

Práctica 2.4. Proxy inverso con Nginx

LA PRÁCTICA 2.1 DEBE ESTAR REALIZADA

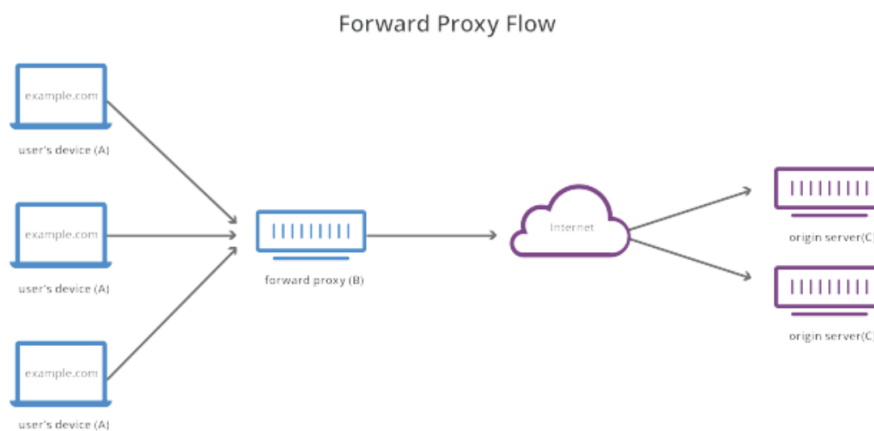
1. Introducción

2.1 ¿Qué es un servidor proxy?

Un proxy de reenvío, a menudo llamado proxy, servidor proxy o proxy web, es un servidor que se encuentra frente a un grupo de máquinas cliente. Cuando esas máquinas realizan solicitudes a sitios y servicios en Internet, el servidor proxy intercepta esas solicitudes y luego se comunica con los servidores web en nombre de esos clientes, como un intermediario.

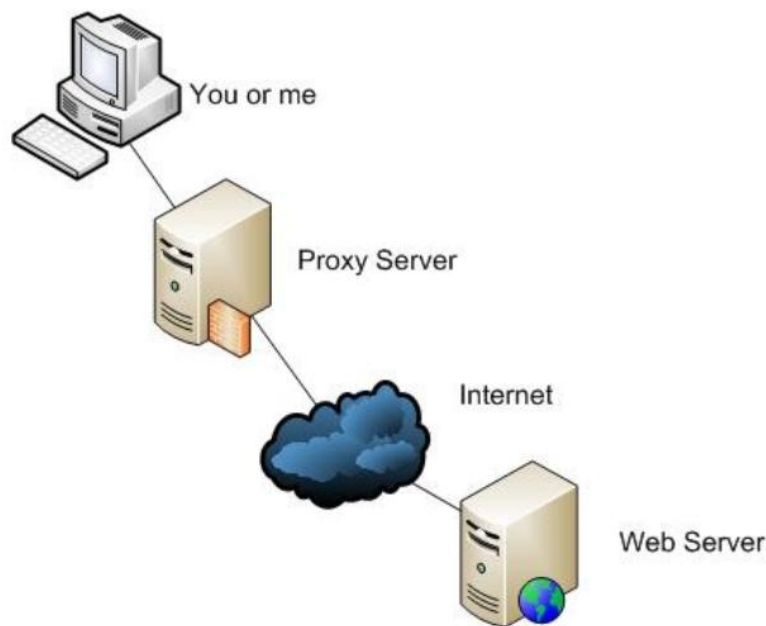
Por ejemplo, tomemos como ejemplo 3 máquinas involucradas en una comunicación típica de proxy de reenvío:

- A: Esta es la máquina del hogar de un usuario.
- B: este es un servidor proxy de reenvío
- C: este es el servidor de origen de un sitio web (donde se almacenan los datos del sitio web)



En una comunicación estándar por Internet, la máquina A se comunicaría directamente con la máquina C, con el cliente enviando solicitudes al servidor de origen y el servidor de origen respondiendo al cliente. Cuando hay un proxy de reenvío, A enviará solicitudes a B, que luego reenviará la solicitud a C. C enviará una respuesta a B, que reenviará la respuesta a A.

¿Por qué agregar este intermediario adicional a nuestra actividad en Internet?



