

A dark blue vertical bar runs along the left edge of the page. A blue arrow-shaped banner points to the right from this bar, containing the text 'DESPLIEGUE DE APLICACIONES WEB'. In the bottom-left corner, several thin, curved lines in dark blue and light grey sweep upwards and to the right.

DESPLIEGUE DE APLICACIONES WEB

CÓDIGOS DE ESTADO HTTP Y USER-AGENT

ÍNDICE

CÓDIGOS DE ESTADO DE HTTP	3
A. 1XX: INFORMACIÓN.....	3
B. 2XX: EXITOSO	3
C. 3XX: REDIRECCIONES	4
D. 4XX: ERRORES DE CLIENTE	5
E. 5XX: ERRORES DE SERVIDOR	8
USER-AGENT	9
A. SINTAXIS	9
B. CADENA UA DE FIREFOX	9
C. CADENA UA DE CHROME	10
D. CADENA UA DE OPERA.....	10
E. CADENA UA DE SAFARI	10
F. CADENA UA DE MICROSOFT EDGE.....	10
OBSERVANDO LOGS	11

CÓDIGOS DE ESTADO DE HTTP

Cuando un navegador solicita un servicio de un servidor web, puede ocurrir un error, y el servidor podría devolver un código de error como "404 Not Found".

Es común nombrar estos errores mensajes de error HTML.

Pero estos mensajes son algo llamado mensajes de estado HTTP. De hecho, el servidor siempre devuelve un mensaje para cada solicitud. El mensaje más común es 200 OK.

A. 1XX: INFORMACIÓN

CÓDIGO	SIGNIFICADO
100 Continue	Esta respuesta provisional indica que todo hasta ahora está bien y que el cliente debe continuar con la solicitud o ignorarla si ya está terminada.
101 Switching Protocol	Este código se envía en respuesta a un encabezado de solicitud Upgrade por el cliente e indica que el servidor acepta el cambio de protocolo propuesto por el agente de usuario.
102 Processing	Este código indica que el servidor ha recibido la solicitud y aún se encuentra procesándola, por lo que no hay respuesta disponible.
103 Early Hints	Este código de estado está pensando principalmente para ser usado con el encabezado Link, permitiendo que el agente de usuario empiece a pre-cargar recursos mientras el servidor prepara una respuesta.

B. 2XX: EXITOSO

CÓDIGO	SIGNIFICADO
200 OK	La solicitud ha tenido éxito. El significado de un éxito varía dependiendo del método HTTP
201 Created	La solicitud ha tenido éxito y se ha creado un nuevo recurso como resultado de ello. Ésta es típicamente la respuesta enviada después de una petición PUT.
202 Accepted	La solicitud se ha recibido, pero aún no se ha actuado. Es una petición "sin compromiso", lo que significa que no hay manera en HTTP que permite enviar una respuesta asíncrona que indique el resultado del procesamiento de la solicitud. Está pensado para los casos en que otro proceso o servidor maneja la solicitud, o para el procesamiento por lotes.

203 Non-Authoritative Information	La petición se ha completado con éxito, pero su contenido no se ha obtenido de la fuente originalmente solicitada, sino que se recoge de una copia local o de un tercero. Excepto esta condición, se debe preferir una respuesta de 200 OK en lugar de esta respuesta.
204 No Content	La petición se ha completado con éxito, pero su respuesta no tiene ningún contenido, aunque los encabezados pueden ser útiles. El agente de usuario puede actualizar sus encabezados en caché para este recurso con los nuevos valores.
205 Reset Content	La petición se ha completado con éxito, pero su respuesta no tiene contenidos y, además, el agente de usuario tiene que inicializar la página desde la que se realizó la petición, este código es útil por ejemplo para páginas con formularios cuyo contenido debe borrarse después de que el usuario lo envíe.
206 Partial Content	La petición servirá parcialmente el contenido solicitado. Esta característica es utilizada por herramientas de descarga como wget para continuar la transferencia de descargas anteriormente interrumpidas, o para dividir una descarga y procesar las partes simultáneamente.
207 Multi-Status	Una respuesta Multi-Estado transmite información sobre varios recursos en situaciones en las que varios códigos de estado podrían ser apropiados. El cuerpo de la petición es un mensaje XML.
208 Already Reported	El listado de elementos DAV ya se notificó previamente, por lo que no se van a volver a listar.
226 IM Used	El servidor ha cumplido una petición GET para el recurso y la respuesta es una representación del resultado de una o más manipulaciones de instancia aplicadas a la instancia actual.

