



# CONFIGURACIÓN DE HOSTS VIRTUALES EN APACHE

Fecha de realización: 14 - 12 - 22

## Introducción

**Apache** es el servidor web más popular para blindar contenido en internet. Cuenta con más de un 50% de los sitios activos en internet y es extremadamente poderoso y flexible. Apache separa su funcionalidades y componentes en unidades distintas que pueden ser personalizadas y configuradas de manera independiente. La unidad básica que describe a un sitio o dominio es denominada **"Virtual host"**. Esto permite al administrador hacer uso de un servidor para alojar múltiples dominios en una única IP utilizando un mecanismo de coincidencias.

Cada dominio configurado envía al visitante a un **directorio** previamente configurado, manteniendo la información del mismo, a simple vista nunca indicará que el mismo servidor es también responsable de otros sitios.

En esta guía voy a configurar un Virtual Host en apache, en un Ubuntu Server 18.04.6 LTS (Bionic Beaver) y voy a hacerlo paso a paso para su correcta comprensión.

<https://releases.ubuntu.com/18.04/>



## Pre-requisitos

Yo he creado un usuario root llamado ubu18, y necesitarás instalar Apache, que lo podrás hacer de la siguiente manera:

```
Sudo apt update && sudo apt install apache2
```

Por último crearemos dos archivos, `pagina1.com` | `pagina2.com`

Ahora dividiremos este tutorial en los siguientes pasos:

- 1 — Crear la Estructura del Directorio
- 2 — Crear Páginas de Prueba para cada Virtual Host
- 3 — Crea un Nuevo Archivo para su Virtual Host
- 4 — Habilitar los Nuevos Archivos Virtual Host
- 5 — Configure su Archivo Hosts Local (Opcional)
- 6 — Pruebe sus Resultados



## 1 — Crear la Estructura del Directorio

---

Lo primero será crear una estructura de directorios que alojará los datos del sitio los datos del sitio que vamos a proporcionar a nuestros visitantes.

Haremos un `documento root` (apache lo interpretará como el directorio más alto y en él buscará contenido para mostrar), será configurado en directorios individuales en el directorio `/var/www`.

Dentro de cada uno de estos directorios, vamos a crear una carpeta, la llamaré `"public_html"`, la cual mantendrá los archivos. No es necesario, pero así aportaremos más flexibilidad a nuestro hosting.

Para nuestros sitios vamos a crear los directorios de la siguiente manera:

```
root@ubu18:/home/ubu18# mkdir -p /var/www/pagina1.com/public_html
root@ubu18:/home/ubu18# mkdir -p /var/www/pagina2.com/public_html
root@ubu18:/home/ubu18# _
```

## 2 — CREAR PÁGINAS DE PRUEBA PARA CADA VIRTUAL HOST

---

Ahora vamos a crear una estructura y vamos a realizar una demostración, con una página `index.html` para cada sitio. Vamos a empezar con `pagina1.com`. Vamos a abrir un archivo `index.html` en un editor de texto, yo usaré `vim`.



```
ubu18@ubu18:~$ vim /var/www/pagina1.com/public_html/index.html_
```

Ahora vamos a crear la estructura, y lo haremos de la siguiente manera, utilizando el lenguaje de marcado HTML:

```
<html>
  <head>
    <title>¡¡Esto es la pagina1.com!!</title>
  </head>
  <body>
    <h1>El virtual host pagina1.com está funcionando bien </h1>
  </body>
</html>
```

Ahora guardamos y cerramos el archivo.

Ahora vamos a copiar el archivo y lo usaremos de base para nuestros sitios escribiendo lo siguiente en nuestra terminal:

```
root@ubu18:/home/ubu18# cp /var/www/pagina1.com/public_html/index.html /var/www/pagina2.com/public_html/index.html_
```

Ahora vamos a configurar el archivo que hemos creado anteriormente y vamos a modificarlo para que sea pagina2.com.

```
root@ubu18:/home/ubu18# vim /var/www/pagina2.com/public_html/index.html
```

```
<html>
  <head>
    <title>¡¡Bienvenido a Pagina2.com!!</title>
  </head>
  <body>
    <h1>¡Bien!, el virtual host pagina2.com funciona perfectamente</h1>
  </body>
</html>
```



Ahora vamos a guardar y cerrar el archivo. Ahora tenemos las páginas necesarias para probar la configuración del virtual host.

### **3 — CREAR NUEVO ARCHIVO PARA NUESTRO VIRTUAL HOST**

---

Los archivos virtual host son archivos que especifican la configuración actual de un virtual host e indican cómo el servidor Apache va a responder a varias solicitudes de dominio.

Apache trae incluido un archivo virtual host por defecto que es nombrado como `000-default.conf` que vamos a utilizar como plantilla. Vamos a copiarlo para crear los archivos virtual host de nuestros sitios.

Estos archivos **SIEMPRE** tienen que acabar con la extensión `.conf`, ya que es un archivo de configuración. Los crearemos en la carpeta `sites-available` y a continuación activaremos creando su enlace en `sites-enable`, usando el comando de apache `a2ensite`.

Ahora vamos a copiar el archivo para usarlo con el primero dominio, usaremos el siguiente comando:

```
root@ubui18:/home/ubui18# cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/pagina1.com.conf
```

Vamos a abrir el archivo con un editor de texto con permisos root:



```
root@ubu18:/home/ubu18# nano /etc/apache2/sites-available/pagina1.com.conf _
```

Nuestro archivo se verá algo como así, le he quitado los comentarios para hacer más legible:

```
<VirtualHost *:80>

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```

Ahora vamos a modificar los parámetros previamente escritos en el archivo anterior. Primero vamos a cambiar la directiva `ServerAdmin` a un correo electrónico en donde el administrador del sitio pueda recibir correos. Después agregamos dos directivas. La primera la llamaremos `ServerName`, la cual establece el dominio base que debe coincidir para la definición de este host virtual. La segunda la llamaremos `ServerAlias`, definimos nombres alternativos, esto será para poder ser encontrado como alternativa al dominio base. Esto es útil para definir dominios alternativos, como `www`. Por último solo tenemos que cambiar la dirección del documento raíz para este dominio. Ya tenemos el directorio creado, por lo que alteramos la dirección del director con el parámetro `DocumentRoot` para que refleje el directorio que hemos creado. El archivo nos quedará de la siguiente manera: