

# APACHE SERVER

## VIRTUAL HOSTS

Salvador Mira

IES Pere Maria Orts



# VIRTUAL HOSTS

Posibilidad de atender desde un único servidor a más de un dominio: **Virtual Hosts**

Para conseguirlo necesitamos que la configuración sea coherente también en DNS

- Varios **nombres de dominio**
- Varios **nombres de máquina** apuntando a la **dirección** o las **direcciones IP** del servidor en el que se ejecuta Apache

También se puede realizar la configuración de **hosts virtuales por IP**

- El servidor web puede tener **varias direcciones IP**, asignando a cada virtual host una de ellas

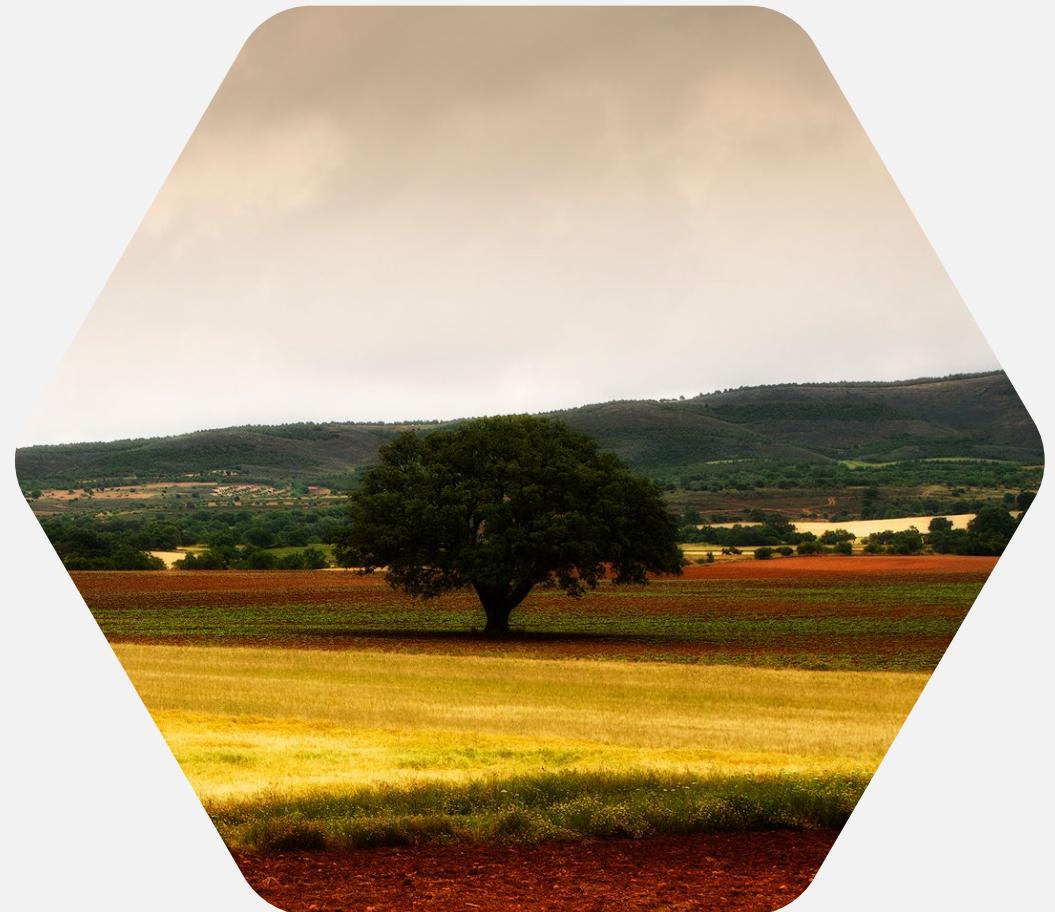


# VIRTUAL HOSTS

Es posible también realizar configurar **Apache** para que escuche en **puertos diferentes**

- Asignar las peticiones a hosts virtuales en función del puerto en el que llegues dichas peticiones

La combinación de todas las posibilidades anteriores puede servir para definir **hosts virtuales**



# Virtual Hosts por IP



# VIRTUAL HOSTS

## CONFIGURACIÓN POR DIRECCIÓN IP

El prerequisito para ello es que el **servidor web** disponga de **diversas direcciones IP**

Para ello se cuenta con varias posibilidades (y la combinación de todas ellas). Algunas de ellas son:

- El host tiene **distintas tarjetas de red** (separación física de dominios de broadcast)
- Se definen distintas **subinterfaces por VLAN** en una única interfaz física (separación lógica de dominios de broadcast)
- Se añaden **interfaces virtuales** sobre las interfaces físicas (mismo dominio de broadcast)

Para realizar el ejemplo, emplearemos la tercera posibilidad, por su mayor sencillez