C. 3XX: REDIRECCIONES

CÓDIGO	SIGNIFICADO
300 Multiple Choice	Esta solicitud tiene más de una posible respuesta. User-Agent o el usuario debe escoger uno de ellos. No hay forma estandarizada de seleccionar una de las respuestas.

301 Moved Permanently	Este código de respuesta significa que la URI del recurso solicitado ha sido cambiado. Probablemente una nueva URI sea devuelta en la respuesta.
302 Found	Este código de respuesta significa que el recurso de la URI solicitada ha sido cambiado temporalmente. Nuevos cambios en la URI serán agregados en el futuro. Por lo tanto, la misma URI debe ser usada por el cliente en futuras solicitudes.
303 See Other	El servidor envía esta respuesta para dirigir al cliente a un nuevo recurso solicitado a otra dirección usando una petición GET.
304 Not Modified	Esta es usada para propósitos de "caché". Le indica al cliente que la respuesta no ha sido modificada. Entonces, el cliente puede continuar usando la misma versión almacenada en su caché.
307 Temporary Redirect	El servidor envía esta respuesta para dirigir al cliente a obtener el recurso solicitado a otra URI con el mismo método que se usó la petición anterior. Tiene la misma semántica que el código de respuesta HTTP 302 Found, con la excepción de que el agente usuario no debe cambiar el método HTTP usado: si un POST fue usado en la primera petición, otro POST debe ser usado en la segunda petición.
308 Permanent Redirect	Significa que el recurso ahora se encuentra permanentemente en otra URI, especificada por la respuesta de encabezado HTTP Location. Tiene la misma semántica que el código de respuesta HTTP 301 Moved Permanently, con la excepción de que el agente usuario no debe cambiar el método HTTP usado: si un POST fue usado en la primera petición, otro POST debe ser usado en la segunda petición.

D. 4XX: ERRORES DE CLIENTE

CÓDIGO	SIGNIFICADO
400 Bad Request	Esta respuesta significa que el servidor no pudo interpretar la solicitud dada una sintaxis inválida.
401 Unauthorized	Es necesario autenticar para obtener la respuesta solicitada. Esta es similar a 403, pero en este caso, la autenticación es posible.
402 Payment Required	Este código de respuesta está reservado para futuros usos. El objetivo inicial de crear este código fue para

	ser utilizado en sistemas digitales de pagos. Sin embargo, no está siendo usado actualmente.
403 Forbidden	El cliente no posee los permisos necesarios para cierto contenido, por lo que el servidor está rechazando otorgar una respuesta apropiada.
404 Not Found	El servidor no pudo encontrar el contenido solicitado. Este código de respuesta es uno de los más famosos dada su alta ocurrencia en la web.
405 Method Not Allowed	El método solicitado es conocido por el servidor pero ha sido deshabilitado y no puede ser utilizado. Los dos métodos obligatorios, GET y HEAD, nunca deben ser deshabilitados y no deberían retornar este código de error.
406 Not Acceptable	Esta respuesta es enviada cuando el servidor, después de aplicar una negociación de contenido servidor-impulsado (en-US), no encuentra ningún contenido seguido por la criteria dada por el usuario.
407 Proxy Authentication Required	Esto es similar al código 401, pero la autenticación debe estar hecha a partir de un proxy.
408 Request Timeout	Esta respuesta es enviada en una conexión inactiva en algunos servidores, incluso sin alguna petición previa por el cliente. Significa que el servidor quiere desconectar esta conexión sin usar. Esta respuesta es muy usada desde algunos navegadores, como Chrome, Firefox 27+, o IE9, usa mecanismos de pre-conexión HTTP para acelerar la navegación. También hay que tener en cuenta que algunos servidores simplemente desconectan la conexión sin enviar este mensaje.
409 Conflict	Esta respuesta puede ser enviada cuando una petición tiene conflicto con el estado actual del servidor.
410 Gone	Esta respuesta puede ser enviada cuando el contenido solicitado ha sido borrado del servidor.
411 Length Required	El servidor rechaza la petición porque el campo de encabezado Content-Length no está definido y el servidor lo requiere.
412 Precondition Failed	El cliente ha indicado pre-condiciones en sus encabezados la cual el servidor no cumple.
413 Payload Too Large	La entidad de petición es más larga que los límites definidos por el servidor; el servidor puede cerrar la

