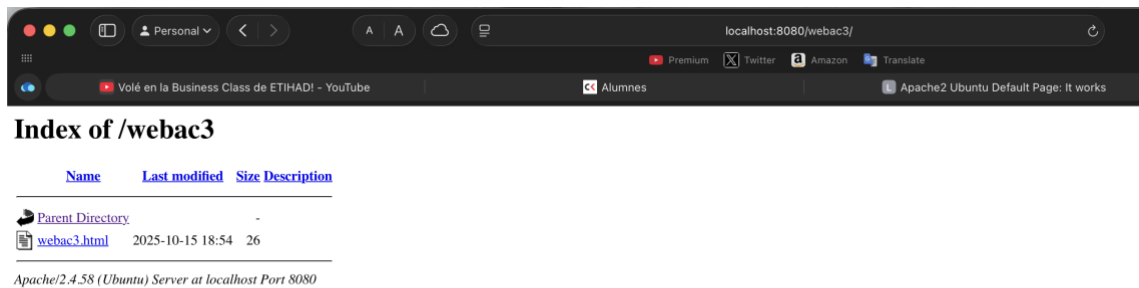
```
GNU nano 7.2
Bienvenido a la ficha AC3

ubuntu@ubuntu:/var/www/html/webac3$
```

- d) Comprueba que Apache sirve esa página, desde la url <http://localhost:8080/webac3/>.



- e) Renombra el fichero *webac3/index.html* a *webac3/webac3.html*, y comprueba qué página sirve ahora Apache desde la máquina anfitriona.



g)¿Por qué ocurre lo anterior?

Porque al llamarse “index.html” ese nombre es especial e indica que apache que cargar al entrar en esa ruta en sí.

h)Investiga en el fichero de configuración principal de Apache qué directiva habilita ese comportamiento (dentro de la configuración del directorio: <Directory /var/www></Directory>).

```
<Directory /var/www/>
#       Options Indexes FollowSymLinks
       AllowOverride None
       Require all granted
</Directory>

#<Directory /srv/>
#       Options Indexes FollowSymLinks
#       AllowOverride None

ubuntu@ubuntu:/etc/apache2$ sudo systemctl restart apache2
[sudo] password for ubuntu:
ubuntu@ubuntu:/etc/apache2$ _
```

i)Deshabilita esa opción y comprueba los cambios al pedir la URL <http://localhost:8080/webac3>.

Forbidden

You don't have permission to access this resource.

Apache/2.4.58 (Ubuntu) Server at localhost Port 8080

Tarea 2. El *DocumentRoot* por defecto está configurado en los ficheros de configuración del servidor virtual por defecto.

a)Compruébalo en

<directorio de configuración de Apache>/sites-available/000-default.conf

```
GNU nano 7.2                                000-default
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html_

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

b) Después de la instalación, el servidor Apache tiene configurado un *único servidor virtual*. Comprueba que sólo en el directorio *sites-enabled* sólo figura el servidor virtual *000-default*.

