



## **RED IP DE TELEFÓNICA DE ESPAÑA GUÍA DE PROVEEDOR DE LA FUNCIONALIDAD DE PROXY-CACHÉ**

# **RED IP DE TELEÓNICA DE ESPAÑA: GUÍA DE PROVEEDOR DE LA FUNCIONALIDAD DE PROXY-CACHÉ**

## **ÍNDICE**

<b>1.</b>	<b>INTRODUCCIÓN .....</b>	<b>3</b>
<b>2.</b>	<b>CARACTERÍSTICAS DE UNA CACHÉ.....</b>	<b>4</b>
2.1	¿CÓMO FUNCIONA UNA CACHÉ?.....	4
2.2	¿CÓMO FUNCIONA UN PROXY-CACHÉ TRANSPARENTE? .....	5
2.3	¿CÓMO CONTROLAR LA ACTUALIZACIÓN DE CONTENIDOS EN EL PROVEEDOR DE CONTENIDOS? .....	6
2.4	CONSEJOS PARA EL PROVEEDOR DE CONTENIDOS PARA APROVECHAR EL EFECTO DE LAS CACHÉS .	10
<b>3.</b>	<b>CÓMO OBTENER INFORMACIÓN DE LOS USUARIOS REALES DE LOS PROXY-CACHÉS .....</b>	<b>12</b>
<b>4.</b>	<b>PREGUNTAS Y RESPUESTAS MÁS FRECUENTES.....</b>	<b>13</b>
<b>5.</b>	<b>EJEMPLOS DE CONFIGURACIÓN EN SERVIDORES WEB.....</b>	<b>16</b>
5.1	APACHE 1.3 .....	16
5.2	NETSCAPE ENTERPRISE 3.6 .....	17
5.3	MICROSOFT IIS 4.0 .....	18
<b>6.</b>	<b>REFERENCIAS.....</b>	<b>19</b>
6.1	PROTOCOLO HTTP 1.1 .....	19
6.2	CACHING TUTORIAL FOR WEB AUTHORS AND WEBMASTERS.....	19
6.3	INTRODUCCIÓN A LAS CACHÉS WEB .....	19
6.4	CACHEABILITY ENGINE .....	19

## 1. INTRODUCCIÓN

Telefónica de España dispone en su red de acceso a Internet de la funcionalidad de proxy-caché transparente. Sobre esta red, se presta el servicio de acceso ADSL de Telefónica y de otros proveedores.

Las soluciones de proxy-caché proporcionan ventajas, tanto para los usuarios finales conectados a la red, como a los proveedores de contenidos para dichos usuarios. Algunas de estas ventajas son:

- Mejora del tiempo de descarga de páginas a los usuarios. Ya que muchos de los objetos web que componen las páginas web se pueden servir localmente desde los servidores de proxy-caché, se evitan las latencias típicas de Internet.
- Reducción de recursos en la infraestructura del proveedor de contenidos, ya que sus servidores tienen que atender a menos peticiones.

Si bien el uso de servidores proxy-caché está muy extendido en los operadores de telecomunicaciones y en las empresas, es una tecnología a veces no lo suficientemente conocida y aprovechada.

El propósito de esta guía es explicar cuál es el funcionamiento y las características de la solución de caché transparente disponible en la red IP de Telefónica de España y cómo los proveedores de contenido pueden sacar el máximo provecho de sus ventajas.

Con este objetivo, en el apartado 2 se describe cuál es el funcionamiento básico de una caché, almacenando copias temporales de los contenidos cerca de los usuarios finales para mejorar el rendimiento. También se describe cómo los proveedores de contenido pueden controlar qué contenidos y por cuánto tiempo se almacenan en la caché, de manera que no pierdan el control sobre qué es lo que los usuarios finales están obteniendo. Para ello se utilizan los mecanismos estándar de http.

En el apartado 3 se describe cómo los proveedores de contenidos pueden implementar mecanismos eficaces para contabilizar el número de usuarios distintos que solicitan sus páginas cuando gran parte de las peticiones pueden venir de los servidores de caché. El uso de pequeños objetos no cacheables y la extracción de las cabeceras http con la dirección IP del cliente introducidas por la caché, permiten realizar esta actividad tan común.

El apartado 4 es una recopilación de las preguntas o dudas más frecuentes respecto a la tecnología de cachés desde el punto de vista de un proveedor de contenidos.

El apartado 5 incluye una mini guía de configuración de cabeceras http en los servidores web más comunes, y en el apartado 6 se ofrecen algunas referencias utilizadas para la confección de este documento.

## 2. CARACTERÍSTICAS DE UNA CACHÉ

### 2.1 ¿CÓMO FUNCIONA UNA CACHÉ?

El funcionamiento general de una caché consiste en almacenar temporalmente los contenidos que son más frecuentemente utilizados en cualquier dispositivo de almacenamiento de ofrezca un acceso más rápido que en el que originalmente se encontraban.

En el caso de Internet se trata de almacenar los contenidos web (html, imágenes, javascript, etc.), en adelante *objetos*, lo más cerca posible del usuario final. El caso más común que todos usamos es la caché de los navegadores web. Esta caché almacena objetos web en nuestro disco duro de forma que, por ejemplo, cuando utilizamos el botón atrás, no tenga que descargar de nuevo de la red todos los objetos de una página ya vista.

