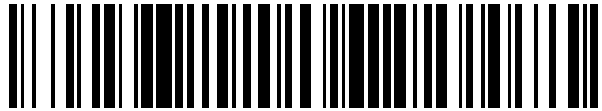


19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 425 565**

51 Int. Cl.:

**H04N 7/24**

(2011.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **28.06.2005 E 05756958 (4)**

97 Fecha y número de publicación de la concesión europea: **19.06.2013 EP 1803299**

54 Título: **Sistema de gestión de meta-datos de programa**

30 Prioridad:

**20.10.2004 GB 0423323**  
**21.10.2004 US 620821 P**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**16.10.2013**

73 Titular/es:

**NDS LIMITED (100.0%)**  
**One London Road**  
**Staines, Middlesex TW18 4EX, GB**

72 Inventor/es:

**BASTABLE, IAN y**  
**PARSONS, IAN**

74 Agente/Representante:

**CARPINTERO LÓPEZ, Mario**

**ES 2 425 565 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**

Sistema de gestión de meta-datos de programa

La presente invención se refiere en general a sistemas de televisión.

**Antecedentes de la invención**

5 Se conocen en la técnica esquemas, tales como esquemas de Emisión de Vídeo Digital - Información de Sistema (DVB-SI), esquemas de TV-Anytime y esquemas del Protocolo de Información de Sistema y Programa (PSIP) del Comité para Sistemas de Televisión Avanzados (ATSC) A65/B, que transportan meta-datos de programa organizados por tiempo.

10 Los esquemas DVB-SI se describen en el documento EN 300 468 del Instituto Europeo de Normas de Telecomunicación (ETSI), que está disponible en el sitio web del ETSI [www.etsi.org](http://www.etsi.org).

Los esquemas de TV-Anytime se describen en el documento TS 102 822-3-1 del ETSI, que está disponible en el sitio web del Foro de TV-Anytime [www.tv-anytime.org](http://www.tv-anytime.org).

Los esquemas PSIP A/65B del ATSC se describen en el sitio web del ATSC [www.atsc.org/standards.html](http://www.atsc.org/standards.html).

15 Los esquemas DVB-SI y TV-Anytime, por ejemplo, también agrupan meta-datos de programa para programas que aparecen más de una vez en un único contenedor virtual. Sin embargo, en tales esquemas no hay acceso a únicamente porciones de un único contenedor virtual y, por lo tanto, el único contenedor virtual no se puede direccionar solo en intervalos de tiempo.

20 El documento US-A-2003/0208761 desvela un procedimiento implementado en un cliente en una arquitectura de emisión en la que un servidor de ficheros de carrusel emite datos de carrusel a uno o más clientes, que comprende recibir una estructura de índice desde el servidor de ficheros de carrusel, teniendo la estructura de datos de índice valores índice derivados del contenido en segmentos correspondientes de los datos de carrusel de emisión, buscar en la estructura de datos de índice para identificar uno de más segmentos particulares de los datos de carrusel de emisión para buscar adicionalmente y recuperar el uno o más segmentos particulares del servidor de ficheros de carrusel.

25 El documento US-A-5844620 desvela un aparato para proporcionar una guía de programa interactiva en la que se transmite información de programación para un periodo de tiempo actual en páginas que se asocian con intervalos de tiempo para posibilitar filtrado en el lado del receptor.

**Sumario de la invención**

30 La presente invención, en realizaciones preferidas de la misma, busca proporcionar un procedimiento de gestión de meta-datos de programa y para posibilitar a un decodificador de salón (STB) manejar meta-datos de programa normalizados de una manera que permita al STB adquirir únicamente porciones de los meta-datos de programa normalizados para presentar una Guía Electrónica de Programas (EPG) que filtra los meta-datos de programa por tiempo.

35 El término "meta-datos" se usa en la presente memoria descriptiva y reivindicaciones para incluir información descriptiva de, o de otra manera, refiriéndose a un flujo de contenido o una porción del mismo. Los meta-datos pueden incluir, por ejemplo, punteros, etiquetas, códigos, banderas, información editorial e información de indexación. Al menos parte de los meta-datos se pueden usar para activar o desactivar operaciones interactivas, tales como operaciones relacionadas con EPG, dependiendo de, por ejemplo, valores asignados a las etiquetas o a las banderas. En el caso donde se asocie el flujo de contenido con al menos un programa de televisión, los punteros, etiquetas, códigos, banderas, información editorial e información de indexación pueden apuntar a, describir y/o indexar segmentos del al menos un programa de televisión así como porciones del flujo de contenido.

40 Se apreciará que los contenidos de los meta-datos pueden pertenecer, preferentemente, a una pluralidad de usuarios o a un usuario individual. Los meta-datos se pueden usar para posibilitar operaciones en el al menos un programa de televisión, si se asocian con el flujo de contenido y/o en contenido en el flujo de contenido. Los meta-datos pueden estar disponibles a partir de una o muchas fuentes de emisión en uno o más momentos cuando se emite el al menos un programa de televisión o en momentos antes de que se emita el al menos un programa de televisión.

45 El término "normalizar", en todas sus formas gramaticales, se usa a lo largo de toda la presente memoria descriptiva y reivindicaciones para referirse a agrupar instancias de programas que comparten al menos un elemento de descripción editorial del contenido.

50 El término "instancia" se usa a lo largo de toda la presente memoria descriptiva y reivindicaciones para referirse a una aparición de un programa o un evento, emitido o transmitido en una fecha y tiempo específicos. Si el mismo programa o evento se emite o transmite más de una vez, cada vez que el programa o el evento se emita o transmita

constituye una instancia del programa o del evento.

La invención proporciona un aparato de acuerdo con la reivindicación 1 y un procedimiento de acuerdo con la reivindicación 16.

Se exponen realizaciones ventajosas en las reivindicaciones dependientes.

## 5 **Breve descripción de los dibujos**

La presente invención se entenderá y apreciará más completamente a partir de la siguiente descripción detallada, tomada en relación con los dibujos en los que:

- La Figura 1 es una ilustración simplificada, parcialmente en diagrama de bloques, parcialmente gráfica de una implementación preferida de un sistema EPG construido y operando de acuerdo con una realización preferida de la presente invención;
- La Figura 2 es una ilustración de diagrama de bloques simplificado de una implementación del aparato para agrupar meta-datos de programa en el sistema EPG de la Figura 1, estando el aparato construido y operando de acuerdo con la presente invención;
- La Figura 3 es una ilustración de diagrama de bloques simplificado de un ejemplo de un diagrama de organización de datos producido mediante el aparato de la Figura 2;
- La Figura 4 es una ilustración de gráfico de barras de un ejemplo de una distribución de ancho de banda entre una pluralidad de flujos PID (Identificación de Paquete);
- La Figura 5 es una ilustración de diagrama de bloques simplificado de una implementación preferida del aparato para acceder a meta-datos de programa.
- La Figura 6 es una ilustración de diagrama de flujo simplificado de un procedimiento de operación preferido del aparato de la Figura 2; y
- La Figura 7 es una ilustración de diagrama de flujo simplificado de un procedimiento de operación preferido del aparato de la Figura 5.

