



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



① Número de publicación: **2 328 998**

② Número de solicitud: 200703321

⑤ Int. Cl.:  
**G06F 17/30** (2006.01)

⑫

PATENTE DE INVENCION CON EXAMEN PREVIO

B2

⑫ Fecha de presentación: **05.12.2007**

⑬ Fecha de publicación de la solicitud: **19.11.2009**

Fecha de la concesión: **07.04.2010**

⑮ Fecha de anuncio de la concesión: **21.04.2010**

⑯ Fecha de publicación del folleto de la patente:  
**21.04.2010**

⑰ Titular/es: **Universidad Politécnica de Valencia  
CTT-Edificio 6G - Camino de Vera, s/n  
46022 Valencia, ES**

⑱ Inventor/es: **Doménech de Soria, Josep;  
Gil Salinas, José Antonio;  
Pont Sanjuán, Ana y  
Sahuquillo Borrás, Julio**

⑳ Agente: **No consta**

㉑ Título: **Método de reducción del tiempo de descarga de páginas Web.**

㉒ Resumen:

Método de reducción del tiempo de descarga de páginas Web.

La invención se refiere a un método de reducción del tiempo de descarga de una página de un servidor Web, en el que:

- el servidor obtiene un identificador de frescura de un objeto asociado a una URL de una relación de URLs dada; caracterizado porque:

- el servidor pone a disposición de un cliente dicha relación de URLs y los identificadores de frescura (300, 400, 500) de dichos objetos asociados a las URLs antes de que el cliente solicite el objeto correspondiente;

y porque:

- el cliente utiliza dichos identificadores de frescura, una vez los reciba del servidor, para actualizar una marca de tiempo de un objeto que se encuentra en una caché del cliente y cuyo objeto tiene asociado una URL para la cual el servidor ha proporcionado el identificador de frescura.

ES 2 328 998 B2

Aviso: Se puede realizar consulta prevista por el art. 40.2.8 LP.

## DESCRIPCIÓN

Método de reducción del tiempo de descarga de páginas Web.

5 **Campo de la invención**

La presente invención se engloba dentro del campo de las redes de ordenadores; más concretamente, se proporciona un método para mejorar los tiempos de descarga de páginas Web.

10 **Antecedentes de la invención**

Desde la aparición de la Web se han realizado numerosos esfuerzos para reducir el tiempo de descarga de las páginas Web. La Web funciona siguiendo la filosofía tradicional de cliente-servidor por medio del protocolo HTTP. La figura 1 muestra un ejemplo de cómo se realiza la comunicación entre ambas partes. En este ejemplo, un cliente solicita al servidor www.myserver.com la página A.html. Generalmente las páginas suelen estar compuestas de varios objetos como imágenes, etc. En el ejemplo, la página A.html contiene la imagen img1.gif, que se solicita al servidor tras haber recibido el documento principal (A.html). La latencia percibida por el usuario (tiempo de descarga de la página) viene dada principalmente por el tamaño de los objetos solicitados, el ancho de banda disponible entre el cliente y el servidor, el tiempo de procesamiento de cada petición por el servidor y el tiempo que tarda la señal en propagarse desde el cliente al servidor.

Una página web suele estar formada por múltiples objetos. De esta forma, el tiempo de descarga de la página web está directamente relacionado con el tiempo de descarga de cada uno de estos objetos que la forman. Se pueden distinguir dos componentes principales del tiempo de descarga de un objeto web: por una parte el tiempo que tarda la señal en ir del cliente al servidor y volver (conocido como RTT) y por otra el tiempo de transferencia del objeto (que depende del ancho de banda entre cliente y servidor). El desarrollo tecnológico ha permitido reducir ambos tiempos, especialmente el dedicado a la transferencia del objeto. Esto ha hecho que el peso de ambos componentes haya variado sensiblemente a lo largo de los años.

Como ejemplo, supongamos que un cliente va a descargar un archivo de 10 KB. Teniendo en cuenta los datos que ofrece la bibliografía sobre las conexiones a Internet en 1996, un valor típico de ancho de banda sería 33 Kbps, mientras que para el RTT sería de 1.130 ms. Descargar 10 KB a través de una conexión de 33 Kbps, suponiendo que no hay sobrecarga de ningún tipo, tardaría 2.483 ms. Por lo tanto, la latencia total de descarga del objeto sería de  $1.130 + 2.483 = 3.613$  ms. En esta situación, el tiempo de transferencia supone el  $2483/3616 = 69\%$  de la latencia total.

En la situación actual, nos encontramos con un ancho de banda típico de 4 Mbps y un RTT de 120 ms. El tiempo de transferencia de un objeto de 10 KB con la conexión moderna sería de 20 ms que, sumado a los 120 ms del RTT, nos daría una latencia total de objeto de 140 ms. En esta situación, el tiempo de transferencia sólo supone el  $20/140 = 14\%$  de la latencia total.

Las principales técnicas desarrolladas hasta ahora para la reducción de latencia son las Redes de Distribución de Contenido (CDN, por sus siglas en inglés), el Web Caching y la prebúsqueda web. La primera consiste en hacer copias del servidor y situar cada una de estas copias cerca de un grupo de usuarios finales, de tal forma que se reduce el tiempo que tarda la señal en propagarse desde el cliente al servidor y, por lo tanto, la latencia percibida por el usuario. Existe una amplia implantación de esta técnica en el mercado, donde destacan empresas con un gran volumen de negocio como Akamai, Amazon S3, BitGravity, etc.

El Web Caching se fundamenta en el hecho de que los usuarios suelen acceder a páginas que habían accedido previamente. En esta técnica, el cliente almacena en su disco duro de forma automática copias las páginas accedidas para poderlas utilizar en el futuro. La reducción de la latencia se consigue por medio de servir el contenido del objeto solicitado de forma local, que es significativamente más rápido que pedirlo al servidor.

Para asegurarse de que sólo se sirven objetos frescos (o no caducados) desde la caché, a cada objeto se le asigna una fecha de caducidad. Generalmente, cuando se solicita un objeto web, el servidor incorpora información sobre la frescura del objeto (por ejemplo, su edad, la marca de tiempo de última modificación o su e-tag), que puede ser utilizada para calcular la fecha de caducidad del mismo. Si el usuario accede a un objeto que tiene en caché y está caducado, el navegador debe validar dicho objeto, es decir, debe comprobar que este objeto sigue siendo válido (es decir, que no ha cambiado) antes de servirlo al usuario. Esta validación se realiza mediante una petición condicional al servidor web (utilizando por ejemplo una cabecera "If-Modified-Since"): el cliente solicita que el servidor le devuelva el objeto sólo si no es el mismo que el cliente descargó y almacenó en caché. A esta petición el servidor responde con el objeto si éste ha sido modificado, o bien con una respuesta "Not Modified" si no ha cambiado desde entonces. Con este modo de actuar lo que se consigue es que, si el objeto no ha cambiado, el usuario se ahorra el tiempo de transferencia del objeto. Teniendo en cuenta la formación del tiempo de descarga de los objetos descrita anteriormente, esta técnica puede reducir drásticamente la latencia percibida considerando la situación de 1996 (un 69% en el ejemplo utilizado). Sin embargo, en la situación actual esta reducción de latencia está mucho más limitada (un 14% en el mismo ejemplo).

La elección de la fecha de caducidad del objeto es importante ya que si se da una fecha muy lejana, se corre el peligro de que el navegador muestre al usuario una versión anterior de la web, mientras que si es muy cercana, se

## ES 2 328 998 B2

reducen los beneficios del Web caching. En la práctica suele primar la consistencia de la web a los beneficios del caching, por lo que se suelen dar tiempos para los que un objeto es fresco muy reducidos, o incluso nulos.

5 El Web Caching se utiliza en prácticamente todos los navegadores Web que existen en la actualidad, además de ser utilizado de forma más global en la mayoría de organizaciones por medio de los servidores proxy. Estos servidores proxy están desarrollados por empresas como Microsoft, Netscape, Sun, etc. y utilizados en grandes organizaciones como Telefónica o la Universidad Politécnica de Valencia entre otras muchas.

10 En un sistema de prebúsqueda como los que existen en la actualidad, el servidor Web mantiene un registro de los accesos realizados para predecir los accesos siguientes de los usuarios. El funcionamiento general de la técnica de prebúsqueda web es el siguiente:

15 1. El navegador web del usuario realiza la petición de un objeto web al servidor.

2. El servidor consulta su histórico de accesos y predice cuáles serán los próximos accesos de este usuario. Esta predicción se incorpora a la respuesta, junto con el objeto web requerido.

20 3. Cuando el navegador web del usuario está ocioso, pide por adelantado los objetos predichos por el servidor y se almacenan localmente.

4. Cuando el usuario accede al objeto predicho, se sirve la copia local del objeto con lo que se reduce el tiempo de descarga de la página.

25 En la patente estadounidense US-6094662 se describe un método para la carga y descarga de páginas HTML, donde se utiliza una marca de tiempo para identificar si la versión guardada en caché está actualizada o no.

30 Por otro lado, la patente estadounidense US-6182133 describe un método que realiza un prebúsqueda y descarga de ciertos elementos de información de acuerdo con aspectos predefinidos, y se le proporciona al usuario una indicación visual que le permite identificar dicha descarga previa.

### Descripción de la invención

35 La invención se refiere a un método de reducción del tiempo de descarga de páginas Web de acuerdo con la reivindicación 1. Realizaciones preferidas del método se describen en las reivindicaciones dependientes.

40 A diferencia de los sistemas descritos anteriormente, el método definido en la presente invención anticipa los identificadores de frescura de los objetos sin esperar siquiera a la petición de algún objeto por parte del cliente.

La invención se refiere a un método de reducción del tiempo de descarga de una página de un servidor Web, en el que el servidor obtiene un identificador de frescura de cada objeto asociado a una URL de una relación de URLs que viene dada bien por el servidor, bien por el cliente.

45 De acuerdo con un primer aspecto de la invención:

- el servidor pone a disposición de un cliente dicha relación de URLs y los identificadores de frescura de dichos objetos asociados a las URLs antes de que el cliente solicite el objeto correspondiente; y,

50 - el cliente utiliza dichos identificadores de frescura, una vez los reciba del servidor, para actualizar una marca de tiempo de un objeto que se encuentra en una caché del cliente y cuyo objeto tiene asociado una URL para la cual el servidor ha proporcionado el identificador de frescura.

55 Dicho identificador de frescura puede ser la fecha de última modificación, la edad o la etiqueta de entidad o e-tag del objeto.

De esta forma, antes de iniciar una petición de un objeto solicitado por un usuario, el cliente recibe una relación de URLs junto con sus identificadores de frescura actualizados.

60 El cliente recibe los identificadores de frescura del servidor bien porque los solicita de forma expresa:

- estableciendo una conexión con el servidor;

65 - solicitando al servidor dicha relación de URLs e identificadores de frescura;

y el servidor le envía al cliente dicha relación de URLs e identificadores de frescura.

## ES 2 328 998 B2

O bien el cliente recibe dichos identificadores de frescura porque el servidor la proporciona de forma predictiva de la siguiente forma:

- 5 - el servidor obtiene, a partir de un algoritmo de predicción, una lista de URLs predichas para cada objeto solicitado por el cliente;
- para cada URL predicha de dicha lista, se adjunta el identificador de frescura del objeto asociado a cada URL predicha; y,
- 10 - el servidor envía al cliente, junto con el objeto solicitado, dicha lista de URLs predichas e identificadores de frescura.

15 Una vez que el cliente dispone de esta información, puede actualizar los identificadores de frescura de los objetos que tiene en caché y las fechas de caducidad de los mismos para que posteriores peticiones a esos objetos tengan mayores probabilidades de encontrarse frescos. Cabe destacar que con esta forma de proceder alargaríamos el tiempo que un objeto permanece fresco en caché.

20 Otra forma de actuar con la información proporcionada de forma anticipada por el servidor es marcar como prevalidados los objetos de la caché que se encuentren en la relación suministrada por el servidor, de la siguiente forma (se incorpora una parte de predicción):

- 25 - el cliente, para cada objeto que se encuentra en caché y que está asociado a una URL proporcionada por el servidor junto con su identificador de frescura, realiza una comparación del identificador de frescura proporcionado por el servidor con una marca de tiempo de descarga en caché y si el objeto en caché sigue siendo fresco, marca dicho objeto como prevalidado; y

30 cuando un usuario solicita una página Web a través de un cliente, en el cliente se descompone dicha página Web en objetos, y para cada objeto:

- se comprueba si el objeto solicitado se encuentra en una caché del cliente, y
  - 35 - si no se encuentra en caché, se efectúa una petición de dicho objeto al servidor de forma estándar;
  - si el objeto se encuentra en la caché, dicho objeto tendrá asociada una marca de tiempo (fecha y hora) de caducidad, y se procede a verificar la caducidad del mismo, y en caso de que el objeto no esté caducado, el cliente sirve al usuario el objeto de la caché.

