

Proxy Inverso

El código de Vagrant para el despliegue sería el siguiente:

```
Vagrantfile
1
2 Vagrant.configure("2") do |config|
3   config.vm.box = "debian/bullseye64"
4   config.vm.provider "virtualbox" do |vb|
5     vb.memory = "256"
6   end
7
8   config.vm.provision "shell", inline: <<-SHELL
9     apt-get update && apt-get install -y nginx
10  SHELL
11
12  config.vm.define "proxy" do |p|
13    p.vm.hostname = "www.example.test"
14    p.vm.network "public_network"
15    p.vm.network "private_network", ip: "192.168.57.10"
16  end
17
18  config.vm.define "web" do |w|
19    w.vm.hostname = "w1.example.test"
20    w.vm.network "private_network", ip: "192.168.57.11"
21  end
22 end
23
24
```

Servidor w1

Configuraremos el servidor w1 con los siguientes pasos:

1. Sobre-escribiremos el archivo de configuración del sitio web:

```
server {
    listen 8080;
    listen [::]:8080;

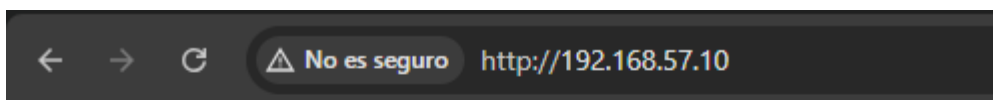
    server_name w1.example.test;
    root /var/www/w1/html;
    index index.html index.htm;

    location / {
        add_header Host w1.example.test;
        try_files $uri $uri/ =404;
    }
}
```

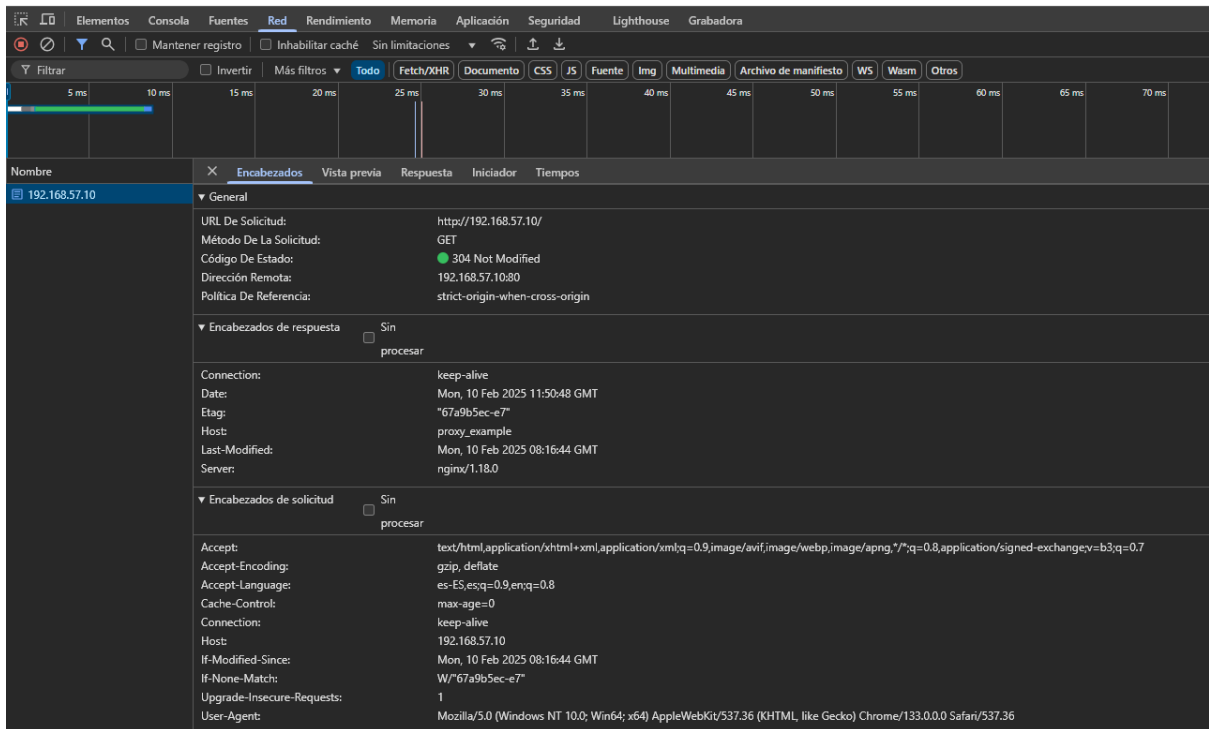
Crearemos una página web de prueba:

```
web > <> test.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9      <h1>Pagina de prueba</h1>
10 </body>
11 </html>
```