## **Descripción detallada de una realización preferida**

- En tecnología EPG convencional, los datos para dibujar una única pantalla de una parrilla EPG se disponen en la EPG en el momento de dibujar la parrilla EPG. Puesto que, típicamente, un conjunto de datos EPG completo no se almacena en una memoria del STB debido a, por ejemplo, limitaciones de capacidad de memoria en el STB, tal conjunto de datos EPG típicamente se divide y únicamente parte del conjunto de datos EPG se mantiene en la memoria del STB en un momento. El conjunto de datos EPG se divide típicamente por tiempo, de modo que únicamente parte del conjunto de datos EPG, que se requiere durante un periodo de tiempo en el que se está mostrando la parrilla EPG, está cargado realmente en la memoria del STB en el momento de visualización.

- Sin embargo, existen sistemas, tales como sistemas de acuerdo con las especificaciones TV-Anytime como se describen en el sitio web del Foro TV-Anytime [www.tv-anytime.org](http://www.tv-anytime.org), en los que los datos de la parrilla EPG se normalizan. En sistemas que están de acuerdo con las especificaciones TV Anytime, por ejemplo, se obtiene la normalización agrupando por "programa" y, por lo tanto, los meta-datos de todas las instancias que comparten los mismos detalles de programa se agrupan juntos.

- Si se hace la agrupación por programa, tal agrupación puede entrar en conflicto con el conjunto de datos EPG que se divide por tiempo, puesto que todas las instancias de un programa pueden estar en diferentes grupos de tiempo. La presente invención, posibilita agrupación de programas por tiempo mientras que mantiene un conjunto de datos EPG normalizado.

- De acuerdo con la normalización del conjunto de datos EPG de la presente invención, o de partes de una EPG, incluye agrupación de instancias de programas que comparten más de un elemento de descripción editorial del contenido y, preferentemente, agrupación de instancias de programas que comparten la mayoría de los elementos de descripción editorial del contenido o todos los elementos de descripción editorial del contenido.

- Se hace referencia ahora a la Figura 1, que es una ilustración simplificada, parcialmente en diagrama de bloques, parcialmente gráfica de una implementación preferida de un sistema 10 EPG construido y operando de acuerdo con una realización preferida de la presente invención.

- En el sistema 10 EPG, un centro de control de cabecera 20 preferentemente transmite a una unidad 30 de usuario o a una pluralidad de unidades 30 de usuario información EPG así como programación de televisión para la que se pretende la información EPG. A modo de ejemplo, lo que no significa que sea limitante, el centro de control de cabecera 20 transmite la información EPG y la programación de televisión mediante un satélite 40. Como alternativa, el centro de control de cabecera 20 puede transmitir la información EPG y la programación de televisión a la unidad 30 de usuario o a la pluralidad de unidades 30 de usuario mediante al menos uno de los siguientes: una red de comunicación basada en cable; una red de televisión de emisión terrestre convencional; una red de comunicación basada en telefonía; una red de emisión de televisión basada en telefonía; una red de emisión de televisión basada en telefonía móvil; una red de emisión de televisión de Protocolo de Internet (IP) y una red de comunicación basada

en ordenador. Un ejemplo de una red de emisión de televisión basada en IP o telefonía apropiada incluye, por ejemplo, un sistema Synamedia™, disponible comercialmente de NDS Limited, One London Road, Staines, Middlesex, TW 18 4EX, Reino Unido.

5 La programación de televisión preferentemente incluye al menos uno de los siguientes: programación de televisión de pago y/o no de pago; información multimedia; programas de audio; datos; juegos e información de redes basadas en ordenador tales como internet.

10 Por simplicidad de representación y descripción, y sin limitar la generalidad de lo anterior, se ilustra únicamente una unidad 30 de usuario en la Figura 1 y se hace referencia a continuación. La unidad 30 de usuario preferentemente incluye un decodificador de salón (STB) 50 y una pantalla 60 que está operativamente asociada con el STB 50 y funciona para mostrar información EPG y programación de televisión recibida y procesada en el STB 50. La pantalla 60 puede comprender cualquier pantalla apropiada tal como una televisión o un monitor de ordenador. A modo de ejemplo, lo que no significa que sea limitante, el STB 50 recibe la información EPG y la programación de televisión mediante una antena 70.

15 En operación, el centro de control de cabecera 20 normaliza los meta-datos de programa de instancias de programas que comparten al menos un elemento de descripción editorial de contenido y, preferentemente, muchos elementos de descripción editorial del contenido, y agrupa los meta-datos de programa para producir meta-datos de programa normalizados-agrupados que constituyen información EPG para una EPG. El centro de control de cabecera 20 a continuación preferentemente emite programación de televisión y los meta-datos de programa normalizados-agrupados a la unidad 30 de usuario.

20 En la unidad 30 de usuario, la programación de televisión se recibe y procesa en el STB 50 y se muestra en la pantalla 60 de acuerdo con selecciones de un usuario (no mostrado) de la unidad 30 de usuario. Los meta-datos de programa normalizados-agrupados también se reciben y procesan en el STB 50 y el usuario puede introducir, preferentemente, una selección EPG, cuya selección EPG se puede introducir mediante al menos uno de los siguientes: un intervalo de tiempo; un servicio; un programa; y al menos un atributo editorial en los meta-datos de programa. El al menos un atributo editorial preferentemente incluye cualquier atributo editorial apropiado tal como, por ejemplo, una categoría o subcategoría de programa.

Se hace referencia ahora a la Figura 2, que es una ilustración de diagrama de bloques simplificado de una implementación preferida del aparato 100 para agrupar meta-datos de programa en el sistema 10 EPG de la Figura 1, estando construido el aparato 100 y operando de acuerdo con la presente invención.

30 El aparato 100 está comprendido preferentemente en el centro de control de cabecera 20 de la Figura 1. El aparato 100 incluye preferentemente un procesador 120 de meta-datos y una unidad 130 de encapsulación. El procesador 120 de meta-datos puede, por ejemplo, incluir un microprocesador convencional o un microcontrolador convencional. El procesador 120 de meta-datos preferentemente procesa meta-datos de programas ofrecidos en una EPG. El procesador 120 de meta-datos preferentemente normaliza meta-datos de programa de instancias de programas que comparten al menos un elemento de descripción editorial de contenido para producir un conjunto de identificadores de programa, y agrupa las instancias de los programas en intervalos de tiempo para producir un conjunto de identificadores de intervalos de tiempo.

40 La unidad 130 de encapsulación preferentemente coloca los meta-datos de programa en contenedores virtuales, y asocia los contenedores virtuales con el conjunto de identificadores de programa y el conjunto de identificadores de intervalo de tiempo. La expresión "contenedor virtual" se usa a lo largo de toda la presente memoria descriptiva y reivindicaciones para referirse a un bloque de información a transportar, bloque de información que incluye carga útil de comunicación, información de direccionamiento e información de tara. La unidad 130 de encapsulación, por lo tanto, prepara los meta-datos de programa para direccionar por un usuario, como se describe a continuación, a través de un mecanismo de direccionamiento que permite al usuario filtrar los meta-datos de programa mediante al menos uno de los siguientes: tiempo; servicio; programa; y al menos un atributo editorial en los meta-datos de programa.

50 Preferentemente, para normalizar los meta-datos de programa, el procesador 120 de meta-datos explora instancias de programas en la EPG y compara la descripción de contenido de las instancias de los programas para determinar igualdad de meta-datos, tal como igualdad del título, igualdad de la sinopsis, etc. El procesador 120 de meta-datos a continuación preferentemente agrupa, para cada programa, todas las instancias del programa que comparten al menos un elemento de descripción editorial del contenido para producir un identificador de programa que se puede usar para hacer referencia al programa. Por ejemplo, una primera instancia de un único episodio de una telenovela que se programa el lunes por la mañana, y una segunda instancia del mismo episodio de la telenovela que se programa el martes por la tarde, se agrupan juntos como un programa con un identificador de programa único. Preferentemente, tal agrupación de todas las instancias de programas en la EPG durante un periodo EPG de, por ejemplo, una semana da como resultado un conjunto de programas normalizado y un conjunto de identificadores de programa.

- El procesador 120 de meta-datos preferentemente también agrupa las instancias de los programas en intervalos de tiempo para producir un conjunto de identificadores de intervalo de tiempo. Por ejemplo, el procesador 120 de meta-datos puede recopilar todos los programas para cada conjunto de días único en el que los programas tienen instancias. Por lo tanto, se obtiene un grupo de programas que, por ejemplo, tiene las siguientes instancias: todas en el día 0 únicamente, todas en el día 1 únicamente, todas en el día 0 y el día 1 únicamente, etc. Un ejemplo de un diagrama de organización de datos de este tipo que se produce mediante el aparato 100 se representa en la Figura 3.
- Preferentemente, el procesador 120 de meta-datos selecciona el intervalo de tiempo de acuerdo con uno de los siguientes: una base de tiempo lineal; y una base de tiempo no lineal.
- La expresión "base de tiempo lineal" se usa a lo largo de toda la presente memoria descriptiva y reivindicaciones para hacer referencia a una base de tiempo en una forma lineal en la que los intervalos de tiempo se calculan como un polinomio de primer grado de una cantidad de unidad de tiempo. Por ejemplo, para una unidad de tiempo de 1 día, el intervalo de tiempo puede ser un número de días entero, tal como 1 día, 2 días, 3 días, etc.
- Preferentemente, la base de tiempo lineal incluye una unidad de tiempo equivalente a al menos uno de los siguientes: una parte de un día; y un día.
- La expresión "base de tiempo no lineal" se usa a lo largo de toda la presente memoria descriptiva y reivindicaciones para hacer referencia a una base de tiempo distinta de una base de tiempo lineal. Por ejemplo, en una base de tiempo no lineal un primer intervalo de tiempo puede ser 6 horas, un segundo intervalo de tiempo puede ser 6 horas, un tercer intervalo de tiempo puede ser 12 horas, un cuarto intervalo de tiempo puede ser 24 horas, un quinto intervalo de tiempo puede ser 24 horas, etc.
- Preferentemente, independientemente de si se usa una base de tiempo lineal o una base de tiempo no lineal para seleccionar intervalos de tiempo, cada intervalo de tiempo preferentemente incluye al menos uno de los siguientes: una parte de un día; un día; más de un día; una semana; y un día semanal específico en un periodo de al menos un mes, tal como el primer sábado de un mes.
- Preferentemente, la unidad 130 de encapsulación asigna identificadores de contenedor (`ids_de_contenedor`) a los contenedores virtuales en los que los meta-datos de programa están colocados para identificar programas contenidos en un grupo. Los identificadores de contenedor se asignan preferentemente usando una asignación de campo de bits.
- La unidad 130 de encapsulación está operativamente asociada, preferentemente, con un transmisor 140 en el centro de control de cabecera 20. El transmisor 140 preferentemente transmite a la unidad 30 de usuario de la Figura 1, por ejemplo, mediante una antena 150, los contenedores virtuales, el conjunto de identificadores de programa asociado y el conjunto de identificadores de intervalo de tiempo asociado. Se apreciará que los contenedores virtuales junto con el conjunto de identificadores de programa asociado y el conjunto de identificadores de intervalo de tiempo asociado se pueden transportar, por ejemplo, como secciones MPEG-2 privadas en una red de emisión, o como datagramas del Protocolo de Internet (IP) en una red IP. En el caso de que los contenedores virtuales se transporten como secciones MPEG-2 privadas, la `tabla_de_id_de_extensión` y otros campos se establecen como sea necesario para transportar un identificador de contenedor. Los campos MPEG-2 que se establecen se determinan, preferentemente, de acuerdo con un número de bits usado para señalar el identificador de contenedor, y los bits usados preferentemente incluyen los primeros 96 bits en los datos privados de la sintaxis de la sección privada MPEG-2. Se apreciará que cualesquiera otros bits adecuados se pueden usar como alternativa o adicionalmente.
- Los contenedores virtuales junto con el conjunto de identificadores de programa asociado y el conjunto de identificadores de intervalo de tiempo asociado se pueden llevar, preferentemente, en uno de muchos flujos PID (PID - Identificación de Paquete) diferentes. Se apreciará que la asignación de identificadores de contenedor a flujos PID se realiza, preferentemente, de una manera que reduce el ancho de banda por flujo de PID distribuyendo los contenedores virtuales entre una pluralidad de flujos PID y también preferentemente permite filtrado de asignación de bits a lo largo de toda la pluralidad de flujos PID donde el bit puede, por ejemplo, corresponder a un intervalo de tiempo tal como un día específico. Tal filtrado puede, por ejemplo, en un caso donde la cantidad de contenedores virtuales es pequeña, posibilitarse transportando en un flujo PID MPEG-2 distinto todos los contenedores virtuales con identificadores que tengan únicamente un único conjunto de bits, o transportando en un flujo PID MPEG-2 distinto todos los contenedores virtuales que tengan un conjunto de bits específico. Todos los demás contenedores virtuales se pueden transportar a continuación en un único flujo PID que se puede filtrar usando un filtrado de sección de máscara de bits por hardware convencional. En otro ejemplo, cada grupo de contenedores virtuales por unidad única de tiempo se puede llevar en un flujo PID distinto y todos los contenedores virtuales que cubran múltiples periodos de tiempo se pueden llevar en un flujo PID separado.
- Se muestra un ejemplo de una distribución de ancho de banda entre una pluralidad de flujos PID con 147 bytes por programa, un ciclo de tiempo de 3,5 segundos y un tiempo gradual de 60 segundos en la Tabla 1 a continuación y se representa en la forma de un gráfico de barras en la Figura 4.

