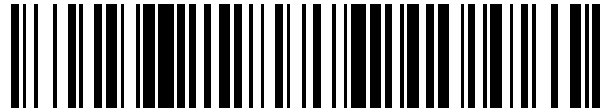


19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 737 073**

51 Int. Cl.:

**G06F 17/30** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **29.11.2012 E 12194865 (7)**

97 Fecha y número de publicación de la concesión europea: **06.03.2019 EP 2600262**

54 Título: **Procedimiento de visualización de páginas por medio de un navegador de un equipo como una caja descodificadora Proveedor de Servicios de Internet**

30 Prioridad:

**02.12.2011 FR 1161072**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:

**10.01.2020**

73 Titular/es:

**FREEBOX (100.0%)  
16 rue de la Ville l'Eveque  
75008 Paris, FR**

72 Inventor/es:

**VASSEUR, CLÉMENT y  
GOURNET, OLIVIER**

74 Agente/Representante:

**FORTEA LAGUNA, Juan José**

**ES 2 737 073 T3**

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

## DESCRIPCIÓN

Procedimiento de visualización de páginas por medio de un navegador de un equipo como una caja descodificadora Proveedor de Servicios de Internet.

5 [0001] La invención se refiere a un procedimiento para mostrar páginas de tipo HTML por un equipo dotado de un sistema operativo cerrado, estando este equipo conectado a un servidor remoto a través de un enlace de red.

10 [0002] En particular, puede tratarse de un equipo formado por una caja descodificadora ISP. Esta caja está conectada por una parte al servidor remoto del proveedor de acceso a través de un enlace de red IP (ADSL o fibra óptica), y está acoplada por otra parte a un televisor que permite visualizar diversas informaciones, en especial páginas HTML generadas localmente por un navegador integrado en el sistema del equipo. Un ejemplo de equipos de este tipo es la caja distribuida por el operador *Free*, París, Francia, bajo la denominación *Freebox Player* o *Freebox HD*. Esta caja integra descodificador, servidor doméstico, etc. En particular, asegura la descodificación de las diferentes señales del flujo de vídeo, que se podrán ver en el televisor al que está enchufado. También está acoplado a una caja de red situada cerca de la toma telefónica de abonado del usuario, que asegura las funciones de interfaz ADSL con la línea telefónica, de módem de telefonía IP, de rúter inalámbrico Wi-Fi, etc.

15 [0003] Este equipo integra además un navegador que permite visualizar en la pantalla del televisor diversas informaciones, formateadas en un lenguaje de programación de tipo HTML, XML, etc. Este navegador permite interrogar al servidor remoto (enviando una solicitud), descargar páginas con etiquetas en el idioma de programación elegido, formatear el contenido de estas páginas y visualizar en la pantalla del televisor el contenido resultante.

20 [0004] Esta página combina de hecho dos tipos de informaciones, a saber:

25 - informaciones recibidas por el navegador provenientes del servidor remoto (el servidor del proveedor de servicios de Internet, para el que el equipo actúa como cliente); y

- informaciones procedentes de una fuente interna del equipo, es decir, informaciones recibidas por el navegador provenientes del cliente (el equipo).

30 [0005] Estas últimas informaciones son informaciones dinámicas, es decir, no fijas.

[0006] Se trata típicamente de datos de estado del equipo, tales como, por ejemplo: versión del microprograma (*firmware*) del sistema del equipo, dirección IP del equipo, indicadores de diagnóstico, configuración de una red en sentido descendente del equipo, estado de un disco duro integrado o acoplado al equipo, registro de fecha y hora de programación de grabaciones, etc.

35 [0007] Teniendo en cuenta los recursos limitados del equipo, en particular en términos de capacidad del microprograma, a veces es deseable limitar las funcionalidades del navegador a un navegador muy simplificado que sólo soporte código estático. Además, algunos navegadores simplificados no integran un motor de lenguaje de script, por ejemplo, un motor *JavaScript*, que es demasiado voluminoso. Este es el caso, en particular, de los navegadores simplificados que sólo admiten código HTML estático. En todas estas hipótesis, la visualización de informaciones dinámicas (en el sentido indicado más arriba) no es posible. En efecto, el servidor no conoce el estado actual del equipo que le envía la solicitud. Por lo tanto, no puede integrar las informaciones dinámicas en la página que le enviará de vuelta.

[0008] El documento US 2009/0100361 A1 divulga un procedimiento para enriquecer aplicaciones instaladas en una caja, pero no proporciona una solución al problema mencionado anteriormente.