1. Accederemos en nuestro navegador a la dirección **192.168.57.10**.

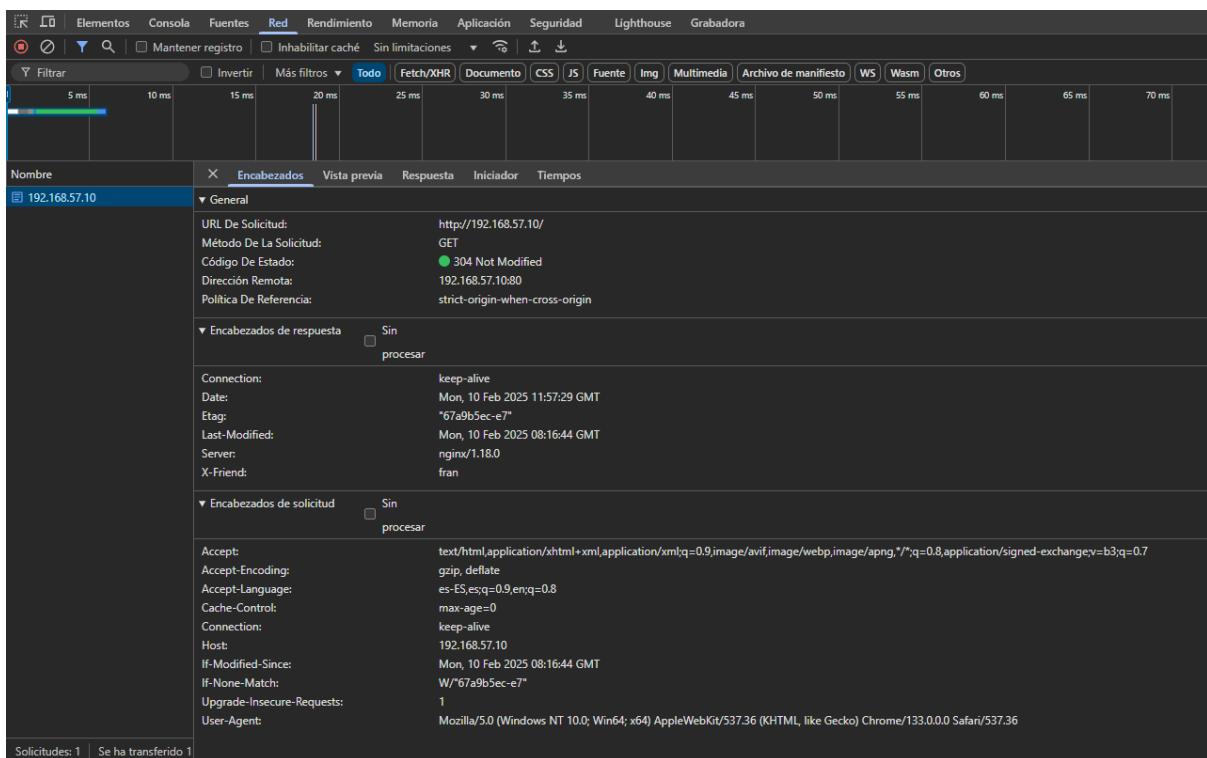


Pagina de prueba



Con las herramientas de desarrollador comprobamos que la petición ha pasado por el proxy inverso que ha añadido la cabecera en la respuesta.

Cabeceras en el servidor web



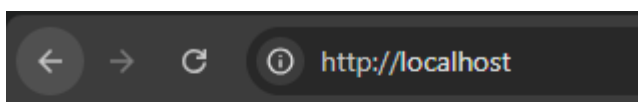
The screenshot shows the Chrome DevTools Network tab. The top toolbar includes buttons for 'Filtrar', 'Invertir', 'Más filtros', and a dropdown menu with 'Todo', 'Fetch/XHR', 'Documento', 'CSS', 'JS', 'Fuente', 'Img', 'Multimedia', 'Archivo de manifiesto', 'WS', 'Wasm', and 'Otros'. Below the toolbar is a timeline showing a single request at 192.168.57.10. The 'Encabezados' (Headers) tab is selected, showing the following details:

- General:**
 - URL De Solicitud: `http://192.168.57.10/`
 - Método De La Solicitud: `GET`
 - Código De Estado: `304 Not Modified`
 - Dirección Remota: `192.168.57.10:80`
 - Política De Referencia: `strict-origin-when-cross-origin`
- Encabezados de respuesta:** ☐ Sin procesar
 - Connection: `keep-alive`
 - Date: `Mon, 10 Feb 2025 11:57:29 GMT`
 - Etag: `"67a9b5ec-e7"`
 - Last-Modified: `Mon, 10 Feb 2025 08:16:44 GMT`
 - Server: `nginx/1.18.0`
 - X-Friend: `fran`
- Encabezados de solicitud:** ☐ Sin procesar
 - Accept: `text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7`
 - Accept-Encoding: `gzip, deflate`
 - Accept-Language: `es-ES;q=0.9,en;q=0.8`
 - Cache-Control: `max-age=0`
 - Connection: `keep-alive`
 - Host: `192.168.57.10`
 - If-Modified-Since: `Mon, 10 Feb 2025 08:16:44 GMT`
 - If-None-Match: `W/"67a9b5ec-e7"`
 - Upgrade-Insecure-Requests: `1`
 - User-Agent: `Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36`

At the bottom, it shows 'Solicitudes: 1' and 'Se ha transferido 1'.

Ampliación

Tras configurar el docker probamos si ha funcionado



Pagina de prueba

Usando las herramientas de desarrollador vemos que toda la configuración se ha aplicado

