

SERVIDORES VIRTUALES BASADOS EN NOMBRE

<VIRTUALHOST>

APACHE 2.4

Apache2 tiene la funcionalidad de trabajar con lo que se denomina servidores virtuales o virtual host. Esto permite tener varios dominios alojados en una misma máquina (por ejemplo, www.aulasmr.org, www.aulaasir.org). Se pueden diferenciar tres tipos de alojamiento virtual:

- **basado en IP's:** el servidor tendrá diferentes direcciones IP por cada servidor web adicional. Para hacer esto, el servidor necesita varias tarjetas de red o asignar varias IP's a una misma tarjeta de red.
- **basado en nombres:** en el servidor alojamos varios nombres de dominio sobre una misma IP. Cada servidor virtual atiende las peticiones de un nombre de dominio.
- **basado en puertos:** cada servidor virtual atiende peticiones en una dirección IP y/o dominio:puerto diferentes.
- **combinaciones:** en un mismo servidor web podemos combinar todas las opciones anteriores.

A la hora de crear un servidor virtual tenemos que hacer tres pasos, no necesariamente en este orden:

- Crear la carpeta del proyecto web y su contenido
- Definir y configurar el servidor virtual o <VirtualHost>
- Asegurarse de que los clientes pueden traducir el nombre de dominio para llegar a la IP del servidor.

OJO, no basta solamente en configurar el servidor web, también debemos de hacer que los clientes web sepan traducir correctamente un nombre de dominio. Para ello, tenemos dos opciones:

OPCION A: Usar un DNS para hacer las traducciones

OPCION B: modificar el archivo estático de traducciones en casa cliente

Windows —> C:\Windows\System32\drivers\etc\hosts (abrir como admin)
Linux —> /etc/hosts

CONFIGURACION COMUN (LINUX Y WINDOWS)

Para definir un nuevo servidor virtual o site, utilizaremos las etiquetas y directivas:

```
<VirtualHost IP:PUERTO>

    ServerName dominio-de-la-web
    DocumentRoot PATH-del-directorio-raiz

    <Directory PATH-del-directorio-raiz>
        directivas a nivel de carpeta
        Require all granted
    </Directory>

    otras directivas y/o etiquetas de control del site

</VirtualHost>
```

- IP: es la IP del servidor donde esta alojada la página. **Se suele poner * para indicar que acepta cualquier IP.** Quizás lo más correcto sería poner de una forma fija la IP del servidor, pero para prevenir el caso de servidores con IP dinámica, se suele dejar el *.
- Puerto: es el puerto por donde va a escuchar mi servidor para ese site. Lo normal es usar el puerto 80, pero podemos usar otros puertos por conveniencia para aumentar la seguridad o por otras razones, como el uso del puerto 8080.
- ServerName: nombre de dominio al que va atender este host virtual.
- DocumentRoot: directorio donde estará el index de mi site
- <Directory>: usaremos esta directiva para especificar comportamiento de la carpeta. No es estrictamente obligatorio.
- Require all granted: para dar permiso de acceso a la carpeta. En Windows es necesario si se monta algo fuera de *htdocs* y en Linux si se monta algo fuera de */var/www*, que son los directorios por defecto en ambos sistemas.

```
<VirtualHost *:80>
    ServerName www.miweb.com
    DocumentRoot "C:/www/miweb.com/"
    <Directory "C:/www/miweb.com">
        Options -Indexes
        Require all granted
    </Directory>
</VirtualHost>
```

CONFIGURACION EN WINDOWS

El archivo de configuración específico (aunque realmente lo podríamos hacer en el general) es `apache/conf/extra/httpd-vhosts.conf`

En este archivo aparecen un par de ejemplos comentados. La directiva `NameVirtualHost` esta en desuso, y en Apache2.4 ya no hay que usarla.

Por otro lado, en Windows, cuando definimos algún `<VirtualHost>` perdemos el acceso a `localhost`, por lo que es interesante definir un `<VirtualHost>` para `localhost`.

```
# Este VirtualHost lo hacemos para no perder el acceso a localhost
<VirtualHost *:80>
    ServerName localhost
    DocumentRoot "C:/xampp/htdocs/"
</VirtualHost>
```

Un ejemplo de un `<VirtualHost>` definido en dicho archivo:

```
35
36 ##<VirtualHost *:80>
37     ##ServerAdmin webmaster@dummy-host2.example.com
38     ##DocumentRoot "C:/xampp/htdocs/dummy-host2.example.com"
39     ##ServerName dummy-host2.example.com
40     ##ErrorLog "logs/dummy-host2.example.com-error.log"
41     ##CustomLog "logs/dummy-host2.example.com-access.log" common
42 ##</VirtualHost>
43
44 <VirtualHost *:80>
45     DocumentRoot "C:/xampp/htdocs"
46 </VirtualHost>
47
48 <VirtualHost externalapp.local:80>
49     DocumentRoot "E:/LaravelApps/my_project_name/public"
50     ServerName externalapp.local
51     <Directory "E:/LaravelApps">
52         .....
53         Require all granted
54     </Directory>
55 </VirtualHost>
```

