

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 391 983**

51 Int. Cl.:
H04M 3/493 (2006.01)
G10L 15/18 (2006.01)
G10L 15/26 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **07019562 .3**
96 Fecha de presentación: **30.11.2001**
97 Número de publicación de la solicitud: **1881685**
97 Fecha de publicación de la solicitud: **23.01.2008**

54 Título: **Procedimiento y sistema para la activación por voz de páginas web**

30 Prioridad:
01.12.2000 US 250809 P

45 Fecha de publicación de la mención BOPI:
03.12.2012

45 Fecha de la publicación del folleto de la patente:
03.12.2012

73 Titular/es:
**THE TRUSTEES OF COLUMBIA UNIVERSITY IN
THE CITY OF NEW YORK (100.0%)
16TH STREET AND BROADWAY
NEW YORK, NY 10027, US**

72 Inventor/es:
**CHARNEY, MICHAEL L. y
STARREN, JUSTIN**

74 Agente/Representante:
ARIAS SANZ, Juan

ES 2 391 983 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 y sistema para la activación por voz de páginas web.

Antecedentes de la invención

5 En la pasada década, los sistemas de reconocimiento automático de voz (ASR) han evolucionado hasta el punto que puede obtenerse un alto grado de precisión de reconocimiento por parte de los sistemas ASR instalados en ordenadores personales y estaciones de trabajo de precio moderado. Esto ha dado lugar a un aumento en el número de sistemas ASR disponibles para clientes y aplicaciones industriales.

10 Los sistemas ASR se basan en gramáticas de voz para reconocer comandos de voz introducidos mediante un micrófono y actuar sobre tales comandos. Las gramáticas de voz se dividen en dos categorías: gramáticas basadas en reglas y gramáticas de libre expresión (*free speech grammar*). Las gramáticas basadas en reglas permiten el reconocimiento de un conjunto limitado de frases predefinidas. Cada gramática basada en reglas, si se invoca, hace que se produzca un evento o conjunto de eventos. Una gramática basada en reglas se invoca si una locución, introducida con un micrófono, coincide con una plantilla de voz correspondiente a una frase almacenada en el conjunto de frases predefinidas. Por ejemplo, el usuario puede decir "guardar archivo" cuando edita un documento
15 en un programa de procesamiento de texto para invocar el comando 'guardar'. Por otro lado, las gramáticas de libre expresión reconocen grandes conjuntos de palabras en un dominio dado, tal como inglés financiero. Estas gramáticas se utilizan generalmente para aplicaciones de dictado, donde algunos ejemplos de estos sistemas son *Dragon Naturally Speaking* e *IBM Viavoice 7 Millennium*. Los sistemas ASR también tienen incorporadas capacidades de conversión de texto a voz (TTS) que permiten a los sistemas ASR reproducir oralmente un texto
20 generado gráficamente utilizando una voz sintetizada. Por ejemplo, un sistema ASR puede leer en voz alta un párrafo resaltado en un procesador de texto a través de los altavoces.

25 Los sistemas ASR se han integrado en los navegadores web para crear navegadores web que soportan voz; véase, por ejemplo, el documento "*Grammar Representation Requirements for Voice Markup Languages*", del W3C. Los navegadores web que soportan voz permiten al usuario navegar por Internet utilizando comandos de voz que invocan gramáticas basadas en reglas. Algunos de los comandos de voz utilizados por estos navegadores incluyen locuciones que hacen que el software ejecute comandos tradicionales utilizados por los navegadores web. Por ejemplo, si el usuario dice "página de inicio" con un micrófono, un navegador que soporta voz ejecutará las mismas rutinas que el navegador web que soporta voz ejecutaría si un usuario pulsase el botón "página de inicio" del navegador web que soporta voz. Además, algunos navegadores web que soportan voz crean gramáticas basadas
30 en reglas en función del contenido de páginas web. Cuando una página web se descarga y se muestra, algunos navegadores web que soportan voz crean gramáticas basadas en reglas en función de los enlaces contenidos en la página web. Por ejemplo, si la página web muestra el enlace "página de inicio de la empresa", este tipo de navegador web que soporta voz creará una gramática basada en reglas, cuando se muestra la página web, de manera que si un usuario dice la frase "página de inicio de la empresa" con un micrófono, el navegador web que soporta voz mostraría la página web asociada al enlace. Un inconveniente de este enfoque es que las reglas generadas a partir del contenido de páginas web son fijas durante largos periodos de tiempo, ya que las páginas web no se rediseñan frecuentemente. Además, las gramáticas basadas en reglas se generan a partir del contenido de páginas web, lo que está orientado principalmente a la visualización. Además, estos sistemas hacen que el usuario se limite a decir lo que aparece en la pantalla.

40 Las páginas web también pueden incluir elementos de audio, los cuales generan una salida de sonido. Actualmente, las páginas web pueden incorporar elementos de audio en sus páginas web de dos maneras. La primera manera de incorporar un elemento de audio es utilizar el contenido de archivos de audio wav para proporcionar una voz humana en una página web. La utilización de archivos de audio wav permite al diseñador de páginas web diseñar las partes visuales y de audio de la página web de manera independiente, pero esta libertad y funcionalidad añadida
45 tienen un alto precio. El ancho de banda requerido para transferir archivos de sonido binarios a través de Internet al usuario final es grande. La segunda manera de incorporar un elemento de audio es incorporar la funcionalidad de un sistema ASR. Los navegadores web que soportan voz pueden utilizar la funcionalidad TTS de un sistema ASR para que el ordenador "diga" el contenido de una página web. La utilización de este enfoque hace que el ancho de banda necesario para ver la página con o sin el elemento de audio sea aproximadamente el mismo, pero limita al contenido de la página web lo que el navegador web puede decir.
50

VoiceXML (VXML) es otra opción para un diseñador de páginas web. VXML permite a un usuario navegar por un sitio web solamente mediante la utilización de comandos de audio utilizados normalmente con un teléfono. VXML requiere que un conversor TTS lea una página web a un usuario convirtiendo la página web visual en una expresión de audio de la página web. El usuario navega por la web diciendo los enlaces a los que el usuario desea acceder.
55 Con este enfoque, un usuario puede navegar por Internet utilizando solamente la voz del usuario, pero el contenido de audio se genera normalmente a partir del contenido de páginas web, el cual está diseñado principalmente para una interpretación visual, eliminándose la interfaz visual de la experiencia del usuario.

Por consiguiente, existe una constante necesidad de crear de manera independiente un componente de audio de una página web que no necesite una gran cantidad de ancho de banda de transmisión y que exista junto con el

componente visual de una página web.

Sumario de la invención

La presente invención se define en las reivindicaciones independientes.

Breve descripción de los dibujos

5 Objetos, características y ventajas adicionales de la invención resultarán evidentes a partir de la siguiente descripción detallada tomada junto con las figuras adjuntas que muestran realizaciones ilustrativas de la invención, en las que:

La FIG. 1 es un diagrama de bloques que ilustra un sistema de la técnica anterior.

10 La FIG. 2A es un diagrama de flujo que ilustra un proceso de generación de contenido de páginas web según la presente invención.

La FIG. 2B es un diagrama de bloques de un sistema de navegador web que soporta un lenguaje de marcado de hipervoz (HSML) según la presente invención.

La FIG. 3A es un diagrama de flujo que ilustra un hilo (*thread*) que supervisa los mensajes enviados a un puerto particular de un ordenador según la presente invención.

15 La FIG. 3B es una tabla que ilustra las condiciones bajo las cuales una cabecera, un delimitador o un pie se añaden al principio o al final de los datos recibidos desde el servidor web según la presente invención.

La FIG. 4 es un diagrama de flujo que ilustra un hilo que recibe datos desde el navegador web y datos enviados al navegador web asociados a las páginas web que se invocaron por el navegador web según la presente invención.

20 La FIG. 5 es un diagrama de flujo que ilustra un hilo que es el procedimiento a través del cual el motor HSML reacciona para recibir una serie de etiquetas desde el motor ASR según la presente invención.

La FIG. 6 es un diagrama de flujo que ilustra un hilo que recibe etiquetas de subreglas desde el motor ASR y datos enviados al navegador web asociados a páginas web que se invocaron por el motor ASR según la presente invención.

25 La FIG. 7 es un diagrama de flujo que ilustra un proceso para crear, almacenar y transmitir gramáticas basadas en reglas y gramáticas TTS según la presente invención.

La FIG. 8 es un diagrama de flujo que ilustra un proceso para analizar sintácticamente conjuntos de bloques de gramáticas de lenguaje de marcado de hipervoz según la presente invención.

La FIG. 9 es un diagrama de flujo que ilustra un proceso para analizar sintácticamente bloques de gramáticas de lenguaje de marcado de hipervoz según la presente invención.

30 La FIG. 10 es un diagrama de flujo que ilustra un proceso para analizar sintácticamente gramáticas de conversión de texto a voz de lenguaje de marcado de hipervoz según la presente invención.

La FIG. 11 es un diagrama de flujo que ilustra un proceso para analizar sintácticamente gramáticas basadas en reglas de lenguaje de marcado de hipervoz según la presente invención.

35 A lo largo de las figuras, a no ser que se indique lo contrario, los mismos números y caracteres de referencia se utilizan para denotar características, elementos, componentes o partes similares de las realizaciones ilustradas. Además, aunque la presente invención se describirá a continuación en detalle con referencia a la figuras y con relación a las realizaciones ilustrativas, varios cambios, modificaciones, alteraciones y sustituciones con respecto a las realizaciones descritas resultarán evidentes para los expertos en la técnica sin apartarse del alcance verdadero de la presente invención definida por las reivindicaciones adjuntas.

40 Descripción detallada de la invención

La FIG. 1 ilustra un sistema de la técnica anterior 100 para visualizar e interactuar con páginas web almacenadas en un servidor web 102. Se proporciona un servidor web 102 que incluye una CPU 104, una interfaz de red 106 y una unidad de almacenamiento de datos 108. La unidad de almacenamiento de datos 108 contiene información que describe una o más páginas web. Una conexión de red 109 conecta el servidor web 102 a una red de comunicaciones 110 a través de la interfaz de red 106, permitiendo al servidor web 102 comunicarse con otros dispositivos de la red de comunicaciones 110. En una determinada realización, la red de comunicaciones 110 es Internet.

Se proporciona un ordenador cliente 120 que incluye una CPU 122, una interfaz de red 124 y una unidad de almacenamiento de datos 126. En una determinada realización, la unidad de almacenamiento de datos 126 puede

ser una memoria. Una conexión de red 111 conecta el ordenador cliente 120 a la red de comunicaciones 110 a través de la interfaz de red 124, permitiendo al ordenador cliente 120 comunicarse con otros dispositivos de la red de comunicaciones 110. El ordenador cliente 120 también está conectado a varios dispositivos de entrada/salida tales como un teclado 128, un ratón 136, un micrófono 130, un altavoz 132 y una pantalla 134, normalmente a través de un bus de sistema 138.

También se proporciona un asistente digital personal (PDA) 140, que incluye una CPU 142 y una memoria 144. El PDA 140 está conectado a y puede comunicarse con otros dispositivos de la red de comunicaciones 110 a través de una conexión de red 148, por medio de una interfaz de red 146.

Para ver una página web, un usuario abre un navegador web, almacenado en una unidad de almacenamiento de datos de un ordenador o en una memoria de un PDA, por ejemplo la unidad de almacenamiento de datos 126 del ordenador cliente 120. Una vez que se haya abierto el navegador web, el usuario puede teclear un localizador universal de recursos (URL), lo que provoca que el ordenador cliente 120 envíe una solicitud a través de la red de comunicaciones 110 relacionada con los archivos de datos que describen el contenido de la página web identificada por el URL. Un servidor web, que almacena los archivos de datos de la página web identificada por el URL, por ejemplo el servidor web 102, recibe la solicitud y envía al ordenador cliente 120 los archivos de datos que describen el contenido de la página web. Los archivos de datos pueden incluir archivos de lenguaje de marcado de hipertexto (HTML), archivos de páginas de servidor activas, archivos de sonido, archivos de vídeo, etc. Después, el navegador web muestra la página web y reproduce archivos de vídeo o de audio en el ordenador cliente 120 según especifique el lenguaje de marcado. El lenguaje de marcado especifica dónde y cuándo colocar o reproducir contenido de texto, de vídeo o de audio.

La FIG. 2A ilustra un proceso 200 para diseñar una página web. El proceso 200 comienza cuando un diseñador de páginas web ejecuta una herramienta de diseño de páginas web en la etapa 202. La herramienta de diseño de páginas web ayuda al diseñador de páginas web durante el proceso de diseño de una página web. La herramienta de diseño de páginas web permite al diseñador de páginas web especificar elementos de la página web a un alto nivel, generándose posteriormente un lenguaje de marcado de bajo nivel en función de los elementos especificados de la página web.

El diseñador de páginas web empieza a diseñar y a comprobar la parte visual de una página web en la etapa 204. Si la parte visual de la página web ya existe y no es necesario actualizarla, el proceso 200 avanza directamente hasta la etapa 206. El diseño y la comprobación de la parte visual de la página web se conocen ampliamente en la técnica. Después de haberse diseñado y comprobado la parte visual de la página web, se ejecuta el bloque de proceso 206.

En la etapa 206 tiene lugar el proceso de diseño y comprobación de una parte de audio de la página web. El diseñador de páginas web puede añadir un lenguaje de marcado de hipervoz (HSML) al lenguaje de marcado de páginas web para mejorar la parte visual de la página web con un componente de audio. El diseñador de páginas web puede añadir HSML directamente o especificarlo a un alto nivel utilizando una herramienta de diseño de páginas web. Si la parte de audio de la página web ya existe y no es necesario actualizarla, el proceso 200 avanza directamente hasta la etapa 208. En la etapa 208, las páginas web pueden hacerse disponibles para los usuarios conectados a la red de comunicaciones 110. Una vez que las páginas web estén disponibles al público, el proceso 200 finaliza.

En una realización alternativa, la parte de audio de la página web puede diseñarse antes de que se diseñe la parte visual de la página web. En otra realización alternativa, la parte de audio y la parte visual de la página web pueden diseñarse al mismo tiempo. En otra realización alternativa adicional, la parte de audio y la parte visual de la página web pueden diseñarse en etapas diferentes.

En una realización alternativa, la parte de audio y la parte visual de la página web pueden generarse mediante una interfaz de pasarela común.

Según la presente invención, el HSML especifica gramáticas basadas en reglas de reconocimiento de voz y gramáticas TTS para la página web. Las gramáticas basadas en reglas de reconocimiento de voz permiten el reconocimiento de un conjunto limitado de frases de voz predefinidas introducidas en el ordenador con un micrófono o similar, que a su vez invocan un evento o conjunto de eventos. Las gramáticas TTS definen texto que puede proporcionarse como contenido audible a través de los altavoces asociados al ordenador que muestra una página web.

Las gramáticas basadas en reglas HSML pueden referirse, pero sin ser necesario que sean idénticas, al texto que aparece en la pantalla. Si se añaden gramáticas basadas en reglas HSML al lenguaje de marcado de una página web mostrada en la pantalla, los usuarios pueden reproducir oralmente de manera aproximada el texto mostrado en la pantalla en lugar de reproducir oralmente de manera literal el texto que aparece en pantalla. Una primera gramática basada en reglas HSML a modo de ejemplo es la siguiente: `<hsm1:JSGFRule href="http://www.columbia.edu" tag="reload" ID = "reload">![CDATA[public <reload> = (reload) {reload}]]` `</hsm1:JSGPRule>`. La primera gramática basada en reglas HSML a modo de ejemplo define que se ejecute un comando, "http://www.columbia.edu", si una frase oral introducida en el ordenador coincide con la frase "reload".

Cada gramática basada en reglas HSML tiene tres partes que pueden configurarse: la parte de comando, la parte de etiqueta y la parte de frase. La parte de comando define el comando que va a ejecutarse una vez que se invoque la gramática basada en reglas HSML. En la primera gramática basada en reglas HSML a modo de ejemplo, la parte de comando es la siguiente: href="http://www.columbia.edu". El comando para la primera gramática basada en reglas HSML a modo de ejemplo es "http://www.columbia.edu". La parte de etiqueta define una manera única de identificar una regla particular de una gramática basada en reglas HSML. En la primera gramática basada en reglas HSML a modo de ejemplo, la parte de etiqueta es la siguiente: tag="reload". La etiqueta para la primera gramática basada en reglas HSML a modo de ejemplo es "reload". La parte de frase define el conjunto de frases que invocarán a la gramática basada en reglas HSML. En la primera gramática basada en reglas HSML a modo de ejemplo, la parte de frase es la siguiente: "CDATA[public <reload> = (reload) {reload} ". La frase para la primera gramática basada en reglas HSML es "reload".

Otro conjunto, más complicado, de gramáticas basadas en reglas HSML es el siguiente: <hsml:block> <hsml:JSGFRule href="http://www.columbia.edu/scripts/cgi/test.cgi" tag="examprep" ID = "examprep"><![CDATA[public <examprep> = [please] (go to| example) (<rad>|<lab>) report {examprep};]]></hsml:JSGFRule> <hsml:JSGFRule tag="RAD"><![CDATA[public <rad> = radiology {RAD};]]></hsml:JSGFRule> <hsml:JSGFRule tag="LAB"><![CDATA[public <lab> = (laboratory | chemistry) {LAB};]]></hsml:JSGFRule> </hsml:block>. Esta gramática proporciona un conjunto anidado de reglas. Esta gramática basada en reglas HSML define un conjunto de comandos, uno de los cuales se ejecutará si una frase oral introducida en el ordenador coincide con una frase asociada de un conjunto de frases. El comando de nivel superior se define mediante la regla HSML "<hsml:JSGFRule href="http://www.columbia.edu/scripts/cgi/test.cgi" tag="examprep" ID = "examprep"> <![CDATA[public <examprep> = [please] (go to | example) (<rad>|<lab>) report {examprep};]]></hsml:JSGFRule>", la cual define el conjunto de frases y el comando de nivel superior "http://www.columbia.edu/scripts/cgi/test.cgi".

El conjunto de frases se define mediante la siguiente parte de la parte de frase de la segunda gramática basada en reglas HSML a modo de ejemplo: [please] (go to | example) (<rad>|<lab>). La frase empieza con la palabra "please" entre corchetes. Una palabra entre corchetes denota una palabra opcional. La regla se invocará tanto si la frase oral empieza por la palabra "please" como si no.

La palabra opcional "please" va seguida de (go to| example). El paréntesis agrupa palabras entre sí. Las palabras "go to" y "example" están separadas por un símbolo "|". El símbolo "|" es un símbolo "OR", el cual indica que puede utilizarse "go to" o "example" para invocar la regla. Por lo tanto, la palabra opcional "please" debe ir seguida de "go to" o de "example" para invocar esta regla HSML.

Las palabras obligatorias "go to" o "example" van seguidas de (<rad>|<lab>). Esta parte de la frase contiene variables. Las variables se definen en subreglas ubicadas en otra parte del bloque HSML. La variable <rad> se define mediante la subregla <hsml:JSGFRule tag="RAD"><![CDATA[public <rad> = radiology {RAD};]]> </hsml:JSGFRule>. Por lo tanto, la variable <rad> se define como "radiology". Si la palabra "radiology" va detrás de "go to" o "example", un separador, en este caso "?", se concatena al comando de nivel superior y después la etiqueta de subregla RAD se concatena al comando de nivel superior, de manera que el comando pasa a ser "http://www.columbia.edu/scripts/cgi/test.cgi?RAD". En una realización alternativa a modo de ejemplo, el separador puede ser "/". La variable <lab> se define mediante la subregla <hsml:JSGFRule tag = "LAB "><![CDATA[public <lab> = (laboratory |chemistry) {LAB};]]></hsml:JSGFRule>. La variable <lab> se define como "laboratory" o "chemistry". Si la palabra "laboratory" o "chemistry" va después de "go to" o "example", un separador, en este caso "?", se concatena al comando de nivel superior y después la etiqueta de subregla LAB se concatena al comando de nivel superior, de manera que el comando pasa a ser "http://www.columbia.edu/scripts/cgi/test.cgiexamprep?LAB".

Las gramáticas TTS definen texto que puede proporcionarse como voz sintética a través de altavoces. Una gramática TTS de HSML de este tipo es la siguiente: <hsml:JSMML> <![CDATA[reloading]]></hsml:JSMML>. Esta gramática TTS proporciona como salida la frase "reloading" como una voz sintética a través de los altavoces. La gramática TTS puede invocarse cuando se invoca una gramática basada en reglas HSML, cuando se carga una página, cuando se carga un sitio web, cuando se sale de una página o cuando se sale de un sitio web.

La FIG. 2B ilustra un diagrama de bloques funcionales de un sistema 200 para visualizar e interactuar con páginas web que tienen contenido HSML. Para ver una página web que tiene contenido HSML, un usuario puede ejecutar un procesador HSML 254, el cual ejecuta a su vez un navegador web 256 y un motor ASR 258 en el ordenador 120. El procesador HSML 254 actúa como un proxy para el navegador web 256. El navegador web 256 debe tener inhabilitado el almacenamiento en caché y debe soportar el marcado "x-mixed replace". En una realización a modo de ejemplo, el navegador web 256 puede ser Netscape Navigator. En la presente realización a modo de ejemplo, el motor ASR 258 puede ser el IBM Viavoice Version 7 Millennium. Otros motores ASR que se ajustan a la norma Java Speech Application Programa Interface, también pueden sustituir al IBM Viavoice Version 7 Millennium. En una determinada realización, puede utilizarse un conjunto de comandos (script) para ejecutar el procesador HSML 254 y el navegador web 256 de manera individual.

En una realización alternativa, el navegador web 256 puede ser Microsoft Internet Explorer. En otra realización alternativa, el procesador HSML 254 se comunica con el navegador web 256, de manera que no es necesario un

marcado "x-mixed replace" para transportar información de control.

También puede incluirse un monitor visual con el navegador que soporta voz. El monitor visual ayuda en la interacción del usuario con las páginas activadas por voz. El monitor visual incluye de manera ventajosa una barra de estado que muestra un nivel de audio de salida, un nivel de audio de entrada, un indicador de estado de motor de voz, un indicador de síntesis de voz, un identificador de reglas, un indicador de estado de datos, un indicador de errores, una línea activa de gramática de página, un área de gramáticas activas, etc. Esto permite al usuario recibir información sobre cómo o por qué su voz se reconoce o no se reconoce. El monitor visual puede implementarse en una ventana diferente al navegador o puede estar integrado como parte del propio navegador. Este monitor extiende el entorno multimedia del navegador de un entorno de contenido gráfico y de texto a un entorno interactivo de texto y voz, generado dinámicamente, de plataforma cruzada.

El nivel de audio de salida informa al usuario sobre el ajuste del volumen de salida. El nivel de audio de entrada indica el volumen actual de la voz del usuario cuando el usuario habla por el micrófono 130 (mostrado en la Fig. 1). El indicador de estado de motor de voz informa al usuario sobre el estado del procesador HSML 254. El procesador HSML 254 puede estar en modo escucha, en modo dormido o inactivo. Si el procesador HSML 254 está en modo escucha, está aceptando todos los comandos del sistema y de las páginas. Si el procesador HSML 254 está en modo dormido, está esperando un comando de sistema específico "despertar" para activar el procesador HSML 254, el cual hará que el indicador de estado de motor de voz pase al modo escucha. Si el procesador HSML 254 está inactivo, no se acepta ningún comando de voz. El indicador de síntesis de voz informa al usuario de si el procesador HSML 254 reaccionará a las gramáticas TTS de HSML asociadas a la página web activa y a los comandos del sistema, solamente a los comandos de salida de voz del sistema o a ningún comando de salida de voz. El identificador de reglas muestra una regla HSML invocada durante un corto periodo de tiempo después de que se haya invocado la gramática HSML. El indicador de estado de datos informa al usuario de si el procesador HSML 254 está cargando datos, de si el procesador HSML 254 está cargando datos de voz, de si el procesador HSML 254 ya ha cargado la parte visual de la página web, o de si el procesador HSML 254 ya ha cargado la página web y los datos de voz. El indicador de errores informa al usuario de cualquier error que haya detectado el procesador HSML 254. La línea activa de gramática de páginas informa al usuario de si existe una gramática de páginas actual y si está activa. El área de gramáticas activas enumera las reglas de gramática HSML que están actualmente activas hasta un determinado número máximo de gramáticas HSML. La apariencia de estas reglas en el área de gramáticas activas permite a los usuarios comprender fácilmente la función y el diseño de una página.

No es necesario que se muestren todos los conjuntos de reglas de grandes conjuntos de reglas. No es necesario que un menú gráfico nombre cada regla HSML o cada gramática basada en reglas HSML. Las reglas HSML individuales o las gramáticas HSML pueden indicar el nivel de importancia de cada regla o gramática, de manera que el monitor visual puede seleccionar de manera inteligente qué reglas y gramáticas son importantes y deberían mostrarse. Cuando se muestra una regla, la pantalla muestra la frase o frases que invocan a la regla. Por ejemplo, si se ha mostrado la gramática basada en reglas HSML `<hsml:block><hsml:JSGFRule href="http://www.columbia.edu" tag="reload" ID = "reload"><![CDATA[public <reload> = (reload) {reload}]]` `</hsml:JSGFRule></hsml:block>`, la pantalla mostrará la frase "reload". Si más de una frase puede invocar a la gramática basada en reglas HSML, la pantalla muestra todas las frases que pueden invocar a la gramática basada en reglas HSML. En una realización alternativa, la pantalla muestra las frases que pueden invocar a la gramática basada en reglas HSML, pero no muestra palabras opcionales. Por ejemplo, si la frase "please reload" invocó una gramática basada en reglas HSML, pero la palabra "please" era opcional, la pantalla mostrará solamente la palabra "reload", no "please reload" y "reload".

En una realización alternativa, el procesador HSML 254 puede ejecutarse por separado con respecto al navegador web 256 y puede interactuar con un navegador web que ya esté ejecutándose.

El motor ASR 258 puede iniciar una solicitud para una página web junto con el procesador HSML o el navegador web 256. Si el motor ASR 258 inicia la solicitud, al menos una gramática basada en reglas de reconocimiento de voz se habrá invocado en el motor ASR 258. El motor ASR 258 transmite una serie de etiquetas al procesador HSML 254, que a su vez envía una solicitud al servidor web 102 (mostrado en la Fig. 1). Este proceso se muestra en mayor detalle en las FIG. 3A, 3B y 5. Con referencia a la Fig. 2B, el navegador web 256 inicia la solicitud y transmite el URL al procesador HSML 254. El procesador HSML 254 recibe el URL desde el navegador web 256 y envía una solicitud al servidor web 102 (mostrado en la Fig. 1). Este proceso se muestra en mayor detalle en las FIG. 3A, 3B y 4.

El ordenador 120 (mostrado en la Fig. 1) recibe una respuesta a la solicitud enviada por el procesador HSML 254 en forma de datos que representan la página web correspondiente al URL. El ordenador 120 comunica al procesador HSML 254 los datos, incluyendo datos de marcado, que representan la página web correspondiente al URL. El procesador HSML 254 procesa los datos de marcado y crea, almacena y transmite al motor ASR 258 las gramáticas basadas en reglas y las gramáticas TTS basadas en los datos de marcado. El proceso para crear, almacenar y enviar gramáticas basadas en reglas y gramáticas TTS se describe en mayor detalle posteriormente con referencia a las FIG. 3A y 3B.

En una realización alternativa, la funcionalidad del procesador HSML 254 puede estar incorporada en el navegador web 256. En otra realización alternativa, la funcionalidad del procesador HSML 254 puede estar incorporada en el

navegador web 256 como un módulo o como subrutinas. En otra realización alternativa adicional, la funcionalidad del motor ASR 258 puede estar incorporada en el navegador web 256.

La FIG. 3A ilustra un hilo 300 que supervisa los mensajes enviados a un puerto particular de un ordenador. Un hilo no es un proceso independiente, sino que es un hilo de ejecución que pertenece a un proceso particular; en este caso, el hilo 300 pertenece al procesador HSML 254. Un único proceso puede tener múltiples hilos. El hilo 300 supervisa el puerto particular para detectar una nueva conexión desde el navegador web 256 o un hilo 400 del procesador HSML 254, como se muestra en la Fig. 5. Una vez que el hilo 300 recibe la nueva conexión desde el navegador web 256 o el hilo 500, el hilo 300 determina si la nueva conexión incluye un flujo de datos. Si la nueva conexión incluye un flujo de datos, el hilo 300 lee la dirección del servidor en el flujo de datos y envía una solicitud al servidor web 102 (mostrado en la Fig. 1). Después de enviar la solicitud, el hilo 300 espera los datos que describen la página web del servidor web. Si el hilo 300 recibe datos desde el servidor web, el hilo 300 procesa los datos extrayendo cualquier gramática HSML y transmitiendo los datos al navegador web 256.

El hilo 300 comienza en la etapa 302, en la que el hilo 300 inicializa un reconocedor (*recognizer*), una variable temporal PREV_VOICE y una variable temporal CUR_VOICE. El hilo 300 utiliza una librería de clases llamada *International Business Machines Java Speech Library* (librería IBMJS), versión 1.0, la cual es una implementación de la norma *Java Speech Application Program Interface*, para interactuar con el motor ASR 258. En una realización alternativa, el hilo 300 puede utilizar una librería de clases diferente a la IBMJS para comunicarse con el motor ASR. Otras librerías de clases que se ajustan a la norma *Java Speech Application Program* también puede sustituir a la librería IBMJS. El hilo 300 inicializa el reconocedor utilizando el procedimiento *createrecognizer* de una clase utilizada para implementar la interfaz *Recognizer* de la librería IBMJS. Después de que se haya inicializado el reconocedor, la variable temporal PREV_VOICE se inicializa a FALSO y la variable temporal CUR_VOICE se inicializa a FALSO. Una vez que se hayan inicializado las variables temporales, el hilo 300 avanza hasta la etapa 304.