TABLA 1

Identificador de contenedor	Programas	Porcentaje	Bytes	kbps*
1 (gradual)	1468	10,53%	215796	29
Impar (gradual)	2181	15,64%	320607	43
2	1367	9,80%	200949	459
4	1208	8,66%	177576	406
8	1155	8,28%	169785	388
16	1117	8,01%	164199	375
32	1157	8,30%	170079	389
64	1302	9,34%	191394	437
128	316	2,27%	46452	106
Par	2675	19,18%	393225	899
Total	13946	100,00%	2050062	3,531

\* kbps - kilobits por segundo

5 Se apreciará que convencionalmente un bit que está establecido se refiere al bit que tiene un valor "1" y un bit que está vacío ser al bit que tiene un valor "0". Se ha de entender, sin embargo, que hacer referencia a un bit establecido como que tiene el valor "0" y a un bit vacío que tiene el valor "1" es aplicable, como alternativa, en la presente invención.

Se aprecia adicionalmente que se pueden usar, como alternativa, otros mecanismos de transporte. Por ejemplo, se puede usar difusión o multidifusión de IP cuando se pueda realizar filtrado en una de las siguientes capas del modelo de siete capas de la Organización Internacional de la Normalización (ISO): una Capa de Enlace de Datos; una Capa de Red; y una Capa de Transporte.

10 Se apreciará adicionalmente todavía que la asignación PID permite que se reduzca el ancho de banda por flujo PID mientras que permite todavía bajos tiempos de acceso a datos llevados en el flujo PID. Puesto que algunos tipos de STB convencionales pueden afrontar únicamente con un ancho de banda limitado en un flujo PID, la asignación PID de acuerdo con la presente invención distribuye los contenedores virtuales sobre unos pocos flujos PID reduciendo por lo tanto el ancho de banda por flujo PID a niveles con los que el STP pueda los pueda afrontar. Se apreciará que  
15 los tiempos de acceso a los datos llevados en los flujos PID preferentemente permanecen aceptables.

Preferentemente, el acceso a los contenedores virtuales, el conjunto de identificadores de programa asociado y el conjunto de identificadores de intervalo de tiempo asociado posibilita permitir a un usuario filtrar los meta-datos de programa para obtener un subconjunto de los meta-datos de programa. El subconjunto de meta-datos de programa preferentemente incluye un subconjunto de meta-datos de programa que corresponde con al menos uno de los  
20 siguientes: un intervalo de tiempo; un servicio; un programa; y al menos un atributo editorial en los meta-datos de programa. Por lo tanto, en un caso donde el usuario filtre los meta-datos de programa para obtener un subconjunto de los meta-datos de programa que correspondan a un intervalo de tiempo específico, se posibilita preferentemente al usuario para visualizar una lista de programas a emitir en el intervalo de tiempo específico. En un caso donde el usuario filtre los meta-datos de programa para obtener un subconjunto de los meta-datos de programa que corresponden a un servicio específico, se posibilita preferentemente al usuario para visualizar una lista de programas ofrecidos en el servicio específico. En un caso donde el usuario filtre los meta-datos de programa para obtener un subconjunto de los meta-datos de programa que corresponden a un programa específico, se posibilita preferentemente al usuario para visualizar diversos intervalos de tiempo en los que el programa específico se emitirá. En un caso donde el usuario filtre los meta-datos de programa para obtener un subconjunto de los meta-datos de programa que corresponden a al menos un atributo editorial en los meta-datos de programa, se posibilita preferentemente al usuario para visualizar una lista de programas que comparten el al menos un atributo editorial.

Se hace referencia ahora a la Figura 5, que es una ilustración de diagrama de bloques simplificado de una implementación preferida del aparato 200 para acceder a meta-datos de programa.

El aparato 200 está comprendido preferentemente en o asociado con el STB 50 de la Figura 1. Preferentemente, el  
35 aparato 200 incluye los siguientes elementos: un receptor 210; una unidad 220 de filtrado y demultiplexación; una unidad 230 de entrada; un procesador 240; y una unidad 250 de presentación en pantalla (OSD). El receptor 210, la unidad 230 de entrada, el procesador 240 y a unidad 250 OSD pueden preferentemente incluir elementos convencionales de un STB convencional. La unidad 220 de filtrado y demultiplexación preferentemente incluye hardware de filtrado y demultiplexación convencional, y en un caso donde se use MPEG-2, la unidad 220 de filtrado y demultiplexación preferentemente incluye hardware de filtrado y demultiplexación MPEG-2 convencional.  
40

El receptor 210 preferentemente recibe una pluralidad de contenedores virtuales de meta-datos de programa, comprendiendo al menos uno de los contenedores virtuales meta-datos de programa normalizados de programas

que tienen instancias en una pluralidad de intervalos de tiempo. El receptor 210 también preferentemente recibe un conjunto de identificadores de programa y un conjunto de identificadores de intervalo de tiempo que están asociados con los contenedores virtuales. Se apreciará que el receptor 210 puede estar comprendido en o asociado con un receptor y decodificador integrado convencional (IRD) (no mostrado) del STB 50.

- 5 La unidad 230 de entrada preferentemente recibe una entrada que indica un intervalo de tiempo seleccionado, la entrada, preferentemente al menos una, comprende y se refiere a un identificador de intervalo de tiempo que identifica el intervalo de tiempo seleccionado. La unidad 230 de entrada puede, por ejemplo, incluir una unidad de interfaz (no mostrada) que recibe la entrada que indica el intervalo de tiempo seleccionado, así como otras posibles instrucciones, desde un usuario (no mostrado) mediante un control 260 remoto que se comunica mediante cualquier canal de comunicaciones apropiado tal como, por ejemplo, un enlace infrarrojo inalámbrico.

10 El procesador 240 preferentemente usa el identificador de intervalo de tiempo para filtrar la pluralidad de contenedores virtuales para obtener un subconjunto de los meta-datos de programa que se corresponden con el intervalo de tiempo seleccionado. El identificador de intervalo de tiempo incluye un bit que identifica el intervalo de tiempo seleccionado y, preferentemente, exactamente un bit que identifica el intervalo de tiempo seleccionado.

- 15 El intervalo de tiempo seleccionado preferentemente incluye al menos uno de los siguientes: una parte de un día; un día; más de un día; una semana; y un día semanal específico en un periodo de al menos un mes, tal como el primer sábado de un mes.

Se apreciará que la unidad 230 de entrada puede, como alternativa o adicionalmente, recibir desde el usuario una entrada que indica un programa seleccionado, la entrada, preferentemente al menos una, comprende y se refiere a un identificador de programa que identifica el programa seleccionado. En tal caso, el procesador 240 preferentemente usa el identificador de programa para filtrar la pluralidad de contenedores virtuales para obtener un conjunto de los meta-datos de programa que se corresponden con el programa seleccionado. El identificador de programa preferentemente incluye un identificador de contenedor virtual (id\_de\_contenedor).

- 20 Además, como alternativa o adicionalmente, la unidad 230 de entrada puede recibir desde el usuario una entrada que indica un servicio seleccionado, la entrada, preferentemente al menos una, comprende y se refiere a un identificador de servicio que identifica el servicio seleccionado. En tal caso, el procesador 240 preferentemente usa el identificador de servicio para filtrar la pluralidad de contenedores virtuales para obtener un subconjunto de los meta-datos de programa que se corresponden con el servicio seleccionado. El identificador de servicio preferentemente incluye un id\_de\_contenedor.