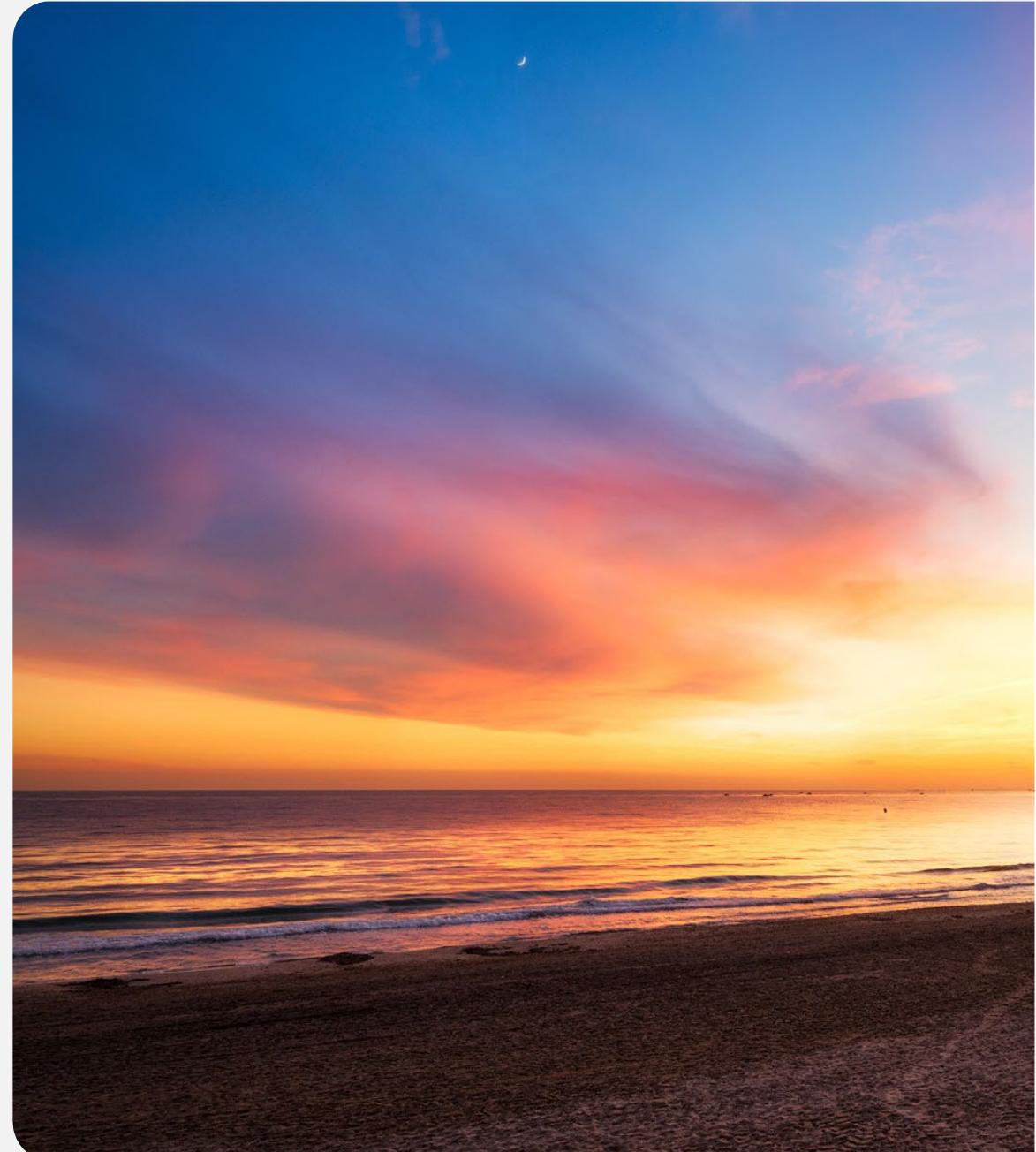


# VIRTUAL HOSTS

## CONFIGURACIÓN POR DIRECCIÓN IP

En Debian 10, la configuración de las interfaces de red se encuentra en el archivo `/etc/network/interfaces`

```
source /etc/network/interfaces.d/*  
  
# The loopback network interface  
auto lo enp0s3  
iface lo inet loopback  
  
# The primary network interface  
allow-hotplug enp0s3  
iface enp0s3 inet static  
    address 192.168.1.170  
    netmask 255.255.255.0  
    network 192.168.1.0  
    gateway 192.168.1.1
```



# VIRTUAL HOSTS

## CONFIGURACIÓN POR DIRECCIÓN IP

```
source /etc/network/interfaces.d/*
# The loopback network interface
auto lo enp0s3
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
    address 192.168.1.170
    netmask 255.255.255.0
    network 192.168.1.0
    gateway 192.168.1.1
```

- En el sistema del ejemplo, su adaptador de red se denomina **enp0s3**
- Su dirección IP es la **192.168.1.170/24**
- Su puerta de enlace es la **192.168.1.1**
- La línea **auto lo enp0s3** hace que las interfaces **lo** y **enp0s3** se habiliten con el arranque del sistema

# VIRTUAL HOSTS

## CONFIGURACIÓN POR DIRECCIÓN IP

Podemos crear adaptadores virtuales asociados a un adaptador principal y darles configuración IP

Para crear un adaptador virtual debemos darle por nombre el mismo que el del adaptador principal, al cual le agregamos un índice

```
# Interfaz virtual
auto enp0s3:0
iface enp0s3:0 inet static
    address 192.168.1.175
    netmask 255.255.255.0
    network 192.168.1.0
    gateway 192.168.1.1
```



# VIRTUAL HOSTS

## CONFIGURACIÓN POR DIRECCIÓN IP

Al reiniciar la red vemos que el adaptador dispone de una nueva dirección IP además de la IP principal

```
administrador@debian-despliegue:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:1d:6d:32 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.170/24 brd 192.168.1.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet 192.168.1.175/24 brd 192.168.1.255 scope global secondary enp0s3:0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe1d:6d32/64 scope link
        valid_lft forever preferred_lft forever
```

# VIRTUAL HOSTS

## CONFIGURACIÓN POR DIRECCIÓN IP

Tras la creación de la interfaz virtual, debemos ser capaces de hacer ping a su IP

```
PS C:\Users\Salva Mira> ping 192.168.1.175
```

Haciendo ping a 192.168.1.175 con 32 bytes de datos:

Respuesta desde 192.168.1.175: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 192.168.1.175:

Paquetes: enviados = 4, recibidos = 4, perdidos = 0  
(0% perdidos),

Tiempos aproximados de ida y vuelta en milisegundos:

Mínimo = 0ms, Máximo = 0ms, Media = 0ms

# VIRTUAL HOSTS

## CONFIGURACIÓN POR DIRECCIÓN IP

Una vez disponemos de múltiples direcciones IP  
hay que definir el virtual host

- Creamos un nuevo archivo de virtual host en **/etc/apache2/sites-available**
- Al archivo de virtual host podemos darle cualquier nombre, pero su extensión debe ser **.conf**
- Una buena práctica puede consistir en que el nombre del archivo sea la dirección IP del virtual host
  - Para el ejemplo el nombre de archivo será **192.168.1.175.conf**