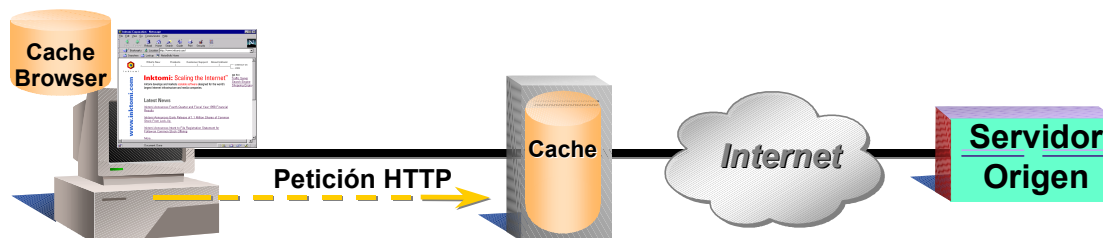
Los servidores proxy-caché utilizados en la red de acceso a Internet de Telefónica, realizan una función similar, pero en lugar de ser una caché privada para cada usuario, son una caché compartida entre un conjunto grande de los usuarios de dicha red.

Al estar compartido el almacenamiento entre muchos usuarios, es más probable que varios usuarios pidan los mismos objetos (los más populares) y por tanto los beneficios sean mayores.

En el caso de la red IP de Telefónica de España, los servidores de caché se encuentran situados lo más cerca posible de los usuarios (en los centros de acceso), de forma que la latencia de las peticiones de estos usuarios se reduce.

#### Escenario de funcionamiento del proxy-caché:

1- El cliente solicita una página web (HTTP) utilizando el puerto 80. Esta petición es recibida por el sistema de proxy-caché.

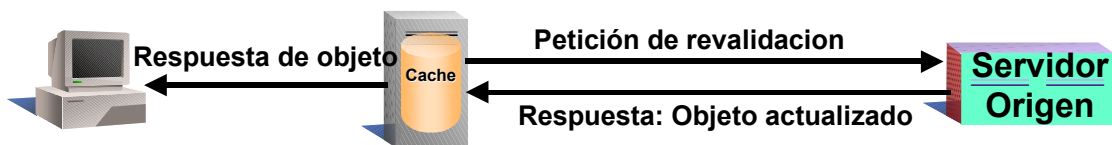


2- El proxy-caché comprueba si el objeto se considera actualizado (y según las políticas descritas anteriormente se considera como “cacheable”).

Si se puede considerar actualizado, devuelve el objeto directamente sin realizar ninguna conexión con el servidor origen (ahorrando por tanto, el tiempo de latencia/transferencia de la página al usuario).



3- Si el objeto se determina como caducado, el proxy-caché realiza la petición al servidor origen para revalidarlo. El servidor origen devuelve el contenido actualizado, que el proxy-caché almacena para posteriores peticiones y sirve el objeto simultáneamente al cliente.



## 2.2 ¿CÓMO FUNCIONA UN PROXY-CACHÉ TRANSPARENTE?

El término proxy hace referencia a una entidad que actúa en representación de otra de cara a utilizar un servicio. En el contexto de Internet, un proxy hace referencia a un servidor al que se conectan los clientes para pedir contenidos webs de Internet, de forma que realmente es este servidor el que los pide a los servidores web origen en representación de los clientes.

Si este servidor proxy además, almacena, copias locales de los contenidos para su posterior reutilización (tal y como se describió en el apartado anterior), es lo que denominamos un proxy-caché.

El ejemplo de proxy más común lo solemos encontrar en las empresas que los colocan en sus Intranets como pasarela hacia Internet. Sin embargo este tipo de proxy suele ser necesario configurarlo explícitamente por el propio usuario en su navegador.

En un entorno de un operador de acceso a Internet como es la red de Telefónica de España la configuración explícita por parte del usuario es inviable y compleja de mantener. Por eso se utilizan los proxies transparentes.

Un proxy-caché transparente se basa en la existencia de unos dispositivos en la red que interceptan las peticiones de los usuarios y las redirigen al servidor de proxy-caché más cercano, sin necesidad de ninguna configuración por parte del cliente.

La identificación de qué peticiones deben ser redirigidas se hace por el puerto destino de las conexiones TCP, que normalmente, permite identificar el protocolo.

En el caso de la red IP de Telefónica de España los protocolos que pueden ser redirigidos transparentemente a las cachés son:

- **Http:** se redirigen a las cachés las peticiones de los usuarios con destino el puerto 80 de TCP. No se redirige cualquier otra petición http que no use el puerto 80. Esto excluye también todo el tráfico HTTPS, que normalmente usa el puerto 443.
- **Streaming de Microsoft Windows Media:** este protocolo de streaming usa el puerto 1755 de TCP y UDP para las conexiones a los servidores. Sólo este puerto es redirigido para este tipo de contenidos.
- **Streaming de Real Networks:** el puerto TCP utilizado es el estándar de RTSP, el 554. Sólo este puerto es redirigido para este tipo de contenidos.
- **Streaming de Apple QuickTime:** en este caso el puerto coincide con el anterior, el 554. Si bien los protocolos no son los mismos pero el puerto sí, la caché averigua automáticamente cuál debe usar. Sólo este puerto es redirigido para este tipo de contenidos.

