

DESPLIEGUE DE APLICACIONES WEB
TÉCNICO EN DESARROLLO DE APLICACIONES WEB

Host virtuales

ÍNDICE

/ 1. Introducción y contextualización práctica	3
/ 2. Hosts virtuales	4
/ 3. Funcionamiento de los Virtual Hosts	4
/ 4. Caso práctico 1: “Instalación ISP Config”	6
/ 5. Virtual Host en Apache	6
/ 6. Virtual Block Nginx	8
/ 7. Redirección HTTPS usando Virtual Host	9
7.1. Redirección en Apache	9
7.2. Redirección en Nginx	10
/ 8. Caso práctico 2: “Denegar acceso carpeta virtualhost”	10
/ 9. Resumen y resolución del caso práctico de la unidad	11
/ 10. Bibliografía	12

OBJETIVOS

Conocer qué son los Host Virtuales

Crear y configurar sitios web con Virtual Hosts

Utilizar Virtual Host para alojar diferentes webs en un mismo servidor

/ 1. Introducción y contextualización práctica

En el tema anterior, aprendimos que Apache utiliza Virtual Host para definir la configuración de los diferentes sitios webs que aloja. En este tema conoceremos en profundidad qué son y para qué se utilizan.

Realizaremos diferentes configuraciones y pondremos en práctica todo lo que vayamos aprendiendo en esta unidad.

Planteamiento del caso práctico inicial

Escucha el siguiente audio donde planteamos la contextualización práctica de este tema, encontrarás su resolución en el apartado Resumen y Resolución del caso práctico.



Fig. 1. Administración de servidores web – VirtualHosts



Audio Intro. "DocumentRoot servidor web"

<https://bit.ly/3h2cXFm>





/ 2. Hosts virtuales

Los **hosts virtuales** o **Virtual Hosts** es una técnica o modalidad que consiste en simular distintas máquinas dentro de un mismo servidor. Esta característica nos brinda la posibilidad de utilizar un único servidor físico para alojar diferentes sitios webs (multisiting). De esta manera, optimizaremos costes y rentabilizaremos la inversión realizada, ya sea al alquilar un VPS Cloud (Servidor Virtual Privado) o adquirir un servidor físico destinado a realizar funciones de servidor web.

En entornos de producción es común tener un único servidor web que aloje diferentes sitios webs. Esta modalidad de negocio se conoce como **hosting** y nos permite dar servicio a tantos clientes como nuestro hardware sea capaz de gestionar.

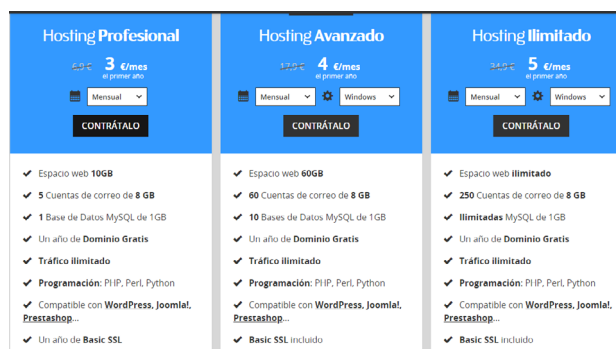


Fig. 2. Oferta Hosting en Arsysis

Para gestionar un hosting a nivel comercial existen aplicaciones que automatizan y facilitan la administración de Apache, MySQL o PHP; implementando además una capa de seguridad entre los diferentes clientes.

Entre los diferentes sistemas de gestión de hosting existentes, podemos destacar ISP Config y cPanel.

- **ISP Config** se define como un panel de control de código abierto con licencia BSD. Desde que apareció en 2005 se ha utilizado para implementar un servidor web multisitio.
- **cPanel** es un potente gestor de servidores web con una cómoda interfaz web. La primera versión se liberó en 1996 y requiere una suscripción mensual para su funcionamiento.