	conexión o retornar un campo de encabezado Retry-After.
414 URI Too Long	La URI solicitada por el cliente es más larga de lo que el servidor está dispuesto a interpretar.
415 Unsupported Media Type	El formato multimedia de los datos solicitados no está soportado por el servidor, por lo cual el servidor rechaza la solicitud.
416 Requested Range Not Satisfiable	El rango especificado por el campo de encabezado Range en la solicitud no cumple; es posible que el rango está fuera del tamaño de los datos objetivo del URI.
417 Expectation Failed	Significa que la expectativa indicada por el campo de encabezado Expect solicitada no puede ser cumplida por el servidor.
418 I'm a teapot	El servidor se rehúsa a intentar hacer café con una tetera.
421 Misdirected Request	La petición fue dirigida a un servidor que no es capaz de producir una respuesta. Esto puede ser enviado por un servidor que no está configurado para producir respuestas por la combinación del esquema y la autoridad que están incluidos en la URI solicitada
422 Unprocessable Entity	La petición estaba bien formada pero no se pudo seguir debido a errores de semántica.
423 Locked	El recurso que está siendo accedido está bloqueado.
424 Failed Dependency	La petición falló debido a una falla de una petición previa.
426 Upgrade Required	El servidor se rehúsa a aplicar la solicitud usando el protocolo actual pero puede estar dispuesto a hacerlo después que el cliente se actualice a un protocolo diferente. El servidor envía un encabezado Upgrade en una respuesta para indicar los protocolos requeridos.
428 Precondition Required	El servidor origen requiere que la solicitud sea condicional. Tiene la intención de prevenir problemas de 'actualización perdida', donde un cliente OBTIENE un estado del recurso, lo modifica, y lo PONE devuelta al servidor, cuando mientras un tercero ha modificado el estado del servidor, llevando a un conflicto.
429 Too Many Request	El usuario ha enviado demasiadas solicitudes en un periodo de tiempo dado.

431 Request Header Fields Too Large	El servidor no está dispuesto a procesar la solicitud porque los campos de encabezado son demasiado largos. La solicitud PUEDE volver a subirse después de reducir el tamaño de los campos de encabezado solicitados.
451 Unavailable For Legal Reasons	El usuario solicita un recurso ilegal, como alguna página web censurada por algún gobierno.

E. 5XX: ERRORES DE SERVIDOR

CÓDIGO	SIGNIFICADO
500 Internal Server Error	El servidor ha encontrado una situación que no sabe cómo manejarla.
501 Not Implemented	El método solicitado no está soportado por el servidor y no puede ser manejado. Los únicos métodos que los servidores requieren soporte (y por lo tanto no deben retornar este código) son GET y HEAD.
502 Bad Gateway	Esta respuesta de error significa que el servidor, mientras trabaja como una puerta de enlace para obtener una respuesta necesaria para manejar la petición, obtuvo una respuesta inválida.
503 Service Unavailable	El servidor no está listo para manejar la petición. Causas comunes puede ser que el servidor está caído por mantenimiento o está sobrecargado. Hay que tomar en cuenta que junto con esta respuesta, una página usuario-amigable explicando el problema debe ser enviada. Estas respuestas deben ser usadas para condiciones temporales y el encabezado HTTP Retry-After: debería, si es posible, contener el tiempo estimado antes de la recuperación del servicio. El webmaster debe también cuidar los encabezados relacionados al caché que son enviados junto a esta respuesta, ya que estas respuestas de condición temporal deben usualmente no estar en el caché.
504 Gateway Timeout	Esta respuesta de error es dada cuando el servidor está actuando como una puerta de enlace y no puede obtener una respuesta a tiempo.
505 HTTP Version Not Supported	La versión de HTTP usada en la petición no está soportada por el servidor.
506 Variant Also Negotiates	El servidor tiene un error de configuración interna: negociación de contenido transparente para la petición resulta en una referencia circular.