### 2.3 ¿CÓMO CONTROLAR LA ACTUALIZACIÓN DE CONTENIDOS EN EL PROVEEDOR DE CONTENIDOS?

Debido al desconocimiento existente respecto a las tecnologías de proxy-caché, algunos proveedores de contenido tienen miedo de perder el control sobre como se sirven sus contenidos a los usuarios finales. En particular temiendo que la caché pueda servir contenidos desactualizados a los usuarios.

Sin embargo el protocolo http[RFC2616] dispone de múltiples formas de permitir a los proveedores de contenido controlar que contenidos se almacenan en las cachés y cuales no, y cada cuanto tiempo se actualiza cada objeto en las cachés. Por ejemplo si una página web contiene información personalizada para cada usuario que accede, se puede marcar que el texto cambiante de esa página para que sea no cacheable y que las imágenes incluidas en dicha página (que suelen ser las mismas para todos los usuarios) si sean cacheadas.

De esta forma el proveedor de contenidos puede crear contenidos que sirvan lo más rápido posible aprovechando la caché, pero con garantía de que el usuario no va a recibir contenidos incorrectos o desactualizados.

El proceso que sigue una caché para decidir si un contenido web es cacheable o no es el siguiente:

1. Si el contenido requiere autenticación o se utiliza una conexión segura (HTTPS) el contenido no es cacheable.

2. En general, si la petición http no es de tipo GET o HEAD, no es cacheable y por tanto se piden al servidor origen. Esto incluye por ejemplo todas las peticiones tipo POST.
3. Si la petición http parece una petición de contenido dinámico (incluye en el URL el carácter ? o cgi o .asp), se marca como no cacheable.
4. Se analizan las cabeceras de http en busca de alguna directiva de cacheabilidad del contenido (explicadas posteriormente), si existen se obedecen. De esta forma, se puede marcar el contenido como cacheable o no.
5. Si no existe ninguna cabecera http de cacheabilidad, se analiza la cabecera Last-modified (fecha de última modificación del contenido), si no existe (suele pasar con las peticiones dinámicas) se marca el contenido como no cacheable.
6. Si la fecha de última modificación es anterior en el tiempo a la fecha actual se marca el contenido como cacheable durante un periodo de tiempo calculado usando una heurística. Típicamente la heurística consiste en un porcentaje del periodo de tiempo transcurrido desde que se modificó el contenido hasta el momento que se almacenó el contenido en la caché, de esta forma se adapta automáticamente a la frecuencia de actualización de cada contenido. Esta heurística utiliza unos valores mínimo y máximo de 5 minutos y 1 día respectivamente.

De este modo, si cuando llega una petición de un objeto a la caché, el objeto es cacheable, y la copia del objeto existente en la caché se encuentra dentro del periodo de validez (ya sea definido explícitamente o mediante la heurística), el objeto se considera actualizado y se sirve directamente desde la caché sin conectar con el servidor origen (evitando la latencia asociada a una petición en Internet y ahorrando el ancho de banda correspondiente al tamaño del objeto).

Si por el contrario el objeto web es cacheable pero se encuentra fuera de su periodo de validez, se considera que el objeto ha expirado de la caché. En este caso la caché conecta con el servidor web origen y realiza una validación para saber si el objeto web ha cambiado desde que lo almacenó en la caché. Para ello existe un tipo de peticiones estándar en http que en caso de que el contenido no haya cambiado tan sólo devuelven un código de retorno (ahorrando la transmisión completa del objeto), y el objeto completo en caso contrario.

### **Ejemplo de funcionamiento de la heurística:**

El cliente solicita un objeto web (HTTP) que no está previamente almacenado en la caché debido a una petición de un usuario anterior utilizando el puerto 80. Esta petición es recibida por el sistema de proxy-caché y se determina que es un objeto cacheable, pero no dispone de cabeceras explícitas de tiempo de validez en la caché (cabeceras http de cacheabilidad). A partir de este momento se pueden dar varias situaciones por ejemplo:

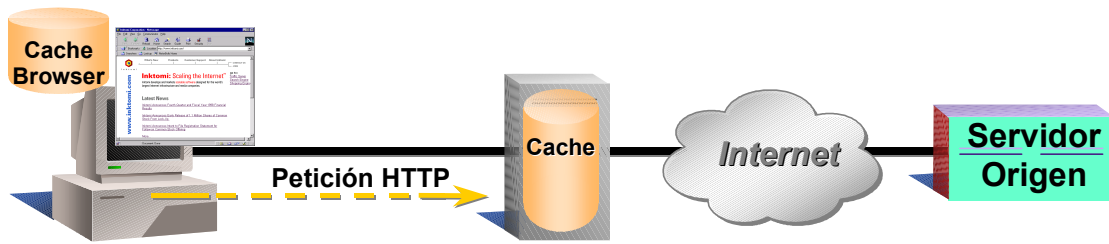


Figura 1: petición de un objeto web al proxy-caché

**A-** Si la fecha de última modificación del objeto es por ejemplo una semana, la caché almacenará el objeto y lo considerará actualizado durante el periodo máximo de validez del procedimiento heurístico, es decir 24 horas. Esto es así porque se considera que este objeto cambia muy raramente y por tanto es razonable almacenarlo por ese periodo de tiempo. Por tanto durante 24 horas, las siguientes peticiones se servirán directamente desde el proxy-caché (ver figura 2). A las 24 horas se realizará una comprobación de validez del contenido (ver figura 3)