Fig. 3. Logo cPanel

/ 3. Funcionamiento de los Virtual Hosts

Dependiendo de su configuración, podemos encontrar dos tipos de Virtual Hosts:

- **Basados en IP:** el servidor tendrá varias direcciones IP y podrá asignar una a un sitio web.
- **Basados en nombre:** el servidor tendrá una única dirección IP y asignará a los sitios web nombres DNS para servir los recursos.

En este tema estudiaremos el caso de los Virtual Hosts basados en nombres DNS.



En la figura 4 podemos analizar el comienzo de un Virtual Host de Apache sin configurar. En este caso todas las peticiones que el servidor escuche por el puerto 80 se redirigirán a la carpeta “/var/www/html”

```
<VirtualHost *:80>

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/
```

Fig. 4 Ejemplo de Virtual Host con la configuración por defecto

Por el contrario, en el siguiente ejemplo el servidor diferenciará las peticiones que reciba, prestando atención al nombre DNS; y únicamente devolverá la información si el host destino es “ejemplo1.local”

```
<VirtualHost ejemplo1.local:80>

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/ejemplo1
```

Fig. 5 Ejemplo de virtual Host configurado para atender peticiones enviadas a ejemplo1.local

De esta manera, al especificar un nombre DNS podremos alojar varios sitios webs. Es importante remarcar que para que el servidor pueda atender peticiones a través de Internet por nombre, en la zona DNS del dominio correspondiente, el registro A debe apuntar a la IP del servidor. En próximos temas hablaremos en profundidad de los diferentes registros DNS que existen y las diferentes configuraciones que podemos realizar.

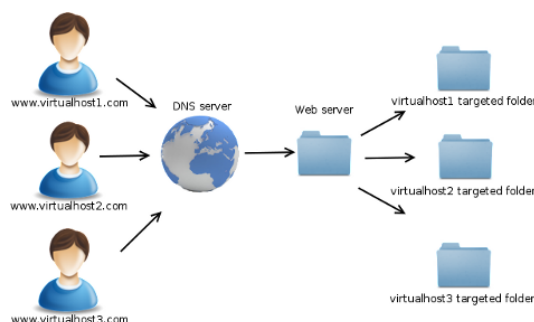


Fig. 6. Funcionamiento virtualHost



Audio 1. “El fichero Host”
<https://bit.ly/3929tAa>





/ 4. Caso práctico 1: “Instalación ISP Config”

Planteamiento: Luis trabaja en una pequeña empresa que se dedica a desarrollar webs para pymes y autónomos. Quiere dar el salto y empezar a vender hosting web por lo que están buscando un sistema de gestión que les permita poner en marcha esta idea.

Luis ha investigado acerca de ISP Config en internet y quiere probarlo.

Nudo: ¿Qué pasos debe seguir? ¿Dónde puede encontrar documentación fiable para ayudarse en este proceso?

Desenlace: Investigando en la web oficial de ISP Config ha localizado un enlace a un instalador automatizado en Ubuntu, además de toda la documentación relativa a instalación, troubleshooting y configuraciones:

<https://www.ispconfig.org/blog/automated-ispconfig-3-installation-on-debian-and-ubuntu/>

Se dispone por tanto a seguir las instrucciones del instalador y ejecuta las siguientes líneas con permisos elevados de root:

```
apt-get update && apt-get upgrade  
apt-get install unzip  
cd /tmp  
wget --no-check-certificate  
https://github.com/servisys/ispconfig_setup/archive/master.zip  
unzip master.zip  
cd ispconfig_setup-master/  
./install.sh
```

Fig. 7. Instalación ISP Config – extraído de <https://www.howtoforge.com/tutorial/ispconfig-install-script-debian/>

Durante la instalación, Luis tendrá que configurar las contraseñas de la base de datos y de PHPMyadmin.

Cuando el script termine de instalar, el ISP quedará listo para realizar todas las pruebas que Luis considere, pudiendo acceder al mismo a través de un navegador web por el puerto 8080.