Hay algunas razones por las que uno podría querer usar un proxy de reenvío:

- **Para evitar restricciones de navegación estatales o institucionales:** algunos gobiernos, escuelas y otras organizaciones usan firewalls para dar a sus usuarios acceso a una versión limitada de Internet. Se puede usar un proxy de reenvío para sortear estas restricciones, ya que permiten que el usuario se conecte al proxy en lugar de directamente a los sitios que está visitando.
- **Para bloquear el acceso a cierto contenido:** a la inversa, los proxies también se pueden configurar para bloquear el acceso de un grupo de usuarios a ciertos sitios. Por ejemplo, una red escolar puede estar configurada para conectarse a la web a través de un proxy que habilita reglas de filtrado de contenido, negándose a reenviar respuestas de Facebook y otros sitios de redes sociales.
- **Para proteger su identidad en línea:** en algunos casos, los usuarios habituales de Internet simplemente desean un mayor anonimato en línea, pero en otros casos, los usuarios de Internet viven en lugares donde el gobierno puede imponer graves consecuencias a los disidentes políticos. Criticar al gobierno en un foro web o en las redes sociales puede dar lugar a multas o encarcelamiento para estos usuarios. Si uno de estos disidentes usa un proxy de reenvío para conectarse a un sitio web donde publica comentarios políticamente sensibles, la dirección IP utilizada para publicar los comentarios será más difícil de rastrear hasta el disidente. Solo estará visible la dirección IP del servidor proxy.

2.2 ¿Qué es un proxy inverso?

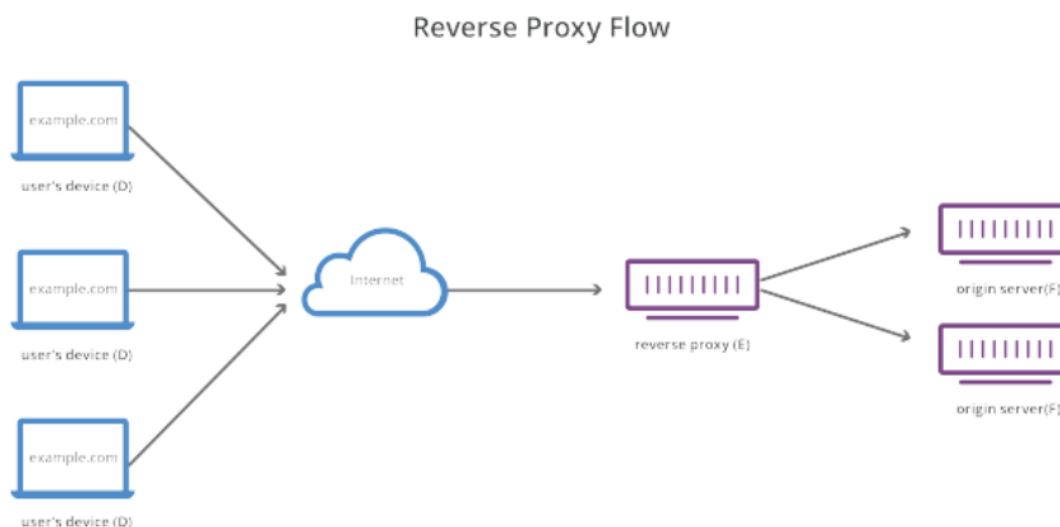
Estaríamos hablando del caso opuesto al anterior.

Un proxy inverso es un servidor que se encuentra frente a uno o más servidores web, interceptando las solicitudes de los clientes. Esto es diferente de un proxy de reenvío, donde el proxy se encuentra frente a los clientes. Con un proxy inverso, cuando los clientes envían solicitudes al servidor de un sitio web, esas solicitudes son interceptadas en la frontera de la red por el servidor proxy inverso. El servidor proxy inverso enviará solicitudes y recibirá respuestas del servidor del sitio web.

La diferencia entre un proxy directo y inverso es sutil pero importante. Una forma simplificada de resumir sería decir que un **proxy de reenvío** se encuentra frente a un cliente y garantiza que ningún servidor de origen se comunice nunca directamente con ese cliente específico. Por otro lado, un **proxy inverso** se encuentra frente a un servidor de origen y garantiza que ningún cliente se comunice nunca directamente con ese servidor de origen.

Una vez más, ilustremos nombrando las máquinas involucradas:

- D: cualquier número de ordenadores domésticos de los usuarios
- E: este es un servidor proxy inverso
- F: uno o más servidores de origen



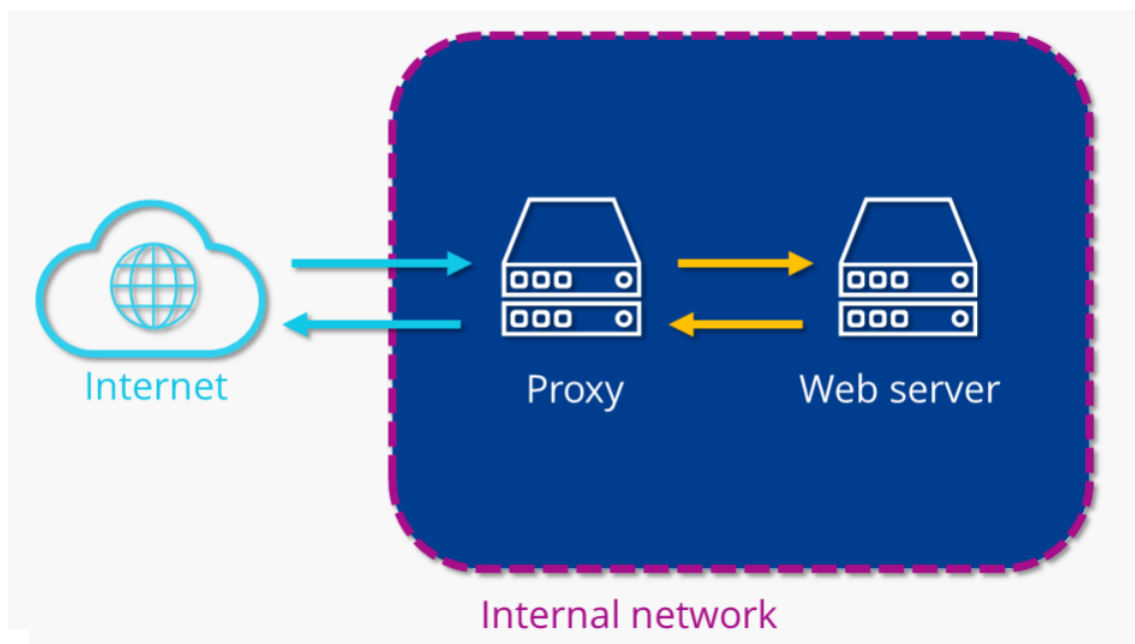
Normalmente, todas las solicitudes de D irían directamente a F, y F enviaría respuestas directamente a D. Con un proxy inverso, todas las solicitudes de D irán directamente a E, y E enviará sus solicitudes y recibirá respuestas de F. E luego transmita las respuestas apropiadas a D.

A continuación, se describen algunos de los beneficios de un proxy inverso:

- **Balanceo de carga:** es posible que un sitio web popular que recibe millones de usuarios todos los días no pueda manejar todo el tráfico entrante del sitio con un solo servidor de origen. En cambio, el sitio se puede distribuir entre un grupo de

servidores diferentes, todos manejando solicitudes para el mismo sitio. En este caso, un proxy inverso puede proporcionar una solución de balanceo de carga que distribuirá el tráfico entrante de manera uniforme entre los diferentes servidores para evitar que un solo servidor se sobrecargue. En el caso de que un servidor falle por completo, otros servidores pueden intensificar para manejar el tráfico.

- **Protección contra ataques:** con un proxy inverso en su lugar, un sitio web o servicio nunca necesita revelar la dirección IP de su (s) servidor (es) de origen. Esto hace que sea mucho más difícil para los atacantes aprovechar un ataque dirigido contra ellos, como un ataque DDoS.
- **Almacenamiento en caché:** un proxy inverso también puede almacenar contenido en caché, lo que resulta en un rendimiento más rápido. Por ejemplo, si un usuario en París visita un sitio web con proxy inverso con servidores web en Los Ángeles, el usuario podría conectarse a un servidor proxy inverso local en París, que luego tendrá que comunicarse con un servidor de origen en Los Ángeles. El servidor proxy luego puede almacenar en caché (o guardar temporalmente) los datos de respuesta. Los usuarios parisinos posteriores que naveguen por el sitio obtendrán la versión en caché local del servidor proxy inverso parisino, lo que dará como resultado un rendimiento mucho más rápido.
- **Cifrado SSL** - Cifrado y descifrado SSL (o TLS comunicaciones) para cada cliente pueden ser computacionalmente caro para un servidor de origen. Se puede configurar un proxy inverso para descifrar todas las solicitudes entrantes y cifrar todas las respuestas salientes, liberando valiosos recursos en el servidor de origen.



