



Despliegue de múltiples sitios webs usando Server Blocks con soporte PHP

1. Enunciado

El alumnado deberá configurar dos sitios web diferentes en Nginx utilizando Server Blocks (virtual hosts), ambos accesibles mediante nombres de dominio distintos y con HTTPS. Además, configurarán PHP para que uno de los sitios utilice archivos PHP como [index.php](#) y los procese correctamente.

2. Guión

2.1. Paso 1: Configuración inicial de Nginx

- Inicia WSL y asegúrate de que Nginx esté instalado y funcionando correctamente. Para ello ejecuta los comandos necesarios

2.2. Paso 2: Crear directorios para los dos sitios webs

- Crear dos directorios web diferentes en [/var/www/sitio1.com/html](#) y [/var/www/sitio2.com/html](#)
- Establece permisos apropiados a los directorios creados.
- Crea un archivo [index.php](#) para **sitio1** y un [index.html](#) para **sitio2**. Podéis ser creativos respecto al contenido que vayáis a introducir en ambos index.



2.3. Paso 3: Instalar PHP, PHP-FPM

- Actualizar los paquetes del sistema.
- Instalar **PHP** y **PHP-FPM** (php-fpm y php-mysql)
- Verificar la instalación de PHP

2.4. Paso 4: Configurar archivos de configuración de Nginx

- Crea el archivo de configuración para el primer sitio, llámalo sitio1.com y guárdalo en el directorio sites-available como en la anterior práctica.
- Modificar la configuración para que soporte PHP. Modifica la información de la captura para adaptarla a tu práctica.

```
GNU nano 7.2
server{
    listen 443 ssl;
    server_name sitio1.com www.sitio1.com;

    ssl_certificate /etc/nginx/ssl/nginx-selfsigned.crt;
    ssl_certificate_key /etc/nginx/ssl/nginx-selfsigned.key;

    root /var/www/sitio1.com/html;
    index index.php index.html;

    location / {
        try_files $uri $uri/ =404;
    }
    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php-fpm.sock;
    }
    location ~ /\.ht {
        deny all;
    }
}

server{
    listen 80;
    server_name sitio1.com www.sitio1.com
    return 301 https://$server_name$request_uri;
}
```



- Configura **sitio2.com**, en este sitio no vamos a configurar PHP, es decir, recuerda cómo configurastes en la práctica anterior el archivo de configuración.
- Activar los **Server Blocks** creando enlaces simbólicos para habilitar ambos sitios webs. Utiliza el comando `ln -s` y recuerda en qué directorio están los archivos de configuración y sobre todo cuál es el directorio destino. (Directorio donde deben estar los archivos de configuración para habilitar o activar los Server Blocks).
- Probar la configuración de Nginx
- Reiniciar Nginx

2.5. Paso 5: Configuración del archivo hosts en Windows

- Para poder acceder a los dos sitios desde tu navegador en Windows, necesitarás configurar el archivo hosts para que los nombres de dominio apunten a la IP de WSL. Para ello:
 - Abre el archivo hosts en Windows (necesitaras permisos de administrador), la ruta es la misma que la de la práctica anterior.
 - Usa la siguiente línea como ejemplo para añadir al archivo hosts:

<IP de tu WSL> sitio1.com

2.6. Paso 6: Redirigir el puerto 443 en Windows (opcional si lo hicistes en la práctica anterior)

- Abre el CMD e inserte el siguiente comando:

```
netsh interface portproxy add v4tov4 listenaddress=0.0.0.0  
listenport=443 connectaddress=<IP de tu WSL> connectport=443
```

- Verifica la redirección.



2.7. Paso 7: Prueba en el navegador

- Abre tu navegador en Windows y prueba a acceder a los siguientes sitios:
 - <https://sitio1.com> y deberías de ver el contenido del archivo PHP (index.php)
 - <https://sitio2.com> y deberías de ver la página estática (index.html).

3. Preguntas

1. ¿Qué función tiene la directiva *server_name* en el bloque *server* de Nginx?

Explica por qué utilizamos *sitio1.com* y *www.sitio1.com* en la configuración de los Server Blocks

2. ¿Qué hace la línea *listen 443 ssl;* dentro del bloque *server*?

Explica qué implicaciones tiene esta línea y por qué es importante para HTTPS.

3. ¿Cuál es la razón de utilizar *return 301 https://\$server_name\$server_uri;* en el bloque del puerto 80?

Explica qué hace esta línea y por qué es importante para la redirección de HTTP a HTTPS.

4. ¿Qué significan las siguientes líneas y por qué son necesarias en el bloque *server* para PHP?

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

5. Explica la función de la directiva *location /* dentro del bloque *server*.

¿Qué sucede cuando no se encuentra un archivo solicitado en el sitio web?