40 Entonces:

- si el objeto se encuentra caducado en caché, se comprueba si éste ha sido marcado como prevalidado hace menos de un tiempo pre-establecido; y,
  - 45 - si el objeto ha sido marcado como prevalidado, el cliente sirve al usuario el objeto de la caché;
  - si el objeto no ha sido marcado como prevalidado, se realiza una petición condicional de dicho objeto al servidor de forma estándar.

50 Dicha petición del objeto al servidor puede realizarse de la siguiente forma:

- el cliente envía al servidor la petición del objeto incluyendo un identificador (la URL) de dicho objeto, e incluyéndose la cabecera correspondiente si la petición es condicional;
- 55 - en el servidor se ejecuta para dicho objeto peticionado por el cliente un método de predicción de objetos Web según ha sido definido en lo anterior;
- el servidor envía al cliente el objeto solicitado junto con dicha lista de predicciones y para cada predicción, dicho  
60 identificador de frescura.

65 Y el cliente puede efectuar dicho marcado de un objeto como prevalidado comparando una marca de tiempo de última modificación derivada de dicho identificador de frescura con la marca de tiempo de descarga en caché de dicho objeto; y si el objeto no ha sido modificado desde que fue descargado (es decir, si dicha marca de tiempo de descarga coincide o es posterior a la marca de tiempo de última modificación proporcionada por el identificador de frescura), se marca dicho objeto como prevalidado.

## ES 2 328 998 B2

Es decir, si el objeto en caché no ha sido modificado desde que fue descargado, ya no es necesario efectuar la petición condicional al servidor de dicho objeto, por lo que el usuario se ahorra el RTT correspondiente a esperar la respuesta "Not Modified" del servidor. Por su parte, el servidor se ahorra tener que contestar a esta petición, por lo que su carga de trabajo también disminuye. Esto tiene un efecto positivo tanto en el usuario que se ha beneficiado de la predicción como indirectamente en el resto de usuarios por la disminución de la carga del servidor.

Es decir, junto con cada respuesta a un objeto solicitado por un cliente, el servidor Web incluye una predicción de cuáles van a ser los próximos objetos que va a pedir el usuario; como es probable que estos objetos ya los tenga el cliente del usuario en su caché, el método de la invención propone que el servidor añada junto con cada objeto predicho el identificador de frescura del mismo. Si en un espacio de tiempo preestablecido (definible en el sistema), el usuario trata de acceder al objeto predicho (identificado por la pista dada en la predicción) y éste se encuentra caducado en la caché, el cliente compara el identificador de frescura recibido junto al objeto con el identificador de frescura del objeto en caché. Si resulta que el objeto no ha sido modificado desde que fue descargado, ya no es necesario efectuar la petición condicional al servidor, con el consiguiente ahorro de recursos.

Es decir, el método propuesto en la presente invención se diferencia de la prebúsqueda tradicional en que, junto con la predicción de accesos, el servidor web anticipa otra información adicional -un identificador de frescura- que hace que el cliente sepa de antemano qué peticiones de acceso debe realizar y cuáles son innecesarias. De esta forma, se consigue reducir la carga de peticiones que recibe el servidor web y el tiempo de descarga de páginas percibido por el usuario. Todo ello sin incrementar el tráfico de red entre clientes y servidores. Según los estudios realizados, con esta nueva técnica, tanto la carga de peticiones del servidor web como el tiempo de descarga se reduce entre un 40% y un 60%.

La invención ha sido descrita para funcionar entre Cliente Web y Servidor Web, pero también puede funcionar entre Cliente Web y Servidor Proxy, entre Servidor Proxy y Servidor Web y entre dos servidores Proxy distintos y entre cualquier otra combinación en la que participen al menos dos elementos de la Arquitectura.

### Descripción de los dibujos

Para complementar la descripción que se está realizando y con objeto de ayudar a una mejor comprensión de las características del invento, de acuerdo con un ejemplo preferente de realización práctica del mismo, se acompaña como parte integrante de dicha descripción, un juego de dibujos en donde con carácter ilustrativo y no limitativo, se ha representado lo siguiente:

La figura 1 muestra el funcionamiento básico de comunicación cliente-servidor en http.

La figura 2 muestra el diagrama de flujo de una posible forma de comunicación entre cliente-servidor según la presente invención.

La figura 3 muestra el diagrama de flujo de otra posible comunicación entre cliente-servidor según la presente invención.

La figura 4 muestra el diagrama de flujo de petición de una página web por parte de un usuario según el método de la presente invención.

La figura 5 muestra el diagrama de flujo de la comunicación entre cliente-servidor según la presente invención.

La figura 6 muestra un gráfico de los resultados obtenidos al aplicar la invención a la web [www.elpais.es](http://www.elpais.es) con el algoritmo de predicción DG.

### Realización preferida de la invención

En la figura 2 se muestra el diagrama de flujo de una posible comunicación entre cliente-servidor según la presente invención.

El servidor genera una relación de URLs e identificadores de frescura correspondientes y la pone a disposición del cliente (300).

El cliente, sin que medie una petición por parte del usuario, solicita dicha relación de URLs e identificadores de frescura correspondientes (310). A continuación, el servidor envía esta relación al cliente (320). El cliente, para cada uno de los objetos que se encuentran en la caché y en la relación proporcionada por el servidor, actualiza la información de caducidad de dichos objetos con los identificadores de frescura recibidos (330).

En la figura 3 se muestra el diagrama de flujo de otra posibilidad de comunicación entre cliente-servidor según la presente invención.

El servidor genera una relación de URLs e identificadores de frescura correspondientes y la pone a disposición del cliente (500).

## ES 2 328 998 B2

El cliente, sin que medie una petición por parte del usuario, solicita los identificadores de frescura para las URLs de aquellos objetos que tuviere caducados en caché (510). A continuación, el servidor genera y envía esta relación al cliente (520). El cliente, para cada uno de los objetos que atendiendo a la información proporcionada por el servidor todavía se encuentren frescos, marca en su caché dichos objetos como prevalidados (530). El cliente utiliza estas marcas de prevalidación según describe la figura 4.

A continuación se describe una realización preferida del método de reducción del tiempo de descarga de páginas web de la invención:

10 En la figura 4 se muestran los pasos seguidos cuando un usuario solicita una página web a través de un cliente.

En el cliente, cuando el usuario solicita una página web, primero la descompone en objetos; y para cada objeto solicitado por el usuario (paso 100) se hace lo siguiente:

15 Se comprueba si el objeto solicitado se encuentra en la caché del cliente (decisión 110). Si no se encuentra (111), se efectúa una petición del objeto al servidor (paso 120; ver figura 5).

20 Si el objeto se encuentra en la caché (112), se verifica la fecha y hora de caducidad del mismo (decisión 130). En caso de que el objeto no esté caducado (131), el cliente sirve al usuario el objeto de la caché (paso 150). Y en caso de que el objeto se encuentre caducado (132), se realiza una comprobación de si éste ha sido prevalidado en los últimos X segundos (decisión 140). Si ha sido prevalidado en ese tiempo preestablecido (por configuración, por ejemplo), se sirve al usuario el objeto de la caché (paso 150). En caso contrario, se realiza una petición condicional al servidor (paso 160; ver figura 5).

25 En la figura 5 se muestra el diagrama de flujo de la comunicación entre cliente-servidor según la presente invención. Así, el cliente inicia una petición al servidor: resolución de nombre de dominio, establecimiento de conexión, etc. (paso 210), proporcionando un identificador de dicho objeto con el método GET de http; si la petición es condicional, se incluye la cabecera correspondiente.

30 El servidor obtiene, a partir de un algoritmo de predicción cualquiera, una lista de objetos candidatos a ser solicitados por el cliente próximamente o predicciones (paso 220).

35 Para cada predicción de la lista, el servidor obtiene un identificador de frescura (paso 230), como por ejemplo: marca de tiempo (fecha y hora) de última modificación, e-tag, edad, etc. y adjunta dicho identificador de frescura a la lista.

Se obtiene la respuesta a la petición de la forma usual y se adjunta la lista de predicciones junto con el identificador de frescura en la respuesta http, enviándose la respuesta al cliente (paso 240).

40 En el cliente se realiza una prevalidación para cada objeto de la lista de predicciones, y aquél objeto está fresco en la caché, se marca como prevalidado en el momento actual (250).

45 Las novedades del método de descarga propuesto residen en la generación de una relación de URLs e identificadores de frescura (pasos 300 y 500), en la toma de decisión respecto a si el objeto ha sido o no prevalidado en los últimos X segundos (paso 140), y en los pasos 230 y 250 relativos a añadir información adicional a la lista de predicciones y realizar la prevalidación para cada predicción, respectivamente.

50 Los experimentos realizados muestran que las mejoras en la latencia percibida por el usuario derivadas del empleo de esta técnica son considerables. Como ejemplo, en la figura 6 se puede apreciar que la latencia percibida por el usuario se puede reducir un 58% y las peticiones al servidor web un 68%. En este ejemplo se ha considerado el servidor web [www.elpais.es](http://www.elpais.es) y el algoritmo de predicción DG, ampliamente utilizado en el área de investigación.

55 En el eje X está representado el índice Object Traffic Increase, que es el resultado de dividir el número de peticiones al servidor cuando se utiliza la técnica por el número de peticiones al servidor cuando no se utiliza. Así pues, un valor de 0.32 significa que se han reducido las peticiones del servidor un 68%.

60 En el eje Y se representa el índice Latency per page ratio, que es el resultado de dividir la latencia percibida cuando se utiliza la técnica por la latencia percibida cuando no se utiliza. De esta forma, un valor de 0.42 significa que se ha reducido la latencia percibida por el usuario un 58%. Cada punto en la línea del gráfico representa una configuración determinada del algoritmo de predicción, por lo que la mejor configuración es la del punto representado en la esquina inferior izquierda de la figura.

65 La invención ha sido descrita para funcionar entre Cliente Web y Servidor Web, pero también puede funcionar entre Cliente Web y Servidor Proxy, entre Servidor Proxy y Servidor Web y entre dos servidores Proxy distintos y entre cualquier otra combinación en la que participen al menos dos elementos de la Arquitectura.

A la vista de esta descripción y juego de figuras, el experto en la materia podrá entender que las realizaciones de la invención que se han descrito pueden ser combinadas de múltiples maneras dentro del objeto de la invención.

## ES 2 328 998 B2

### REIVINDICACIONES

1. Método de reducción del tiempo de descarga de una página de un servidor Web, en el que:

- el servidor obtiene un identificador de frescura de un objeto asociado a una URL de una relación de URLs dada;

**caracterizado** porque:

- el servidor pone a disposición de un cliente dicha relación de URLs y los identificadores de frescura (300, 400, 500) de dichos objetos asociados a las URLs antes de que el cliente solicite el objeto correspondiente;

y porque:

- el cliente utiliza dichos identificadores de frescura, una vez los reciba del servidor, para actualizar una marca de tiempo de un objeto que se encuentra en una caché del cliente y cuyo objeto tiene asociado una URL para la cual el servidor ha proporcionado el identificador de frescura (330).

2. Método según la reivindicación 1, en el que dicho identificador de frescura es la fecha de última modificación, la edad o la etiqueta de entidad o e-tag del objeto.

3. Método según cualquiera de las reivindicaciones 1-2, en donde la función del cliente es realizada por un servidor proxy.

4. Método según cualquiera de las reivindicaciones 1-3, en donde la función del servidor es realizada por un servidor proxy.

5. Método según cualquiera de las reivindicaciones anteriores, en el que el cliente recibe los identificadores de frescura del servidor:

- estableciendo una conexión con el servidor;

- solicitando al servidor dicha relación de URLs e identificadores de frescura (310, 410);

y el servidor envía al cliente dicha relación de URLs e identificadores de frescura (320, 420).

6. Método según cualquiera de las reivindicaciones 1-4, en el que el cliente recibe los identificadores de frescura del servidor de la siguiente forma:

- el servidor obtiene, a partir de un algoritmo de predicción, una lista de URLs predichas para cada objeto solicitado por el cliente (220);

- para cada URL predicha de dicha lista, se adjunta el identificador de frescura del objeto asociado a cada URL predicha (230); y,

- el servidor envía al cliente, junto con el objeto solicitado, dicha lista de URLs predichas e identificadores de frescura (240).

7. Método según cualquiera de las reivindicaciones anteriores, en el que:

- el cliente, para cada objeto que se encuentra en caché y que está asociado a una URL proporcionada por el servidor junto con su identificador de frescura, realiza una comparación del identificador de frescura proporcionado por el servidor con una marca de tiempo de descarga en caché y si el objeto en caché sigue siendo fresco, marca dicho objeto como prevalido (430); y

cuando un usuario solicita una página Web a través de un cliente, dicho cliente descompone dicha página Web en objetos y para cada objeto:

- se comprueba si el objeto solicitado se encuentra en una caché del cliente (110), y:

- si no se encuentra en caché (111), se efectúa una petición de dicho objeto al servidor (120);

- si el objeto se encuentra en la caché (112), dicho objeto tiene asociada una marca de tiempo de caducidad, y se procede a verificar la caducidad del mismo (130), y en caso de que el objeto no esté caducado (131), el cliente sirve al usuario el objeto de la caché (150); y

## ES 2 328 998 B2

- en caso de que el objeto se encuentre caducado en caché (132), se comprueba si éste ha sido marcado como prevalidado hace menos de un tiempo preestablecido (140); y

5

- si el objeto ha sido marcado como prevalidado (141), el cliente sirve al usuario el objeto de la caché;

- si el objeto no ha sido marcado como prevalidado (142), se realiza una petición condicional de dicho objeto al servidor (160).