CONFIGURACION EN LINUX (Debian / Ubuntu / derivados)

En Linux Debian el directorio principal de apache una vez instalado es **/etc/apache2**. Los principales usuarios, directorios y archivos implicados en la instalación son:

Principales elementos

- ➡ Directorio **/var/www**: su propietario es el root y su grupo es el root. Es el directorio raíz por defecto del servidor donde se meterán todas los website que haya en el servidor. Es decir, es el directorio donde se montarán o subirán todas las webs.
- ➡ Directorio **/etc/apache2/sites-available** : directorio donde se van a definir por medio de archivos todos los VirtualHost. Cada site/web debería ser un archivo diferente. Estos archivos deben de terminar con la extensión **.conf**. En este directorio, hay definidos un par de archivos por defecto que nos sirven de esqueleto para nuestros VirtualHost. Dicho archivo se llama *000-default.conf*
- ➡ Directorio **/etc/apache2/sites-enabled** : en este directorio aparecen todos los sites que tenemos activados y por tanto funcionando. Yo puedo tener varios sites definido pero solo activados algunos de ellos. La activación se hace con el comando **a2ensite**
- ➡ Archivo **/etc/apache2/ports.conf**: se definen los puertos de escucha del servidor. Por defecto ya está definido el puerto 80.
- ➡ Archivo **/etc/apache2/apache2.conf**: archivo de configuración principal de Apache.

Existen otras carpetas y elementos de configuración, pero no son necesarios para hacer una web básica

Paso 1. Crear carpeta del site y crear/subir la web

A partir de **/var/www** vamos a crearnos las carpetas donde alojaremos nuestro proyecto. A la carpeta le daremos el nombre de la web, es una buena forma de diferenciar las carpetas. Opcionalmente, y para que quede más organizado, podemos crear otra subcarpeta llamada **public_html**. Es aquí dentro donde colocaremos nuestro **index.html**.

Así, si mi web se va a llamar **www.jacuela.com**, podemos crearnos la ruta

/var/www/www.jacuela.com/public_html

y dentro de aquí colocar el **index.html**. También podemos subir todo mi proyecto y descomprimirlo, que será lo más normal, pues el proyecto se suele programar en local.

Un aspecto típico de como puede quedar este directorio sería el que puedes ver a la derecha.

```
/var/www/  
├── domain1.com  
│   └── public_html  
├── domain2.com  
│   └── public_html  
├── domain3.com  
│   └── public_html
```

Paso 2. Definir el <VirtualHost>

Al igual que hacíamos en Windows, debemos definir nuestros sites usando la etiqueta <VirtualHost>. Pero a diferencia de windows, cada site se definirá en un archivo diferente.

Los archivos se deben definir en el directorio *sites-available* visto anteriormente y deben de terminar en la extensión **.conf**. Por ejemplo, para crear el archivo correspondiente a la web **www.jacuela.com**, nos crearíamos el archivo **www.jacuela.com.conf**

El nombre del archivo no es relevante, pero de esta forma podemos identificar claramente a que web hace referencia cada archivo. En lugar de crear el archivo desde cero (que podríamos), haremos una copia del básico llamado 000-default.conf

El comando usado (estando ya dentro del directorio sites-available) sería:

```
#cp 000-default.conf www.jacuela.com.conf
```

y una vez creado, lo editaremos con el comando nano

```
#nano www.jacuela.com.conf
```

Un posible ejemplo sería el siguiente (esto es común con Windows)

```
/etc/apache2/sites-available/example.com.conf
```

```
<VirtualHost *:80>
    ServerName example.com
    ServerAlias www.example.com
    ServerAdmin webmaster@example.com
    DocumentRoot /var/www/example.com/public_html

    <Directory /var/www/example.com/public_html>
        Options -Indexes +FollowSymLinks
        AllowOverride All
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/example.com-error.log
    CustomLog ${APACHE_LOG_DIR}/example.com-access.log combined
</VirtualHost>
```

Paso 3. Activar el <VirtualHost>

Una vez creado y definido cada archivo correspondiente a cada VirtualHost, debemos de activarlo con el comando a2ensite. (solo lo puede hacer el usuario root)

```
#a2ensite www.jacuela.com
```

Esto nos creara otro archivo en el directorio sites-enabled. Es aquí donde estarán todos los sites activados.

Si quisiéramos, por ejemplo, porque no queremos que una web esté disponible temporalmente, podemos desactivar un site con el comando:

```
#a2dissite www.jacuela.com
```

Paso 4. Probar la web

Ya solo nos quedaría abrir un navegador y acceder a la web

<http://www.jacuela.com>

Para que funcione, hay que conseguir que dicho nombre se traduzca en la IP del servidor Apache donde esté alojada dicha web. Esto se hace configurando el DNS, pero esto es algo totalmente ajeno al servidor Apache.