**B-** Por el contrario, si el objeto ha sido modificado por última vez hace quince minutos. Se considera que este objeto cambia muy frecuentemente y se almacenará el objeto por el periodo mínimo del procedimiento heurístico, es decir 5 minutos. Durante cinco minutos las peticiones de los clientes se servirán directamente desde el proxy-caché y a los 5 minutos se realizará una consulta al servidor web sobre si el objeto web ha cambiado. Si ha cambiado se almacenará y devolverá el nuevo objeto. Si no ha cambiado permanecerá el antiguo aumentándose ligeramente el tiempo de validez en el proxy-caché (ya que ha pasado más tiempo desde la fecha de última modificación)



Figura 2: objeto actualizado servido desde el proxy-caché

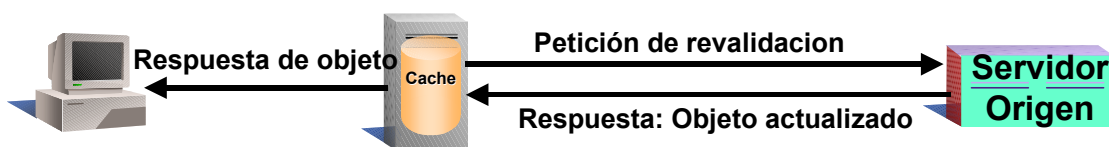


Figura 3 objeto desactualizado, comprobación desde el proxy-caché

Por tanto el tiempo de estancia de un objeto en el proxy-caché como actualizado, no es un valor fijo y constante para cada objeto. Si no que varía a lo largo del tiempo según aumente o disminuya el tiempo desde la fecha de última modificación

Dentro de una página web hay múltiples objetos (imágenes, texto html, javascript,...), siendo el caso más típico que cuando se pide una página web completa, algunos se marquen como cacheables y actualizados y se sirvan desde la caché mientras que otros se pidan y se sirvan desde el servidor origen. Cuantos más objetos y de mayor volumen se puedan servir desde la caché, mayor será el beneficio observado por los clientes, y por tanto, el usuario tendrá una mejor impresión sobre la velocidad de navegación por el servidor web correspondiente.

El estándar http define el uso de cabeceras http para controlar cómo las cachés (tanto las de los navegadores como los proxy-cachés) van a manejar los distintos contenidos. Las cabeceras http no son visibles en las páginas HTML, sino que son generadas automáticamente por los servidores web en las respuestas o por los navegadores en las peticiones. Normalmente todos los servidores web permiten controlar que cabeceras quieres incluir para cada contenido en el servidor web. Ejemplos de cómo configurar estas cabeceras en los servidores web más comunes se incluyen en el apartado 5.

Es importante resaltar que las cabeceras sobre cacheabilidad que se pueden introducir en las páginas HTML (por ejemplo: `<META HTTP-EQUIV="Pragma" CONTENT="no-cache">`) no son interpretadas por los proxy-caches. Esto es así porque los proxies-caches no se detienen a analizar el contenido de las páginas HTML que almacenan y por tanto no “ven” estas cabeceras de HTML. Estas cabeceras en HTML están pensadas para controlar la caché existente en los navegadores.

**Pragma: no-caché:** esta cabecera puede ser incluida por el navegador web en la petición http para forzar que la caché contacte el servidor origen para servir el contenido más actualizado posible. Es incluida automáticamente por el navegador al forzar un refresco de la página web. La especificación http no describe qué ocurre cuando se incluye esta cabecera en la respuesta, así que no se recomienda usar esta cabecera en los servidores web.

**Expires: Fri, 30 Oct 1998 14:19:41 GMT:** con esta cabecera se puede definir exactamente cuando queremos que caduque cada objeto que se almacene en la caché. Es muy útil para marcar como cacheables imágenes que cambien muy de vez en cuando o contenidos que cambien en unos momentos específicos en el tiempo.

**Cache-Control:** Aunque la cabecera Expires es útil, en algunas circunstancias no permite suficiente control sobre la cacheabilidad. Por ello HTTP 1.1 introdujo esta cabecera que puede tomar los siguientes valores:

- **max-age**=[segundos] – especifica el número máximo de segundos que un objeto se considerará actualizado en una caché desde que se pidió el objeto por primera vez.
- **s-maxage**=[segundos] – similar a max-age pero solo aplica a proxy-cachés no a la caché del navegador
- **public** – marca un contenido como cacheable incluso si la política normal de la caché lo marcaría como no cacheable. SE puede usar por ejemplo para marcar contenido autenticado como cacheable.

- **no-cache** – Fuerza a las cachés (tanto del navegador como del proxy-caché) a confirmar la vigencia del contenido en el servidor origen.
- **must-revalidate** – Fuerza a la caché a obedecer cualquier indicación sobre el tiempo de validez proporcionada. La especificación http permite a las cachés cierto margen al definir este periodo, con esta directiva las cachés deben obedecer lo especificado en las cabeceras.
- **proxy-revalidate** - similar a must-revalidate pero sólo aplica a los proxy-cachés.