- 25 Aún más, como alternativa o adicionalmente, la unidad 230 de entrada puede recibir desde el usuario una entrada que indica al menos un atributo editorial en los meta-datos de programa. En tal caso, el procesador 240 preferentemente filtra la pluralidad de contenedores virtuales para obtener un subconjunto de los meta-datos de programa que se corresponden con el al menos un atributo editorial.

- 30 En operación, el receptor 210 preferentemente recibe transmisiones en un formato MPEG-2. Las transmisiones preferentemente incluyen una pluralidad de contenedores virtuales de meta-datos de programa así como un conjunto de identificadores de programa y un conjunto de identificadores de intervalo de tiempo que están asociados con los contenedores virtuales. El procesador 240 preferentemente procesa los contenedores virtuales, el conjunto de identificadores de programa y el conjunto de identificadores de intervalo de tiempo de acuerdo con entradas introducidas por el usuario mediante el control 260 remoto. Los resultados de procesamiento del procesador 240 se proporcionan, por ejemplo, a la unidad 250 OSD para mostrar en la pantalla 60 de la Figura 1.

- 35 Preferentemente, el hardware de filtrado y demultiplexación MPEG-2 convencional en la unidad 220 de filtrado y demultiplexación permite adquisición de secciones usando un "filtro". Un filtro de este tipo preferentemente se aplica mediante el procesador 240 y permite que un número de bits de un comienzo de una sección MPEG-2 se compare con un valor de filtro de una manera tal que algunos bits se puedan ignorar, y los bits que no se ignoran se pueden comparar con un valor deseado. Por lo tanto, el hardware de filtrado y demultiplexación MPEG-2 en el STB 50 y el procesador 240 se pueden usar mediante el aparato 200 para filtrar secciones MPEG-2 en una tabla\_de\_id\_de\_extensión PID y en otros campos para un bit o bits dados que se establecen o vacían en un id\_de\_contenedor, y como un resultado para adquirir únicamente programas que se han de presentar en una EPG para un intervalo de tiempo pre-especificado.

- 40 Un experto en la materia apreciará que se puede realizar filtrado similar en datagramas IP en una red IP.

Se apreciará que el aparato 200 de la Figura 5 junto con el aparato 100 de la Figura 2 se pueden denominar como un mecanismo de direccionamiento que permite al usuario filtrar los contenedores virtuales de meta-datos de programa por al menos uno de los siguientes: tiempo; servicio; programa; y al menos un atributo editorial en los meta-datos de programa.

- 55 Un ejemplo de la operación del aparato 100 de la Figura 2 y el aparato 200 de la Figura 5 se describirá ahora brevemente.

Un programa tiene, por ejemplo, instancias que ocurren los días  $d_0, d_1, \dots, d_n$ , donde  $n$  es el número total de días discretos en el que el programa tiene una instancia, y  $d_m$  es un día dado en el que el programa tiene una instancia. El programa, preferentemente, se coloca a continuación en un contenedor virtual cuyo identificador está basado en números de bits  $d_0, d_1, \dots, d_n$  como sigue:

$$\text{id\_de\_contenedor} = 2^{d_0} + 2^{d_1} + \dots + 2^{d_n} \quad (1)$$

Por ejemplo, si el programa tiene instancias en  $d_0$  y  $d_3$ , el  $\text{id\_de\_contenedor}$  será:

$$\text{id\_de\_contenedor} = 2^{d_0} + 2^{d_3}$$

Para adquirir todos los contenedores virtuales para todos los programas que tengan instancias en un día dado  $d_m$ , se adquieren preferentemente todos los contenedores virtuales que tengan el  $\text{id\_de\_contenedor}$   $d_m$  establecido. Por lo tanto, si el usuario introduce una instrucción para mostrar programas que tienen instancias el día  $d_m$  en la EPG, el aparato 200 preferentemente filtra realizando la siguiente operación lógica:  $\text{id\_de\_contenedor AND } 2^{d_m}$ .

Con referencia, por ejemplo, a un conjunto de datos EPG de 8 días que comprende los días 0 - 7, existen 8 posibles posiciones de bit. El  $\text{id\_de\_contenedor}$  puede, por lo tanto, ser cualquier valor de 1 a 255 para todas las permutaciones de los 8 días ( $\text{id\_de\_contenedor}$  0 no aparece debido a que los programas que aparecen en ningún día no existen de ninguna manera). Todos los contenedores virtuales para el día 0 tienen el bit 0 establecido, de esta manera 1, 3, 5, 7, ..., 253 y 255. Por lo tanto, el día 0 tendría 128 contenedores virtuales. Lo mismo se aplica para cualquier otro día  $d_x$  que tendría el bit  $x$  establecido y 128 permutaciones de los 7 bits restantes. Por lo tanto, se apreciará que se requieren los 128 contenedores virtuales para cualquier día.

Si todos los 128 contenedores virtuales para el día 0 se almacenan en una memoria caché (no mostrada), a continuación para cualquier otro día únicamente se adquirirán 64 contenedores virtuales. La razón para esto es que ya se han adquirido 128 contenedores virtuales para el día 0 de los cuales el 50% tienen cualquier bit dado distinto del bit 0 establecido, y el 50% tienen el bit vacío, y por lo tanto restan 64 contenedores virtuales más por adquirir (todos los del bit 0 vacío). A continuación, se realiza filtrado mediante una operación lógica que comprende  $\text{id\_de\_contenedor AND } (2^0 | 2^m)$  para obtener los contenedores virtuales con el bit establecido que no están en la memoria caché, es decir los del bit 0 vacío.

Como se ha mencionado anteriormente, la base de tiempo no tiene que ser lineal y no tiene que hacer referencia a días enteros. Por ejemplo, las unidades se pueden elegir en un modo no lineal para incluir, por ejemplo, lo siguiente:

- Unidad 0 - las primeras 6 horas
- Unidad 1 - las siguientes 12 horas
- Unidad 2 - las siguientes 24 horas
- Unidad 3 - las siguientes 24 horas
- Etc.

En tal caso, en la expresión (1) anterior la Unidad 0 preferentemente reemplaza  $d_0$ , la Unidad 1 reemplaza  $d_1$ , etc.

Los contenedores virtuales almacenados en caché se pueden organizar de una manera que es más adecuada para almacenar datos en caché. Con referencia, por ejemplo, a los contenedores virtuales almacenados en caché en el intervalo de tiempo 0, un número dado de contenedores virtuales, por ejemplo 64 contenedores virtuales, preferentemente se selecciona y se rellenan con datos igualmente. Por lo tanto, todos los 64 contenedores virtuales tienen aproximadamente el mismo tamaño. Como resultado, un número entero  $n$  de contenedores virtuales se puede almacenar en caché teniendo únicamente suficiente capacidad de almacenamiento de memoria de acceso aleatorio (RAM) para almacenar un total de  $n+1$  contenedores virtuales y mantener acceso todavía a los datos durante una actualización.