3. Configuraciones

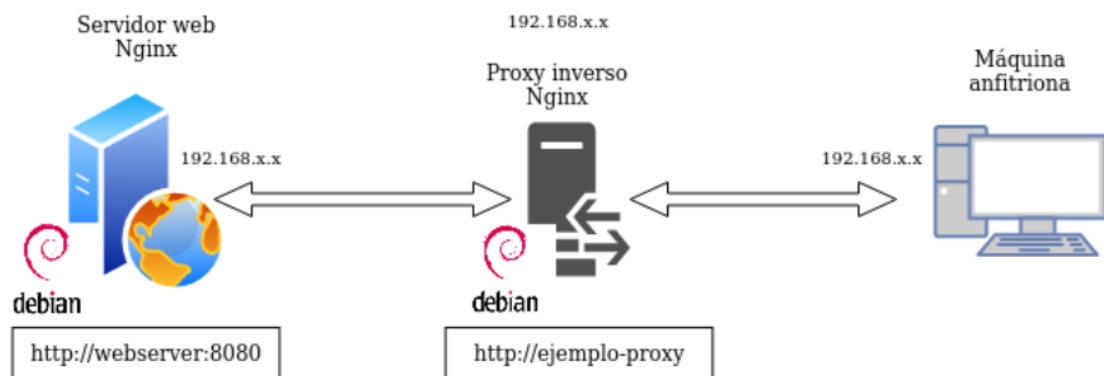
En esta práctica necesitaremos crear otra máquina virtual Debian. Podéis clonar la que ya funciona correctamente.

Por lo tanto, tendremos 2 máquina virtuales:

1. **Servidor web.** La máquina que servirá las páginas web que ya hemos configurado, así pues, utilizaremos el servidor que ya tenemos configurado de la Práctica 2.1.
2. **Proxy-inverso.** El nuevo servidor clon Debian con Nginx que **tendremos que configurar.**

Ojo al clonar las máquinas virtuales porque hay que darle a crear una nueva MAC, de lo contrario no tendréis IP en esa máquina.

El diagrama de red quedará así:



Para que todo quede más diferenciado y os quede más claro que la petición está pasando por el proxy inverso y llega al servidor web destino, vamos a hacer que cada uno de los servidores escuche las peticiones en un puerto distinto.

Cambios en el servidor web

1. En primer lugar, debéis cambiar el nombre que tuviera vuestra web por el de *webserver*, ello implica:
 - Cambiar el nombre del archivo de configuración de sites-available para Nginx
 - Cambiar el nombre del sitio web (*server_name*) dentro de este archivo de configuración donde haga falta
 - No os olvidéis de eliminar el link simbólico antiguo con el comando `unlink nombre_del_link` dentro de la carpeta *sites-enabled* y crear el nuevo para el nuevo nombre de archivo.
2. En el archivo de configuración del **sitio web**, en lugar de hacer que el servidor escuche en el puerto 80, cambiadlo al 8080.

3. Reiniciar Nginx

Proxy-inverso

Ahora, cuando intentamos acceder a *http://ejemplo-proxy* (o el nombre que tuvieráis de vuestra web de las prácticas anteriores), en realidad estaremos accediendo al proxy, que nos redirigirá a *http://webserver:8080*, el servidor web que acabamos de configurar para que escuche con ese nombre en el puerto 8080.