# VIRTUAL HOSTS

## CONFIGURACIÓN POR DIRECCIÓN IP

Las únicas declaraciones de directivas necesarias en este archivo son:

```
<VirtualHost IP_del_host_virtual>
    ServerName "direccion_ip_u_otro_nombre"
    DocumentRoot "/directorio/principal/recursos"
</VirtualHost>
```

```
<VirtualHost 192.168.1.175>
    ServerName "192.168.1.175"
    DocumentRoot "/var/www/vhip"
</VirtualHost>
```

# VIRTUAL HOSTS

## CONFIGURACIÓN POR DIRECCIÓN IP

Tras su definición, habrá que enlazarlo en **/etc/apache2/sites-enabled** mediante la utilidad **a2ensite**

Además habrá que reiniciar Apache para que la configuración tenga efecto

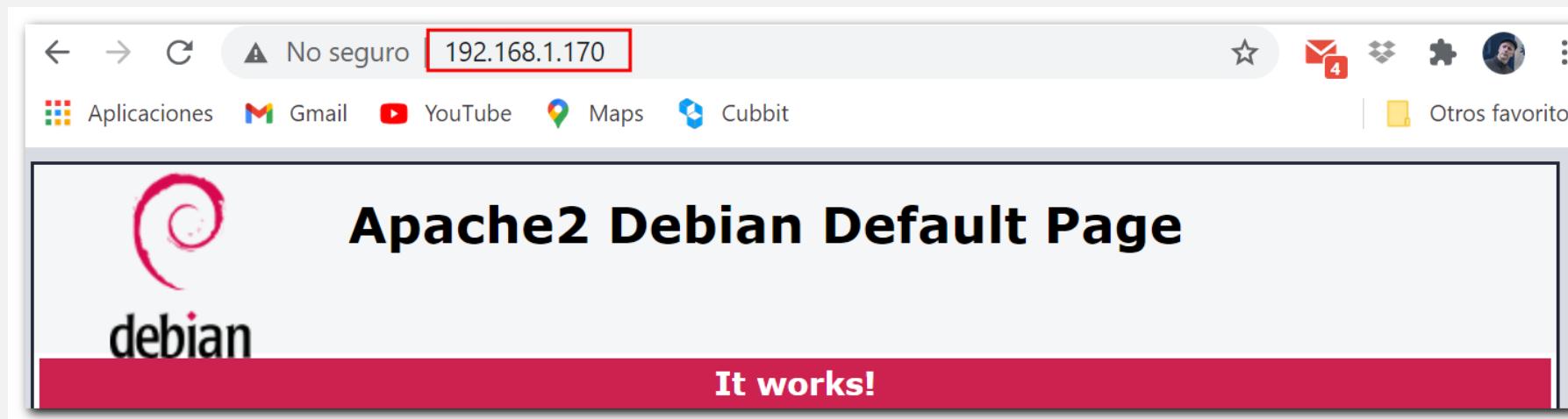


```
administrador@debian-despliegue:/etc/apache2/sites-available$ sudo a2ensite 192.168.1.175.conf
Enabling site 192.168.1.175.
To activate the new configuration, you need to run:
  systemctl reload apache2
administrador@debian-despliegue:/etc/apache2/sites-available$ |
```

# VIRTUAL HOSTS

## CONFIGURACIÓN POR DIRECCIÓN IP

Por último, habrá que cargar los recursos al directorio de publicación definido en DocumentRoot





# Virtual Hosts por Puerto

# VIRTUAL HOSTS

## CONFIGURACIÓN POR PUERTO

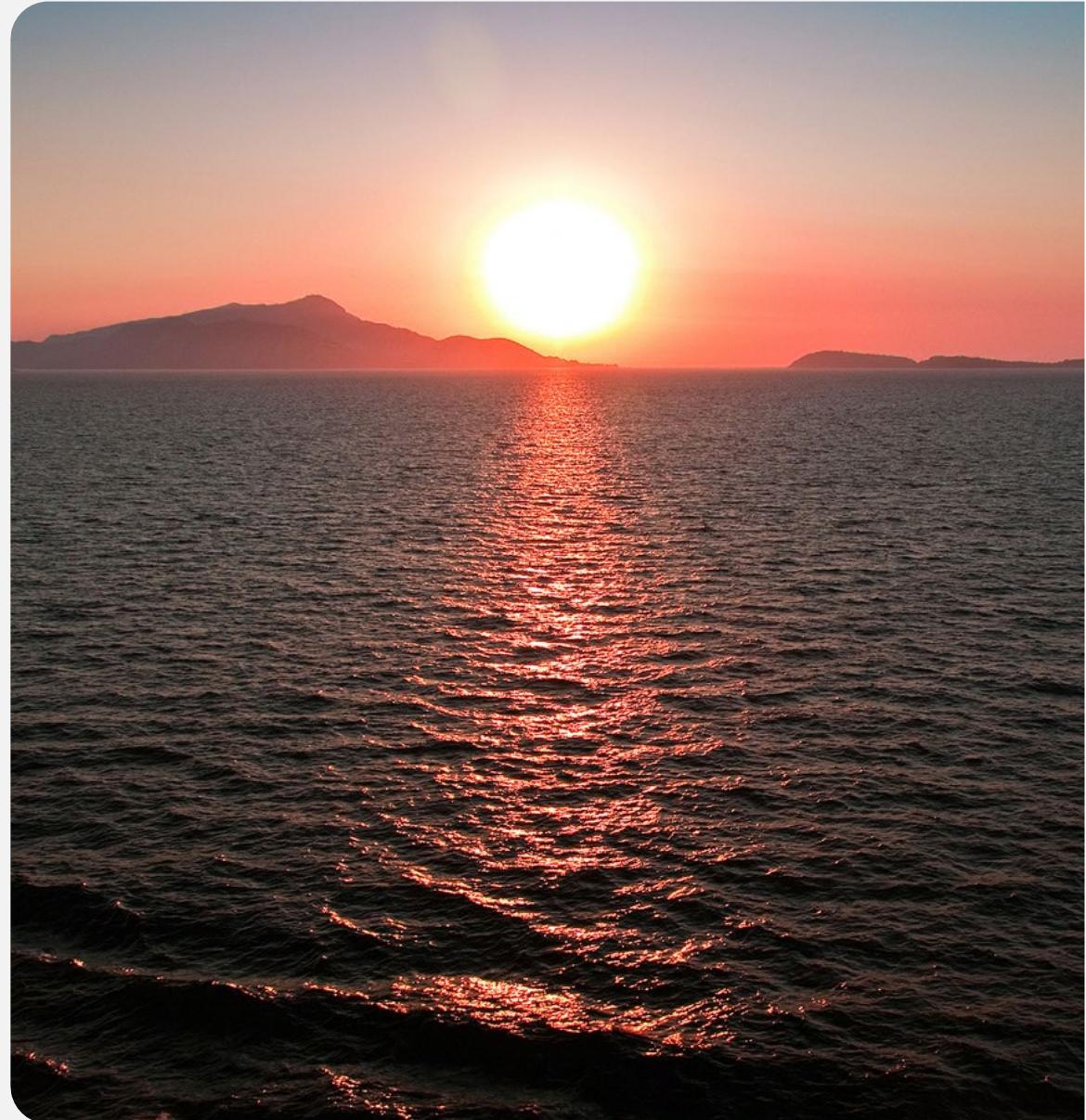
En primer lugar debemos permitir que **Apache** escuche peticiones en el puerto o los puertos que queremos hacer servir para los virtual host