El uso de esta cabecera se encuentra extensamente documentado en el RFC de http 1.1.

## 2.4 CONSEJOS PARA EL PROVEEDOR DE CONTENIDOS PARA APROVECHAR EL EFECTO DE LAS CACHÉS

Como complemento al uso de cabeceras http para marcar los contenidos cacheables, hay una serie de recomendaciones para aprovechar al máximo los beneficios de las cachés:

- **Nombrar a los objetos de forma consistente:** ésta es la regla de oro para mejorar la cacheabilidad de un servidor web. Siempre que se utilice un mismo contenido en distintas partes de un web, se deben referir a ese objeto usando exactamente la misma URL. De esta manera se garantiza la máxima reusabilidad de ese objeto dentro de los sistemas de caché.
- **Usar una librería de imágenes comunes** y otros elementos (javascripts,...), y referirse a ellos desde diferentes sitios.
- **Marcar como cacheables las imágenes y los objetos que no cambien frecuentemente** mediante el uso de la cabecera Expires.
- **Si un objeto cambia (especialmente ficheros descargables o voluminosos) cambia, cambia su nombre.** De esa forma se puede definir un periodo de validez largo en la caché para ese objeto, y asegurar que se sirve el contenido correcto. Tan sólo la página que contiene el enlace a ese objeto debe cambiar por lo tanto ser definida para que expire frecuentemente. Esto no implica que obligatoriamente se tenga que cambiar el nombre de todos los ficheros cuando cambian, tan sólo es una forma eficiente de actualizar inmediatamente un objeto que pueda tener un periodo de expiración largo y sea necesario que se actualice inmediatamente.
- **No modificar objetos innecesariamente.** Al hacerlo se modifica la fecha de última modificación aunque el contenido no haya cambiado.
- **Sólo usar cookies cuando se necesiten.** Por defecto las páginas html con cookies no se cachean. Se recomienda utilizar las cookies para usarlas sólo en páginas dinámicas.
- **Minimizar el uso de SSL.** El contenido seguro no es cacheable, así que sólo debería usarse cuando sea realmente necesario.

Si se desea comprobar la cacheabilidad de cualquier página web y saber que objetos serán cacheados, cuales no y por qué, existe una utilidad online de uso público en Internet llamada Cacheability Engine (<http://www.mnot.net/cacheability/>). Basta con proporcionar el URL de la página analizar para que genere un informe de cacheabilidad de todos los objetos contenidos en ella.

Ejemplo de información de cacheabilidad para un objeto obtenida con la utilidad Cacheability Engine:

- [http://www.unapagina.es/images/una\\_imagen.gif](http://www.unapagina.es/images/una_imagen.gif)

Date	Tue, 07 May 2002 23:27:28 GMT
Expires	-
Cache-Control	-
Last-Modified	16 weeks 4 days ago (Fri, 11 Jan 2002 09:17:14 GMT) <b>validated</b>
ETag	"089bdc1809ac11:8f0"
Content-Length	0.4K (406)
Server	Microsoft-IIS/5.0

- This object doesn't have any explicit freshness information set, so a cache may use Last-Modified to determine how fresh it is with an adaptive TTL (at this time, it could be, depending on the adaptive percent used, considered fresh for: 3 weeks 2 days (20%), 8 weeks 2 days (50%), 16 weeks 4 days (100%)). It can be validated with Last-Modified.

### 3. CÓMO OBTENER INFORMACIÓN DE LOS USUARIOS REALES DE LOS PROXY-CACHÉS

Una característica general de los proxy-cachés es que son ellos los encargados de realizar la conexión a los servidores origen para pedir los contenidos, en representación de todos los clientes que dependen de ellos. Al hacer estas peticiones a los servidores web la dirección IP origen que utilizan es la suya propia, tal y como marcan las reglas generales de TCP/IP. Al acceder a los servidores web los proxy-cachés vuelven a resolver por DNS el nombre del servidor. Por tanto es necesario que los nombres que utilizemos para acceder a nuestros servidores web estén dados de alta en los DNS públicos.

Debido a que las peticiones se realizan con las IPs de los proxy-cachés, el registro de la IP origen en los ficheros históricos de los servidores web no permite conocer la IP del usuario final que solicitó dicha página.

En general muchos websites son conscientes de esta situación y utilizan mecanismos como las cookies para poder contabilizar usuarios únicos. Debido a que las cookies se almacenan en cada navegador proporcionan una forma más fiable de identificación de usuarios únicos. Hay que tener en cuenta que el uso de servidores proxy-caché está muy extendido, especialmente en el entorno empresarial.

Si aun así es necesario el registro de la IP origen para la realización de, por ejemplo auditorias de audiencia, los servidores de proxy-caché pueden ser configurados para mandar la IP real del cliente en unas cabeceras http especiales. Estas cabeceras varían dependiendo del proveedor de la solución de proxy-caché, pero las dos más comunes son Client-IP y X-Forwarded-For. Algunos administradores de los proxy-cachés deshabilitan el envío de estas cabeceras para ocultar la topología de su red interna.