507 Insufficient Storage	El servidor tiene un error de configuración interna: la variable de recurso escogida está configurada para acoplar la negociación de contenido transparente misma, y no es por lo tanto un punto final adecuado para el proceso de negociación.
508 Loop Detected	El servidor detectó un ciclo infinito mientras procesaba la solicitud.
510 Not Extended	Extensiones adicionales para la solicitud son requeridas para que el servidor las cumpla.
511 Network Authentication Required	El código de estado 511 indica que el cliente necesita autenticar para obtener acceso a la red.

USER-AGENT

El User-Agent (UA) request header es una cadena característica que le permite a los servidores y servicios de red identificar la aplicación, sistema operativo, compañía, y/o la versión del user agent que hace la petición.

A. SINTAXIS

```
User-Agent: <producto> / <producto-version> <comentario>
```

- <producto> Un identificador de producto — su nombre o nombre clave de desarrollo.
- <producto-version> Número de versión de producto.
- <comentario> Cero o más comentarios conteniendo más detalles. Por ejemplo, información del subproducto.

El formato común para los navegadores es el siguiente:

```
User-Agent: Mozilla/5.0 (<información-de-sistema>) <plataforma> (<plataforma-detalles>)<extensiones>
```

B. CADENA UA DE FIREFOX

```
Mozilla/5.0 (platform; rv:geckoversion) Gecko/geckotrail Firefox/firefoxversion
```

Se divide en 5 componentes:

1. Mozilla/5.0 es el token general que indica que el navegador es compatible con Mozilla. Por razones históricas, prácticamente todos los navegadores actuales lo envían.
2. platform describe la plataforma nativa en la que está corriendo el navegador (Windows, Mac, Linux, Android, etc.) y si es un teléfono

móvil. Los teléfonos con Firefox OS (en-US) dicen Mobile — la web es la plataforma. Nótese que platform puede consistir de múltiples tokens separados por ";". Léase debajo para más detalles y ejemplos.

3. rv:geckoversion indica la versión de Gecko (como "17.0"). En navegadores recientes, geckoversion es lo mismo que firefoxversion.
4. Gecko/geckotrail indica que el navegador está basado en Gecko. (En el escritorio, geckotrail siempre es la cadena fija 20100101.)
5. Firefox/firefoxversion indica que el navegador es Firefox y provee la versión (como "17.0").

```
Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:47.0) Gecko/20100101 Firefox/47.0
Mozilla/5.0 (Macintosh; Intel Mac OS X x.y; rv:42.0) Gecko/20100101 Firefox/42.0
```

C. CADENA UA DE CHROME

La cadena de agente de usuario para Chrome (o motores basados en Chromium/Blink) es similar a la de Firefox. Por compatibilidad, añade cadenas como KHTML, like Gecko y Safari.

```
Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/51.0.2704.103 Safari/537.36
```

D. CADENA UA DE OPERA

El navegador Opera también está basado en el motor Blink, que es la razón por la cual luce exactamente como la cadena UA de Chrome, pero añade "OPR/<version>".

```
Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/51.0.2704.106 Safari/537.36 OPR/38.0.2220.41
```

E. CADENA UA DE SAFARI

En este ejemplo, la cadena de agente de usuario es la versión móvil de Safari. Contiene la palabra "Mobile".

```
Mozilla/5.0 (iPhone; CPU iPhone OS 13_5_1 like Mac OS X) AppleWebKit/605.1.15
(KHTML, like Gecko) Version/13.1.1 Mobile/15E148 Safari/604.1
```