La configuración de los puertos de escucha está en el archivo `/etc/apache2/ports.conf`

A través de la directiva **Listen** se definen los puertos de escucha

La directiva Listen se puede aplicar

- De forma global para todas las direcciones IP
- De forma particular para cierta dirección IP
- De forma condicional



# VIRTUAL HOSTS

## CONFIGURACIÓN POR PUERTO

En el ejemplo se observa la siguiente configuración:

- El servidor escucha en el puerto 80 para todas sus direcciones IP
- El servidor escucha en el puerto 8080 para la dirección IP 192.168.1.175
- El servidor escuchará en el puerto 443 si el módulo **ssl\_module** o el módulo **mod\_gnutls.c** están cargados

```
Listen 80
Listen 192.168.1.175:8080

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>
```

# VIRTUAL HOSTS

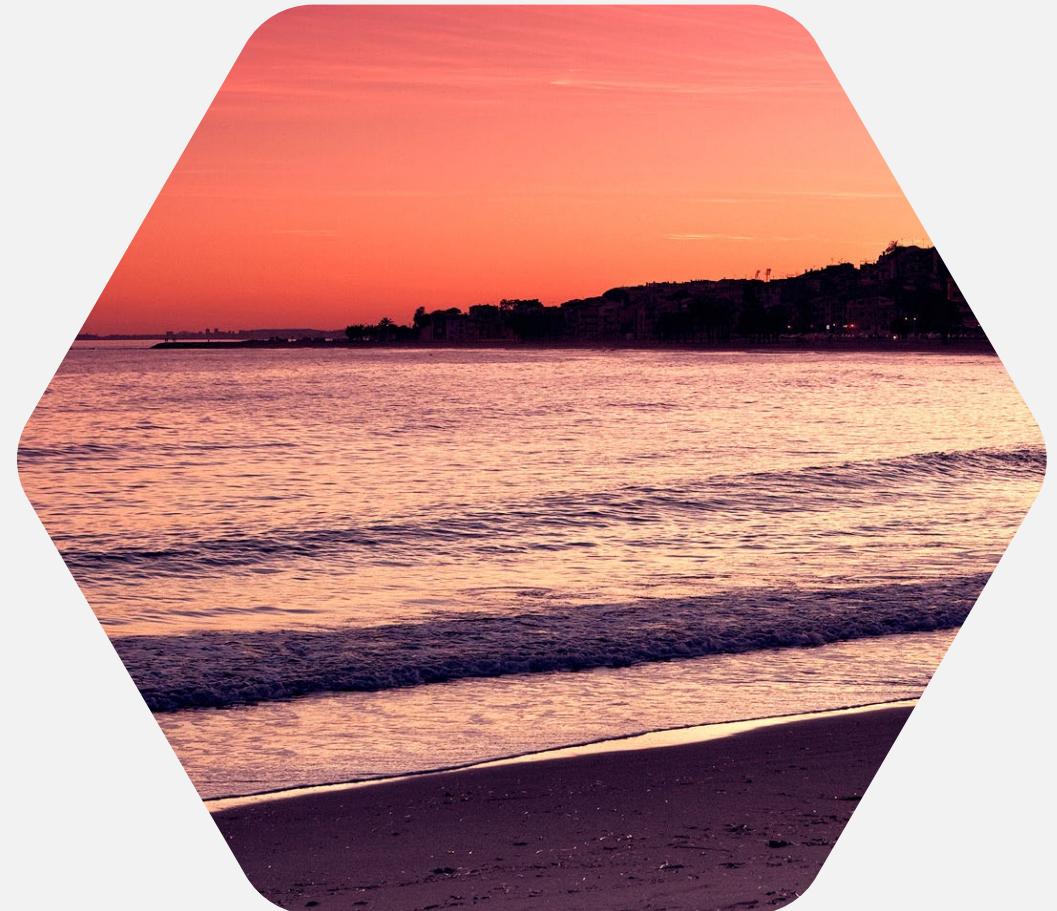
## CONFIGURACIÓN POR PUERTO

Para añadir puertos de escucha debemos añadir líneas con la directiva **Listen** en el archivo `/etc/apache2/ports.conf`

```
Listen 80
Listen 8080

<IfModule ssl_module>
    Listen 443
</IfModule>
```

Tras añadir puertos en este archivo se debe reiniciar el servidor Apache para que la configuración tenga efecto