En el caso de la red IP de Telefónica de España ambas cabeceras son enviadas por la solución de proxy-caché, de forma que los proveedores de contenidos puedan usarlas para sus análisis de accesos

Un caso típico es la contabilidad de visitas a una página web por usuario distinto:

1. En primer lugar para asegurar que cada vez que un usuario pide una página queda algún tipo de registro en el servidor web, es necesario asegurar que al menos un objeto contenido en la página va a ser no cacheable (usando por ejemplo las cabeceras http descritas anteriormente). Típicamente se elige un objeto lo más pequeño posible, para no perjudicar los tiempos de descarga.
2. En segundo lugar se contabilizan los accesos desde usuarios distintos a ese objeto, ya sea mediante el uso de cookies, y/o la obtención de la IP del cliente.

Es importante resaltar que las cabeceras http pueden ser modificadas fácilmente por cualquier cliente, así que no deben usarse como único medio para autenticar el acceso, salvo que provengan de las IPs de servidores proxy-caché confiables. En cualquier caso el uso de autenticación por IP origen es algo poco fiable en Internet debido al uso extensivo de IPs dinámicas, proxies, firewalls, etc... Alternativas como el uso de HTTPS son una mejor garantía para la seguridad de los servidores web, y por tanto de la confianza de los usuarios en los mismos.

## 4. PREGUNTAS Y RESPUESTAS MÁS FRECUENTES

### ¿Cuáles son las cosas más importantes que deben ser cacheables?

En primer lugar se deben identificar los objetos más grandes y más populares y trabajar sobre ellos

### ¿Cómo puedo hacer que mis páginas carguen lo más rápidamente posible con cachés?

Los objetos son más cacheables cuanto mayor sea su tiempo de validez en la caché, es decir su tiempo de expiración sea mayor. Aunque cuando la caché tiene que validar un objeto el ancho de banda consumido es pequeño, la latencia de contactar al servidor origen puede ser importante, por tanto es mejor si la caché no tiene que contactar muy a menudo al servidor origen.

### Entiendo el efecto positivo de las cachés, pero ¿cómo mantengo estadísticas de acceso a mis páginas?

Si necesitas conocer cada vez que una página es accedida, crea un pequeño objeto en cada página (o incluso la misma página), y hazlo no cacheable utilizando las cabeceras de http adecuadas. Un ejemplo es referirse en cada página a una imagen de 1x1 pixels transparente que sea no cacheable. La cabecera Referer de esa imagen contendrá la información sobre la página que el usuario pidió.

No se debe abusar del uso de estos objetos, ya que la caché debe contactar al servidor origen por cada objeto de este tipo incluido en una página.

### Tengo una página que se actualiza muy frecuentemente, ¿cómo puedo asegurar que las cachés no sirvan contenido desactualizado?

La cabecera Expires es la mejor forma de conseguirlo. Configurando esta cabecera para que expire el documento en función de la fecha de última modificación del mismo, puedes conseguir que las cachés lo marquen como desactualizado (y soliciten una nueva copia) al cabo de un tiempo después de su última modificación.

Por ejemplo, si la página principal cambia todos los días a las 8:00 am, configura la cabecera Expires a 23 horas después de la última modificación. De esta manera, tus usuarios siempre obtendrán una copia actualizada desde la caché.

También es posible usar la cabecera Caché-Control: max-age.

### ¿Cómo puedo ver las cabeceras Http de un objeto?

Para ver todas las cabeceras de un objeto puedes conectarte manualmente al servidor web utilizando un cliente de Telnet. Dependiendo del programa que uses, puedes necesitar introducir el puerto en un campo separado, o conectarte a `www.miservidor.com:80` o `www.miservidor.com 80` (con un espacio). Consulta la documentación de tu cliente de telnet.

Una vez conectado, escribe la petición del objeto web. Por ejemplo, si quieres ver las cabeceras de `http://www.miservidor.com/blabla.html`, conéctate a `www.miservidor.com`, puerto 80, y escribe:

```
GET /blabla.html HTTP/1.1 [return]
Host: www.miservidor.com [return][return]
```

Presiona la tecla `return` cada vez que ves `[return]`, asegúrate de presionarla dos veces al final. Esto mostrará las cabeceras, y el objeto solicitado. Si sólo quieres ver las cabeceras utiliza `HEAD` en lugar de `GET`.

### **Mis páginas están protegidas por usuario y password, ¿cómo las tratan las cachés?**

Por defecto las páginas protegidas mediante autenticación `http` son marcadas como no cacheables. Aunque puedes marcarlas como cacheables con la cabecera `Caché-Control`, si quieres que las cachés las almacenen.

Si quieres hacer las páginas cacheables, pero que se realice la autenticación para cada usuarios, combina las cabeceras de `http Cache-control: public` y `no-caché`. Esto le dice a las cachés que debe mandar la información de autenticación al servidor origen antes de devolver el objeto desde la caché.

En cualquier caso es recomendable reducir el uso de autenticación. Por ejemplo si hay imágenes que no contienen información sensible, ponlas en directorio separado que no requiera autenticación. De esa forma las imágenes serán cacheables.