/ 5. Virtual Host en Apache

El siguiente procedimiento describe los **procesos básicos necesarios para alojar dos webs independientes en un mismo servidor Apache utilizando Virtual Hosts**.

Como vimos en puntos anteriores, para poder acceder a las webs por el nombre DNS será necesario modificar el archivo ‘hosts’ de nuestro equipo añadiendo una línea con la IP y el FQDN (Fully Qualified Domain Name; o sea, la dirección completa y única que nos identifica en internet) al que queremos referenciar.



```
# For example:
#      102.54.94.97      rhino.acme.com      # source server
IP_SERVERDOR      web1.local|      www.web1.local
IP_SERVERDOR      web2.local      www.web2.local
```

Fig. 8. Ejemplo fichero Hosts

Lo primero que haremos será crear las carpetas de los diferentes sitios en la carpeta pública de Apache, y asignar la propiedad de éstas al usuario de www-data (**El usuario que utilizarán los servidores web para acceder a los archivos**). Para comprobar que los Virtual Hosts funcionan y poder diferenciarlos crearemos un archivo HTML con algún contenido.

```
usuario@lnxsrv-web-01:~$ sudo mkdir /var/www/web1
[sudo] password for usuario:
usuario@lnxsrv-web-01:~$ sudo mkdir /var/www/web2
usuario@lnxsrv-web-01:~$ sudo chown www-data:www-data /var/www/web1/
usuario@lnxsrv-web-01:~$ sudo chown www-data:www-data /var/www/web2
usuario@lnxsrv-web-01:~$ sudo nano /var/www/web1/index.html
usuario@lnxsrv-web-01:~$ sudo nano /var/www/web2/index.html
```

Fig. 9. Configuración virtual Host multisite 1

A continuación, creamos un **nuevo archivo de configuración en la carpeta** “sites-available” de Apache definiendo dos Virtual Host dentro del mismo fichero, aunque sería más correcto realizar la configuración de cada sitio en archivos independientes.

Tendremos que especificar el ServerName de cada web, así como el documentRoot correspondiente.

Una vez creado el fichero de configuración, **activamos el sitio con el comando a2ensite y reiniciamos Apache**.

```
usuario@lnxsrv-web-01:~$ cat /etc/apache2/sites-available/multisitio.conf
<VirtualHost *:80>
    ServerName www.web1.local

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/web1

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
<VirtualHost *:80>
    ServerName www.web2.local

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/web2

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

usuario@lnxsrv-web-01:~$ sudo a2ensite multisitio.conf
Enabling site multisitio.
To activate the new configuration, you need to run:
    systemctl reload apache2
usuario@lnxsrv-web-01:~$ sudo apache2ctl restart
```

Fig. 10. Configuración de VirtualHost preparado para alojar dos sitios webs

Para finalizar, comprobaremos el correcto funcionamiento de los dos sitios accediendo a través de un navegador web.



Fig. 11. Comprobación del funcionamiento de los virtual Host

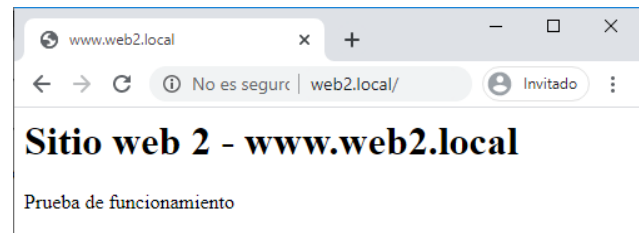


Fig. 12. Comprobación del funcionamiento de los virtual Host



/ 6. Virtual Block Nginx

Nginx cuenta con un sistema similar a los Virtual Hosts para hacer multisiting. No obstante, Virtual Hosts se considera un término propio de Apache. La nomenclatura correcta para definir los hosts virtuales en Nginx sería Virtual Block.