50 [0009] Una primera solución a este problema consiste en enviar al servidor, al mismo tiempo que la solicitud, las informaciones de estado a integrar en la página a visualizar.

[0010] Sin embargo, esto implica un coste adicional de tráfico en la red, especialmente en el sentido ascendente (de cliente a servidor), donde el rendimiento es bajo en el caso de los enlaces asimétricos (ADSL). Además, esta solución implica cargar el servidor con funciones de gestión de los equipos conectados al mismo, a menudo del orden de varios millones, lo que no es deseable desde el punto de vista del ahorro general de los recursos informáticos implementados por el proveedor de acceso.

60 [0011] Una segunda solución consiste en añadir un soporte de lenguaje de script al navegador, por ejemplo, integrando un motor *JavaScript*.

[0012] Un ejemplo de técnica de adaptación de un navegador para permitirle presentar páginas web que contengan a la vez informaciones estáticas e informaciones dinámicas se describe, por ejemplo, en el documento WO 99/08182 A1.

65

[0013] Dado que el navegador HTML admite un lenguaje de script, se hace posible entonces incorporar en la página HTML devuelta por el servidor scripts que se ejecutarán por el navegador después de descargar esa página, en particular para obtener de forma interna informaciones sobre el estado del equipo y cambiar el contenido de la página integrando en la misma esas informaciones.

[0014] Sin embargo, esta solución requiere modificar el navegador, lo que no siempre es posible o deseable, ya sea porque el código fuente no está disponible, ya sea porque tal modificación sería costosa en términos de desarrollo, o porque queremos mantener una versión muy simplificada y muy compacta del navegador. En efecto, las cajas descodificadoras de los proveedores de servicios de Internet integran generalmente un navegador no estándar, especialmente desarrollado por el diseñador de ese equipo con el fin de optimizar la gestión de los recursos informáticos del mismo.

[0015] El documento US2009/0100361 A1 describe una caja de tipo caja descodificadora ("*set-top box*") que incluye un software que permite la visualización en una pantalla de televisión de *widgets* dinámicos añadidos bajo diversas formas a la imagen emitida presente en la pantalla. Un renderizador combina informaciones provenientes de diferentes fuentes para formar el *widget* que se visualizará en la pantalla. La invención atañe a la situación muy particular en la que, por las razones explicadas anteriormente, se desea utilizar sólo un navegador muy simplificado, no susceptible de ejecutar un *script*, pero que, sin embargo, debe ser capaz de mostrar contenido dinámico. Esta limitación muy importante no se tiene en cuenta en las soluciones propuestas por los documentos WO 99/08182 A1 y US2009/0100361 A1 antes citados, que no dan ninguna indicación al entendido en la materia para resolver el problema particular de la invención.

[0016] La invención tiene por objeto superar las limitaciones de las técnicas conocidas expuestas más arriba proponiendo otra solución que permita resolver este problema generando, al nivel del equipo cliente, contenido dinámico en la página sin modificar el navegador y sin comunicar al servidor remoto las informaciones dinámicas del equipo cliente.

[0017] Esencialmente, la solución de la invención consiste en introducir en el equipo un servidor intermedio, local, de tipo HTTP, que admita un lenguaje de script capaz de recuperar las informaciones dinámicas a visualizar y de generar contenido para permitir que el navegador muestre una página que integre esas informaciones.

[0018] Del lado del servidor, para mostrar esas informaciones, el código de navegación será reemplazado por archivos que contengan el código fuente que será ejecutado del lado del cliente por el servidor intermedio.

[0019] El cliente (el navegador) en lugar de hacer una solicitud de red en el servidor remoto, hará una solicitud en el servidor local, el cual a su vez hará una solicitud en el servidor remoto. El código cargado en el servidor como resultado de esa solicitud se utilizará para crear contenido mediante la ejecución del lenguaje de script integrado en ese servidor intermedio.

[0020] En otras palabras, se trata de añadir un intermediario dentro del equipo (es decir, del lado del cliente, para el servidor remoto), sin modificar el navegador (por ejemplo, el cliente HTML).

[0021] El uso de un servidor intermedio (por ejemplo, *proxy* HTTP) es ciertamente conocido, pero generalmente en servidores remotos y no localmente dentro del cliente. Además, y sobre todo, la originalidad de la presente invención radica en el hecho de que este servidor intermedio se utiliza del lado del cliente con el fin de generar el contenido de las páginas a transmitir al navegador para su visualización, en función de informaciones dinámicas originadas por el cliente. Esta manera de proceder es especialmente ventajosa para los navegadores que no son modificables o que no se desea modificar, por las razones expuestas más arriba.

[0022] Otra ventaja de la presente invención es que permite utilizar cualquier lenguaje de script, o cualquier medio para generar contenido, sin estar limitado por el o los lenguajes proporcionados por un determinado navegador. Esta característica es particularmente interesante para los sistemas antiguos, o en el caso de integración de sistemas que no son modificables, o que no admiten el lenguaje de script.

[0023] Más precisamente, la invención propone un procedimiento de visualización de páginas por parte de un equipo cliente equipado con un sistema cerrado, conectado a un servidor remoto, integrando este sistema cerrado un navegador. Las páginas a mostrar combinan i) código de navegación estático recibido del servidor remoto y ii) código de navegación dinámico representativo de datos de estado del equipo, entregado por una fuente interna a ese equipo.

[0024] De manera característica de la invención, el navegador es un navegador sin lenguaje de script, que sólo admite código de navegación estático, y los datos de estado no se envían al servidor remoto. El equipo cliente está equipado con un servidor local. Además, el equipo cliente está equipado con un servidor local intermedio *proxy* equipado con un lenguaje de script e interconectado con el servidor remoto, el navegador y la fuente interna. El procedimiento consiste en ejecutar los siguientes pasos:

- 1°) emisión por el navegador de una solicitud de página, dirigida al servidor local intermedio;
- 2°) consulta al servidor remoto por el servidor local;
- 3°) obtención de vuelta por el servidor local de un archivo que contiene código de navegación estático y código fuente ejecutable;
- 4°) consulta a la fuente interna por el servidor local;
- 5°) obtención de vuelta por el servidor local de dichos datos de estado;
- 6°) ejecución por el servidor local de un script de generación del código de navegación dinámico a partir del código fuente ejecutable recibido en el paso 3°) y de los datos de estado;
- 7°) transmisión desde el servidor local al navegador de la combinación del código de navegación estático obtenido en el paso 3°) y del código de navegación dinámico generado en el paso 6°); y
- 8°) por el navegador, formateo en una página del código de navegación recibido del servidor local, para su visualización en una pantalla de visualización acoplada al equipo.

**[0025]** En una forma de implementación preferencial, la solicitud enviada por el navegador al servidor local en el paso 1°) contiene i) la dirección IP del servidor remoto y ii) la ruta de acceso al archivo a obtener en el paso 3°), y en el paso 2°) el servidor local consulta al servidor remoto con esa dirección y con esa ruta de acceso.

**[0026]** Además, el script de generación ejecutado en el paso 6°) está ventajosamente codificado en lenguaje *JavaScript*, conteniendo el servidor local un entorno de ejecución de ese lenguaje.

**[0027]** Ahora vamos a describir un ejemplo de implementación del dispositivo de la invención, con referencia a los dibujos adjuntos en los que las mismas referencias numéricas designan de una figura a la otra elementos idénticos o funcionalmente similares.

**[0028]** La Figura 1 es una representación esquemática de los diferentes bloques funcionales y elementos materiales implementados en el estado de la técnica. La Figura 2 es una representación esquemática similar a la de la Figura 1, en el caso del procedimiento de la invención.

**[0029]** En las figuras, la referencia 10 designa de forma general a un equipo cliente, como una caja decodificadora ISP, por ejemplo una caja de tipo *Freebox Player* distribuido por el operador *Free*, París, Francia.

**[0030]** Ese equipo está conectado a un servidor remoto 20 mediante un enlace IP, por ejemplo un enlace ADSL o de fibra óptica. Por otro lado, el equipo 10 está conectado a un televisor 30 para visualizar diferentes informaciones bajo la forma de páginas HTML generadas por un navegador HTML 12 incorporado en el equipo 10.

**[0031]** En la Figura 1, se ha ilustrado de forma esquemática la manera convencional de proceder según el estado de la técnica.

**[0032]** El navegador HTML 12, que es un navegador estático, consulta al servidor 20 enviando una solicitud (paso 1) de la forma:  
`http://A.B.C.D/.../fichier.html`,  
 donde "A.B.C.D." es la dirección IP del servidor remoto y "... " representa la ruta de acceso al archivo "*fichier.html*" a descargar.

**[0033]** El archivo recibido del servidor (paso 2) contiene código HTML estático, que está formateado por el navegador 12 y transmitido (paso 3) al televisor 30 para su visualización en forma de página HTML.