### **¿Debo preocuparme por la seguridad de mis usuarios al acceder mediante un proxy-caché?**

Las páginas situadas en servidores seguros (`https`) no son interceptadas por los servidores proxy-caché, y por tanto no son almacenadas ni descriptadas.

En los servidores no seguros, las medidas de seguridad han de ser las mismas que generalmente se utilizan cuando se transmite información por una red pública como es Internet. En particular el envío de usuarios y contraseñas embebidos en el URL se desaconseja, ya que es fácil de capturar.

### **Mis imágenes expiran en un mes, pero necesito cambiarlas en la caché ahora mismo, ¿cómo lo hago?**

La solución más efectiva es cambiarlas de nombre, de esta forma serán objetos nuevos para la caché y las pedirán al servidor origen. Recuerda que las páginas que las referencian pueden ser cacheables también (lo cual podría hacer que no se solicitasen las nuevas imágenes). Por este motivo se recomienda hacer las imágenes muy cacheables (periodos de validez largos), y las páginas estáticas que las referencien que se refresquen más frecuentemente.

Si quieres forzar la recarga de una página en la caché, se puede hacer desde el navegador pulsando la tecla `Control` y el botón `reload` (en el caso del Internet Explorer). Esto fuerza la inclusión de una cabecera “no-caché” en la petición haciendo que la caché solicite el contenido al servidor origen. En algunas circunstancias de navegación con frames anidados el navegador no refresca cada uno de los frames visualizados, sí los contenidos por defecto dentro del frame más externo.

**He introducido en mis páginas HTML las cabeceras de cacheabilidad del tipo `<META HTTP-EQUIV="Pragma" CONTENT="no-cache">` y sin embargo los proxy-cachés siguen cacheando mis páginas, ¿están funcionando mal las caches?**

Es importante resaltar que las cabeceras sobre cacheabilidad que se pueden introducir en las páginas HTML (por ejemplo: <META HTTP-EQUIV="Pragma" CONTENT="no-cache">) no son interpretadas por los proxy-cachés. Esto es así porque los proxy-cachés no se detienen a analizar el contenido de las páginas HTML que almacenan y por tanto no “ven” estas cabeceras de HTML. Estas cabeceras en HTML están pensadas para controlar la caché existente en los navegadores.

Las cabeceras de cacheabilidad se tienen que definir por tanto, a nivel del protocolo HTTP. Los servidores web más comunes proporcionan mecanismos sencillos para configurar estas cabeceras HTTP. En el capítulo 5 se describe como hacerlo en algunos de estos servidores web.

### **En mi servidor web utilizo autenticación mediante NTLM de Microsoft, ¿existe algún problema con el proxy-caché?**

El sistema de autenticación NTLM de Microsoft no funciona en un entorno de proxy-caché. Este hecho es conocido por Microsoft y por tanto sólo recomienda este tipo de autenticación en entornos con conexión punto a punto.

En el entorno de proxy-caché de Telefónica de España, Microsoft recomienda el uso de otros mecanismos de autenticación alternativos a NTLM.

En particular existen casos reportados en Microsoft para las siguientes aplicaciones:

Microsoft Exchange OWA: <http://support.microsoft.com/default.aspx?scid=kb;en-us;198509>

Administración de un servidor web autenticado por NTLM:

<http://support.microsoft.com/default.aspx?scid=kb;en-us;183730>

Otros enlaces y recomendaciones sobre métodos de autenticación en diferentes entornos de red aparecen publicados en las siguientes páginas de Microsoft:

[http://officeupdate.microsoft.com/frontpage/WPP/serk/scwin\\_2.htm](http://officeupdate.microsoft.com/frontpage/WPP/serk/scwin_2.htm)

<http://support.microsoft.com/support/kb/articles/Q264/9/21.ASP>

## 5. EJEMPLOS DE CONFIGURACIÓN EN SERVIDORES WEB

### 5.1 APACHE 1.3

Apache (<http://www.apache.org>) usa módulos opcionales para incluir cabeceras http, incluyendo Expires y Caché-Control. Estos módulos están disponibles en versiones 1.2 y superiores.

Es necesario incluir los módulos dentro de Apache, ya que aunque están incluidos en la distribución, no están habilitados por defecto. Para saber los módulos que están habilitados en tu servidor, busca el ejecutable httpd y ejecuta `httpd -l`, esto mostrará la lista de los módulos habilitados. Los módulos que necesitas son `mod_expires` y `mod_headers`.

- Si no están habilitados, puedes recompilar Apache para incluirlos. Esto se puede hacer descomentando las líneas adecuadas en el fichero de configuración, o usando los argumentos `-enable-module=expires` y `-enable-module=headers` del comando `configure`. Consulta el fichero `INSTALL` de tu distribución de Apache

Una vez que tengas los módulos habilitados en Apache, puedes especificar que objetos tienen cabeceras Expires tanto en el fichero `.htaccess` o en fichero `access.conf` del servidor. Puedes especificar tiempos de expiración basados tanto la fecha de última modificación como en la de acceso. Consulta [http://www.apache.org/docs/mod/mod\\_expires.html](http://www.apache.org/docs/mod/mod_expires.html) para más información.