En la etapa 304, el hilo 300 espera una nueva conexión desde el navegador web 256 (mostrado en la Fig. 2B) o un hilo 500 (mostrado en la Fig. 5). El hilo 500 se describe en mayor detalle con relación a la FIG. 5. La nueva conexión recibida por el hilo 300 será un flujo de datos. Una vez que el hilo 300 reciba una nueva conexión desde el navegador web 256 o el hilo 500, el hilo 300 avanza hasta la etapa 306.

En la etapa 306, el hilo 300 determina si ha recibido algún dato desde la nueva conexión. Si el hilo 300 ha recibido datos, el hilo 300 avanza hasta la etapa 308. Si el hilo 300 no ha recibido datos, el hilo 300 avanza hasta la etapa 304.

En la etapa 308, el hilo 300 fija una variable temporal DNSTR_COMP a un valor apropiado. La variable temporal DNSTR_COMP debe fijarse igual a un valor que representa el conector (*socket*) asociado al navegador web 256 o al hilo 500 que inició la nueva conexión con el hilo 300. Hacer que el valor del conector esté asociado al iniciador de la nueva conexión con el hilo 300 permite al hilo 300 transmitir datos a y leer datos en el iniciador de la nueva conexión. Una vez que se haya fijado la variable temporal DNSTR_COMP, el hilo 300 avanza hasta la etapa 312.

En la etapa 312, el hilo 300 inhabilita las gramáticas basadas en reglas y las gramáticas TTS actuales del motor ASR 258 (mostrado en la Fig. 2B). Las gramáticas basadas en reglas que están asociadas a cualquier página web y especificadas en el motor ASR 258 se inhabilitan utilizando el procedimiento *deleteRuleGrammar* de la instancia *recognizer* de una clase que implementa la interfaz *Recognizer* de la librería IBMJS. Una vez que se hayan inhabilitado las gramáticas existentes, el hilo 300 avanza hasta la etapa 314.

En la etapa 314, la dirección del servidor se lee en el flujo de datos recibido por el hilo 300 durante la nueva conexión. Después de que la dirección del servidor se lea en el flujo de datos, el hilo 300 formatea y envía una solicitud de conexión al servidor web 102 (mostrado en la Fig. 1) que especifica la dirección del servidor. Una vez que se haya enviado la solicitud al servidor web 102, el hilo 300 avanza hasta la etapa 316.

En la etapa 316, el hilo 300 inicia una ejecución en paralelo de un hilo distinto 400 del procesador HSML 254. El hilo 400 se describe en mayor detalle con referencia a la FIG. 4. Cuando el hilo 300 inicia la ejecución del hilo 400, el hilo 300 comunica al hilo 400 el nombre de servidor leído en el flujo de datos. Una vez que se haya comunicado el nombre del servidor al hilo 400, el hilo 300 avanza hasta la etapa 318.

En la etapa 318, el hilo 300 determina si está abierta la conexión al servidor web 102 (mostrado en la Fig. 1). Con el fin de leer en el servidor web 102, la conexión tiene que estar abierta. El hilo 300 lleva a cabo una lectura en el servidor web 102. Si se devuelve un error, la conexión está cerrada; en caso contrario, la conexión está abierta. Si la conexión al servidor web está abierta, el hilo 300 avanza hasta la etapa 320. Si la conexión al servidor web no está abierta, el hilo 300 avanza hasta la etapa 304.

En la etapa 320, el hilo 300 realiza una lectura en el servidor web 102 (mostrado en la Fig. 1). El hilo 300 recibe los datos que describen la página web solicitada durante la lectura en el servidor web 102. Una vez que el hilo 300 haya llevado a cabo la lectura en el servidor web 102, el hilo 300 avanza hasta la etapa 322.

En la etapa 322, el hilo 300 añade al principio o al final de los datos leídos en el servidor web 102 una cabecera, un

5 delimitador o un pie apropiados y envía la información que incluye una cabecera, un delimitador o un pie apropiados al conector definido por la variable temporal DNSTR_COMP. Las condiciones bajo las cuales el hilo 300 añade al principio o al final una cabecera, un delimitador o un pie, se ilustran gráficamente en una tabla 350 ilustrada en la FIG. 3B. Haciendo referencia a la FIG. 3B, la tabla 350 incluye un grupo de columnas: una columna de página actual 352, una columna de página anterior 354, una columna de inicio de página actual 356, una columna de primera directiva 358, una columna de segunda directiva 360 y una columna de adición al principio/al final 361. La tabla 350 también incluye un grupo de entradas: una primera entrada 362, una segunda entrada 364, una tercera entrada 366, una cuarta entrada 368, una quinta entrada 370, una sexta entrada 372, un séptima entrada 374 y una octava entrada 376.

10 Las entradas en la columna de página actual 352 pueden ser un Sí o un No. Un Sí en la columna de página actual 352 indica que la página actual incluye al menos una gramática HSML. Un No en la columna de página actual 352 indica que la página actual no incluye ninguna gramática HSML.

15 Las entradas en la columna de página anterior 354 pueden ser un Sí o un No. Un Sí en la columna de página anterior 354 indica que la página anterior incluye al menos una gramática HSML. Un No en la columna de página anterior 354 indica que la página anterior no incluye ninguna gramática HSML.

Las entradas en la columna de inicio de página actual 356 pueden ser Voz o Clic de Ratón. El valor Voz en la columna de inicio de página actual 356 indica que la página actual se inició mediante una regla de gramática. El valor Clic de Ratón en la columna de inicio de página actual 356 indica que la página actual se inició mediante un clic de ratón.

20 Las entradas en la columna de primera directiva 358 y en la columna de segunda directiva 360 pueden tener valores iguales a Cabecera, Delimitador, Pie, Cerrar Conector y Nada. Cada valor almacenado en la columna de primera directiva 358 y en la columna de segunda directiva 360 indica una directiva particular que el hilo 300 debe transmitir al componente aguas abajo especificado por la variable temporal DNSTR_COMP. La directiva especificada en la columna de primera directiva 358 debe transmitirse antes de la directiva especificada en la columna de segunda directiva 360. Por ejemplo, la primera entrada 362 indica que si la página actual tiene una gramática, si la página anterior tiene una gramática y si la página actual se inició por voz, entonces solo debe añadirse un delimitador como segunda directiva al final de los datos antes de que se transmitan al componente aguas abajo especificado por la variable temporal DNSTR_COMP.

30 Una cabecera, como se especifica en la columna de primera directiva 358 de la segunda entrada 364, la tercera entrada 366 y la cuarta entrada 368, es una cabecera del protocolo de transferencia de hipertexto (HTTP). La cabecera HTTP debe incluir un delimitador inicial. La cabecera HTTP define el inicio de los datos *x-mixed replace*. A continuación se muestra una cabecera HTTP a modo de ejemplo:

```
HTTP/1.1 200
Content-Type: multipart/x-mixed-replace;boundary=123456789
Cache-Control: no-cache
-123456789
```

40 Un delimitador, como se especifica en la columna de primera directiva 358 de la primera entrada 362, y en la columna de segunda directiva 360 de la segunda entrada 364, la tercera entrada 366 y la cuarta entrada 368, es un delimitador HTTP. El delimitador HTTP define el final de una sección dada de datos *x-mixed replace*. A continuación se muestra un delimitador HTTP a modo de ejemplo:

```
-123456789
```

Un pie, como se especifica en la columna de primera directiva 358 de la quinta entrada 370, es un pie HTTP. El pie HTTP define el final de todos los datos *x-mixed replace*. A continuación se muestra un pie HTTP a modo de ejemplo:

```
-123456789-
```

45 La cabecera HTTP a modo de ejemplo puede definir el comienzo de los datos *x-mixed replace*, el delimitador HTTP a modo de ejemplo puede definir el final de una parte de los datos *x-mixed replace*, y el pie HTTP a modo de ejemplo puede definir el final de todos los datos *x-mixed replace*. La cabecera HTTP a modo de ejemplo, el delimitador HTTP a modo de ejemplo y el pie HTTP a modo de ejemplo están relacionados entre sí mediante el número 123456789, el cual aparece en la segunda línea de la cabecera HTTP a modo de ejemplo, en el delimitador HTTP a modo de ejemplo y en el pie HTTP a modo de ejemplo. El navegador web 256 se muestra en la pantalla 134 (mostrada en la Fig. 1) acoplada al ordenador 120 (mostrado en la Fig. 1) según los datos que describen el sitio web, los cuales aparecen después de la cabecera HTTP a modo de ejemplo. Si el navegador web 256 recibe nuevos datos desde el procesador HSML 254, el navegador web 256 se muestra en la pantalla 134 según los nuevos datos. Si el navegador web 256 recibe un delimitador HTTP, el navegador web 256 “borra” la pantalla 134 acoplada al

- ordenador 120 con la recepción de nuevos datos y vuelve a mostrarse en la pantalla 134 según los datos que siguen al delimitador HTTP. El navegador web también puede cerrar su conector una vez que reciba un pie HTTP. Si el navegador web 256 recibe una indicación que señala que es necesario obtener nueva información, por ejemplo si un usuario hace clic sobre un hipere enlace, el navegador web 256 transmite un URL al procesador HSML 254 (mostrado en la Fig. 2B), y cierra el anterior conector antes de abrir una nueva conexión.
- Un conector cercano, como se especifica en la columna de segunda directiva 360 de la quinta entrada 370, cierra el conector para el navegador web. El procesador HSML 254 no puede establecer contacto con el navegador web 256 una vez que se haya cerrado el conector. El conector se cierra utilizando el procedimiento *close* de la instancia *socket* de la clase *Socket* de *JAVA 2 Platform, Standard Edition, v1.2.2*.
- Las entradas de la columna de adición al principio/al final 361 puede tener valores iguales a "antes" y "después". Cada valor almacenado en la columna de adición al principio/al final 361 indica si la primera directiva asociada debe añadirse al principio o al final de los datos que están enviándose al componente aguas abajo especificado por la variable temporal *DNSTR_COMP*. Si la entrada de la columna de adición al principio/al final 361 es "después", la primera directiva debe añadirse al final de los datos recibidos desde el servidor web 102 (mostrado en la Fig. 1); sin embargo, si la entrada de la columna de adición al principio/al final 361 es "antes", la primera directiva debe añadirse al principio de los datos recibidos desde el servidor web 102. La segunda directiva debe añadirse siempre al final de los datos recibidos desde el servidor web 102 (mostrado en la Fig. 1).
- El hilo 300 determina si los datos recibidos por el hilo 300 durante la nueva conexión incluyen alguna gramática HSML. El hilo 300 lee datos del flujo de datos, sin eliminar ningún dato del flujo de datos, buscando una gramática HSML. Después de que el hilo 300 haya revisado los datos recibidos desde el servidor web 102, el hilo 300 añade las directivas especificadas en la tabla 350 al principio o al final de los datos recibidos desde el servidor web 102 dependiendo de si la página anterior contenía una gramática HSML, de si la página actual contenía una gramática HSML y de cómo se inició la página actual. Una vez que se hayan añadido las directivas al principio o al final, el hilo 300 avanza hasta la etapa 324.
- En la etapa 324, el hilo 300 determina si la variable temporal *CUR_VOICE* es igual a *VERDADERO*. Si la variable temporal *CUR_VOICE* es igual a *VERDADERO*, el hilo 300 avanza hasta la etapa 328. Si la variable temporal *CUR_VOICE* no es igual *VERDADERO*, el hilo 300 avanza hasta la etapa 326.
- En la etapa 326, el hilo 300 fija la variable temporal *PREV_VOICE* igual *FALSO*. Una vez que la variable temporal *PREV_VOICE* se haya fijado a *FALSO*, el hilo 300 avanza hasta la etapa 330.
- En la etapa 328, el hilo 300 fija la variable temporal *PREV_VOICE* igual a *VERDADERO*. Una vez que la variable temporal *PREV_VOICE* se haya fijado a *VERDADERO*, el hilo 300 avanza hasta la etapa 330.
- En la etapa 330, el hilo 300 fija la variable temporal *CUR_VOICE* igual *FALSO*. Una vez que la variable temporal *CUR_VOICE* se haya fijado igual a *FALSO*, el hilo 300 avanza hasta la etapa 332.
- En la etapa 332, el hilo 300 determina si el flujo de datos, recibido por el hilo 300 durante la nueva conexión, incluye alguna gramática HSML. El hilo 300 lee datos del flujo de datos, sin eliminar ningún dato del flujo de datos, buscando una gramática HSML. Si el hilo 300 encuentra el comienzo de alguna gramática HSML, hay gramáticas HSML en el flujo de datos y el hilo 300 avanza hasta la etapa 334. La etapa 334 se muestra en mayor detalle en la FIG. 7. Una vez que haya finalizado la etapa 334, el hilo 300 avanza hasta la etapa 304. Si el hilo 300 no encuentra el comienzo de ninguna gramática HSML, no hay gramáticas HSML en el flujo de datos y el hilo 300 avanza hasta la etapa 304.
- La FIG. 4 ilustra el hilo 400, el cual inicia una ejecución en paralelo con el hilo 300 en la etapa 316. El hilo 400 recibe datos desde el navegador web 256 (mostrado en la Fig. 2B). El hilo 400 comienza en la etapa 401 enviando una solicitud a la dirección de servidor incluyendo el URL y algunos parámetros. Una vez que se haya enviado la solicitud al servidor, el hilo 400 avanza hasta la etapa 402.
- En la etapa 402, el hilo 400 lleva a cabo una lectura en el navegador web 256. Leyendo en el navegador web, el hilo 400 puede determinar si el navegador web 256 especifica algún argumento adicional. Una vez que el hilo 400 haya realizado una lectura en el navegador web 256, el hilo 400 avanza hasta la etapa 404.
- En la etapa 404, el hilo 400 determina si el hilo 400 ha recibido una interrupción desde el hilo 500, el cual se explica en mayor detalle con relación a la Fig. 5. Si ha recibido una interrupción, el hilo 400 avanza hasta la etapa 406. Si no ha recibido ninguna interrupción, el hilo 400 avanza hasta la etapa 410.
- En la etapa 406, el hilo 400 cierra su conexión con el servidor web 102. Una vez que el hilo 400 haya cerrado su conexión con el servidor web 102, el hilo 400 avanza hasta la etapa 408.
- En la etapa 408, el hilo 400 no establece más comunicaciones con el navegador web 256 (mostrado en la Fig. 2B). El hilo 400 no cierra la conexión con el navegador web 256. La conexión con el navegador web 256 permanece abierta y espera otro contacto desde el procesador HSML 254 (mostrado en la Fig. 2B). Una vez que el hilo 400

haya dejado de comunicarse con el navegador web 256, el hilo 400 finaliza.

En la etapa 410, el hilo 400 determina si se han recibido datos adicionales desde el navegador web 256. Si se han leído datos desde el navegador web 256, el hilo 400 avanza hasta la etapa 412. Si no se han recibido datos desde el navegador web 256, el hilo 400 avanza hasta la etapa 402.