El procedimiento para crear Server Blocks y alojar varios sitios webs es el siguiente:

Al igual que en Apache, empezaremos creando las carpetas, posteriormente realizaremos la asignación de permisos al usuario “www-data,” y como anteriormente, crearemos los “index.html”.

```
sudo mkdir -p /var/www/nginx1.com/html
sudo mkdir -p /var/www/nginx2.com/html

sudo chown www-data:www-data /var/www/nginx1.com/ /var/www/nginx2.com/
sudo chmod 755 -R /var/www
sudo nano /var/www/nginx1.com/html/index.html
```

Fig. 13. Creando Server Blocks

A continuación, creamos los archivos Server Block en la carpeta “sites-availables” y los editaremos prestando especial atención en eliminar default_server en las líneas “listen...”.

Si no se hiciera esto, al validar la configuración Nginx, mostrará un error indicando que el default server está duplicado.

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;
```

Fig. 14. Default server



```
server {
    listen 80;
    listen [::]:80;

    root /var/www/nginx1.com/html;
    index index.html index.htm index.nginx-debian.html;

    server_name nginx1.com www.nginx1.com;

    location / {
        try_files $uri $uri/ =404;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php7.0-fpm.sock;
    }
}
```

Fig. 15. Server

Por último, activaremos los sitios realizando un enlace simbólico a la carpeta sites-enabled y reiniciaremos Nginx para activar los cambios.

```
sudo ln -s /etc/nginx/sites-available/nginx1.com /etc/nginx/sites-enabled/.
sudo ln -s /etc/nginx/sites-available/nginx2.com /etc/nginx/sites-enabled/.
```

Fig. 16. Enlace a sites-enabled

Encontramos el manual completo en el siguiente enlace:

<https://www.digitalocean.com/community/tutorials/how-to-set-up-nginx-server-blocks-virtual-hosts-on-ubuntu-16-04>

/ 7. Redirección HTTPS usando Virtual Host

Utilizando los hosts virtuales podemos reenviar una petición de un cliente a otro sitio web o forzar el uso de HTTPS. Esto es especialmente útil cuando hacemos un cambio de dominio, queremos hacer una landing page (página de bienvenida), o simplemente queremos reforzar la seguridad de nuestro sitio web evitando el uso de conexiones inseguras.

Existen **dos tipos de redirecciones**:

- **Permanentes (301)**: se utilizan cuando el cambio es definitivo. Se trata de la opción más habitual y la reputación SEO del dominio se traspasará en la redirección.
- **Temporales (302)**: se utilizan en tareas de mantenimiento. El posicionamiento no se verá afectado ni se traspasará al nuevo dominio.

7.1. Redirección en Apache

En Apache se puede realizar una **redirección a HTTPS** de varias maneras, pero la más común es **a través del módulo mod_rewrite**, que nos permite establecer y evaluar una serie de condiciones para que se produzca la redirección. La activación de dicho módulo se realiza con el comando "a2enmod rewrite"



Si quisiéramos forzar el uso de HTTPS tendríamos que añadir el siguiente **código en el Virtual Host** que escuche por el puerto 80. De esta manera todas las peticiones se trasladarán automáticamente al puerto 443.

```
RewriteEngine On  
  
RewriteCond %{HTTPS} !=on  
  
RewriteRule ^/?(.*) https://%{SERVER_NAME}/$1 [R,L]
```

Fig. 17. Rewrite HTTP to HTTPS extraído de <https://wiki.apache.org/confluence/display/HTTPD/RewriteHTTPToHTTPS>

Encontramos más información del uso del módulo rewrite en el siguiente enlace:

https://httpd.apache.org/docs/current/mod/mod_rewrite.html

7.2. Redirección en Nginx

Para **realizar redirecciones en Nginx** solo tenemos que añadir en el Virtual Block correspondiente las líneas indicando la redirección:

```
server_name www.nginx1.com;  
  
rewrite ^/(.*)$ http://www.nginx2.com/$1 permanent;
```

Fig. 18. Redirección de dominio en Nginx

Escaneando el siguiente QR podrás **ampliar información** sobre el uso de redirecciones en Nginx



Vídeo 2. "Redirecciones Nginx"
<https://bit.ly/32jviK0>



/ 8. Caso práctico 2: "Denegar acceso carpeta virtualhost"

Planteamiento: Isabel es responsable del mantenimiento de un repositorio web de documentación en la empresa donde trabaja. En ella, tienen configuradas diferentes subredes para cada departamento. Desde Dirección le han solicitado que deniegue el acceso a la carpeta "Laboral2" de cualquier IP que proceda del departamento de mantenimiento (172.27.15.0/24).

Nudo: ¿Cómo debería hacerlo?

Desenlace: Entre las diferentes opciones que tiene, Isabel opta por limitar el acceso a la carpeta laboral2 utilizando el archivo Virtual Host del sitio web.



Si la estructura de carpetas es:

REPOSITORIO

|__CARPETA1

|__CARPETA2

|__LABORAL1

|__LABORAL2

Modificaremos el archivo correspondiente en la carpeta /etc/apache2/sites-available para añadir las siguientes líneas al Virtual Host :

```
<Directory /var/www/REPOSITORIO/LABORAL2/>  
  
Order allow,deny  
  
Allow from ALL  
  
Deny from 172.27.15.0/24  
  
</Directory>
```

Fig. 19. Información ampliada de redirecciones en Nginx

De esta manera ya estaremos bloqueando el acceso al rango 172.27.15.0/24 y permitiendo el resto de IP's de la empresa.

Al intentar acceder desde el rango denegado obtendremos un error de Apache indicando que no tenemos permisos para acceder a este recurso:

Forbidden

You don't have permission to access /laboral2 on this server.

Fig. 20. Error 403 de Apache

/ 9. Resumen y resolución del caso práctico de la unidad

En esta unidad hemos aprendido qué son los **Virtual Hosts**, su **funcionamiento teórico** y cómo **utilizarlos para poder dar servicio a varios clientes** empleando un único servidor web.

Hemos comentado también la utilidad de los hostings que, basándose en los fundamentos del Virtual Host, **comercializan un producto**.

Además, se han realizado **diversos supuestos prácticos** de implementación de Virtual Hosts tanto en Apache como en Nginx, para aplicar la teoría y conseguir hacer multisiting.

Por último, hemos hablado de las **diferentes funciones de redirección y seguridad** que podemos establecer en los Virtual Hosts.



Resolución del caso práctico de la unidad

En el caso práctico inicial Marín tenía el siguiente problema: al solicitar una web al servidor, éste no le devolvía la ruta en la que está guardada la aplicación web. En su lugar se cargaba, por defecto, el **documentRoot**.

Para solucionar el problema Martín tendría que modificar el Virtual Host correspondiente indicando al servidor web el **documentRoot** correcto.

Una vez que aplicamos los cambios, reiniciando el servicio Apache, se abrirá correctamente la aplicación web que hemos desarrollado.

/ 10. Bibliografía

«Virtual Hosting» en Apache. (s. f.). Recuperado de <https://www.osmosislatina.com/apache/vhosting.htm>

ISPConfig Hosting Control Panel. (s. f.). Recuperado de <https://www.ispconfig.org/>

colaboradores de Wikipedia. (s. f.). CPanel. Recuperado de <https://es.wikipedia.org/wiki/CPanel>

Papiernik, M. (s. f.). How To Create Temporary and Permanent Redirects with Nginx. Recuperado de <https://www.digitalocean.com/community/tutorials/how-to-create-temporary-and-permanent-redirects-with-nginx>

How To Set Up Nginx Server Blocks (Virtual Hosts) on Ubuntu 16.04 | DigitalOcean. (s. f.). Recuperado de <https://www.digitalocean.com/community/tutorials/how-to-set-up-nginx-server-blocks-virtual-hosts-on-ubuntu-16-04>