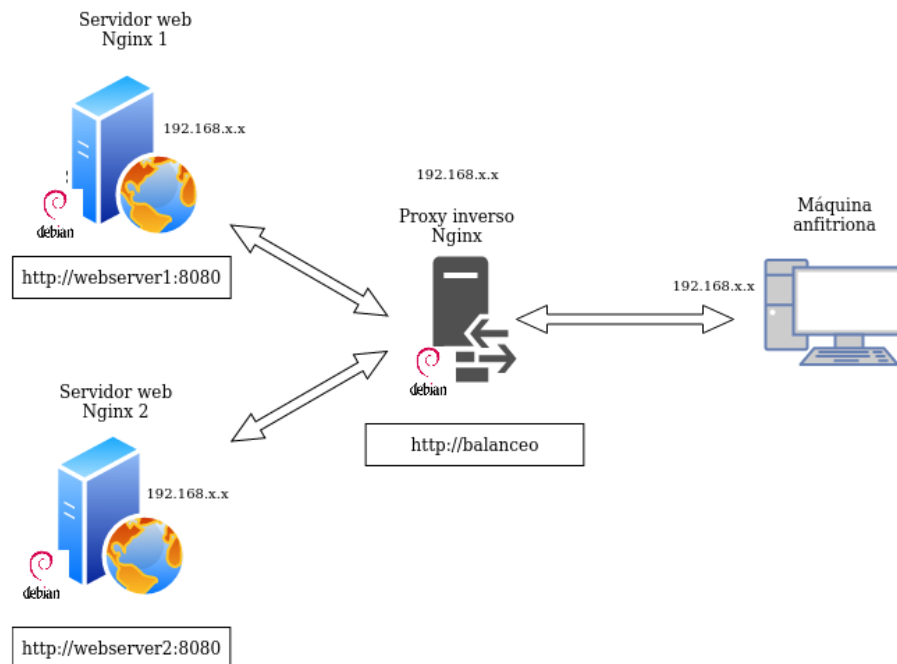
F. CADENA UA DE MICROSOFT EDGE

El navegador Edge también está basado en el motor Blink. Añade "Edg/<version>".

```
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/91.0.4472.124 Safari/537.36 Edg/91.0.864.59
```

OBSERVANDO LOGS

En este esquema de servidores web con balanceo y proxy inverso:



Partimos de estas configuraciones de red:

```
mortadelo@servidor:~$ 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 noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,DYNAMIC,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:8b:f5:a1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.106/24 brd 192.168.56.255 scope global enp0s3
        valid_lft forever preferred_lft forever

mortadelo@servidor2:~$ 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 noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,DYNAMIC,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:a7:46:ca brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.110/24 brd 192.168.56.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fea7:46ca/64 scope link
        valid_lft forever preferred_lft forever

mortadelo@proxyINV:~$ 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 noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,DYNAMIC,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:03:cc:07 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.107/24 brd 192.168.56.255 scope global enp0s3
        valid_lft forever preferred_lft forever
```

El cliente http utiliza esta configuración:

```
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: enp5s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
   link/ether e8:11:32:1a:a1:8e brd ff:ff:ff:ff:ff:ff
3: wlp3s0b1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
   link/ether 4c:ed:de:91:59:7a brd ff:ff:ff:ff:ff:ff
   inet 192.168.0.10/24 brd 192.168.0.255 scope global dynamic noprefixroute wlp3s0b1
       valid_lft 80779sec preferred_lft 80779sec
   inet6 fe80::b3b2:648d:9622:8bcc/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
4: vboxnet0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 0a:00:27:00:00:00 brd ff:ff:ff:ff:ff:ff
   inet 192.168.56.1/24 brd 192.168.56.255 scope global vboxnet0
       valid_lft forever preferred_lft forever
   inet6 fe80::800:27ff:fe00:0/64 scope link
       valid_lft forever preferred_lft forever
```

1. Revisamos los registros de acceso /var/log/nginx/access.log

```
192.168.56.1 - - [03/Nov/2023:07:22:45 +0100] "GET / HTTP/1.1" 200 169 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36"
192.168.56.1 - - [03/Nov/2023:07:22:45 +0100] "GET /favicon.ico HTTP/1.1" 404 187 "http://balanceo/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36"
192.168.56.1 - - [03/Nov/2023:07:22:55 +0100] "GET / HTTP/1.1" 304 0 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36"
192.168.56.1 - - [03/Nov/2023:07:22:57 +0100] "GET / HTTP/1.1" 499 0 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36"
```

- ¿En qué servidor estoy viendo este log?