# VIRTUAL HOSTS

## CONFIGURACIÓN POR PUERTO

Tras el reinicio podemos ver si el sistema está a la escucha en los puertos definidos mediante **nmap localhost**

```
administrador@debian-despliegue:/etc/apache2$ nmap localhost
Starting Nmap 7.70 ( https://nmap.org ) at 2020-12-12 19:24 CET
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00013s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 994 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
53/tcp    open  domain
80/tcp    open  http
631/tcp   open  ipp
8080/tcp  open  http-proxy
10000/tcp open  snet-sensor-mgmt

Nmap done: 1 IP address (1 host up) scanned in 0.08 seconds
```



# VIRTUAL HOSTS

## CONFIGURACIÓN POR PUERTO

Tras la configuración de los puertos podemos añadir configuraciones de Virtual Host creando los archivos que sean necesarios en `/etc/apache2/sites-available`

- Indicamos `<VirtualHost IP:Puerto>` para una IP específica
- Indicamos `<VirtualHost *:Puerto>` para todas las IPs

Como en el servidor de ejemplo disponemos de dos direcciones IP (una principal y otra con una interfaz virtual) hemos creado dos archivos para host virtuales que escuchen en el puerto 8080, uno para la IP 192.168.1.170 y otro para la IP 192.168.1.175

```
administrador@debian-despliegue:/etc/apache2/sites-available$ ls -l
total 24
-rw-r--r-- 1 root root 2078 dic 11 19:09 000-default.conf
-rw-r--r-- 1 root root 117 dic 12 19:38 192.168.1.170_8080.conf
-rw-r--r-- 1 root root 119 dic 12 19:39 192.168.1.175_8080.conf
-rw-r--r-- 1 root root 102 dic 12 18:28 192.168.1.175.conf
-rw-r--r-- 1 root root 6338 ago  8 09:47 default-ssl.conf
```

# VIRTUAL HOSTS

## CONFIGURACIÓN POR PUERTO

El contenido de estos archivos es el que se muestra a continuación

Archivo 192.168.1.170\_8080.conf

```
<VirtualHost 192.168.1.170:8080>
    ServerName "principal.8080"
    DocumentRoot "/var/www/principal_8080"
</VirtualHost>
```

Archivo 192.168.1.175\_8080.conf

```
<VirtualHost 192.168.1.175:8080>
    ServerName "secundario.8080"
    DocumentRoot "/var/www/secundario_8080"
</VirtualHost>
```

# VIRTUAL HOSTS

## CONFIGURACIÓN POR PUERTO

Tras crear los archivos, hay que enlazarlos en `/etc/apache2/sites-enabled` con la utilidad `a2ensite`

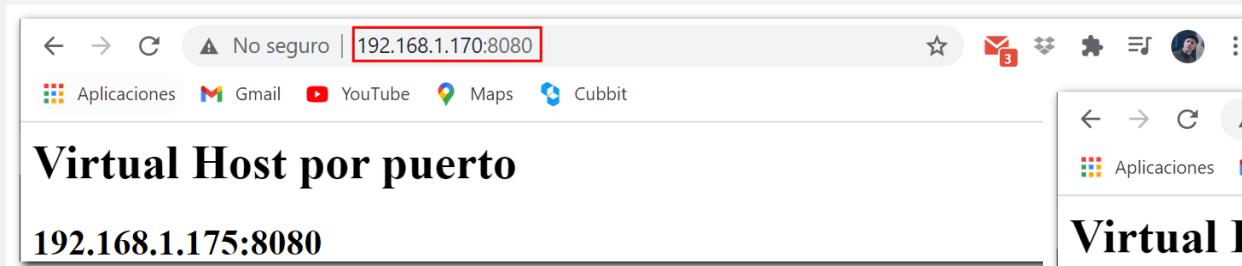
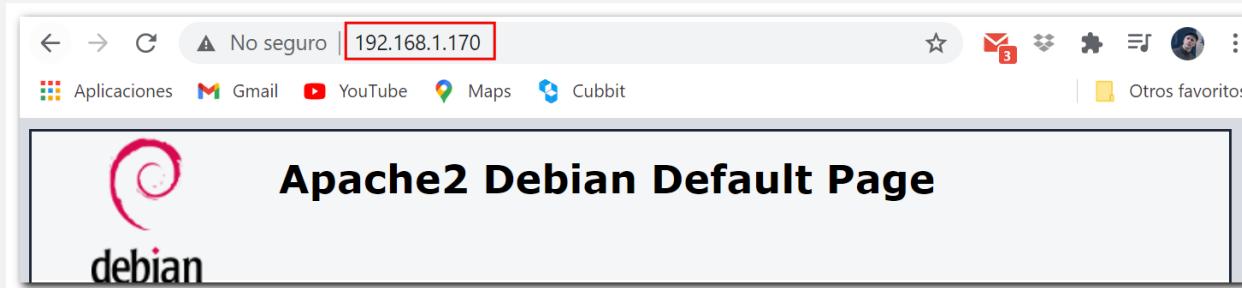
```
administrador@debian-despliegue:/etc/apache2/sites-available$ sudo a2ensite  
192.168.1.170_8080.conf  
site 192.168.1.170_8080 already enabled  
administrador@debian-despliegue:/etc/apache2/sites-available$ sudo a2ensite  
192.168.1.175_8080.conf  
site 192.168.1.175_8080 already enabled
```

Tras reiniciar Apache y crear los recursos correspondientes a cada virtual host podremos obtener las distintas páginas