```
ubuntu@ubuntu:/etc/apache2/sites-enabled$ ls /etc/apache2/sites-enabled/
000-default.conf
```

c) Apache, no obstante, es capaz de servir distintos sitios webs, cada uno de ellos (y esto es lo importante) con su propio *DocumentRoot*. Cada uno de esos sitios webs tendrá, además, su propia configuración y sus propios controles de acceso: a ojos del usuario, aparecerá como un servidor totalmente independiente. Para ello se utiliza la técnica de los servidores virtuales.

d) Investiga la diferencia entre un servidor virtual por nombre de dominio y un servidor virtual por IP. Investiga si pueden configurarse en Apache ambos tipos de servidor virtual.

Dominio: Son varios sitios web que tienen la misma ip y vienen del mismo servidor pero apache diferencia el dominio al que entras y te da acceso a uno o a otro

IP: Son varios sitios web que tienen diferente IP y depende de por cual entres apache decide que página web devolverte.

Tarea 3. En primer lugar, vamos a probar la técnica de los servidores virtuales por nombre de dominio.

Necesitamos que la IP de nuestra MV esté asociada a más de un nombre de dominio, para que nuestro servidor Apache pueda servir páginas webs diferentes para diferentes nombres de dominio. Por ejemplo: nos gustaría que si accedemos a la URL www.dominio1.com se nos muestre una web que dé la bienvenida al dominio 1; pero si accedemos a la URL www.dominio2.com, se nos muestre una web que dé la bienvenida a dominio 2.

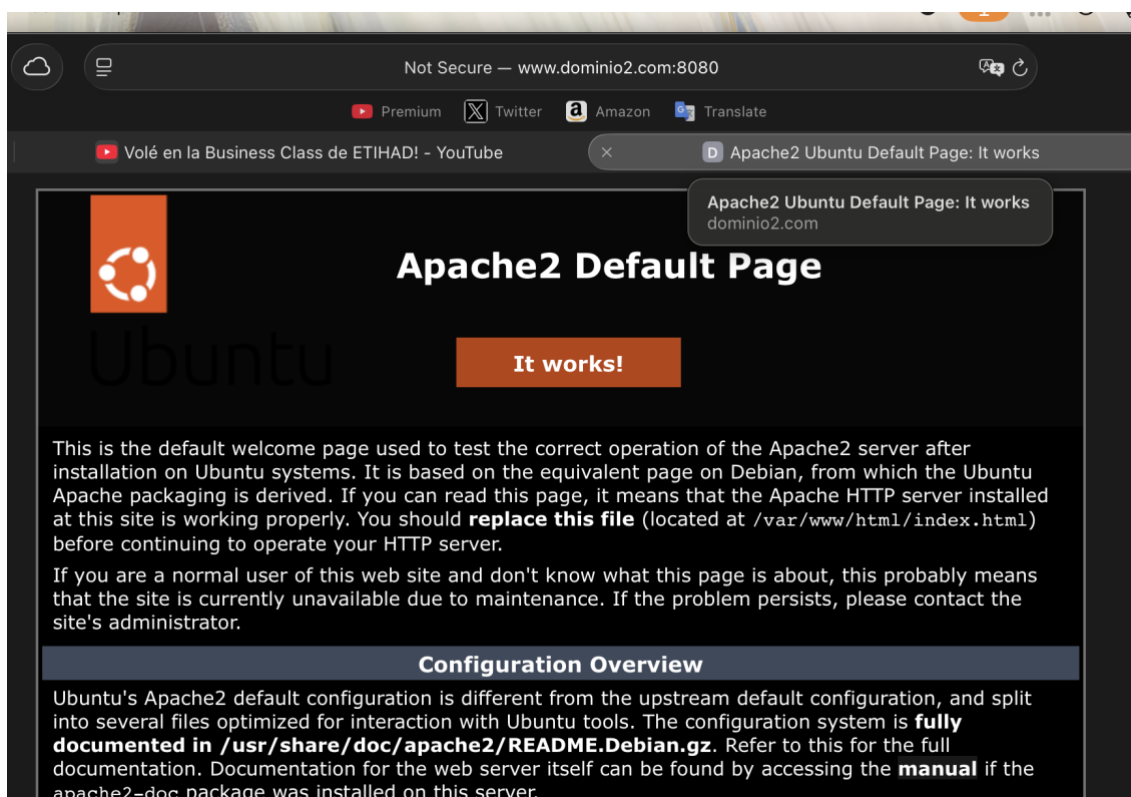
a) Para ello, hay que añadir una nueva sinonimia *nombre de dominio* <-> *dirección IP* en la red. Como aún no sabemos hacerlo a nivel de DNS, vamos a simular esa traducción en nuestra máquina *host*.

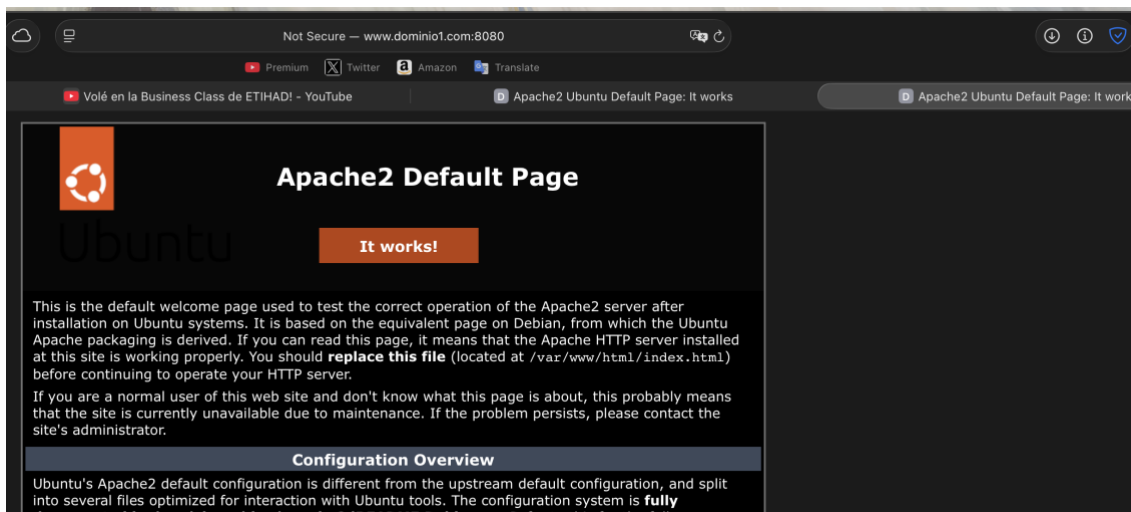
Por ejemplo (para una MV en modo NAT):

127.0.0.1	www.dominio1.com
127.0.0.1	www.dominio2.com

b) Comprueba que podemos acceder a nuestro servidor web a través de las URLs <http://www.dominio1.com> y <http://www.dominio2.com>

Si tienes la MV en NAT, recuerda acceder al puerto al que estés redireccionando el tráfico de Apache.





Tarea 4. A continuación, debemos crear dos directorios independientes para alojar las diferentes webs que queremos que sean servidas en los nombres de dominio www.dominio1.com y www.dominio2.com, respectivamente.

a) En el directorio `/var/www/html/` crea dos estructuras de ficheros como las siguientes:

`www.dominio1.com/index.html`

`www.dominio2.com/index.html`