Con referencia, por ejemplo, a un conjunto de datos EPG de 7,25 días, el número total de programas es, por ejemplo 13946, y se estima una media de 135 bytes sin comprimir por programa para cadenas de título y sinopsis. Una instancia de un programa es, por ejemplo 10 bytes, y las cadenas se pueden comprimir al 70%. Se hace normalización por título y sinopsis únicamente. Tomando, por ejemplo, las instancias que han comenzado o comienzan en las primeras 6 horas para ser 1357 y la cantidad de cadenas sin comprimir requeridas para esas instancias para ser 183189 bytes, una capacidad de memoria del STB requerida es  $((183189 * 70\%) + (1357 * 10)) * 65/64 = 141$  Kbytes (aproximadamente).

Se hace referencia ahora a la Figura 6, que es una ilustración de diagrama de flujo simplificado de un procedimiento de operación preferido del aparato 100 de la Figura 2.

Preferentemente, se normalizan los meta-datos de programa de instancias de programas que comparten al menos un elemento de descripción editorial del contenido (etapa 300) para producir un conjunto de identificadores de



5 programa. La normalización preferentemente se realiza para reducir el ancho de banda de transmisión de los meta-datos de programa. Las instancias de los programas preferentemente se agrupan (etapa 310) en intervalos de tiempo para producir un conjunto de identificadores de intervalo de tiempo. Los meta-datos de programa a continuación preferentemente se colocan (etapa 320) en contenedores virtuales, y los contenedores virtuales preferentemente se asocian (etapa 330) con el conjunto de identificadores de programa y el conjunto de identificadores de intervalo de tiempo.

Se hace referencia ahora a la Figura 7, que es una ilustración de diagrama de flujo simplificado de un procedimiento de operación preferido del aparato 200 de la Figura 5.

10 Preferentemente, se reciben (etapa 400) una pluralidad de contenedores virtuales de meta-datos de programa. Al menos uno de los contenedores virtuales preferentemente incluye meta-datos de programa normalizados de programas que tienen instancias en una pluralidad de intervalos de tiempo. Preferentemente se recibe (etapa 410) una entrada que indica un intervalo de tiempo seleccionado, donde la entrada, al menos una, comprende y se refiere a un identificador de intervalo de tiempo que identifica el intervalo de tiempo seleccionado. El identificador de intervalo de tiempo preferentemente se usa (etapa 420) para filtrar la pluralidad de contenedores virtuales para  
15 obtener un subconjunto de los meta-datos de programa que corresponden con el intervalo de tiempo seleccionado.

Se apreciará que diversas características de la invención que se describen por claridad en los contextos de realizaciones separadas, se pueden proporcionar también en combinación en una única realización. Al contrario, diversas características de la invención que se describen por brevedad en el contexto de una única realización, se pueden proporcionar también por separado o en cualquier subcombinación adecuada.

20 Se apreciará por los expertos en la materia que la presente invención no está limitada por lo que se ha mostrado y descrito particularmente anteriormente en el presente documento. En su lugar, el alcance de la invención se define mediante las reivindicaciones adjuntas.

**REIVINDICACIONES**

1. Aparato para agrupar meta-datos de programa que relaciona todos los programas ofrecidos para visualizar en una Guía Electrónica de Programas, comprendiendo el aparato:

un procesador (120) de meta-datos que es operativo para realizar lo siguiente:

5 explorar los meta-datos de programa para determinar instancias de programa que comparten al menos un elemento de meta-datos editorial en los meta-datos de programa;  
agrupar los meta-datos de programa en grupos de modo que cada grupo se refiere a un conjunto de instancias de programa que comparten al menos un elemento de meta-dato editorial de los meta-datos de programa para producir meta-datos de programa normalizados;  
10 asignar a cada grupo de meta-datos de programa agrupado un identificador de programa único, produciendo de esta manera un conjunto de identificadores de programa; y  
agrupar conjuntos de instancias de programa mediante intervalos de tiempo para producir un conjunto de identificadores de intervalos de tiempo en el que cada identificador de intervalo de tiempo se asocia con al menos uno de los intervalos de tiempo; y

15 una unidad (130) de encapsulación que funciona para colocar los meta-datos de programa en contenedores virtuales, y para asociar cada contenedor virtual con al menos un identificador de programa a partir del conjunto de identificadores de programa y un identificador de intervalo de tiempo a partir del conjunto de identificadores de intervalo de tiempo, en el que al menos uno de los contenedores virtuales comprende metadatos de programa normalizados de programas que tienen instancias en una pluralidad de intervalos de tiempo.

20 2. Aparato de acuerdo con la reivindicación 1 y que comprende también:

un transmisor (140) asociado operativamente con la unidad (130) de encapsulación y que es operativo para transmitir los contenedores virtuales, el conjunto de identificadores de programa y el conjunto de identificadores de intervalos de tiempo.

25 3. Aparato de acuerdo con la reivindicación 2 y en el que el transmisor (140) también es operativo para transportar los contenedores virtuales, el conjunto de identificadores de programa y el conjunto de identificadores de intervalos de tiempo como secciones privadas MPEG-2.

4. Aparato de acuerdo con la reivindicación 2 y en el que el transmisor (140) también es operativo para transportar los contenedores virtuales, el conjunto de identificadores de programa y el conjunto de identificadores de intervalos de tiempo en al menos un flujo PID (Identificación de Paquete).

30 5. Aparato de acuerdo con la reivindicación 4 y en el que cada uno de los contenedores virtuales se asigna a un identificador de contenedor, y el transmisor (140) es operativo también para transportar en un flujo PID distinto todos los contenedores virtuales con identificadores de contenedor que tienen un conjunto de bits específico.

6. Aparato de acuerdo con la reivindicación 4 y en el que el transmisor (140) también es operativo para transportar todos los contenedores virtuales que cubren múltiples periodos de tiempo en un flujo PID distinto.

35 7. Aparato de acuerdo con la reivindicación 4 y en el que el transmisor (140) también es operativo para transportar en cada flujo PID distinto un grupo de contenedores virtuales que cubre una única unidad de tiempo.

8. Aparato de acuerdo con la reivindicación 2 y en el que el transmisor (140) también es operativo para reducir el ancho de banda por flujo PID distribuyendo los contenedores virtuales entre una pluralidad de flujos PID.