Para ello:

- Crear un archivo de configuración en *sites-available* con el nombre ejemplo-proxy
- Este archivo de configuración será más simple, tendrá la siguiente forma

```
server {  
    listen ____;  
    server_name _____;  
    location / {  
        proxy_pass http://_____:____;  
    }  
}
```

Donde, ***mirando el diagrama de red y teniendo en cuenta la configuración hecha hasta ahora***, debéis completar:

- El puerto donde está escuchando el proxy inverso
- El nombre de vuestro dominio o sitio web original al que accedemos en el proxy
- La directiva *proxy_pass* indica a dónde se van a redirigir las peticiones, esto es, al servidor web. Por tanto, debéis poner la **IP y número de puerto adecuados** de vuestro sitio web configurado en el apartado anterior.
- Crear el link simbólico pertinente

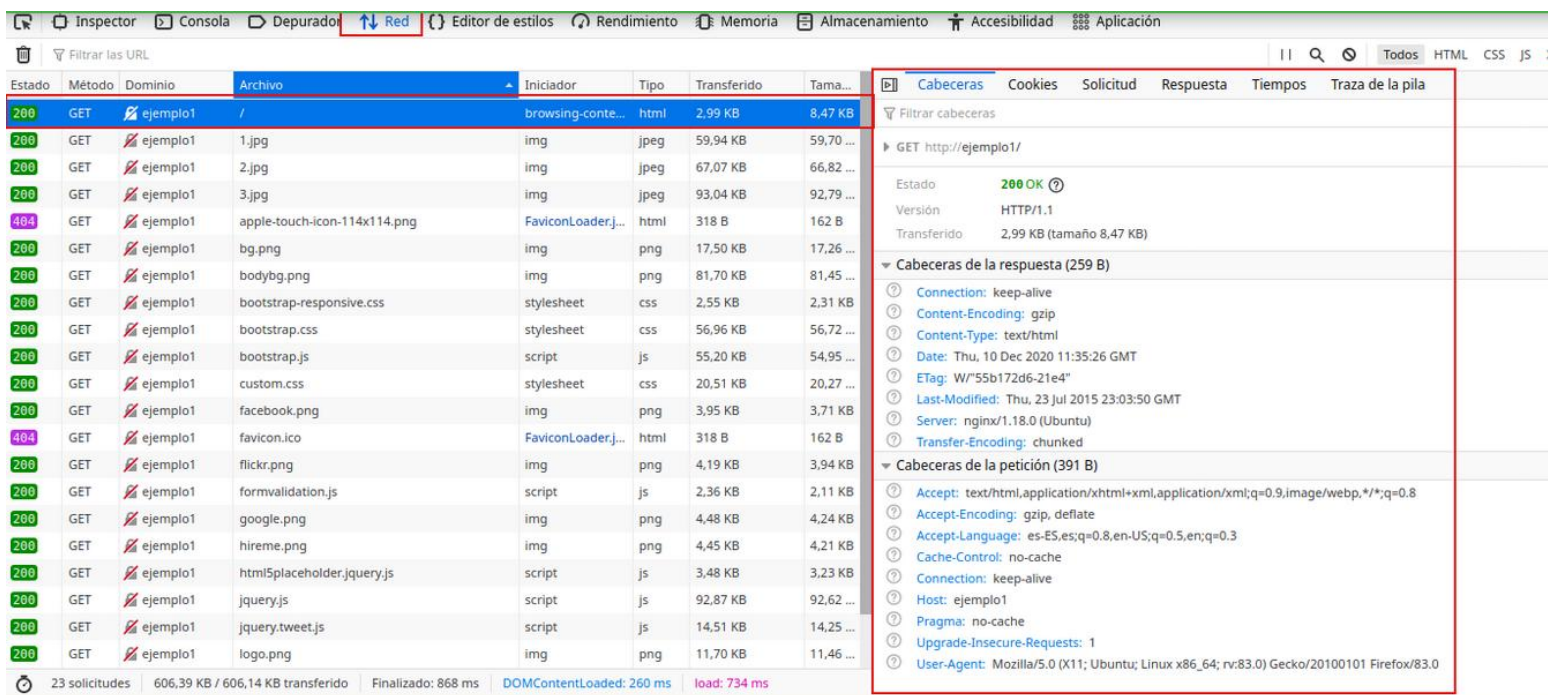
Esto es para simular la situación en la que nosotros, como clientes, cuando accedamos a nuestro sitio web, no necesitemos saber cómo está todo configurado, sólo necesitamos saber el nombre de la web.