5 En la etapa 412, el hilo 400 transmite al servidor web 102 (mostrado en la Fig. 1) los argumentos adicionales recibidos desde el navegador web 256. Los argumentos adicionales deben enviarse al servidor web 102 como argumentos asociados con el nombre de servidor que se envió al servidor web mediante el hilo 300. Después de que el hilo 400 transmita al servidor web 102 los argumentos recibidos desde el navegador web 256, el hilo 400 avanza hasta la etapa 402.

10 La FIG. 7 ilustra en mayor detalle el procedimiento 334 para crear, almacenar y enviar gramáticas basadas en reglas y gramáticas TTS. El procedimiento 334 junto con sus subprocedimientos analizan sintácticamente el lenguaje de marcado de una página web extrayendo las gramáticas HSML que definen las gramáticas basadas en reglas y las gramáticas TTS para esa página web, que almacenan las gramáticas basadas en reglas y las gramáticas TTS y que transmiten las gramáticas basadas en reglas al motor ASR 258. El propio procedimiento 334 analiza sintácticamente el lenguaje de marcado de la página web buscando un conjunto de bloques (*blockset*) de HSML. El conjunto de bloques de HSML es la sentencia de más alto nivel de cualquier gramática HSML.

El procedimiento 334 comienza en la etapa 702 inicializando una variable temporal SPEECH, una variable temporal HREF y una variable temporal ASRCMDS al valor nulo (*null*). Una vez que las variables temporales SPEECH, HREF y ASRCMDS se hayan inicializado, el procedimiento 334 avanza hasta la etapa 704.

20 En la etapa 704, el procedimiento 334 comienza leyendo el lenguaje de marcado incluido en los datos que describen la página web. Si el procedimiento 334 lee la cadena de caracteres "<hsm:blockset>", el procedimiento 334 avanza hasta la etapa 706. La etapa 706 se muestra en mayor detalle en la FIG. 8. Una vez que haya finalizado la etapa 706, el procedimiento 334 avanza hasta la etapa 704. Si el procedimiento 334 no lee la cadena de caracteres "<hsm:blockset>", el procedimiento avanza hasta la etapa 708.

25 En la etapa 708, el procedimiento 334 compara lo que el procedimiento 334 ha leído con EOF o final del archivo. EOF indica cuándo se ha llegado al final de un archivo. Si el procedimiento 334 ha leído el carácter EOF, el procedimiento 334 avanza hasta la etapa 710. Si el procedimiento 334 no ha leído el carácter EOF, el procedimiento 334 avanza hasta la etapa 704.

30 En la etapa 710, el procedimiento 334 transmite las gramáticas basadas en reglas al motor ASR 258 (mostrado en la Fig. 2B). El proceso 700 utiliza una librería de clase llamada *International Business Machines Java Speech Library* (librería IBMJS), versión 1.0, para interactuar con el motor ASR 258. Después, el procedimiento 334 inicializa un lector, que es una superclase JAVA, como un lector de cadenas de caracteres (*stringreader*). El lector de cadenas de caracteres es una clase JAVA y un miembro de la superclase JAVA *reader*. El lector (*reader*) está dotado de las gramáticas basadas en reglas que se suministrarán al motor ASR 258. Las gramáticas basadas en reglas se almacenan en la variable temporal ASRCMDS. Una vez que se hayan almacenado las gramáticas basadas en reglas, el procedimiento 334 inicializa una instancia *rgGram* de la clase *Rule Grammar*, la cual es una clase IBMJS. La instancia *rgGram* se inicializa con el valor de salida del procedimiento *loadJSGF*, el cual es un procedimiento de la instancia *recognizer* de una clase que implementa la interfaz *Recognizer* de la librería IBMJS. La variable temporal ASRCMDS se proporciona como un argumento al procedimiento *loadJSGF*. El procedimiento *loadJSGF* transmite las gramáticas basadas en reglas al motor ASR 258. Una vez que las gramáticas basadas en reglas se hayan enviado al motor ASR 258, el procedimiento 334 avanza hasta la etapa 712.

35 En la etapa 712, el procedimiento 334 especifica un escuchador de resultados (*result listener*), el cual recibe la salida del motor ASR 258. El escuchador de resultados se añade utilizando el procedimiento *addResultListener*, el cual es un procedimiento de la instancia *rgGram* de una clase que implementa la interfaz *Rule Grammar* de la librería IBMJS. Una vez que se haya inicializado el escuchador de resultados, el procedimiento 334 utiliza el procedimiento *commitChanges* de la clase que implementa la interfaz *Recognizer* de la librería IBMJS para confirmar los cambios especificados al motor ASR 258, y finaliza.

40 La FIG. 8 ilustra el procedimiento 706 para analizar sintácticamente etiquetas conjuntos de bloques de gramáticas HSML en mayor detalle. El procedimiento 706 comienza en la etapa 802 leyendo el lenguaje de marcado incluido en los datos que describen la página web. Si el procedimiento 706 lee la cadena "<hsm:JSML>", el procedimiento 706 avanza hasta la etapa 804. La etapa 804 se muestra en mayor detalle en la FIG. 10. Una vez que haya finalizado la etapa 804, el procedimiento 706 avanza hasta la etapa 802. Si el procedimiento 706 no lee la cadena de caracteres "<hsm:JSML>", el procedimiento 706 avanza hasta la etapa 806.

45 En la etapa 806, el procedimiento 706 compara lo que ha leído el procedimiento 706 con la cadena de caracteres "<hsm:JSGFRule>". Si el procedimiento 706 ha leído la cadena de caracteres "<hsm:JSGFRule>", el procedimiento 706 avanza hasta la etapa 808. La etapa 808 se muestra en mayor detalle en la FIG. 11. Una vez que haya finalizado la etapa 808, el procedimiento 706 avanza hasta la etapa 802. Si el procedimiento 706 no ha leído la

cadena de caracteres "<hsml:JSGFRule>", el procedimiento 706 avanza hasta la etapa 810.

5 En la etapa 810, el procedimiento 706 compara lo que ha leído el procedimiento 706 con la cadena de caracteres "<hsml:block>". Si el procedimiento 706 ha leído la cadena de caracteres "<hsml:block>", el procedimiento 706 avanza hasta la etapa 812. La etapa 812 se muestra en mayor detalle en la FIG. 9. Una vez que haya finalizado la etapa 812, el procedimiento 706 avanza hasta la etapa 802. Si el procedimiento 706 no ha leído la cadena de caracteres "<hsml:block>", el procedimiento 706 avanza hasta la etapa 814.

10 En la etapa 814, el procedimiento 706 compara lo que ha leído el procedimiento 706 con la cadena de caracteres "</hsml:blockset>". Si el procedimiento 706 ha leído la cadena de caracteres "</hsml:blockset>", el procedimiento 706 termina y el procedimiento 706 finaliza su ejecución. Si el procedimiento 706 no ha leído la cadena de caracteres "</hsml:blockset>", el procedimiento 706 avanza hasta la etapa 816.

En la etapa 816, el procedimiento 706 notifica un error del lenguaje de marcado de los datos que describen la página web. Si no aparece en este punto la cadena de caracteres "</hsml:blockset>", debe notificarse un error. Una vez que se haya notificado el error, el procedimiento 706 finaliza y, a su vez, el procedimiento 334 finaliza.

15 La FIG. 9 ilustra en mayor detalle el procedimiento 812 para analizar sintácticamente bloques de gramáticas HSML. El procedimiento 812 comienza en la etapa 901, en la que el procedimiento 812 inicializa la variable temporal VOICE_GRAM a FALSO. Una vez que se haya inicializado la variable temporal VOICE_GRAM, el procedimiento 812 avanza hasta la etapa 902.

20 En la etapa 902, el procedimiento 812 lee el lenguaje de marcado incluido en los datos que describen la página web. Si el procedimiento 812 lee la cadena de caracteres "<hsml:JSML>", el procedimiento 812 avanza hasta la etapa 804. La etapa 804 se muestra en mayor detalle en la FIG. 10. Una vez que haya finalizado la etapa 804, el procedimiento 812 avanza hasta la etapa 902. Si el procedimiento 812 no lee la cadena de caracteres "<hsml:JSML>", el procedimiento 812 avanza hasta la etapa 904.

25 En la etapa 904, el procedimiento 812 sigue leyendo los datos de marcado incluidos con los datos. Si el procedimiento 812 ha leído una cadena de caracteres que coincide con la cadena de caracteres "<hsml:JSGFRule>", el procedimiento 812 avanza hasta la etapa 905. Si el procedimiento 812 no ha leído una cadena de caracteres que coincida con la cadena de caracteres "<hsml:JSGFRule>", el procedimiento 812 avanza hasta la etapa 906.

30 En la etapa 905, el procedimiento 812 fija la variable temporal VOICE_GRAM igual a VERDADERO para indicar que hay una gramática basada en reglas en el bloque HSML. Una vez que se haya fijado la variable temporal VOICE_GRAM, el procedimiento 812 avanza hasta la etapa 808. La etapa 808 se muestra en mayor detalle en la FIG. 10. Una vez que haya finalizado la etapa 808, el procedimiento 812 avanza hasta la etapa 902.

35 En la etapa 906, el procedimiento 812 compara lo que ha leído el procedimiento 812 con la cadena de caracteres "</hsml:block>". Si el procedimiento 812 ha leído una cadena de caracteres que coincide con la cadena de caracteres "</hsml:block>", el procedimiento 812 avanza hasta la etapa 910. Si el procedimiento 812 no ha leído una cadena de caracteres que coincida con la cadena de caracteres "</hsml:block>", el procedimiento 812 avanza hasta la etapa 908.

40 En la etapa 908, el procedimiento 812 notifica un error del lenguaje de marcado de los datos que describen la página web. Si no aparece en este punto la cadena de caracteres "</hsml:block>", debe notificarse un error de lenguaje de marcado. Una vez que se haya notificado el error, el procedimiento 812 finaliza y, a su vez, los procesos 334 y 706 finalizan.

En la etapa 910, el procedimiento 812 determina si la variable temporal VOICE_GRAM es igual a VERDADERO. Si la variable temporal VOICE_GRAM es igual a FALSO, el procedimiento 812 avanza hasta la etapa 912. Si la variable temporal VOICE_GRAM no es igual a FALSO, el procedimiento 812 finaliza.

45 En la etapa 912, el procedimiento 812 transmite al motor ASR 258 reglas de formato de gramática de voz de Java (reglas JSGF) correspondientes a la cadena de caracteres almacenada en la variable temporal SPEECH. La cadena de caracteres almacenada en la variable temporal SPEECH se proporciona al procedimiento *speak* de la instancia *synthesizer* (sintetizador) de una clase que implementa la interfaz *Synthesizer* de la librería IBMJS. Una vez que la regla JSGF se haya transmitido al motor ASR 258 (mostrado en la Fig. 2B), el procedimiento 812 finaliza.

50 La FIG. 10 ilustra en mayor detalle el procedimiento 804 para analizar sintácticamente gramáticas TTS de HSML. El procedimiento 804 comienza en la etapa 1002 leyendo el lenguaje de marcado incluido en los datos que describen la página web. Si el procedimiento 804 lee la cadena de caracteres "<!CDATA[", el procedimiento 804 avanza hasta la etapa 1006. Si el procedimiento 804 no lee la cadena de caracteres "<!CDATA[", el procedimiento 804 avanza hasta la etapa 1004.

55 En la etapa 1004, el procedimiento 804 notifica un error del lenguaje de marcado de los datos que describen la página web. Si la cadena de caracteres "<hsml:JSML>" no va seguida de la cadena de caracteres "<!CDATA[", debe

notificarse un error. Una vez que se haya notificado el error, el procedimiento 804 finaliza y, a su vez, los procesos 334 y 706 finalizan, y, si está ejecutándose, el procedimiento 812 finaliza.

5 En la etapa 1006, el procedimiento 804 captura el texto del lenguaje de marcado incluido en los datos que describen la página web hasta que el procedimiento 804 lee un "]". El lenguaje de marcado capturado se almacena en la variable temporal SPEECH como una cadena de caracteres. Una vez que el lenguaje de marcado capturado se haya almacenado en la variable temporal SPEECH, el procedimiento 804 avanza hasta la etapa 1008.

10 En la etapa 1008, el procedimiento 804 sigue leyendo el lenguaje de marcado incluido con los datos que describen el sitio web. Si el procedimiento 804 lee una cadena de caracteres que coincide con la cadena de caracteres "]></hsml:JSML>", el procedimiento 804 finaliza y el procedimiento 804 termina su ejecución. Si el procedimiento 804 no ha leído una cadena de caracteres que coincida con la cadena de caracteres "]></hsml:JSML>", el procedimiento 804 avanza hasta la etapa 1010.

15 En la etapa 1010, el procedimiento 804 notifica un error del lenguaje de marcado de los datos que describen la página web. Si la cadena de caracteres "]" no va seguida de la cadena de caracteres "]></hsml:JSML>", debe notificarse un error. Una vez que se haya notificado el error, el procedimiento 804 finaliza y, a su vez, los procesos 334 y 706 finalizan, y, si está ejecutándose, el procedimiento 812 finaliza.

20 La FIG. 11 ilustra en mayor detalle el procedimiento 808 para analizar sintácticamente gramáticas basadas en reglas HSML. El procedimiento 808 comienza en la etapa 1102 leyendo el lenguaje de marcado incluido en los datos que describen la página web. Si el procedimiento 808 lee la cadena de caracteres "href=", el procedimiento 808 avanza hasta la etapa 1104. Si el procedimiento 808 no lee la cadena de caracteres "href=", el procedimiento 808 avanza hasta la etapa 1106.

25 En la etapa 1104, el procedimiento 808 captura el texto del lenguaje de marcado incluido en los datos que describen la página web hasta que el procedimiento 808 lee una cadena de caracteres completa delimitada por dobles comillas; por ejemplo, en la primera gramática basada en reglas HSML a modo de ejemplo, "www.columbia.edu". El lenguaje de marcado capturado incluido en los datos que describen la página web se almacena en la variable temporal HREF como una cadena de caracteres. Una vez que el lenguaje de marcado capturado se haya almacenado en la variable temporal HREF, el procedimiento 808 avanza hasta la etapa 1106.

30 En la etapa 1106, el procedimiento 808 sigue leyendo el lenguaje de marcado incluido con los datos que describen el sitio web. Si el procedimiento 808 ha leído una cadena de caracteres que coincide con la cadena de caracteres "tag=", el procedimiento 808 avanza hasta la etapa 1110. Si el procedimiento 808 no ha leído una cadena de caracteres que coincida con la cadena de caracteres "tag=", el procedimiento 808 avanza hasta la etapa 1108.

En la etapa 1108, el procedimiento 808 notifica un error del lenguaje de marcado de los datos que describen la página web. Si la cadena de caracteres "tag=" no está presente en este punto en el lenguaje de marcado, debe notificarse un error. Una vez que se haya notificado el error, el procedimiento 808 finaliza y, a su vez, los procesos 334 y 706 finalizan, y, si está ejecutándose, el procedimiento 812 finaliza.

35 En la etapa 1110, el procedimiento 808 captura el texto del lenguaje de marcado incluido en los datos que describen la página web hasta que el procedimiento 808 lee una cadena de caracteres completa delimitada por dobles comillas; por ejemplo, en la primera gramática basada en reglas HSML a modo de ejemplo, "reload". El lenguaje de marcado capturado incluido en los datos que describen la página web se almacena en la variable temporal TAG como una cadena de caracteres. Una vez que el lenguaje de marcado capturado se haya almacenado en la variable temporal TAG, el procedimiento 808 crea un nuevo registro en una base de datos ubicada en la unidad de almacenamiento de datos 126 (mostrada en la Fig. 1). El nuevo registro incluye un primer campo, un segundo campo y un tercer campo. El primer campo se rellena con la cadena de caracteres almacenada en la variable temporal TAG, el segundo campo se rellena con la cadena de caracteres "HREF", y el tercer campo se rellena con la cadena de caracteres almacenada en la variable temporal HREF. Después de haberse relleno el primer registro de base de datos, el procedimiento 808 avanza hasta la etapa 1112.

En una realización alternativa, la cadena de caracteres "tag=" y una etiqueta asociada pueden eliminarse de la regla HSML, pero debe haber al menos una cadena de caracteres "tag=" y una etiqueta asociada por gramática.

50 En la etapa 1112, el procedimiento 808 sigue leyendo el lenguaje de marcado incluido con los datos que describen el sitio web. Si el procedimiento 808 ha leído una cadena de caracteres que coincide con la cadena de caracteres "><!CDATA[", el procedimiento 808 avanza hasta la etapa 1116. Si el procedimiento 808 no ha leído una cadena de caracteres que coincida con la cadena de caracteres "><!CDATA[", el procedimiento 808 avanza hasta la etapa 1114.

55 En la etapa 1114, el procedimiento 808 notifica un error del lenguaje de marcado de los datos que describen la página web. Si la cadena de caracteres "><!CDATA[" no está presente en este punto en el lenguaje de marcado, debe notificarse un error. Una vez que se haya notificado el error, el procedimiento 808 finaliza y, a su vez, los procesos 334 y 706 finalizan, y, si está ejecutándose, el procedimiento 812 finaliza.

En la etapa 1116, el procedimiento 808 captura el texto del lenguaje de marcado incluido en los datos que describen la página web hasta que el procedimiento 808 lee "]". El lenguaje de marcado capturado incluido en los datos que describen la página web se almacena en la variable temporal ASRCMDS como texto, añadiéndose un carácter de retorno de carro al final del texto capturado. Una vez que el texto se haya almacenado en la variable temporal ASRCMDS, el procedimiento 808 avanza hasta la etapa 1118.

En la etapa 1118, el procedimiento 808 sigue leyendo el lenguaje de marcado incluido con los datos que describen el sitio web. Si el procedimiento 808 ha leído una cadena de caracteres que coincide con la cadena de caracteres "]]></hsml:JSGFRule>", el procedimiento 808 avanza hasta la etapa 1122. Si el procedimiento 808 no ha leído una cadena de caracteres que coincida con la cadena de caracteres "]]></hsml:JSGFRule>", el procedimiento 808 avanza hasta la etapa 1120.

En la etapa 1120, el procedimiento 808 notifica un error del lenguaje de marcado de los datos que describen la página web. Si la cadena de caracteres "]]></hsml:JSGFRule" no está presente en este punto en el lenguaje de marcado, debe notificarse un error. Una vez que se haya notificado el error, el procedimiento 808 finaliza y, a su vez, los procesos 334 y 706 finalizan, y, si está ejecutándose, el procedimiento 812 finaliza.

En la etapa 1122, el procedimiento 808 compara la variable temporal SPEECH con NULL. Si la variable temporal SPEECH es igual a NULL, el procedimiento 808 avanza hasta la etapa 1126. Si la variable temporal SPEECH no es igual a NULL, el procedimiento 808 avanza hasta la etapa 1124.

En la etapa 1124, el procedimiento 808 crea otro nuevo registro en la base de datos ubicada en la unidad de almacenamiento de datos 126 (mostrada en la Fig. 1). Este nuevo registro incluye un primer campo, un segundo campo y un tercer campo. El primer campo se rellena con la cadena de caracteres almacenada en la variable temporal TAG, el segundo campo se rellena con la cadena de caracteres "SPEECH", y el tercer campo se rellena con la cadena de caracteres almacenada en la variable temporal SPEECH. Después de haberse rellenado este nuevo registro, el procedimiento 808 avanza hasta la etapa 1126.

En la etapa 1126, el procedimiento 808 vuelve a fijar el valor de algunas de las variables temporales. Las variables temporales SPEECH, TAG y HREF se fijan a NULL. Una vez que estas variables temporales se hayan fijado a NULL, el procedimiento 808 finaliza y termina su ejecución.

La FIG. 5 ilustra el hilo 500, el cual recibe series de etiquetas desde el motor ASR 258 (mostrado en la Fig. 2B). Más en particular, el hilo 500 recibe series de etiquetas desde el motor ASR 258, genera URL completos basándose en las series de etiquetas recibidas desde el motor ASR 258 y transmite los URL completos al servidor web. Cada etiqueta de una serie corresponde a una regla de gramática que se ha invocado en el motor ASR 258.

El hilo 500 comienza en la etapa 502 determinando si hay algún comando de sistema en la serie de etiquetas recibidas desde el motor ASR 258. Si se ha recibido algún comando de sistema, el hilo 500 avanza hasta la etapa 504. Si no se ha recibido ningún comando de sistema, el hilo 500 avanza hasta la etapa 506.

En la etapa 504, el hilo 500 ejecuta cualquier comando de sistema recibido por el hilo 500. Los comandos de sistema son comandos de nivel de sistema invocados por las gramáticas basadas en reglas almacenadas en el motor ASR 258, por ejemplo, *exit* (salir). Las gramáticas basadas en reglas, asociadas a los comandos de sistema, siempre están activas en el motor ASR 258 independientemente de la página web que esté activa actualmente. En el caso del comando de sistema *exit*, el motor ASR 258 tratará de hacer corresponder alguna locución con la cadena de caracteres "exit", independientemente de la página web que esté visualizándose actualmente. Si se invoca la gramática basada en reglas asociada a la cadena de caracteres "exit", el motor ASR 258 emite una serie de etiquetas, una de las cuales está asociada con el comando de sistema *exit*, y el hilo 500 ejecuta los comandos de sistema. Una vez que se hayan ejecutado los comandos de sistema, el hilo 500 avanza hasta la etapa 502.

En la etapa 506, el hilo 500 busca en la base de datos ubicada en el dispositivo de almacenamiento de datos 126 (mostrado en la Fig. 1) registros de base de datos con una primera etiqueta de la lista de etiquetas almacenadas en el primer campo de los registros de base de datos. Si la base de datos no contiene ningún registro en el que el primer campo del registro de base de datos coincida con la primera etiqueta de la serie de etiquetas, o más de dos registros en los que el primer campo de los registros de base de datos coincide con la primera etiqueta de la serie de etiquetas, la base de datos contiene un número inapropiado de registros de base de datos, y el hilo 500 avanza hasta la etapa 508. Si la base de datos contiene uno o dos registros en los que el primer campo del registro de base de datos coincide con la primera etiqueta de la serie de etiquetas, la base de datos contiene un número apropiado de registros de base de datos y el hilo 500 avanza hasta la etapa 510.

En la etapa 508, el hilo 500 notifica un error del lenguaje de marcado de los datos que describen la página web. Si la base de datos no contiene ningún registro correspondiente a la primera etiqueta de la serie de etiquetas o la base de datos contiene más de dos registros correspondientes a la primera etiqueta de la serie de etiquetas, debe notificarse un error. Una vez que se haya notificado el error, el hilo 500 finaliza y, a su vez, el hilo 300 finaliza.

En la etapa 510, el hilo 500 busca en la base de datos ubicada en el dispositivo de almacenamiento de datos 126 (mostrado en la Fig. 1) un registro de base de datos con una cadena de caracteres que coincida con la primera

etiqueta de la serie de etiquetas almacenadas en el primer campo del registro de base de datos y con la cadena de caracteres "SPEECH" almacenada en el segundo campo del registro de base de datos. Si el hilo 500 encuentra tal registro de base de datos, el hilo 500 avanza hasta la etapa 512. Si el hilo 500 no encuentra tal registro de base de datos, el hilo 500 avanza hasta la etapa 514.

5 En la etapa 512, el hilo 500 transmite al motor ASR 258 (mostrado en la Fig. 2B) una regla JSGF correspondiente a la cadena de caracteres almacenada en el tercer campo del registro de base de datos hallado durante la etapa 510. La cadena de caracteres almacenada en el tercer campo del registro de base de datos hallado durante la etapa 510 se proporciona al procedimiento *speak* de la instancia *synthesizer* de una clase que implementa la interfaz *Synthesizer* de la librería IBMJS. Una vez que la regla JSGF se haya transmitido al motor ASR 258, el hilo 500
10 avanza hasta la etapa 514.

En la etapa 514, el hilo 500 determina si la variable temporal PREV_VOICE es igual a VERDADERO. Si la variable temporal PREV_VOICE es igual a VERDADERO, el hilo 500 avanza hasta la etapa 518. Si la variable temporal PREV_VOICE no es igual a VERDADERO, el hilo 500 avanza hasta la etapa 516.

15 En la etapa 516, el hilo 500 asigna el valor de una variable temporal DNSTR_COMP_2 igual al valor de la variable temporal DNSTR_COMP. La variable temporal DNSTR_COMP_2 representa ahora el conector del componente aguas abajo, el cual se utilizará por el hilo 600, descrito en mayor detalle con relación a la FIG. 6, para transmitir datos al navegador web. Una vez que el valor de la variable temporal DNSTR_COMP_2 se haya actualizado, el hilo 500 avanza hasta la etapa 518.

20 En la etapa 518, el hilo 500 fija la variable temporal CUR_VOICE a VERDADERO. Una vez que la variable temporal CUR_VOICE se haya fijado a VERDADERO, el hilo 500 avanza hasta la etapa 520.

En la etapa 520, el hilo 500 determina si el hilo 400 está ejecutándose actualmente. Si el hilo 400 está ejecutándose actualmente, el hilo 500 avanza hasta la etapa 522. Si el hilo 400 no está ejecutándose actualmente, el hilo 500 avanza hasta la etapa 524.

25 En la etapa 522, el hilo 500 transmite una interrupción al hilo 400. La interrupción hará que el hilo 400 finalice sin cerrar su conexión con el navegador web 256. Una vez que se haya enviado la interrupción al hilo 400, el hilo 500 avanza hasta la etapa 524.

30 En la etapa 524, el hilo 500 busca en la base de datos ubicada en el dispositivo de almacenamiento de datos 126 (mostrado en la Fig. 1) un registro de base de datos con la primera etiqueta de la serie de etiquetas almacenadas en el primer campo del registro de base de datos y la cadena "HREF" almacenada en el segundo campo del registro de base de datos. Si el hilo 500 encuentra tal registro de base de datos, el hilo 500 almacena la cadena de caracteres almacenada en el tercer campo del registro de base de datos en una variable temporal y añade cualquier etiqueta recibida desde el motor ASR 258 al final de la cadena de caracteres almacenada en la variable temporal. Una vez que se hayan añadido las etiquetas, el hilo 500 transmite la cadena de caracteres almacenada en la variable temporal al hilo 300. Después de que la cadena de caracteres se haya enviado al hilo 300, el hilo 500 avanza hasta
35 al etapa 526.

En la etapa 526, el hilo 500 hace que el hilo 600 inicie su ejecución. Una vez que el hilo 600 inicia su ejecución, el hilo 500 finaliza.

40 La FIG. 6 ilustra un hilo 600 que recibe y transmite datos enviados al navegador web 256 (mostrado en la Fig. 2B) asociados con páginas web invocadas por el motor ASR 258. El hilo 600 comienza en la etapa 602 determinando si el hilo 600 ha recibido un flujo de datos. Si se ha recibido un flujo de datos, el hilo 600 avanza hasta la etapa 604. Si no se ha recibido ningún flujo de datos, el hilo 600 avanza hasta la etapa 608.

En la etapa 604, el hilo 600 debe transmitir los datos contenidos en el flujo de datos recibido desde el hilo 300 al conector especificado en la variable temporal DNSTR_COMP_2. Una vez que los datos recibidos por el hilo 600 se hayan transmitido al navegador web 256, el hilo 600 avanza hasta la etapa 606.

45 En la etapa 606, el hilo 600 determina si el conector de navegador especificado por la variable temporal DNSTR_COMP_2 está cerrado. Si el hilo 600 recibe un error cuando intenta transmitir datos al valor de la variable de conector de navegador en la etapa 604, el conector de navegador está cerrado y el hilo 600 debe finalizar. Si el hilo 600 no recibe ningún error, el conector de navegador está abierto y el hilo 600 avanza hasta la etapa 602.

50 En la etapa 608, el hilo 600 no establece más comunicaciones con el navegador web 256. El hilo 600 no cierra la conexión el navegador web 256. La conexión con el navegador web 256 permanece abierta y espera contactos de otro proceso. Una vez que el hilo 600 haya dejado de comunicarse con el navegador web 256, el hilo 600 finaliza.

55 En una realización alternativa, el procesador HSML 254 (mostrado en la Fig. 2B) puede recibir datos que describen una páginas web con múltiples cuadros (*frames*) incluidos en esa página web. Cada cuadro puede incluir un conjunto diferente de gramáticas basadas en reglas HSML y de gramáticas TTS de HSML. El procesador HSML 254 puede configurarse para supervisar múltiples conjuntos de gramáticas basadas en reglas HSML y gramáticas TTS de HSML.

REIVINDICACIONES

1.- Un procedimiento para proporcionar páginas web visuales con una interfaz de audio, que comprende:

recibir, en un navegador (256), primeros datos de página web transmitidos desde un servidor (102), incluyendo los primeros datos de página web:

5 (a) primeros componentes de marcado que describen y pueden convertirse en una primera página web visual que incluye enlaces representados visualmente que pueden seleccionarse cada uno por un usuario para navegar a otra página web respectiva; y

10 (b) segundos componentes de marcado diferentes de los primeros componentes de marcado y que describen y pueden convertirse en una primera gramática basada en reglas generada independientemente del contenido de los primeros componentes de marcado, donde:

la primera gramática basada en reglas define para cada uno de uno o más de los enlaces una regla respectiva que incluye una parte de comando y una parte de frase;

la parte de comando define un comando que puede ejecutarse para navegar hasta otra de las páginas web respectivas correspondientes a los enlaces; y

15 la parte de frase identifica una pluralidad de frases alternativas que (i) no se muestran en la página web visual, (ii) no están incluidas en los enlaces y (iii) cada una, cuando se pronuncia por un usuario, invoca por separado la ejecución del comando respectivo de la regla a la que pertenece la parte de frase;

convertir los primeros componentes de marcado en la primera página web visual;

20 recibir una entrada de voz que incluye una frase pronunciada;

hacer corresponder la frase pronunciada con una de las frases alternativas identificadas por una de las reglas definidas por los segundos componentes de marcado; y

25 como respuesta a la correspondencia, navegar desde la primera página web visual hasta la página web respectiva especificada por la parte de comando de la regla que identifica la frase con la que se hizo corresponder la frase pronunciada.

2.- El procedimiento según la reivindicación 1, en el que cada regla de la primera gramática basada en reglas incluye además una primera parte de etiqueta que identifica de manera unívoca la regla respectiva de la primera gramática basada en reglas.

3.- El procedimiento según la reivindicación 1, que comprende además:

30 abrir una conexión al navegador (256) desde el servidor (102); y

enviar la primera página web visual al navegador (256) a través de la conexión.

4.- El procedimiento según la reivindicación 3, en el que la conexión es una conexión persistente.

5.- El procedimiento según la reivindicación 4, en el que la navegación desde la primera página web visual hasta la página web respectiva incluye:

35 recibir segundos datos de página web desde el servidor (102), especificando los segundos datos de página web una segunda página web visual y una segunda gramática basada en reglas asociada; y

enviar la segunda página web visual al navegador a través de la conexión.

6.- El procedimiento según la reivindicación 1, en el que los primeros componentes de marcado están en forma de lenguaje de marcado de hipertexto.

40 7.- El procedimiento según la reivindicación 1, en el que la primera gramática basada en reglas incluye una primera parte de prioridad de visualización.

8.- El procedimiento según la reivindicación 1, que comprende además:

proporcionar un nivel de audio de salida.

9.- El procedimiento según la reivindicación 1, que comprende además:

45 proporcionar como salida de manera audible contenido relacionado con la primera página web visual.

- 10.- El procedimiento según la reivindicación 9, en el que la salida audible se produce como respuesta a la recepción de la entrada de voz.
- 11.- El procedimiento según la reivindicación 1, en el que la conversión se produce en un navegador (256).
- 12.- El procedimiento según la reivindicación 1, en el que:
- 5 los segundos componentes de marcado se proporcionan en un lenguaje de marcado de voz;
comprendiendo además el procedimiento:
- abrir una conexión entre el navegador (256) y un módulo que procesa la entrada de voz; y
 - enviar los primeros componentes de marcado al navegador (256) a través de la conexión; y
- 10 la primera página web visual y la página web respectiva hacia la que se navega desde la primera página web visual se muestran en el navegador (256).
- 13.- El procedimiento según la reivindicación 1, en el que la entrada de voz se procesa en un motor de reconocimiento automático de voz (258).
- 14.- El procedimiento según la reivindicación 1, en el que los primeros datos de página web incluyen datos que pueden convertirse en una primera gramática basada en voz, comprendiendo además el procedimiento proporcionar de manera audible una salida de voz basada en la primera gramática basada en voz.
- 15
- 15.- El procedimiento según la reivindicación 14, en el que la salida audible se produce como respuesta a la entrada de voz recibida.
- 16.- El procedimiento según la reivindicación 2, en el que:
- 20 los primeros datos de página web incluyen además terceros componentes de marcado que pueden convertirse en una gramática de voz que incluye reglas para proporcionar una salida de audio;
- cada regla de la gramática de voz está asociada a una parte de etiqueta; y
- una regla particular de la gramática de voz se carga en una memoria en asociación con una regla particular de la primera gramática basada en reglas mediante la utilización de una misma parte de etiqueta por parte de las reglas particulares de la gramática de voz y la primera gramática basada en reglas.
- 25
- 17.- El procedimiento según la reivindicación 16, que comprende además:
- como respuesta a la recepción, cuando la primera página web visual está activa, de una entrada de voz correspondiente a la parte de frase de la regla particular de la primera gramática basada en reglas, ejecutar el comando de la regla particular de la primera gramática basada en reglas y proporcionar como salida el audio de la regla particular de la gramática de voz.
- 30
- 18.- El procedimiento según la reivindicación 1, que comprende además:
- determinar si añadir una o más directivas a los datos de la primera página web recibidos y, si se determina que van a añadirse una o más directivas, si la una o más directivas se añaden al principio o al final, donde las determinaciones se basan en si los datos de la página web actualmente recibidos incluyen una gramática, en si los datos de una página web recibidos anteriormente incluyen una gramática, y en un procedimiento a través del cual se iniciaron los datos de la página web actualmente recibidos.
- 35
- 19.- El procedimiento según la reivindicación 1, en el que los segundos componentes de marcado incluyen un componente que proporciona una instrucción de recarga invocada por voz para recargar la primera página web visual.
- 20.- El procedimiento según la reivindicación 1, en el que cada una de al menos un subconjunto de reglas de la primera gramática basada en reglas incluye una etiqueta inicial, una asociación de un localizador uniforme de recursos, al menos un elemento de voz para invocar un comando y una etiqueta final.
- 40
- 21.- El procedimiento según la reivindicación 1, en el que los componentes de gramática se proporcionan como datos binarios.
- 22.- El procedimiento según la reivindicación 1, en el que la parte de frase identifica además una pluralidad de alternativas de entrada de voz correspondientes a variables respectivas.
- 45
- 23.- El procedimiento según la reivindicación 1, en el que la parte de frase define además una sección de entrada de voz opcional que no afecta a la ejecución del comando respectivo de la regla que incluye la parte de frase.

24.- Un sistema que proporciona páginas web visuales con una interfaz de audio, que comprende:

una disposición de entrada para recibir, desde un servidor (102), primeros datos de página web que incluyen:

5 (a) primeros componentes de marcado que describen y pueden convertirse en una página web visual que incluye enlaces representados visualmente que pueden seleccionarse cada uno por un usuario para navegar a otra página web respectiva; y

(b) segundos componentes de marcado diferentes de los primeros componentes de marcado y que describen y pueden convertirse en una primera gramática basada en reglas generada independientemente del contenido de los primeros componentes de marcado, donde:

10 la primera gramática basada en reglas define para cada uno del uno o más de los enlaces una regla respectiva que incluye una parte de comando y una parte de frase;

la parte de comando define un comando que puede ejecutarse para navegar hasta una de las páginas web respectivas correspondientes a los enlaces; y

15 la parte de frase identifica una pluralidad de frases alternativas que (i) no se muestran en la página web visual, (ii) no están incluidas en los enlaces y (iii) cada una, cuando se pronuncia por un usuario, invoca por separado la ejecución del comando respectivo de la regla a la que pertenece la parte de frase;

un navegador (256) para convertir los primeros componentes de marcado en los primeros componentes de marcado en la primera página web visual;

un módulo para:

20 recibir una entrada de voz que incluye una frase pronunciada;

hacer corresponder la frase pronunciada con una de las frases alternativas identificadas por una de las reglas definidas por los segundos componentes de marcado; y

25 como respuesta a la correspondencia, navegar desde la primera página web visual hasta la página web respectiva especificada por la parte de comando de la regla que identifica la frase con la que se hizo corresponder la frase pronunciada; y

una base de datos para almacenar al menos una parte de la primera gramática basada en reglas con relación a la primera página web visual.

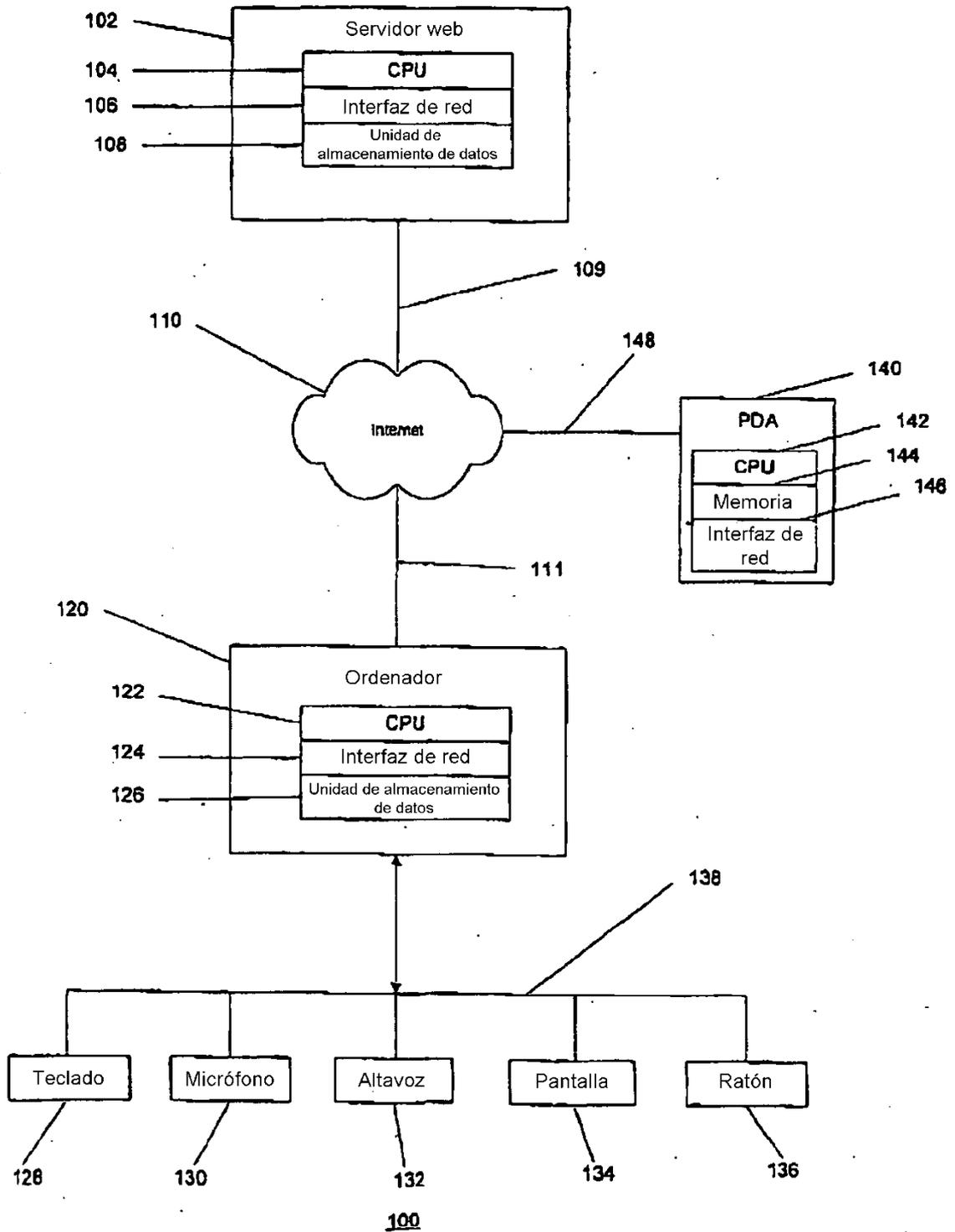
25.- El sistema según la reivindicación 24, en el que los primeros componentes de marcado están en forma de lenguaje de marcado de hipertexto.

30 26.- El sistema según la reivindicación 24, en el que la navegación incluye recibir segundos datos de página web que especifican una segunda página web visual y una segunda gramática basada en reglas asociada.

27.- El sistema según la reivindicación 24, que comprende además:

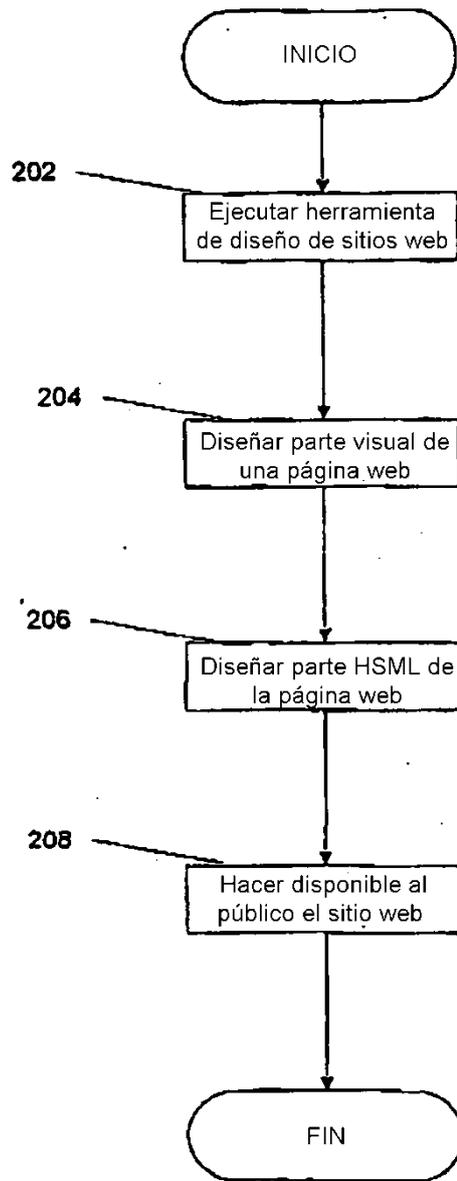
una disposición configurada para proporcionar como salida de manera audible contenido relacionado con la primera página web visual.

35 28.- El sistema según la reivindicación 27, en el que la salida audible se produce como respuesta a la recepción de la entrada de voz.



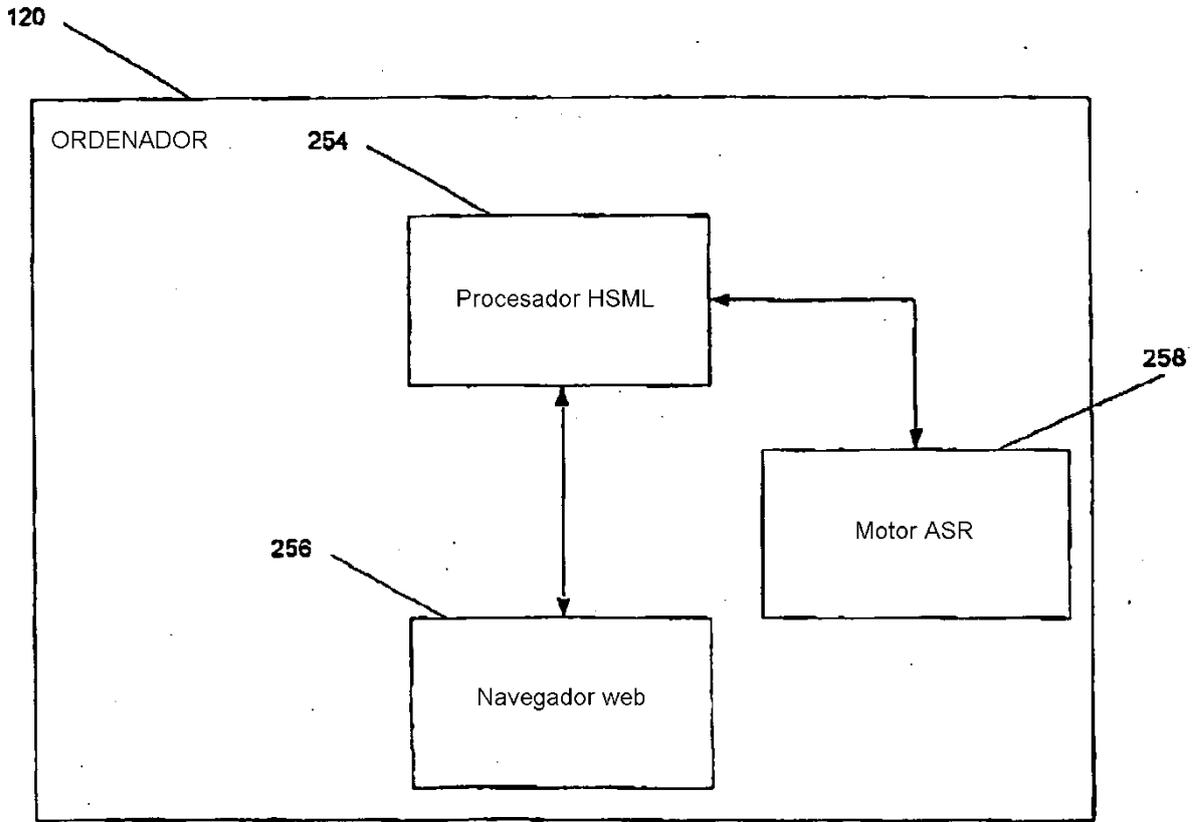
TÉCNICA ANTERIOR

Figura 1



200

Figura 2A



250

Figura 2B

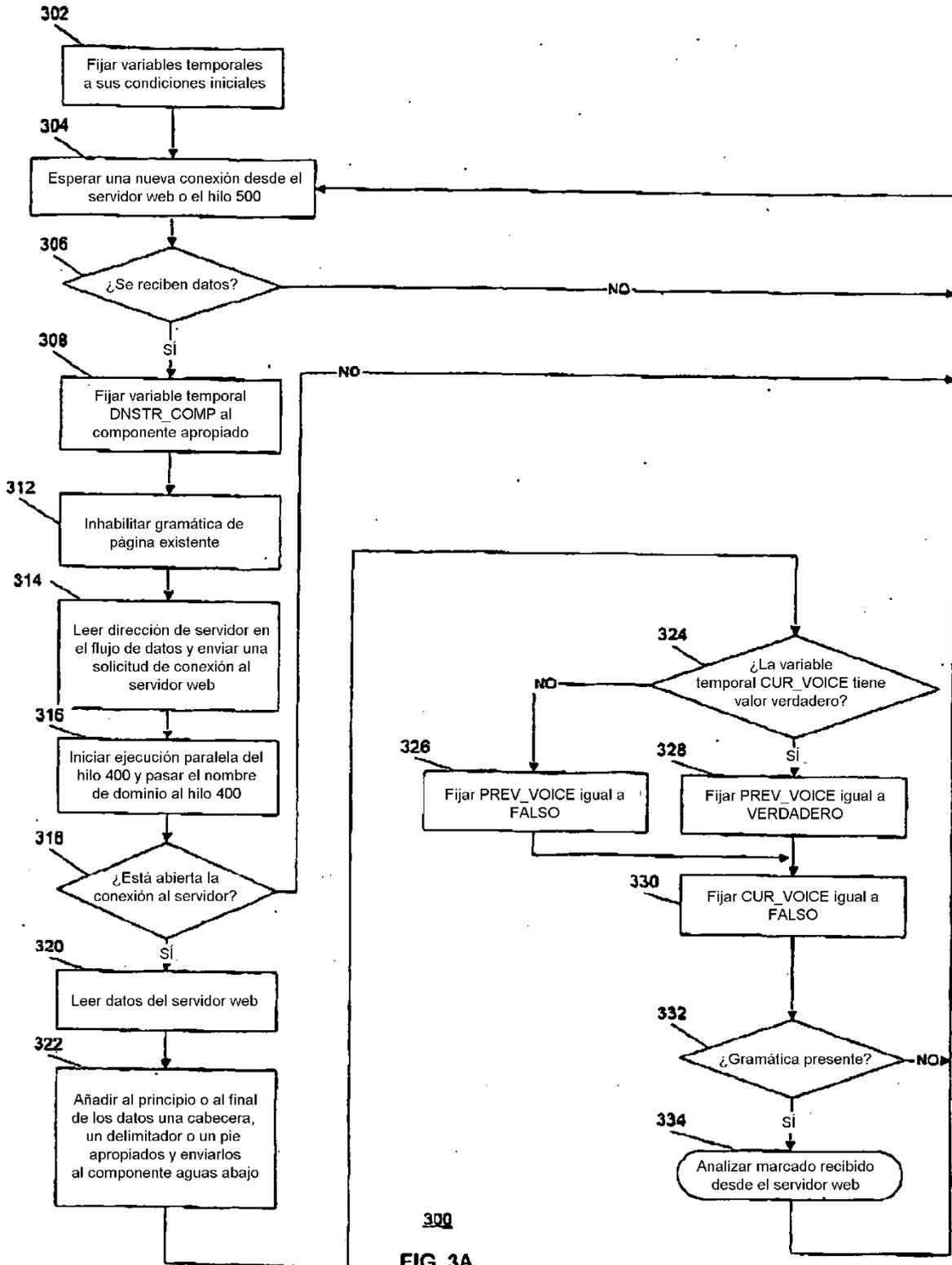
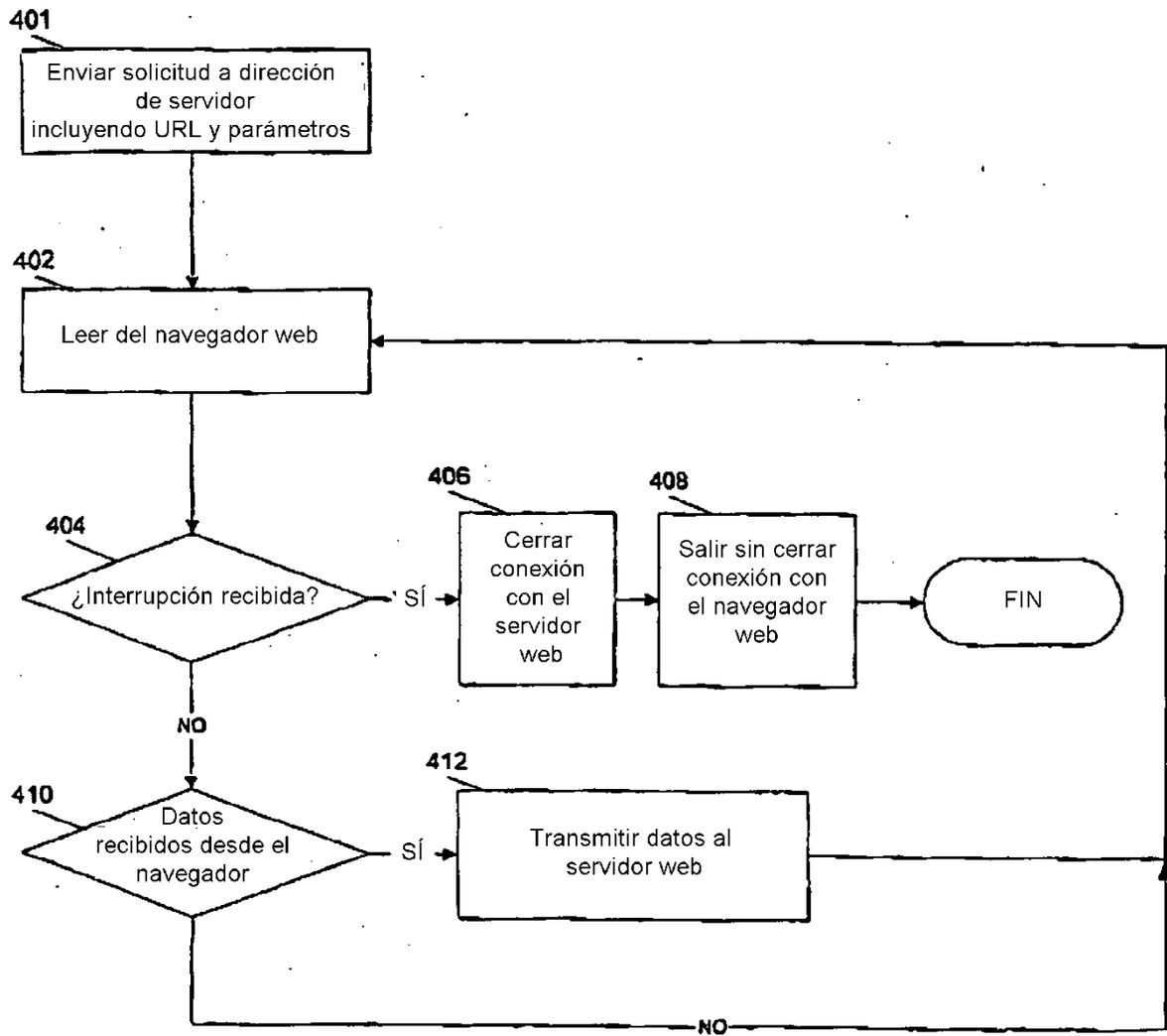


FIG. 3A

	352	354	356	358	360	361
	La página actual tiene gramática	La página anterior tenía gramática	¿Cómo se ha iniciado la página actual?	Primera directiva	Segunda directiva	Añadir al principio/ al final
362	Si	Si	Voz	Nada	Delimitador	Después
364	Si	Si	Clic de Ratón	Cabecera	Delimitador	Antes
366	Si	No	Voz	Cabecera	Delimitador	Antes
368	Si	No	Clic de Ratón	Cabecera	Delimitador	Antes
370	No	Si	Voz	Pie	Cerrar conector	Después
372	No	Si	Clic de Ratón	Nada	Nada	No aplicable
374	No	No	Voz	Nada	Nada	No aplicable
376	No	No	Clic de Ratón	Nada	Nada	No aplicable