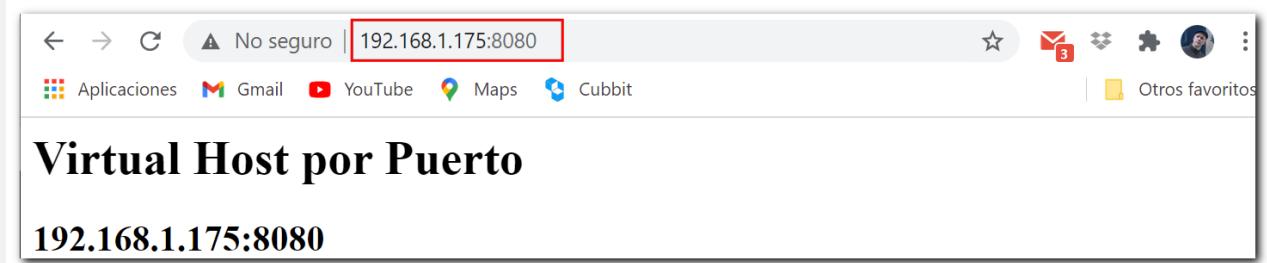
# VIRTUAL HOSTS

## CONFIGURACIÓN POR PUERTO

Virtual Hosts en la IP 192.168.1.170

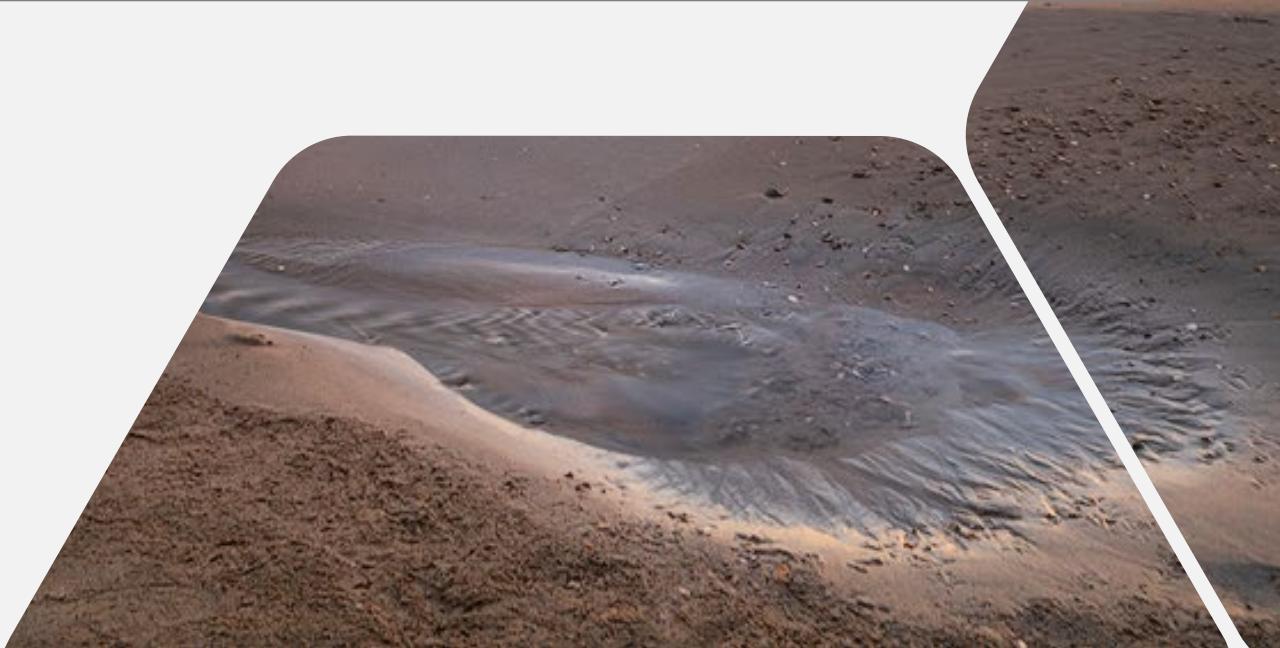


Virtual Hosts en la IP 192.168.1.175





# Virtual Hosts por Nombre



# VIRTUAL HOSTS

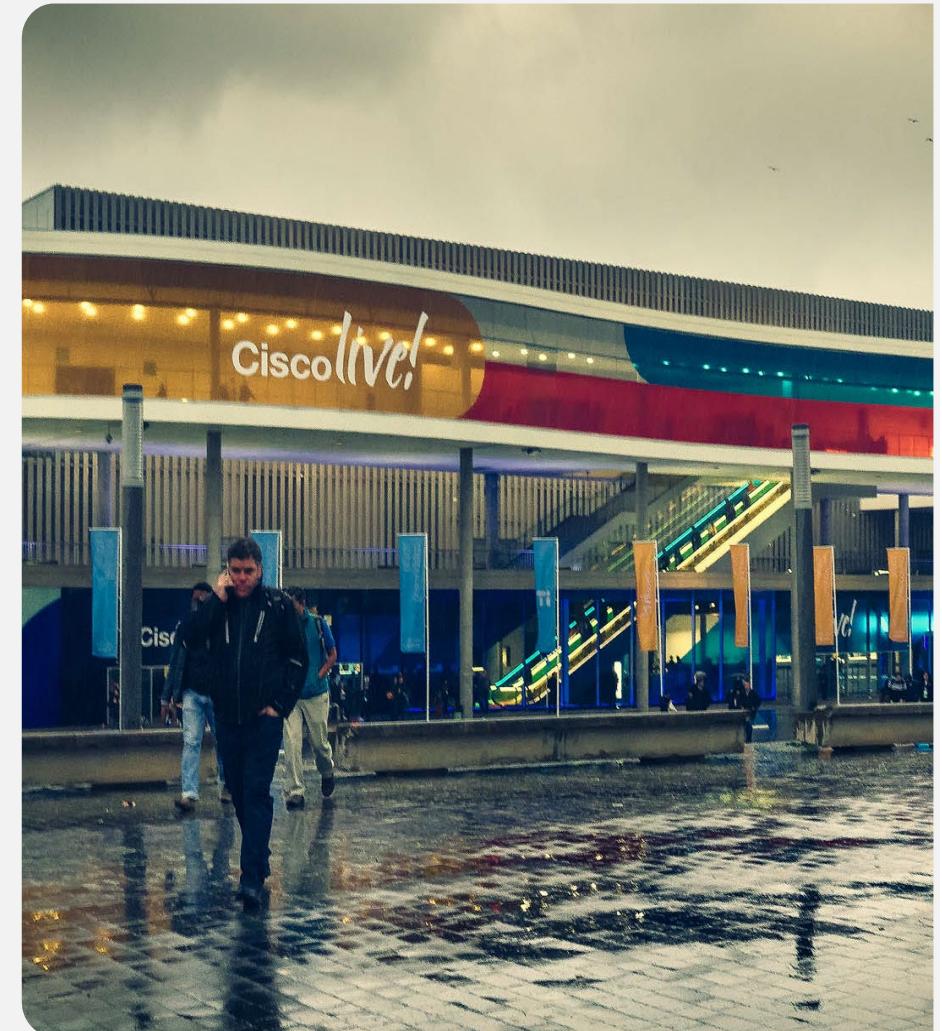
## CONFIGURACIÓN POR NOMBRE

Esta configuración ha de ser coherente con los nombres DNS de los servidores virtuales