Debéis modificar el archivo host que configurastéis en la práctica 2.1. Si miráis el diagrama de red, ahora el nombre de vuestro sitio web se corresponderá con la IP de la nueva máquina clon que hace de proxy. Será ésta la encargada de redirigirnos automáticamente al verdadero sitio web.

4. Comprobaciones

Si accedéis a vuestro sitio web, debéis poder seguir accediendo sin problemas.

- Comprobad en los access.log de los dos servidores que llega la petición
- Comprobad además la petición y respuesta con las herramientas de desarrollador de Firefox en Xubuntu. Pulsando F12 en el navegador os aparecerán estas herramientas



The screenshot shows the Firefox Developer Tools interface. The 'Network' tab is active, displaying a list of requests. The first request, 'GET http://ejemplo1/', is highlighted in red. The 'Cabezas' (Headers) sub-tab is selected, showing the response status '200 OK' and various headers like 'Content-Type: text/html', 'Date: Thu, 10 Dec 2020 11:35:26 GMT', 'ETag: W/"55b172d6-21e4"', 'Last-Modified: Thu, 23 Jul 2015 23:03:50 GMT', 'Server: nginx/1.18.0 (Ubuntu)', and 'Transfer-Encoding: chunked'.

En la primera petición (marcada en rojo), utilizando el apartado “Red” (también marcado en rojo) y también en rojo está señalado dónde se puede ver la respuesta de la petición GET HTTP (200 OK).

También vemos las cabeceras que se incluyen en la petición (método GET) y en la respuesta a esta petición.

4.1 Añadiendo cabeceras

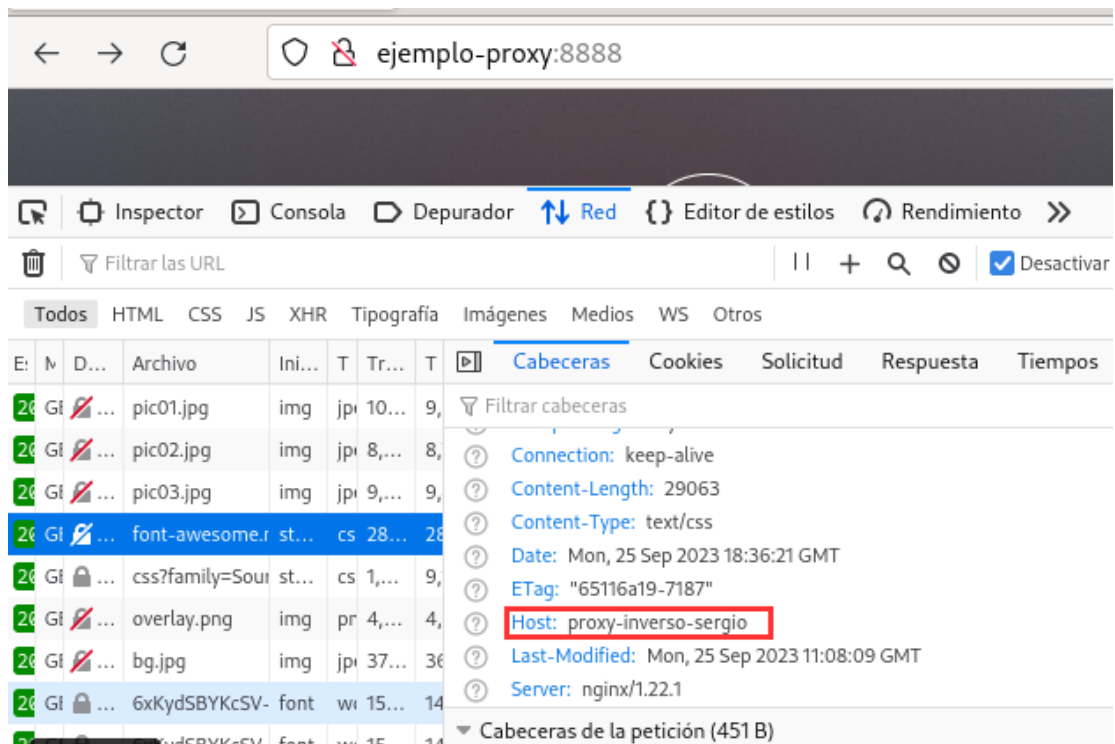
Además de haber mirado los logs, vamos a demostrar aún de forma más clara que la petición está pasando por el proxy inverso y que está llegando al servidor web y que vuelve por el mismo camino.