**[0034]** La solución de la invención consiste en modificar esta configuración, conservando el navegador HTML estático 12 e introduciendo en el equipo 12 un servidor HTTP intermedio local 14 . Este servidor local 14 está interconectado con el navegador 12 y con el servidor remoto 20 para servir de intermediario (*proxy*) entre esos dos elementos. El servidor local 14 también tiene acceso a una fuente 16 de informaciones dinámicas, interna al equipo 10, que permite recuperar informaciones dinámicas representativas de datos de estado de ese equipo tales como, como se ha indicado más arriba: versión del microprograma del sistema del equipo, dirección IP del equipo, indicadores de diagnóstico, configuración de una red descendente del equipo, estado de un disco duro integrado o acoplado al equipo, registro de fecha y hora de programación de grabaciones, etc.

**[0035]** Cuando el sistema del equipo pide al navegador 12 que muestre una página HTML, éste envía (paso 1) una solicitud al servidor local 14 (en lugar de enviarla directamente al servidor remoto como en el caso de la Figura 1). Esta solicitud será en la forma:  
`http://127.0.0.1/remote/A.B.C.D/.../fichier.html`

**[0036]** La dirección IP "127.0.0.1" indica que la solicitud debe aplicarse localmente, y el resto de la solicitud permite especificar la dirección IP del servidor remoto (A.B.C.D.), así como la ruta de acceso del archivo a descargar. A partir de esa solicitud, el servidor local genera por su lado otra solicitud, dirigida al servidor remoto 20 (paso 2), en la forma:  
`http://A.B.C.D/.../fichier.html`.

**[0037]** Al recibir esta solicitud, el servidor remoto 30 devuelve (paso 3) un archivo que contiene etiquetas HTML para el código HTML estático, así como un código fuente ejecutable que permitirá, del lado cliente, la generación del contenido de la página.

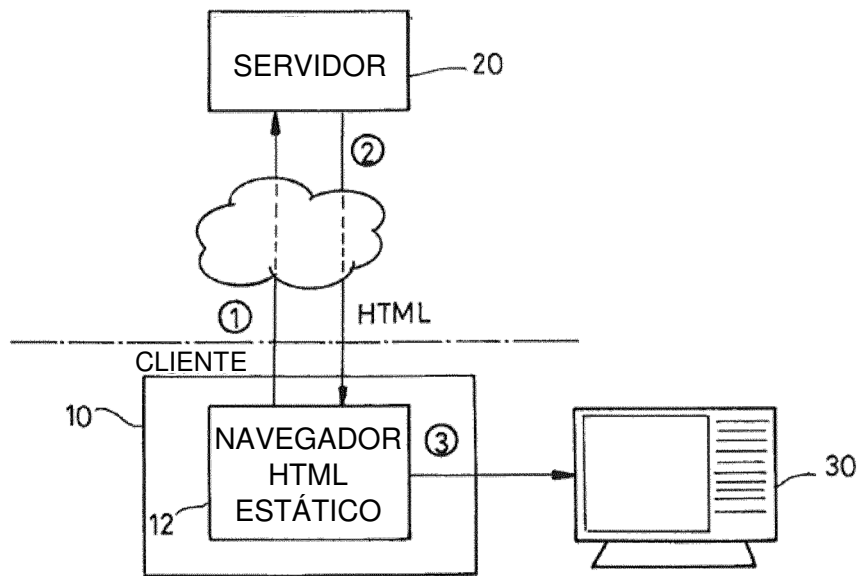
5 **[0038]** Al recibir ese archivo, el servidor local 14 consulta a la fuente interna 16 (paso 4) para recuperar (paso 5) las informaciones dinámicas representativas de los datos de estado del equipo a integrar en la página a visualizar.

**[0039]** El script de código fuente recibido del servidor 20 es ejecutado (paso 6) por el servidor local 14, que genera así código HTML transmitido al navegador 12 (paso 7).