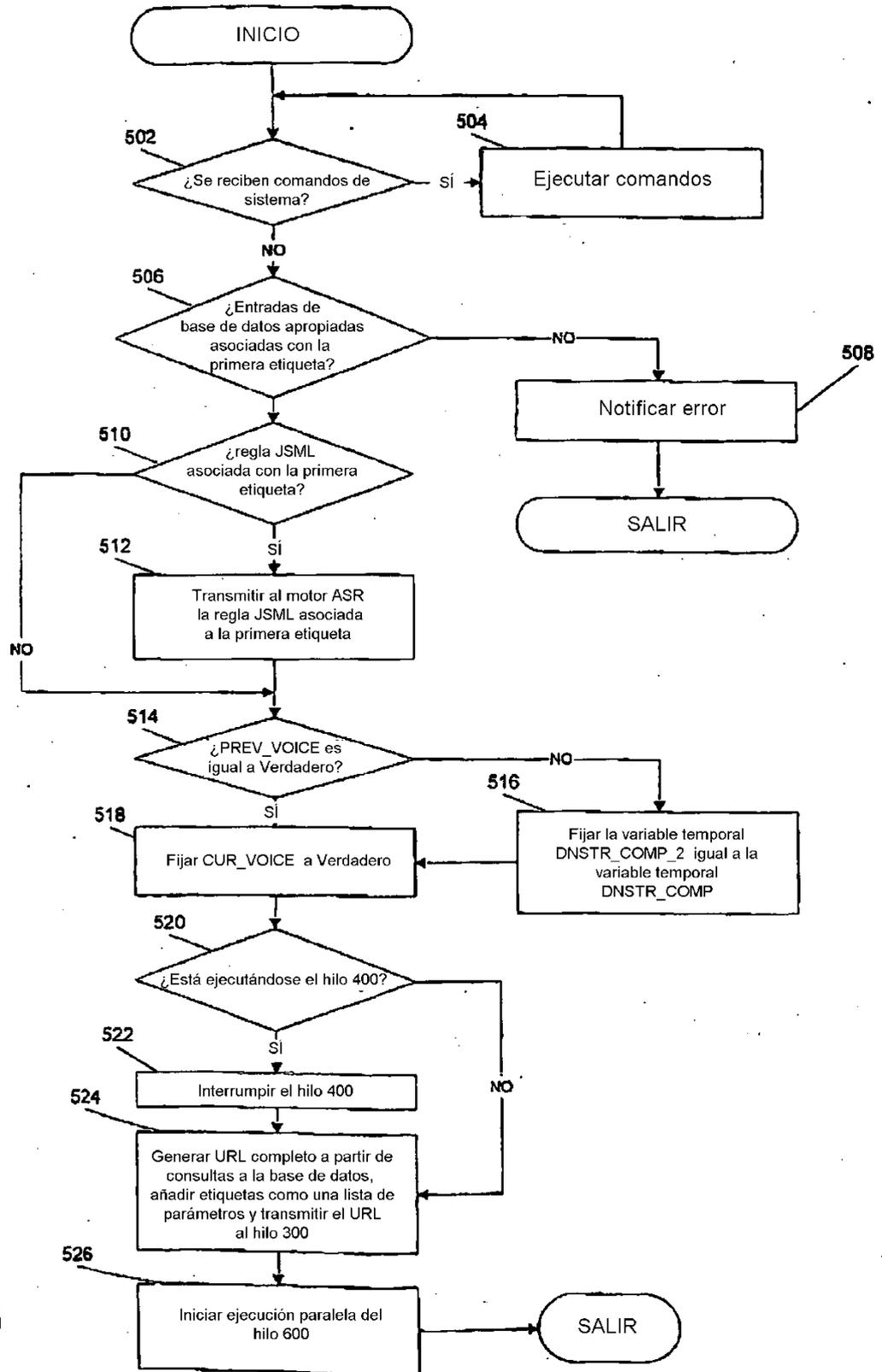
350

FIG. 3B

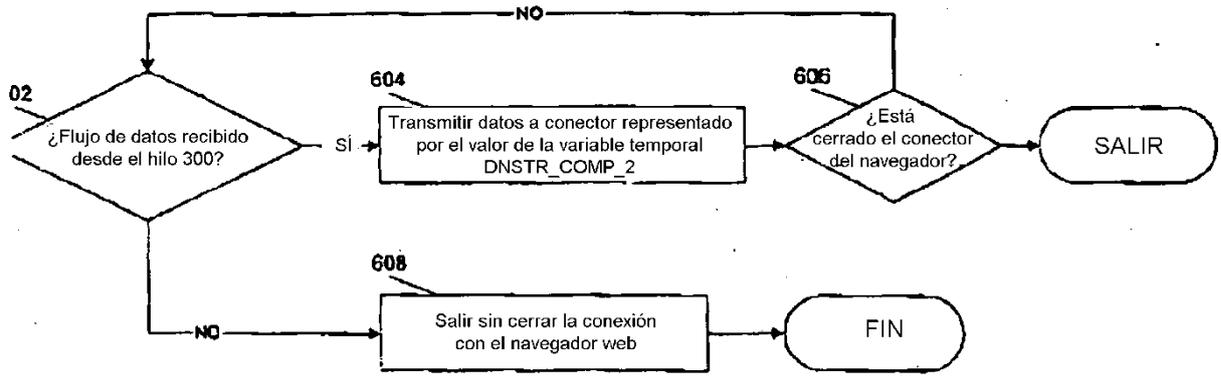


400

FIG. 4

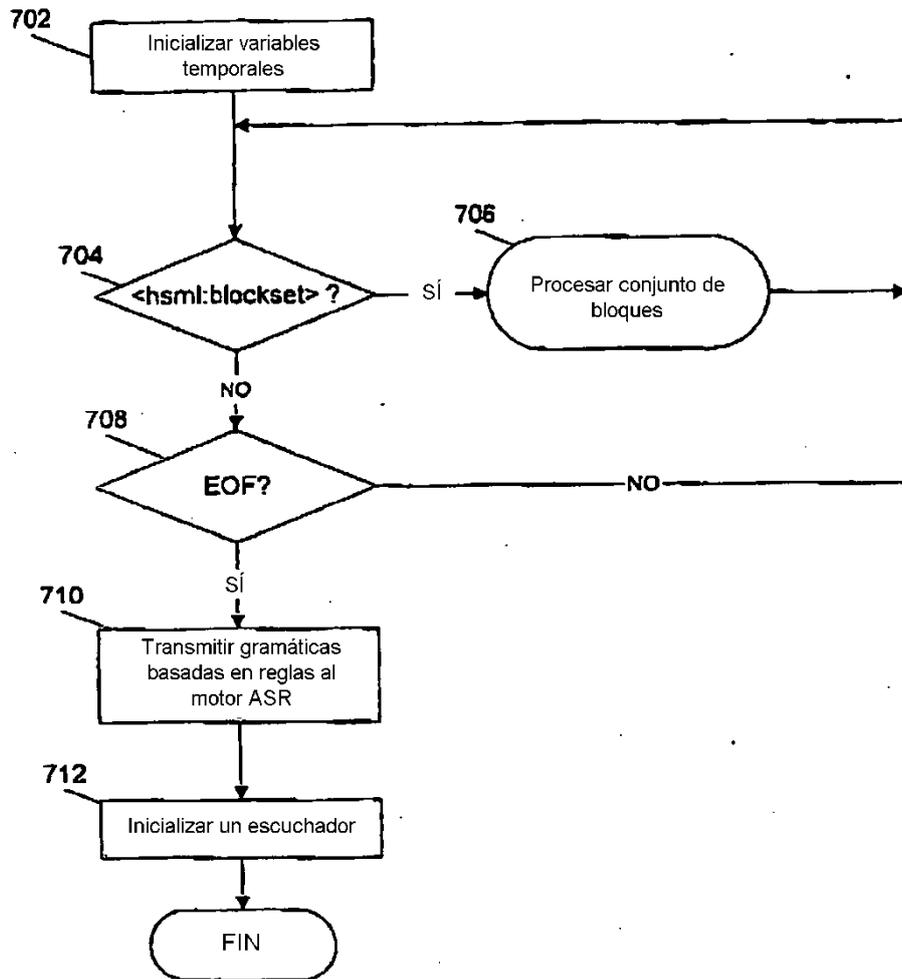


500
FIG. 5



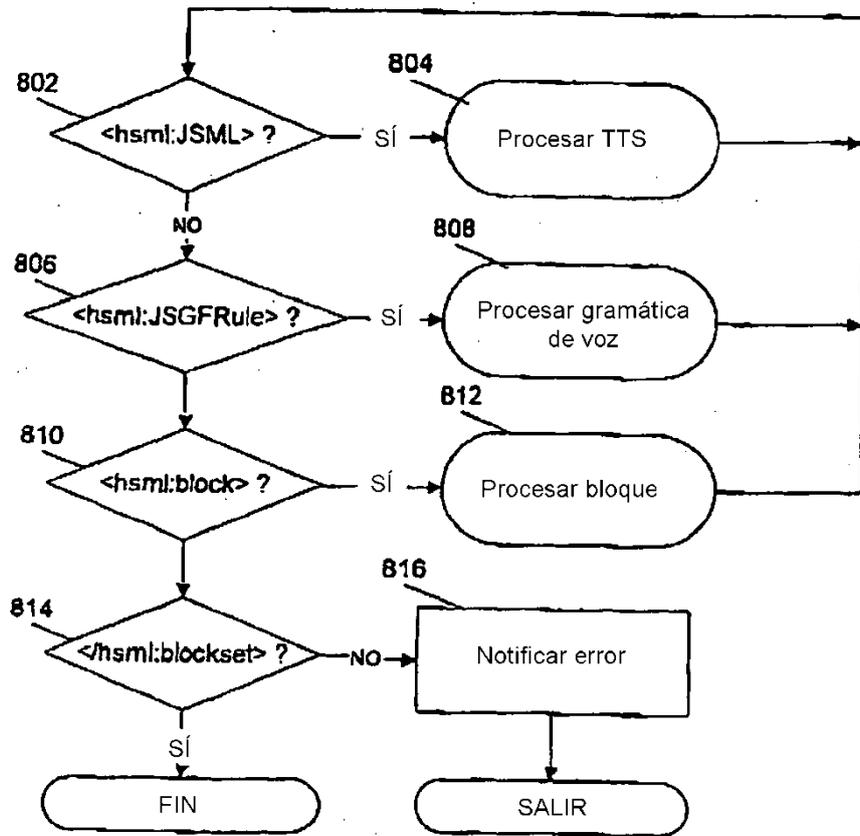
600

FIG. 6



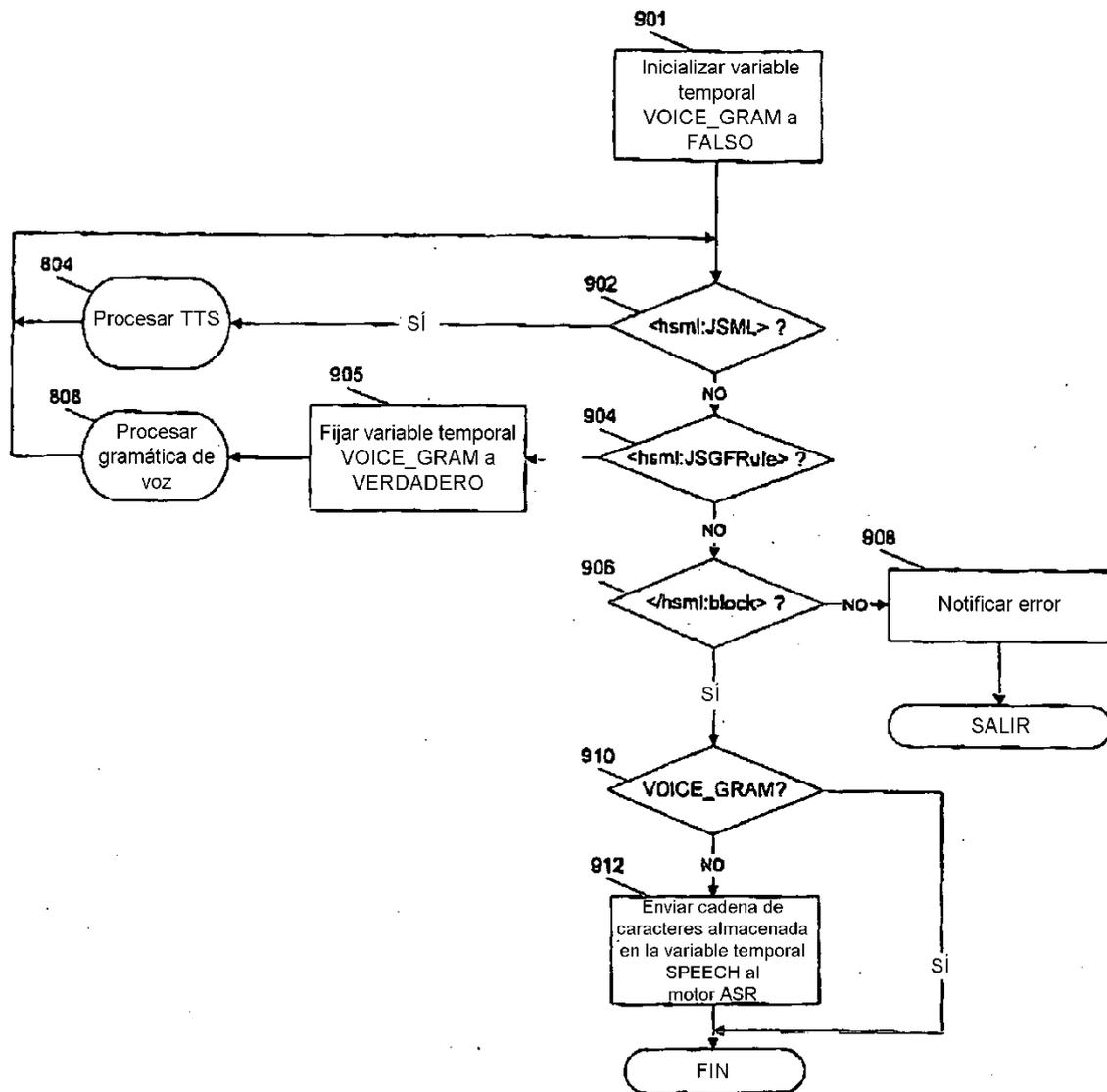
334

FIG. 7



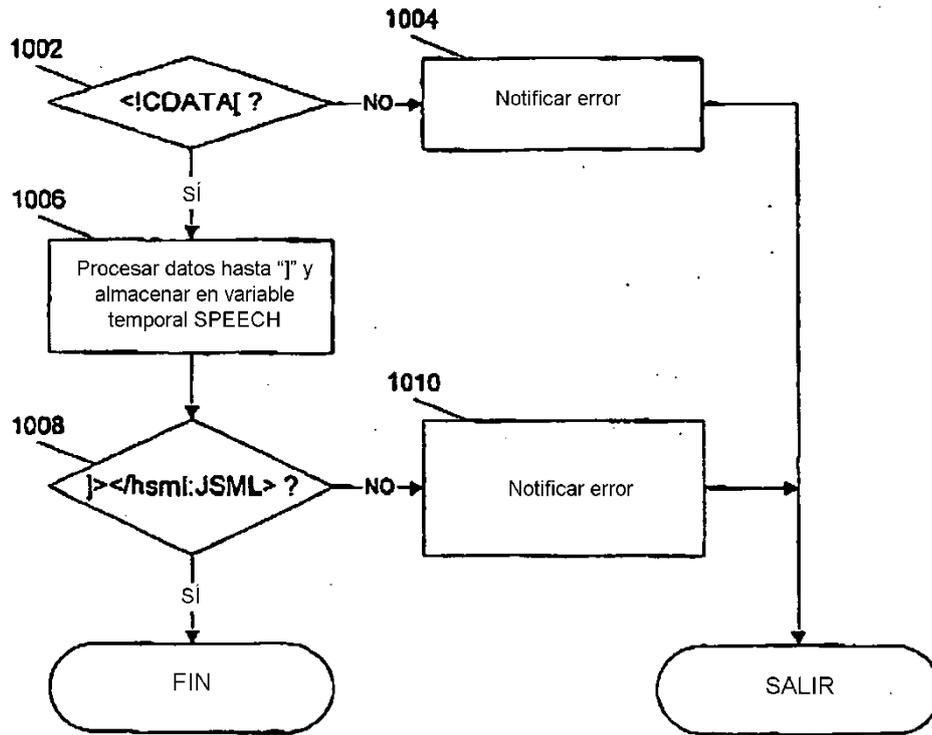
706

FIG. 8



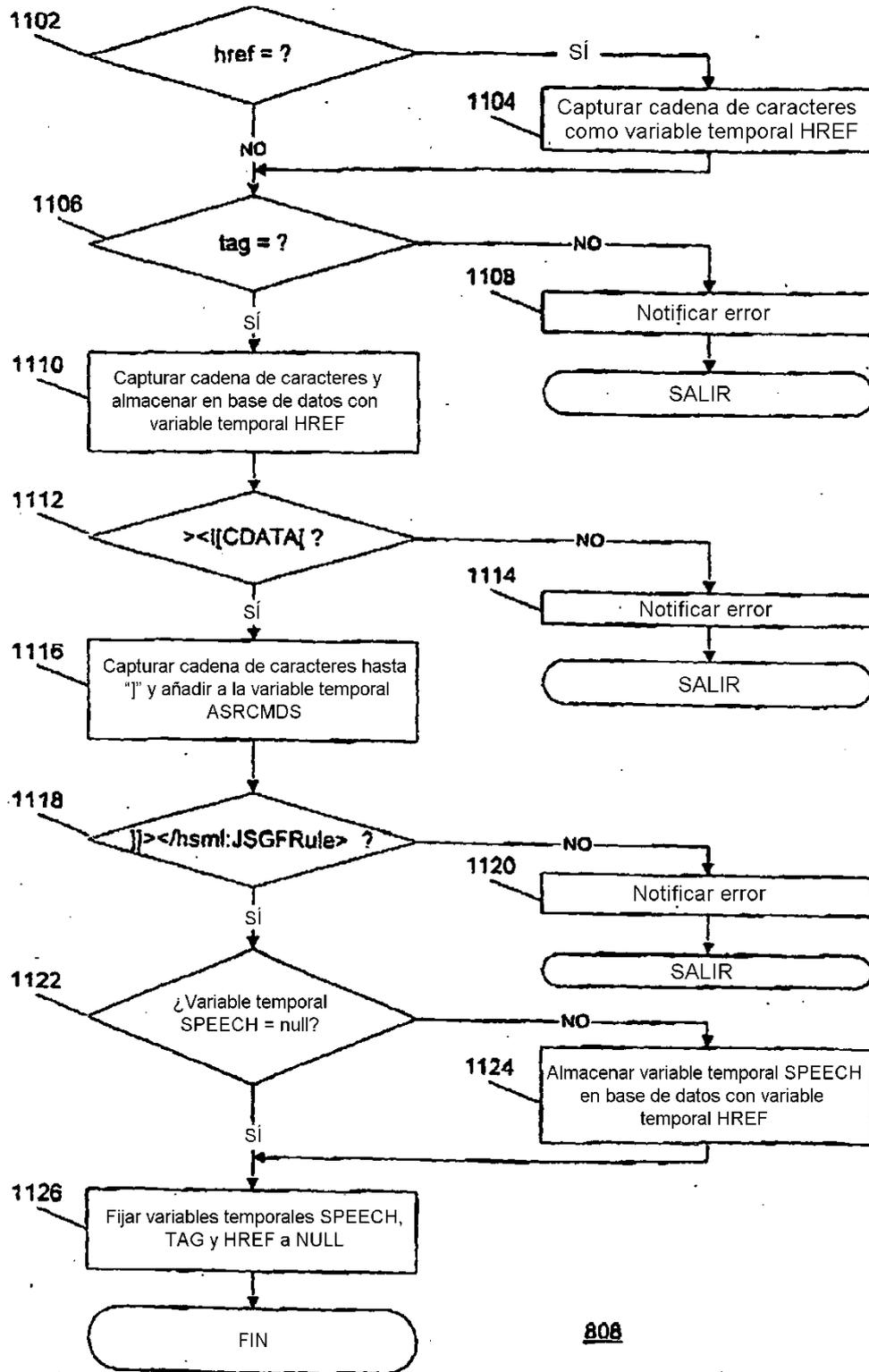
812

FIG. 9



804

FIG. 10



808

FIG. 11