El servidor web es capaz de añadir cabeceras en las respuestas a las peticiones. Así pues, vamos a **configurar tanto el proxy inverso como el servidor web** para que añadan cada uno la **cabecera “Host”**.

Para añadir cabeceras, en el archivo de configuración del sitio web debemos añadir dentro del bloque location / { ... } debemos añadir la directiva:


```
add_header Host nombre_del_host;
```

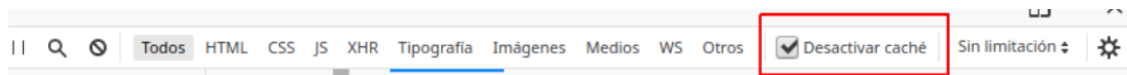
1. Añadiremos primero esta cabecera únicamente en el archivo de configuración del sitio web del proxy inverso. El Nombre_del_host será Proxy_inverso_vuestronombre.
2. Reiniciamos Nginx
3. Comprobamos que podemos acceder al sitio web sin problemas
4. Con las herramientas de desarrollador comprobamos que la petición ha pasado por el proxy inverso que ha añadido la cabecera en la respuesta:



5. Hacemos lo propio con el servidor web. Esta vez el Nombre_del_host será servidor_web_vuestronombre.
6. Si todo está configurado correctamente, al examinar las peticiones y respuestas, os aparecerán las dos cabeceras que han incluido en la respuesta tanto el proxy inverso como el servidor web. .

Almacenamiento				Accesibilidad		Aplicación	
<div> <div> </div> <div>+</div> <div>Q</div> <div>⌂</div> </div>				<div> <div>Todos</div> <div>HTML</div> <div>CSS</div> <div>JS</div> <div>XHR</div> <div>Tipografía</div> <div>Imágenes</div> <div>Medios</div> <div>WS</div> <div>Otros</div> <div>Desactivar caché</div> </div>			
Tipo	Transferido	Tamaño	<div> <div>Cabeceras</div> <div>Cookies</div> <div>Solicitud</div> <div>Respuesta</div> <div>Tiempos</div> </div>				
html	4,15 KB	14,52 KB	<div> <div>Filtrar cabeceras</div> <div> <div>GET http://daw_nginx:8888/images/pic02.jpg</div> <div> <div>Estado</div> <div>200 OK</div> <div> <div>Versión</div> <div>HTTP/1.1</div> <div> <div>Transferido</div> <div>9,17 KB (tamaño 8,90 KB)</div> <div> <div>Política de referencia</div> <div>strict-origin-when-cross-origin</div> <div> <div>Prioridad de la solicitud</div> <div>Low</div> <div> <div>Resolución DNS</div> <div>Sistema</div> </div> </div> </div> </div> </div> </div></div></div>				
css	32,90 KB	32,63 KB					
jpeg	10,33 KB	10,06 KB					
jpeg	9,17 KB	8,90 KB					
jpeg	9,97 KB	9,70 KB					
js	96,24 KB	95,96 KB					
js	9,37 KB	9,09 KB					
js	12,71 KB	12,43 KB					
js	9,08 KB	8,80 KB					
css	29,33 KB	29,06 KB					
css	1,37 KB	9,35 KB					
png	4,65 KB	4,39 KB					
jpeg	38,13 KB	37,86 KB					
woff2	15,66 KB	14,82 KB					
woff2	15,60 KB	14,78 KB					
woff2	72,17 KB	71,90 KB					
js...	300 B	153 B					

Es muy importante que para realizar estas comprobaciones tengáis marcado el checkbox *Desactivar caché* o en una ventana privada del navegador.



Si no marcáis esto, la página se guardará en la memoria caché del navegador y no estaréis recibiendo la respuesta del servidor sino de la caché del navegador, lo que puede dar lugar a resultados erróneos.

ACTIVIDADES

1. Para esta actividad, es importante que en el vídeo mostréis las 2 máquinas virtuales, así como las dos cabeceras que se añaden a la respuesta HTTP.