10

15

20

25

30

35

40

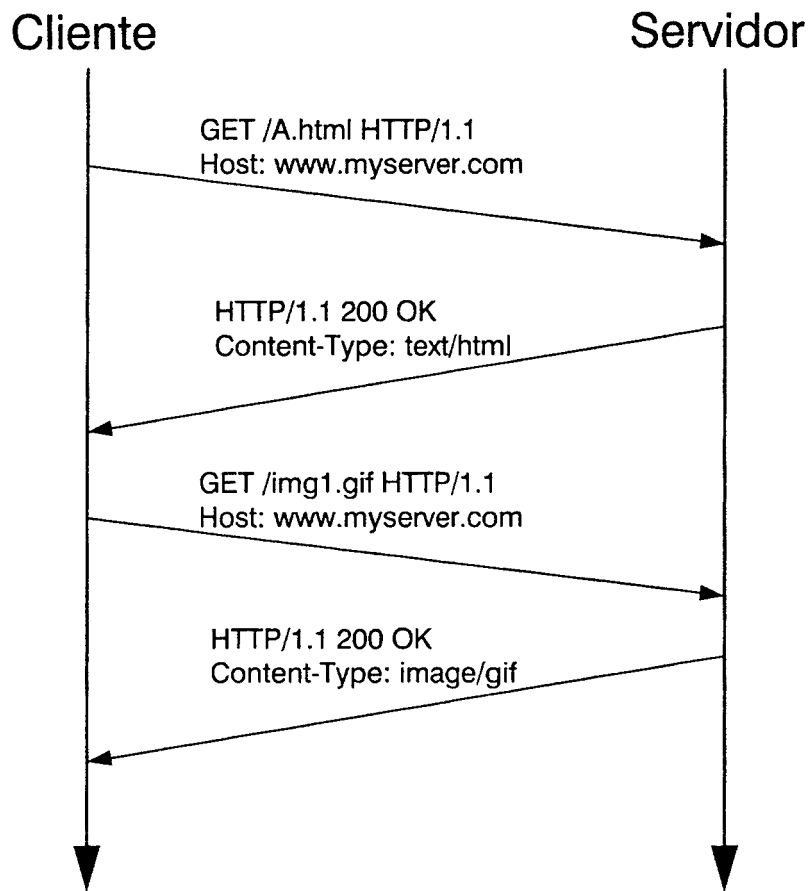
45

50

55

60

65



**FIG. 1**

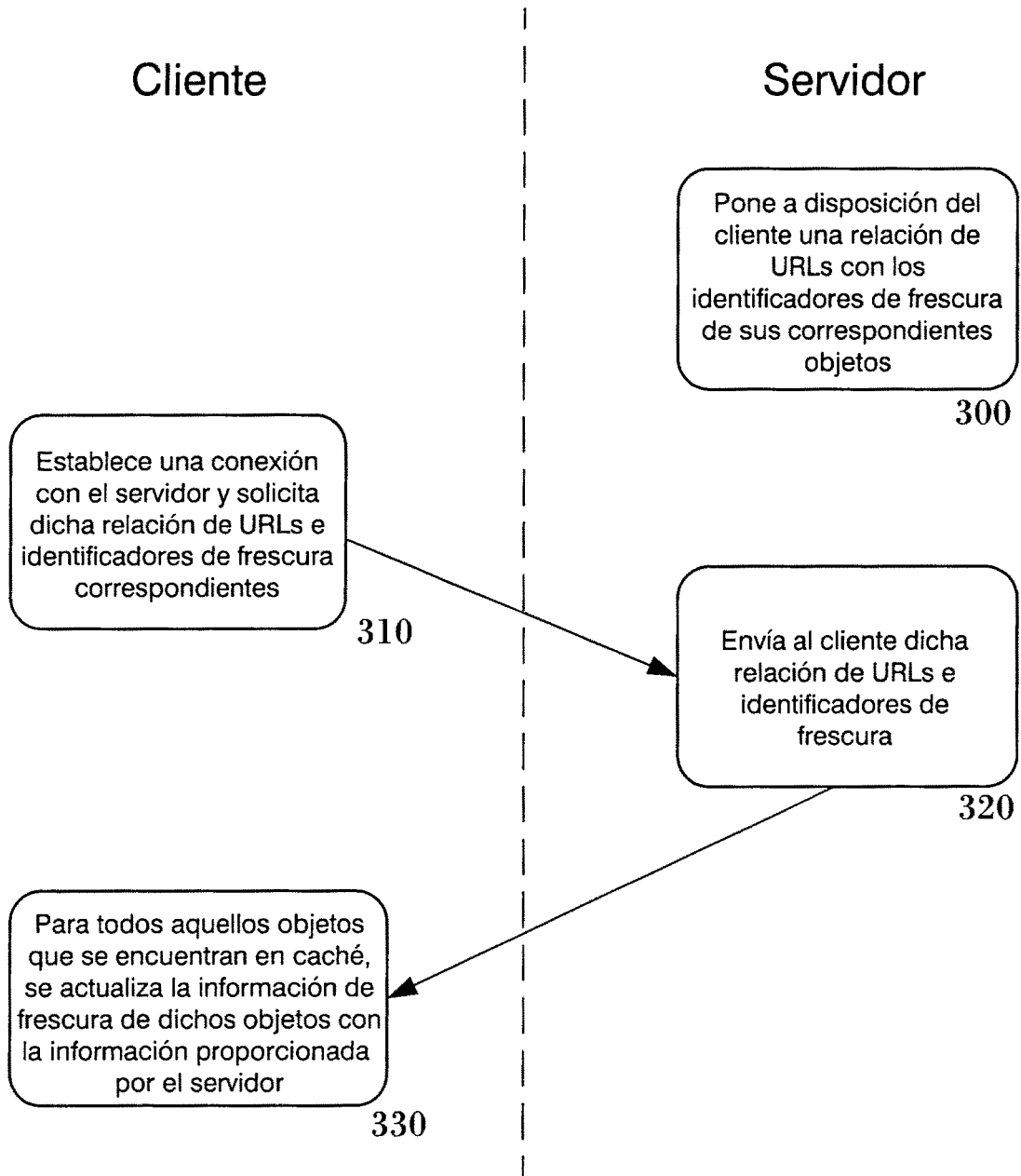
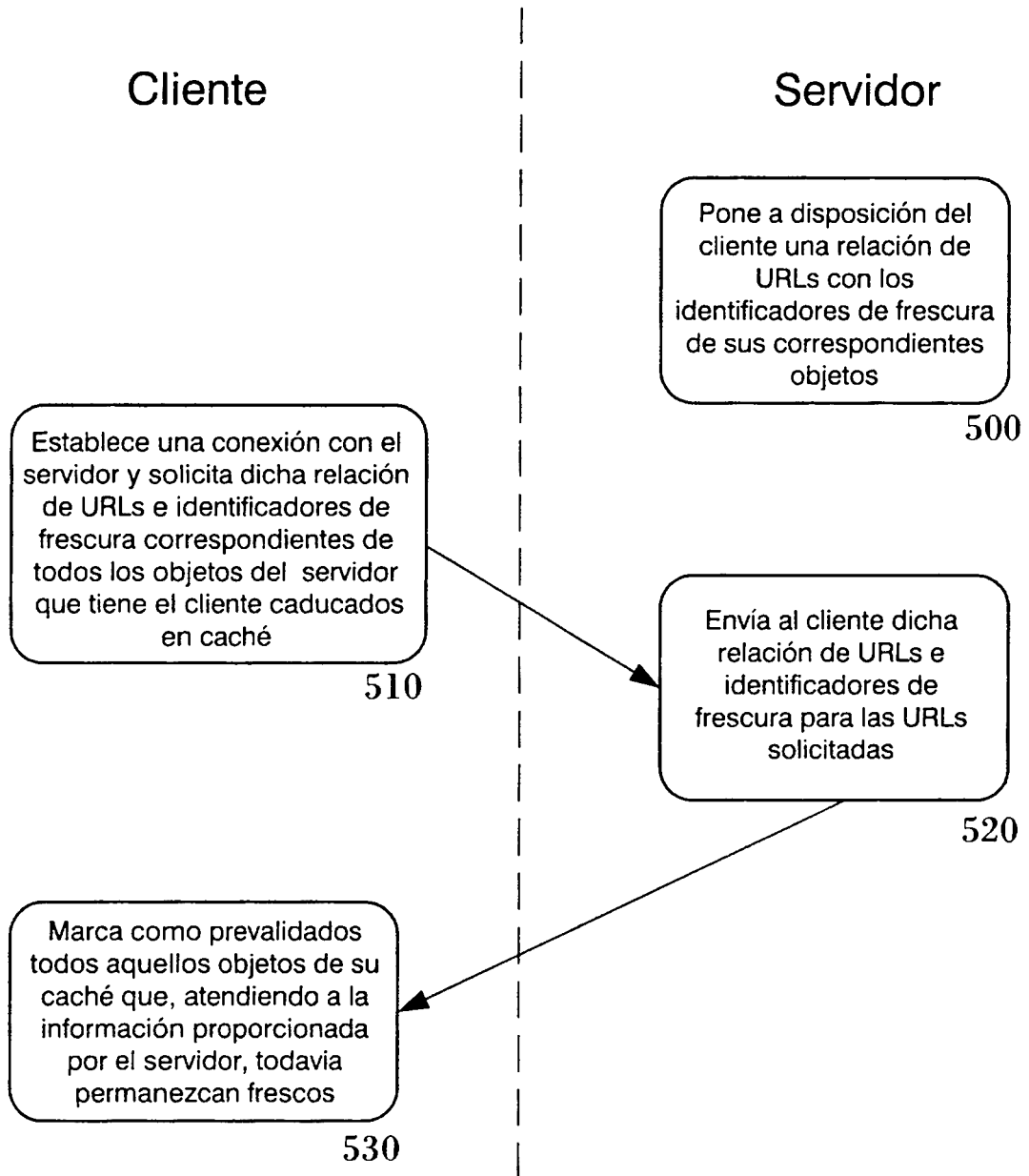