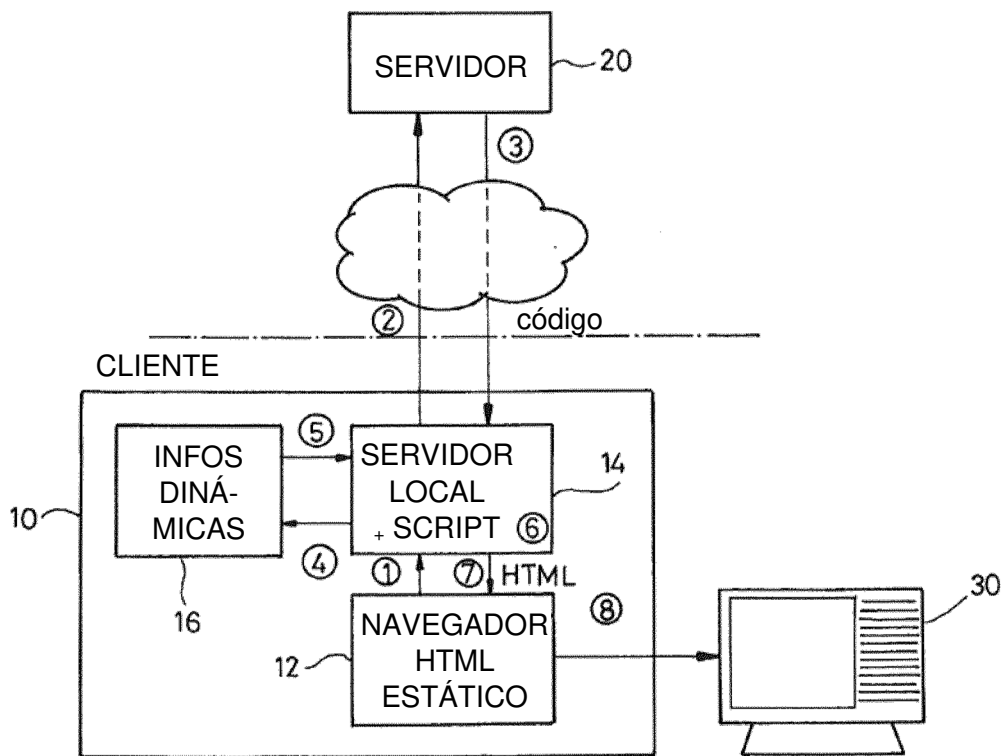
10 **[0040]** El navegador 12 puede entonces formatear el código HTML recibido del servidor local y enviarlo al televisor para su visualización en la pantalla de visualización (paso 8).

**REIVINDICACIONES**

- 5 1. Un procedimiento de visualización de páginas por un equipo cliente (10) equipado de un sistema cerrado, conectado a un servidor remoto (20), integrando dicho sistema cerrado un navegador (12), combinando las páginas a visualizar combinando i) código de navegación estática recibido del servidor remoto y ii) código de navegación dinámico representativo de datos de estado del equipo, proporcionado por una fuente (16) interna a dicho equipo, estando este procedimiento **caracterizado por que:**
- 10 - el navegador es un navegador sin lenguaje de script, que sólo admite código de navegación estático;  
 - los datos de estado no se envían al servidor remoto;  
 - el equipo cliente (10) está equipado con un servidor local intermedio *proxy* (14) provisto de un lenguaje de script e interconectado con el servidor remoto (20), el navegador (12) y la fuente interna (16),  
 - y **por que** comprende los pasos siguientes:
- 15 1° emisión por el navegador de una solicitud de página, dirigida al servidor local intermedio;  
 2° consulta al servidor remoto por el servidor local ;  
 3° obtención de vuelta por el servidor local de un archivo que contiene código de navegación estático y código fuente ejecutable;  
 20 4° consulta a la fuente interna por el servidor local;  
 5° obtención de vuelta por el servidor local de dichos datos de estado;  
 6° ejecución por el servidor local de un script de generación del código de navegación dinámico a partir del código fuente ejecutable recibido en el paso 3°) y de los datos de estado;  
 7° transmisión desde el servidor local al navegador de la combinación del código de navegación estático obtenido en el paso 3°) y del código de navegación dinámico generado en el paso 6°); y  
 25 8° por el navegador, formateo en una página del código de navegación recibido del servidor local, para su visualización en una pantalla de visualización (30) acoplada al equipo
- 30 2. El procedimiento de la reivindicación 1, en el que los datos de estado son datos del grupo formado por: versión del microprograma (*firmware*) del sistema del equipo, dirección IP del equipo, indicadores de diagnóstico, configuración de una red en sentido descendente del equipo, estado de un disco duro integrado o acoplado al equipo, registro de fecha y hora de programación de grabaciones.
- 35 3. El procedimiento de la reivindicación 1, en el que el equipo es una caja decodificadora ISP apta para ser acoplada a un televisor para la visualización de la página.
- 40 4. El procedimiento de la reivindicación 1, en el que la solicitud enviada por el navegador al servidor local en el paso 1°) contiene i) la dirección IP del servidor remoto y ii) la ruta de acceso al archivo a obtener en el paso 3°), y en el que en el paso 2°) el servidor local consulta al servidor remoto con esa dirección y esa ruta de acceso.
5. El procedimiento de la reivindicación 1, en el que el script de generación ejecutado en el paso 6°) está codificado en lenguaje *JavaScript*, y en el que el servidor local contiene un entorno de ejecución de *JavaScript*.



**FIG. 1**  
(ESTADO DE LA TÉCNICA)



**FIG. 2**