```
ubuntu@ubuntu:/etc/apache2/sites-available$ sudo mkdir /var/www/html/www.dominio1.com
[sudo] password for ubuntu:
Sorry, try again.
[sudo] password for ubuntu:
ubuntu@ubuntu:/etc/apache2/sites-available$ sudo mkdir /var/www/html/www.dominio2.com
ubuntu@ubuntu:/etc/apache2/sites-available$
```

b) Crea un contenido HTML distinto para cada una de las dos páginas web. Por ejemplo, que contengan un mensaje de bienvenida al dominio 1 o al dominio 2, respectivamente.

```
GNU nano 7.2 /var/www/html/www.dominio2.com/index.html
Dominio 2

GNU nano 7.2 /var/www/html/www.dominio1.com/index.html
Dominio 1
```

Tarea 5. El siguiente punto que debemos conseguir es que el servidor Apache sirva páginas distintas para las peticiones a uno u otro nombre de dominio. Para ello deberemos crear dos nuevos servidores virtuales.

a) Accede al directorio `/etc/apache2/sites-available`. Realiza una copia del fichero `000-default.conf` que se llame `001-www.dominio1.com.conf`, y editala para que:

-Conste en la directiva `ServerName` el nombre de dominio con el que queremos asociar el nuevo servidor virtual.

-Conste en la directiva *DocumentRoot* el directorio donde se aloja el sitio web que queremos asociar al nombre de dominio www.dominio1.com, en este caso:

`/var/www/html/www.dominio1.com`

```
GNU nano 7.2                                002-www.dominio2.com
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    ServerName www.dominio2.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/www.dominio2.com

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

ubuntu@ubuntu:/etc/apache2/sites-available$ _
```



```

GNU nano 7.2                                001-www.dominio1.c
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
ServerName www.dominio1.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/www.dominio1.com