El acceso al log se ha realizado en el proxy ya que la petición entrante proviene de la IP de la máquina que tiene el rol de cliente (192.168.56.1).

- ¿Qué cliente web se ha utilizado para realizar la conexión?

Se está usando el navegador Chrome porque el user-agent coincide con el usado en Chrome ya que añade al final las cadenas propias de Chrome y por compatibilidad las de Safari.

2. Revisamos los registros de acceso /var/log/nginx/access.log

```
192.168.56.107 - - [03/Nov/2023:07:43:19 +0100] "GET / HTTP/1.0" 200 213 "-" "Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MR58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Mobile Safari/537.36"
192.168.56.107 - - [03/Nov/2023:07:43:27 +0100] "GET / HTTP/1.0" 200 213 "-" "Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MR58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Mobile Safari/537.36"
```

- ¿En qué servidor estoy viendo este log?

El acceso al log se está haciendo desde uno de los servidores porque la IP que vemos en cada una de las entradas del log (192.168.56.107) coincide con la IP del proxy.

No podemos saber exactamente en cuál de los servidores estamos viendo la información.

- ¿Qué cliente web se ha utilizado para realizar la conexión?

Se está usando el navegador Chrome en su versión móvil porque el user-agent coincide con el usado en Chrome ya que añade al final las cadenas propias de Chrome y por compatibilidad las de Safari.

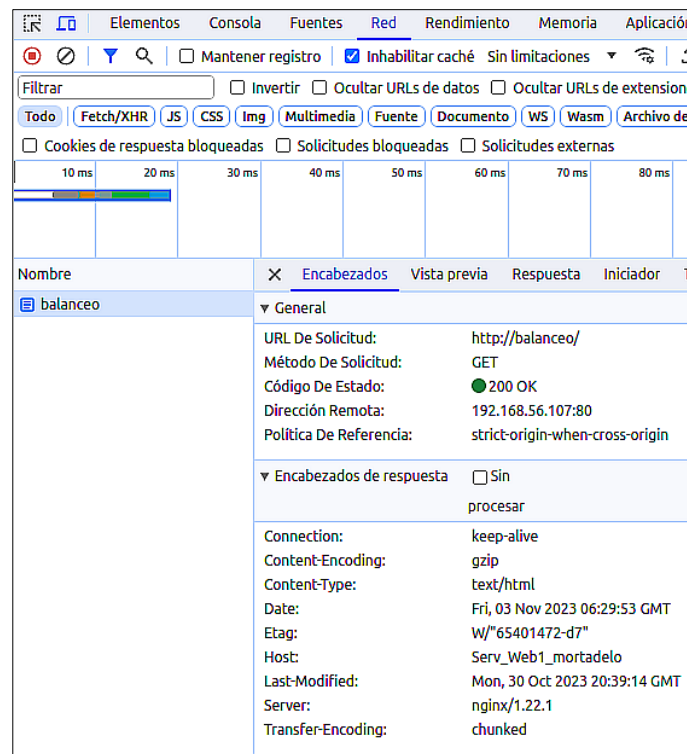
Además, sabemos que es la versión móvil porque en la cadena de Safari vemos la modificación "Mobile".