40 9. Aparato de acuerdo con la reivindicación 8 y en el que el transmisor (140) también es operativo para transportar uno de los siguientes en uno separado de la pluralidad de flujos PID para posibilitar filtrado de asignación de bits de la pluralidad de flujos PID:

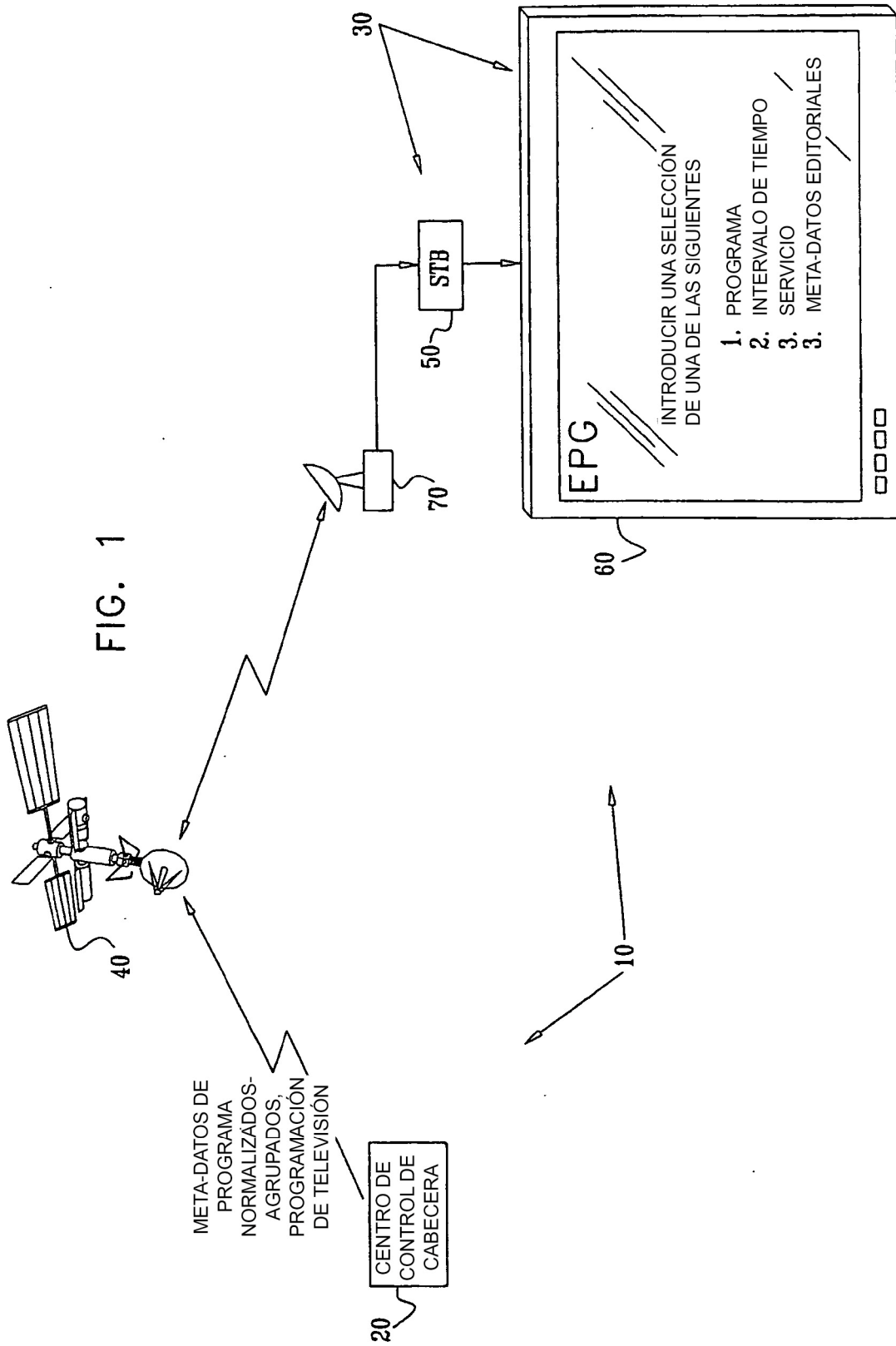
todos los contenedores virtuales que comprenden identificadores de contenedor que tienen uno de los siguientes: únicamente un único conjunto de bits; y un conjunto de bits específico; y  
un grupo de contenedores virtuales que cubre una única unidad de tiempo.

45 10. Aparato de acuerdo con cualquiera de las reivindicaciones 1 - 9 y en el que el procesador (120) de meta-datos también es operativo para seleccionar los intervalos de tiempo de acuerdo con una base de tiempo lineal.

11. Aparato de acuerdo con la reivindicación 10 y en el que la base de tiempo lineal comprende una unidad de tiempo equivalente a al menos una de las siguientes: una parte de un día; y un día.

50 12. Aparato de acuerdo con cualquiera de las reivindicaciones 1 - 9 y en el que el procesador (120) de meta-datos también es operativo para seleccionar los intervalos de tiempo de acuerdo con una base de tiempo no lineal.

13. Aparato de acuerdo con cualquiera de las reivindicaciones 1 - 12 y en el que al menos uno de los intervalos de tiempo comprende al menos uno de los siguientes: una parte de un día; un día; más de un día; una semana; y un día semanal específico en un periodo de al menos un mes.
- 5 14. Aparato de acuerdo con cualquiera de las reivindicaciones 1 - 13 y en el que la unidad (130) de encapsulación también es operativo para posibilitar acceso a los contenedores virtuales, el conjunto de identificadores de programa y el conjunto de identificadores de intervalo de tiempo para permitir a un usuario filtrar los meta-datos de programa para obtener un subconjunto de los meta-datos de programa.
- 10 15. Aparato de acuerdo con la reivindicación 14 y en el que el subconjunto de los meta-datos de programa comprende un subconjunto de meta-datos de programa que corresponde con al menos uno de los siguientes: un intervalo de tiempo; un servicio; un programa; y al menos un elemento de meta-datos editorial en los meta-datos de programa.
16. Un procedimiento para agrupar meta-datos de programa que relaciona todos los programas ofrecidos para visualizar en una Guía Electrónica de Programas, comprendiendo el procedimiento:
- 15 explorar los meta-datos de programa para determinar instancias de programa que comparten al menos un elemento de meta-datos editorial en los meta-datos de programa;
- agrupar los meta-datos de programa en grupos de modo que cada grupo se refiere a un conjunto de instancias de programa que comparten al menos un elemento de meta-datos editorial en los meta-datos de programa para producir meta-datos de programa normalizados;
- 20 asignar a cada grupo de meta-datos de programa agrupado un identificador de programa único, produciendo de esta manera un conjunto de identificadores de programa; insertar parte correspondiente de la reivindicación 1;
- agrupar conjuntos de instancias de programa mediante intervalos de tiempo para producir un conjunto de identificadores de intervalo de tiempo;
- colocar los meta-datos de programa en contenedores virtuales; y
- 25 asociar cada contenedor virtual con al menos un identificador de programa a partir del conjunto de identificadores de programa y un identificador de intervalo de tiempo a partir del conjunto de identificadores de intervalo de tiempo, en el que al menos uno de los contenedores virtuales comprende metadatos de programa normalizados de programas que tienen instancias en una pluralidad de intervalos de tiempo.