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>

```

b) Una vez creado su fichero de configuración, debemos proceder a habilitar el sitio. Para ello, utilizamos el comando `a2ensite <fichero-de-conf-del-sitio>`

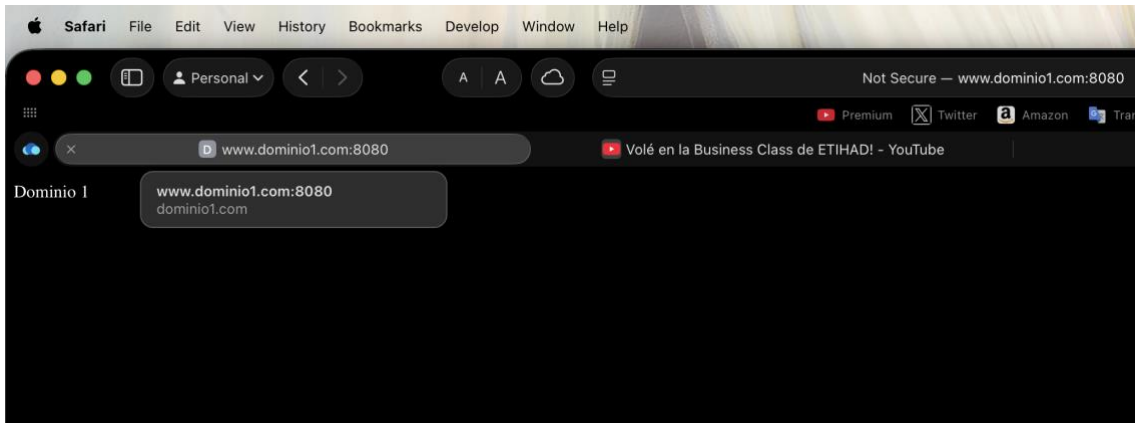
Comprueba que se ha añadido un enlace simbólico del nuevo servidor virtual a la carpeta `sites-enabled`. Reinicia el servicio de Apache.

```

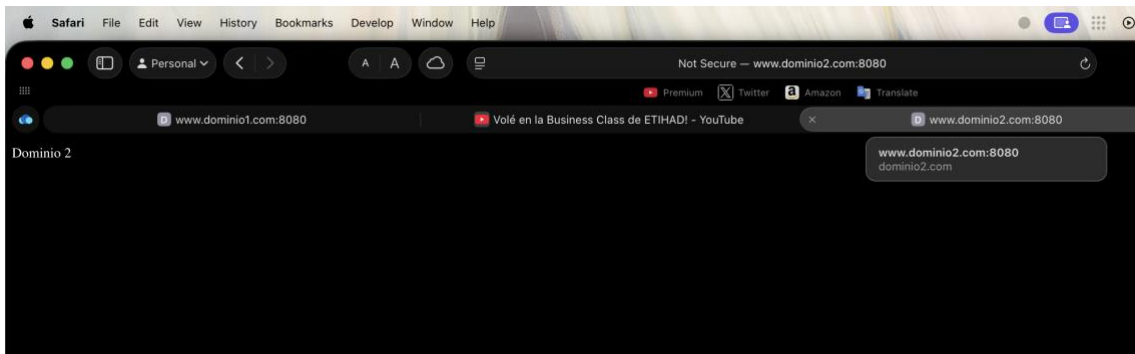
ubuntu@ubuntu:/etc/apache2/sites-available$ sudo a2ensite 00
000-default.conf          001-www.dominio1.com.conf  002-www.dominio2.com.conf
ubuntu@ubuntu:/etc/apache2/sites-available$ sudo a2ensite 001-www.dominio1.com.conf
Enabling site 001-www.dominio1.com.
To activate the new configuration, you need to run:
    systemctl reload apache2
ubuntu@ubuntu:/etc/apache2/sites-available$ sudo a2ensite 002-www.dominio2.com.conf
Enabling site 002-www.dominio2.com.
To activate the new configuration, you need to run:
    systemctl reload apache2
ubuntu@ubuntu:/etc/apache2/sites-available$ sudo sys
syscount-bpfcc             systemd-cgls               systemd-firstboot
syscount.bt                systemd-cgtop              systemd-hwdb
sysctl                     systemd-confext            systemd-id128
systemctl                  systemd-creds              systemd-inhibit
systemd                    systemd-cryptenroll        systemd-machine-id-setup
systemd-ac-power           systemd-cryptsetup         systemd-mount
systemd-analyze            systemd-delta               systemd-notify
systemd-ask-password       systemd-detect-virt        systemd-path
systemd-cat                 systemd-escape              systemd-repart
ubuntu@ubuntu:/etc/apache2/sites-available$ sudo systemctl reload ap
apache2.service  apparmor.service
ubuntu@ubuntu:/etc/apache2/sites-available$ sudo systemctl reload apache2
ubuntu@ubuntu:/etc/apache2/sites-available$

```

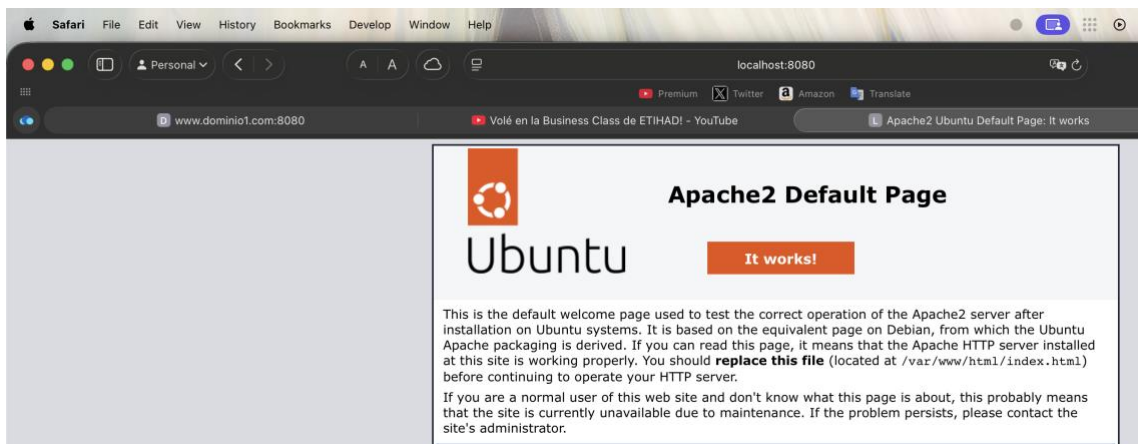
c) Comprueba que la URL <http://www.dominio1.com> sirve ahora la web con el mensaje de bienvenida al dominio 1.



d) Configura, habilita y comprueba el servidor virtual asociado a <http://www.dominio2.com>



f) Comprueba que en <http://localhost> se sigue sirviendo la web por defecto. Cuando la petición no va asociada a algún *ServerName* conocido, seguirá siendo atendida... ¿por qué servidor virtual?



g)

f) Deshabilita los dos sitios virtuales creados con el comando *a2dissite*

<archivo-de-conf-del-sitio>

```
ubuntu@ubuntu:/etc/apache2/sites-available$ sudo a2dissite 002-www.dominio2.com.conf
Site 002-www.dominio2.com disabled.
To activate the new configuration, you need to run:
    systemctl reload apache2
ubuntu@ubuntu:/etc/apache2/sites-available$ sudo a2dissite 001-www.dominio1.com.conf
Site 001-www.dominio1.com disabled.
To activate the new configuration, you need to run:
    systemctl reload apache2
ubuntu@ubuntu:/etc/apache2/sites-available$ sudo systemctl reload apache2
ubuntu@ubuntu:/etc/apache2/sites-available$ _
```

, y comprueba cómo desaparecen del directorio *sites-enabled*, pero permanecen en el directorio *sites-available*. Comprueba que las URLs <http://www.dominio1.com> y <http://www.dominio2.com> ya no son atendidas por nuestro servidor.