FIG. 2



**FIG. 3**

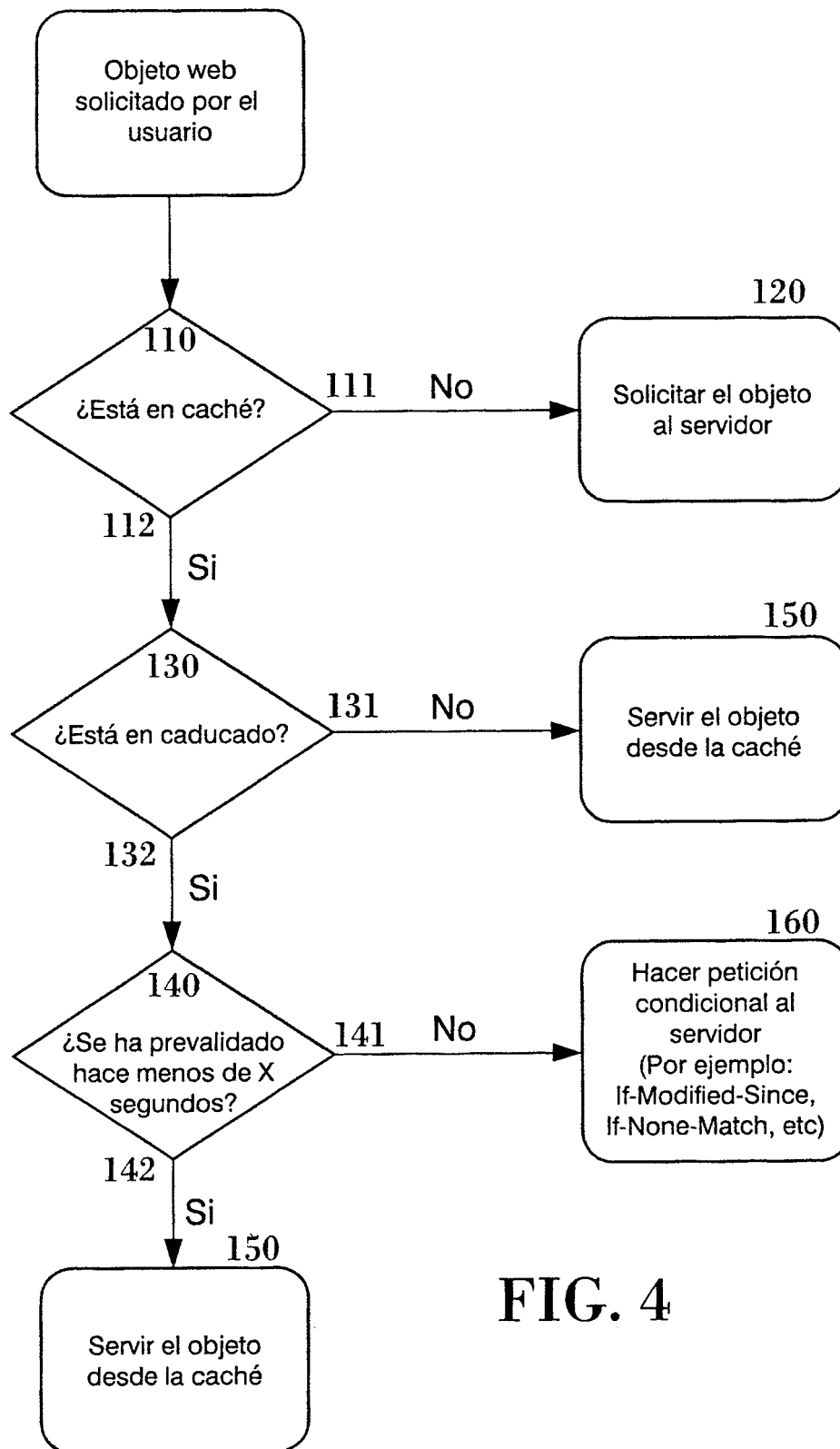