3. ¿Y en este caso?

```
192.168.56.1 - - [03/Nov/2023:07:43:19 +0100] "GET / HTTP/1.1" 499 0 "-" "Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Mobile Safari/537.36"
192.168.56.1 - - [03/Nov/2023:07:43:19 +0100] "GET / HTTP/1.1" 200 169 "-" "Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Mobile Safari/537.36"
192.168.56.1 - - [03/Nov/2023:07:43:27 +0100] "GET / HTTP/1.1" 200 169 "-" "Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Mobile Safari/537.36"
192.168.56.1 - - [03/Nov/2023:07:44:49 +0100] "GET / HTTP/1.1" 200 169 "-" "Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Mobile Safari/537.36"
```

Se está realizando un acceso al access.log del proxy porque la IP de las peticiones es la del cliente (192.168.56.1). En este caso se está usando el navegador Chrome.

Observando la siguiente información en el navegador:



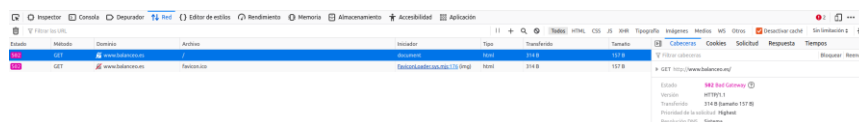
- ¿A qué servidor hice la petición http?

Se hizo al servidor proxy porque la dirección remota del bloque "General" es la 192.168.56.107.

- ¿Qué servidor web me está respondiendo?

Está respondiendo el servidor 1 porque la cabecera "Host" del bloque "Encabezados de respuesta" nos devuelve el valor Serv_Web1_mortadelo.

- Consigue que te responda con un código 500 desde el servidor. Comenta qué has hecho para conseguirlo.



Para conseguir el error he paralizado los servicios NGINX tanto en el servidor 1 como en el servidor 2 (`sudo systemctl stop nginx.service`), de esta manera, aunque la conexión con el proxy se completa no está recibiendo una respuesta correcta desde el otro servidor.

¿Y en este otro caso?

✕	Encabezados	Vista previa	Respuesta	Iniciador	Tier
▼ General					
URL De Solicitud:		http://balanceo/			
Método De Solicitud:		GET			
Código De Estado:		● 200 OK			
Dirección Remota:		192.168.56.107:80			
Política De Referencia:		strict-origin-when-cross-origin			
▼ Encabezados de respuesta		<input type="checkbox"/> Sin procesar			
Connection:		keep-alive			
Content-Encoding:		gzip			
Content-Type:		text/html			
Date:		Fri, 03 Nov 2023 06:44:49 GMT			
Etag:		W/"654017d8-d5"			
Host:		Serv_Web2_doraemon			
Last-Modified:		Mon, 30 Oct 2023 20:53:44 GMT			
Server:		nginx/1.22.1			
Transfer-Encoding:		chunked			

En este caso se hizo una petición al proxy por la misma razón que en el anterior, la dirección remota coincide con la IP del proxy 192.168.56.107 y la respuesta está llegando desde el servidor 2 porque en este caso la cabecera “Host” devuelve el valor `Serv_Web2_doraemon`.

Tras configurar el servidor proxy para acceso con SSL, tras realizar un par de accesos desde un cliente captura un par de líneas en el fichero de log con los accesos, pégalo aquí y comenta la información que puedes extraer de lo que estás observando:

```
albertom-servidor@debian-deaw:~$ sudo tail /var/log/nginx/https_access.log
10.0.2.6 - - [03/Nov/2023:17:44:12 +0100] "GET / HTTP/1.1" 200 257 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/119.0"
10.0.2.6 - - [03/Nov/2023:17:44:13 +0100] "GET / HTTP/1.1" 200 258 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/119.0"
```

Se ha producido un doble acceso desde la máquina cliente (IP 10.0.2.6) el 3 de noviembre de 2023 a las 17:44:12 y a las 17:44:13 el cual ha realizado solicitudes de tipo GET utilizando protocolo HTTP/1.1 al servidor proxy.

Estas solicitudes han recibido el código 200 (OK – implica que la solicitud ha sido procesada de forma correcta) y se han realizado desde un navegador Firefox (versión 119.0).

Así mismo se puede saber que se está utilizando una distro de Linux (Ubuntu para ser exactos) con arquitectura x86_64.