Para usar la cabecera Caché-Control se usa el módulo `mod_headers`, consulta [http://www.apache.org/docs/mod/mod\\_headers.html](http://www.apache.org/docs/mod/mod_headers.html).

Este es un ejemplo de fichero `.htaccess` :

El fichero `.htaccess` permite el uso de comandos que normalmente sólo se encuentran en los ficheros de configuración del servidor. Esos comandos sólo aplican al directorio y subdirectorios donde se encuentre el fichero `.htaccess`.

```
### activate mod_expires
ExpiresActive On
### Expire .gif's 1 month from when they're accessed
ExpiresByType image/gif A2592000
### Expire everything else 1 day from when it's last modified
### (this uses the Alternative syntax)
ExpiresDefault "modification plus 1 day"
### Apply a Cache-Control header to index.html
<Files index.html>
Header append Cache-Control "public, must-revalidate"
</Files>
```

Para añadir a los ficheros de registro de accesos estándar de apache la dirección IP real del cliente que envían los proxy-cachés de Telefónica de España en las cabeceras de http X-Forwarded-For y client-ip,

es necesario modificar el fichero de configuración httpd.conf. La configuración estándar de registro de accesos en dicho fichero es (se ha omitido el resto del contenido del fichero httpd.conf por claridad):

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent} i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer LogFormat "%{User-agent}i" agent
```

Para añadir por ejemplo la cabecera client-ip como un campo adicional al final del fichero de registro la configuración sería:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent} i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Client-ip}i" common
LogFormat "%{Referer}i -> %U" referer LogFormat "%{User-agent}i" agent
```

Se recomienda mantener las dos IPs (la de la conexión TCP y la enviada por la caché en la cabecera) para poder identificar posibles intentos de envío de cabeceras http falsas para ocultar la dirección IP. Hecho esto sólo se debería confiar en las cabeceras http cuando provienen desde la IP de los proxy-cachés de Telefónica de España.

## 5.2 NETSCAPE ENTERPRISE 3.6

Netscape Enterprise Server (<http://www.netscape.com>) no proporciona ningún mecanismo sencillo para configurar la cabecera Expires. Sin embargo si soporta funciones de http 1.1 desde su versión 3.0. Lo cual permite el uso de la cabecera Caché-Control.

Para usar esta cabecera, selecciona Content Management | Caché Control Directives en el servidor de administración. Ahora usando el Resource Picker, selecciona el directorio para el que quieres configurar las cabeceras. Una vez configuradas pulsa OK. Para mas información consulta: <http://developer.netscape.com/docs/manuals/enterprise/admnunix/content.htm#1006282>

En el caso de querer usar la funcionalidad de registro de accesos flexible de Netscape, se debe seleccionar la opción de “Custom logs” en la web de administración del servidor. Un ejemplo del formato “custom” a utilizar para añadir al final del fichero de registro la dirección IP real del cliente enviada por el proxy-caché en las cabeceras http es:

```
%Ses->client.ip% - %Req->vars.auth-user% [%SYSDATE%] \"%Req->reqpb.clf-request%\"
%Req->srvhdrs.clf-status% %Req->srvhdrs.content-length% %Req->headers.Client-ip%
```

Para más información sobre la configuración de registro de accesos flexible consultar el URL: [http://enterprise.netscape.com/docs/enterprise/611/nsapi/07\\_magnu.htm#33046](http://enterprise.netscape.com/docs/enterprise/611/nsapi/07_magnu.htm#33046) y <http://developer.netscape.com/docs/manuals/enterprise/admnunix/manage.htm#1016920> .

### 5.3 MICROSOFT IIS 4.0

Microsoft Internet Information Server (<http://www.microsoft.com>) permite configurar cabeceras de una forma muy sencilla y flexible.

Para configurar las cabeceras a una parte concreta del servidor, selecciona esa parte en la interfaz de administración y abre sus propiedades. En la pestaña `HTTP Headers` hay dos áreas interesantes, `Enable Content Expiration` y `Custom HTTP headers`. La primera es autoexplicativa y la segunda se puede usar para configurar la cabecera `Cache-control`

En esta versión de IIS no es posible configurar ficheros de registro de acceso personalizados que puedan incluir las cabeceras `http` que envía la caché con la dirección IP real del cliente. En este caso, una posible solución es el desarrollo de un filtro ISAPI que añada dicha cabecera a cada registro de acceso. Telefónica de España dispone de un ejemplo de dicho filtro que podría ser suministrado bajo demanda.

## 6. REFERENCIAS

### 6.1 PROTOCOLO HTTP 1.1

El RFC 2616 especifica el protocolo HTTP 1.1 y dedica el apartado 13 a los sistemas de proxy-cachés

### 6.2 CACHING TUTORIAL FOR WEB AUTHORS AND WEBMASTERS

[http://www.mnot.net/cache\\_docs/](http://www.mnot.net/cache_docs/)

Guía utilizada para la confección de este documento.

### 6.3 INTRODUCCIÓN A LAS CACHÉS WEB

<http://www.web-caching.com/>

Introducción a los conceptos de las cachés web, y enlaces a otras Fuentes de información

### 6.4 CACHEABILITY ENGINE

<http://www.mnot.net/cacheability/>

Herramienta online para examinar páginas web y analizar su cacheabilidad.