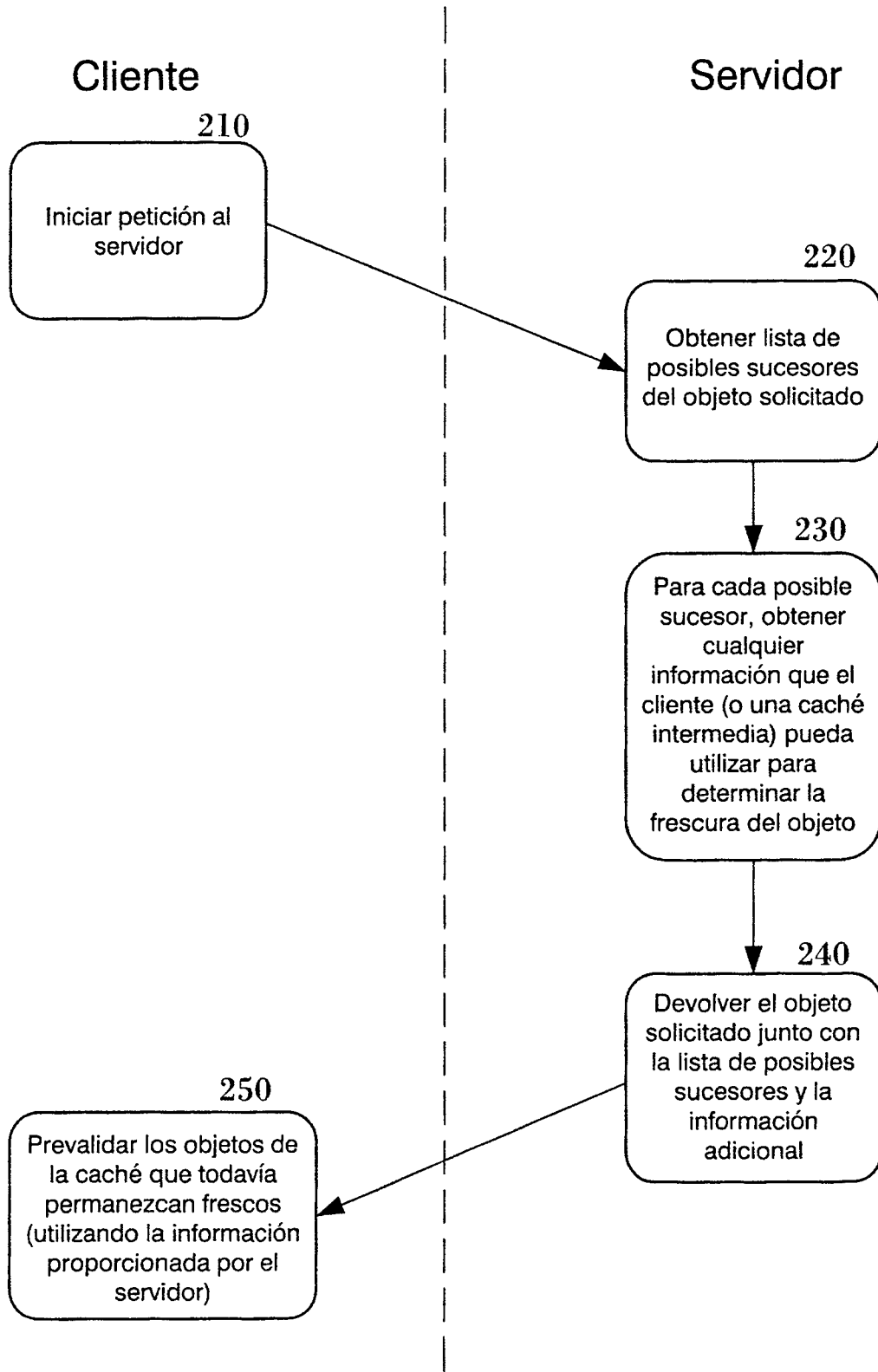
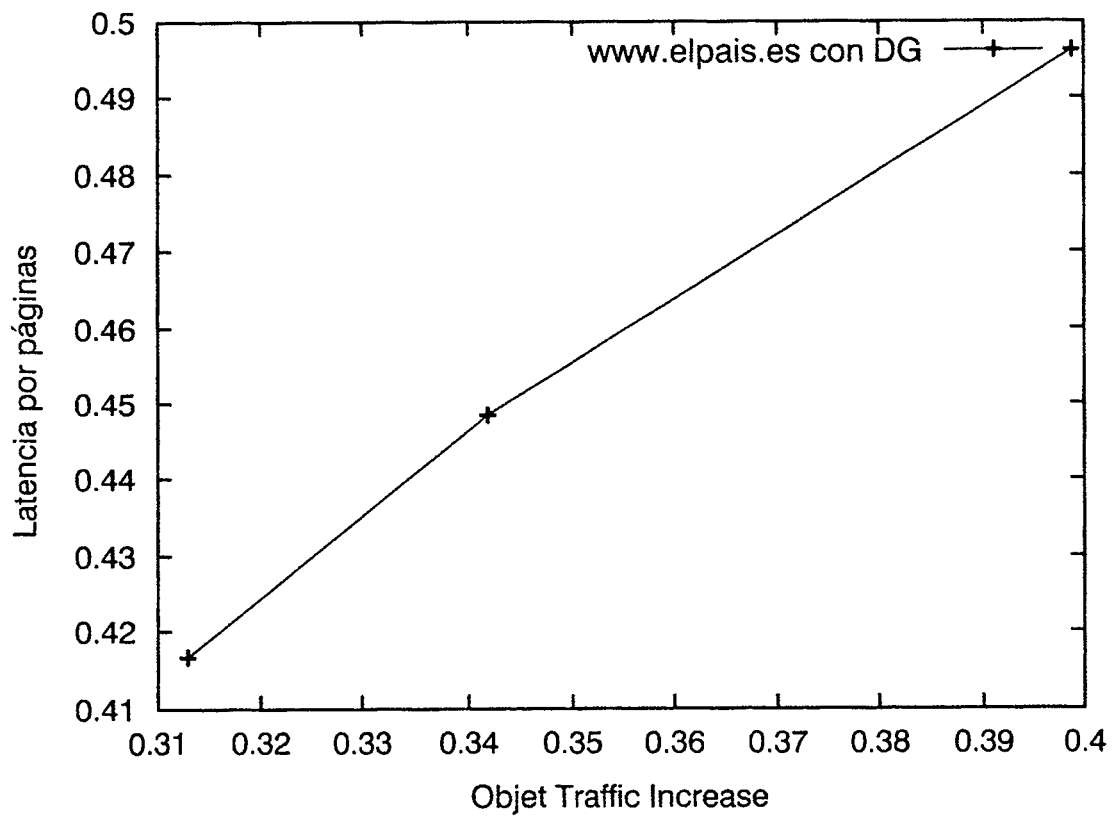