Añadimos la configuración del virtual host en un nuevo archivo en  
`/etc/apache2/sites-available`

En el archivo de configuración será necesario definir al menos las siguientes **directivas**:

```
<VirtualHost nombre_del_servidor o comodínnombre_del_servidor
    [ServerAlias otro_nombre_mismo_servidor]
    DocumentRoot /directorio/con/recursos
</VirtualHost>
```



# VIRTUAL HOSTS

## CONFIGURACIÓN POR NOMBRE

Para hacer los ejemplos de configuración vamos a contar con las zonas DNS **olimpo.daw** y **smira.web**



# VIRTUAL HOSTS

## CONFIGURACIÓN POR NOMBRE

Podemos ver los registros de ambas zonas en las imágenes siguientes:

◀

### ★ All Records

In olimpo.daw

Show records matching:  Search

Select all  Invert selection

Name	Type	TTL	Values
olimpo.daw.	NS	Default	ns1.olimpo.daw.
olimpo.daw.	A	Default	192.168.1.170
ns1.olimpo.daw.	A	Default	192.168.1.170
www.olimpo.daw.	CNAME	Default	olimpo.daw.
ftp.olimpo.daw.	CNAME	Default	www
linux.olimpo.daw.	A	Default	192.168.1.171
windows.olimpo.daw.	A	Default	192.168.1.172
mac.olimpo.daw.	A	Default	192.168.1.173
olimpo.daw.	NS	Default	ns2.olimpo.daw.
ns2.olimpo.daw.	A	Default	192.168.1.180
catalina.olimpo.daw.	CNAME	Default	mac
alumnos.olimpo.daw.	NS	Default	ns-alumnos.olimpo.daw.
ns-alumnos.olimpo.daw.	A	Default	192.168.1.180



# VIRTUAL HOSTS

## CONFIGURACIÓN POR NOMBRE

Podemos ver los registros de ambas zonas en las imágenes siguientes:

All Records  
In smira.web

Show records matching:  Search

Select all  Invert selection

Name	Type	TTL	Values
smira.web.	NS	Default	ns1.smira.web.
ns1.smira.web.	A	Default	192.168.1.170
smira.web.	A	Default	192.168.1.170
www.smira.web.	CNAME	Default	smira.web.

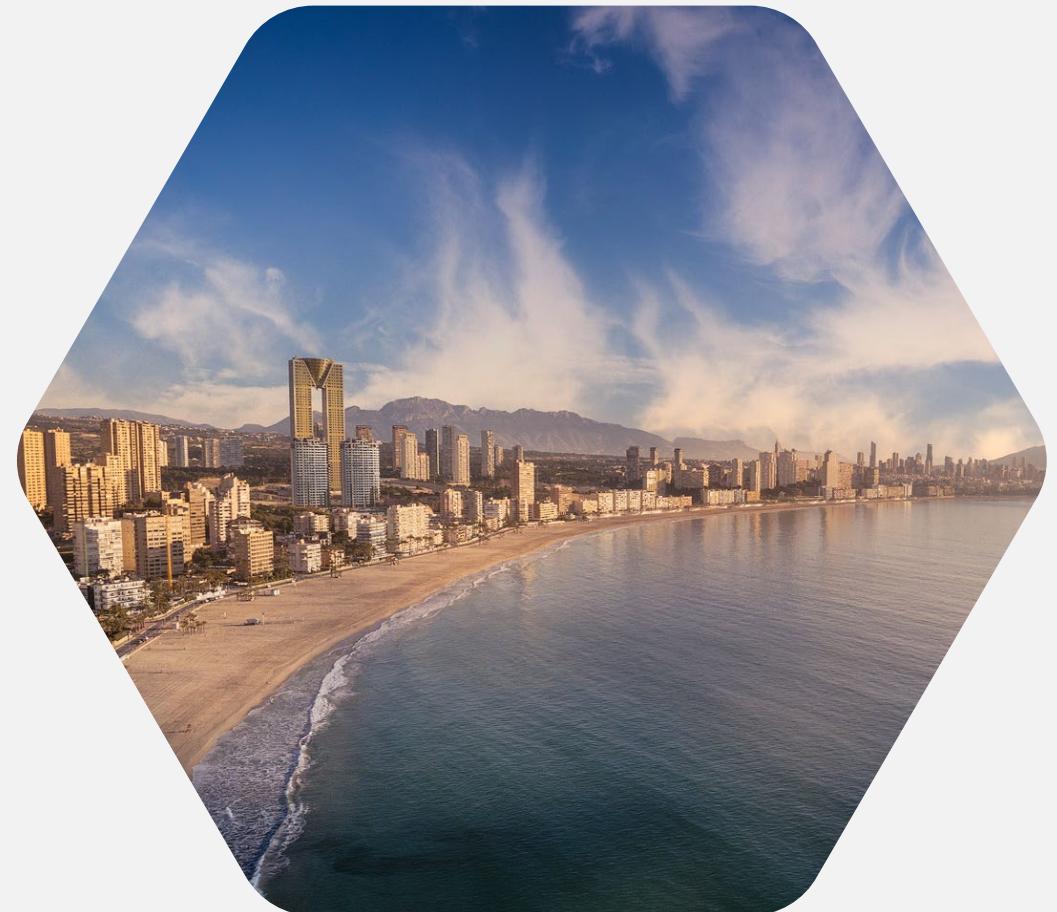


# VIRTUAL HOSTS

## CONFIGURACIÓN POR NOMBRE

En la configuración del archivo del Virtual Host, es importante establecer las directivas:

- **ServerName *nombre\_del\_servidor*.** Nombre DNS del servidor
- **ServerAlias *otro\_nombre\_mismo\_servidor*.** Nombres alternativos para el mismo servidor virtual. Deben existir también en el DNS



# VIRTUAL HOSTS

## CONFIGURACIÓN POR NOMBRE

El contenido del archivo del virtual host para esta configuración es el siguiente:

Archivo olimpo.daw.conf

```
<VirtualHost olimpo.daw:80>
    ServerName "olimpo.daw"
    ServerAlias "www.olimpo.daw" "smira.web"
    DocumentRoot "/var/www/olimpo_daw"
</VirtualHost>
```

Tras crear el archivos, hay que enlazarlo en `/etc/apache2/sites-enabled` con la utilidad `a2ensite`

# VIRTUAL HOSTS

## CONFIGURACIÓN POR NOMBRE

Además, para evitar errores, hay que modificar el archivo del virtual host por defecto, `000-default.conf`:

Extracto del archivo `000-default.conf`

```
<VirtualHost 192.168.1.170:80 *:80>
    ServerName "192.168.1.170"
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
```

En él se puede ver que ahora atiende preferentemente al nombre `192.168.1.170`, pero también a cualquier petición que no coincida con ningún otro virtual host

# VIRTUAL HOSTS

## CONFIGURACIÓN POR NOMBRE

Virtual Host por defecto, URL http://192.168.1.170



Virtual Host olimpo.daw, URL http://olimpo.daw



Virtual Host olimpo.daw accedido por alias www.olimpo.daw



Virtual Host olimpo.daw accedido por alias smira.web



Virtual Host por defecto, URL http://www.smira.web



# VIRTUAL HOSTS

## CONFIGURACIÓN POR NOMBRE

Para solucionar el problema de que un mismo **virtual host** pueda tener **varios nombres** dentro de un mismo subdominio, podemos emplear el comodín \* en la definición de los **ServerAlias**

```
<virtualHost olimpo.daw:80>
    ServerName "olimpo.daw"
    ServerAlias "www.olimpo.daw" "smira.web" "* .smira.web"
    DocumentRoot "/var/www/olimpo_daw"
</virtualHost>
```



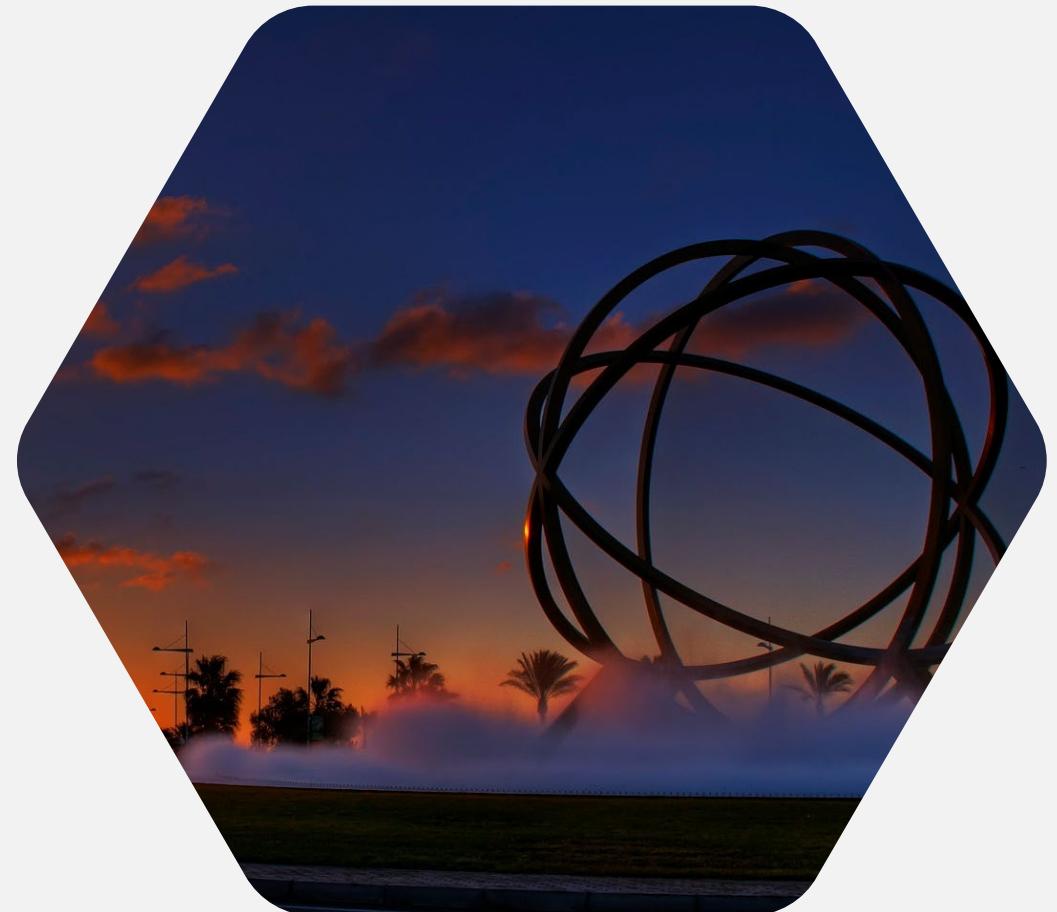
# VIRTUAL HOSTS

## CONFIGURACIÓN POR NOMBRE

Para completar el ejemplo, se ha añadido el alias **w3.smira.web** que apunta a **smira.web**

```
administrador@linux:~/Escritorio$ nslookup w3.smira.web
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
w3.smira.web    canonical name = smira.web.
Name:   smira.web
Address: 192.168.1.170
```



# VIRTUAL HOSTS

## CONFIGURACIÓN POR NOMBRE

Virtual Host olimpo.daw accedido por el alias www.smira.web



Virtual Host olimpo.daw accedido por el alias w3.smira.web