FIG. 4



**FIG. 5**



**FIG. 6**



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA

① ES 2 328 998

② Nº de solicitud: 200703321

③ Fecha de presentación de la solicitud: **05.12.2007**

④ Fecha de prioridad:

## INFORME SOBRE EL ESTADO DE LA TÉCNICA

⑤ Int. Cl.: **G06F 17/30** (2006.01)

### DOCUMENTOS RELEVANTES

Categoría	⑥ Documentos citados	Reivindicaciones afectadas
A	US 20020194382 A1 (KAUSIC et al.) 19.12.2002, todo el documento.	1-7
A	US 6766422 B2 (BEYDA) 20.07.2004, todo el documento.	1-7
A	US 20050033926 A1 (DUMONT) 10.02.2005, todo el documento.	1-7
A	22.04.2001, "Refreshment policies for web content caches". COHEN E; KAPLAN H. Proceedings IEEE Infocom 2001. Conference On Computer Communications. Twentieth Annual Joint Conference Of The IEEE Computer And Communications Society IEEE Piscataway, NJ, USA; ISBN 0-7803-7016-3.	1-7

#### Categoría de los documentos citados

X: de particular relevancia

Y: de particular relevancia combinado con otro/s de la  
misma categoría

A: refleja el estado de la técnica

O: referido a divulgación no escrita

P: publicado entre la fecha de prioridad y la de presentación  
de la solicitud

E: documento anterior, pero publicado después de la fecha  
de presentación de la solicitud

#### El presente informe ha sido realizado

para todas las reivindicaciones

para las reivindicaciones nº:

Fecha de realización del informe

30.10.2009

Examinador

M. Alvarez Moreno

Página

1/4

Documentación mínima buscada (sistema de clasificación seguido de los símbolos de clasificación)

G06F

Bases de datos electrónicas consultadas durante la búsqueda (nombre de la base de datos y, si es posible, términos de búsqueda utilizados)

INVENES, EPODOC, WPI

## OPINIÓN ESCRITA

Nº de solicitud: 200703321

Fecha de Realización de la Opinión Escrita: 30.10.2009

### Declaración

<b>Novedad (Art. 6.1 LP 11/1986)</b>	Reivindicaciones	1-7	<b>SÍ</b>
	Reivindicaciones		<b>NO</b>
<b>Actividad inventiva (Art. 8.1 LP 11/1986)</b>	Reivindicaciones	1-7	<b>SÍ</b>
	Reivindicaciones		<b>NO</b>

Se considera que la solicitud cumple con el requisito de **aplicación industrial**. Este requisito fue evaluado durante la fase de examen formal y técnico de la solicitud (Artículo 31.2 Ley 11/1986).

### Base de la Opinión:

La presente opinión se ha realizado sobre la base de la solicitud de patente tal y como ha sido publicada.

**1. Documentos considerados:**

A continuación se relacionan los documentos pertenecientes al estado de la técnica tomados en consideración para la realización de esta opinión.

Documento	Número Publicación o Identificación	Fecha Publicación
D01	US 20020194382 A1	19.12.2002
D02	US 6766422 B2	20.07.2004
D03	US 20050033926 A1	10.02.2005
D04	"Refreshment policies for web content caches". Cohen E; Kaplan H. Proceedings IEEE Infocom 2001. Conference On Computer Communications. Twentieth Annual Joint Conference Of The IEEE Computer And Communications Society IEEE Piscataway, NJ, USA; ISBN 0-7803-7016-3	22.04.2001

**2. Declaración motivada según los artículos 29.6 y 29.7 del Reglamento de ejecución de la Ley 11/1986, de 20 de marzo, de patentes sobre la novedad y la actividad inventiva; citas y explicaciones en apoyo de esta declaración**

Se considera que los documentos D01 a D04 constituyen el estado de la técnica. En todos los documentos citados el cliente realiza una petición condicional al servidor, indicándole las marcas de tiempo que tiene almacenadas en su caché. Es en este momento cuando el servidor compara dichas marcas con los valores actuales y decide si remite de nuevo la información. El documento D04 indica que se realiza un refresco proactivo, mediante el cual la caché realiza validaciones offline. En su apartado V describe dos tipos de políticas de refresco, pero en ambos casos es el cliente el encargado de transmitir al servidor las marcas de tiempo almacenadas, y es el servidor el responsable de identificar su validez.

Ninguno de los anteriores documentos muestra la característica de identificar, por parte del servidor, la relación de URLs junto con los identificadores de frescura asociados, previamente a su solicitud por el cliente; junto con la utilización posterior por parte del cliente de la anterior información para actualizar una marca de tiempo que permita, al propio cliente, identificar si el objeto contenido en la caché ha sido prevalidado.

Así, la invención reivindicada en las reivindicaciones 1 a 7 es, con referencia a los documentos anteriores, nueva y se considera que implica actividad inventiva.