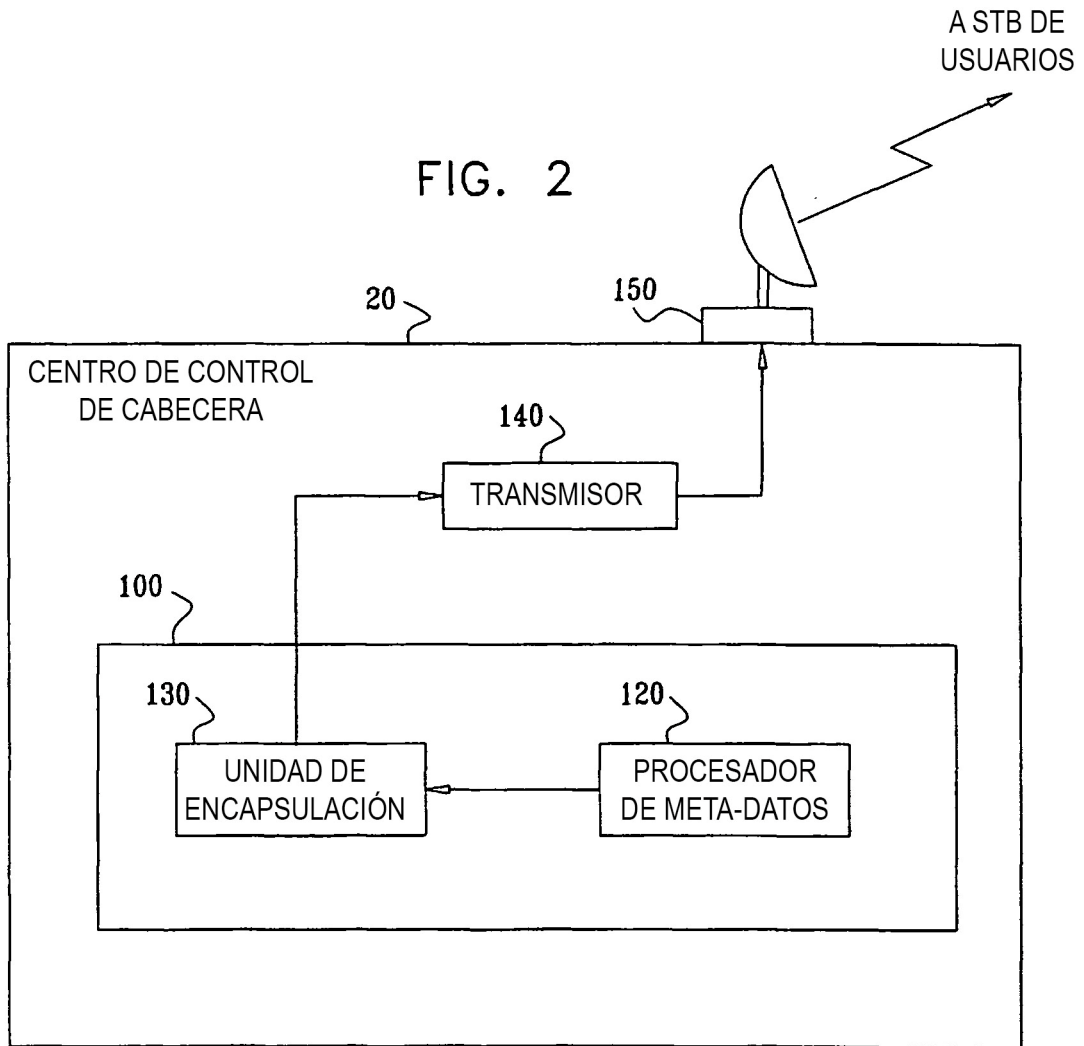


FIG. 3

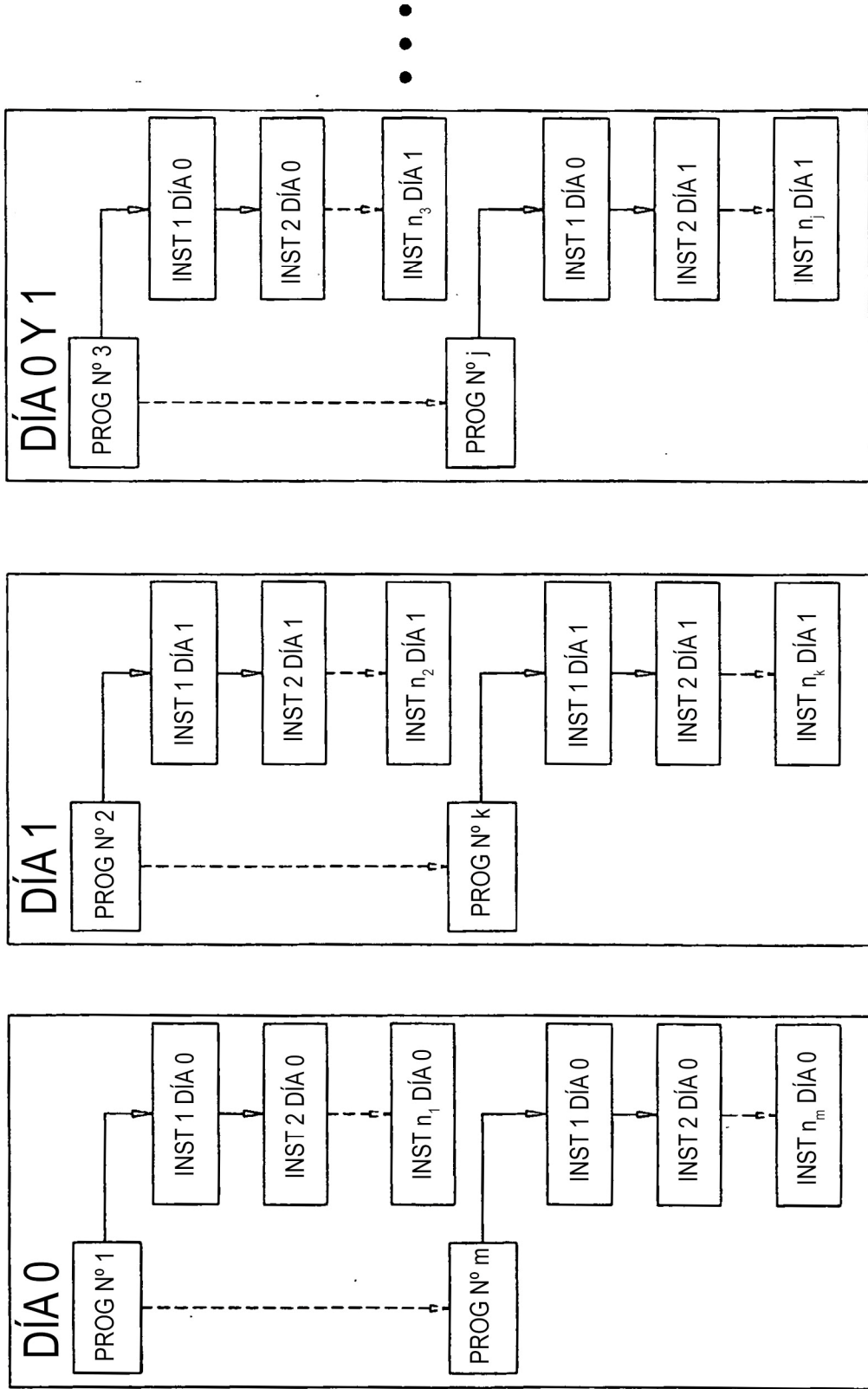


FIG. 4

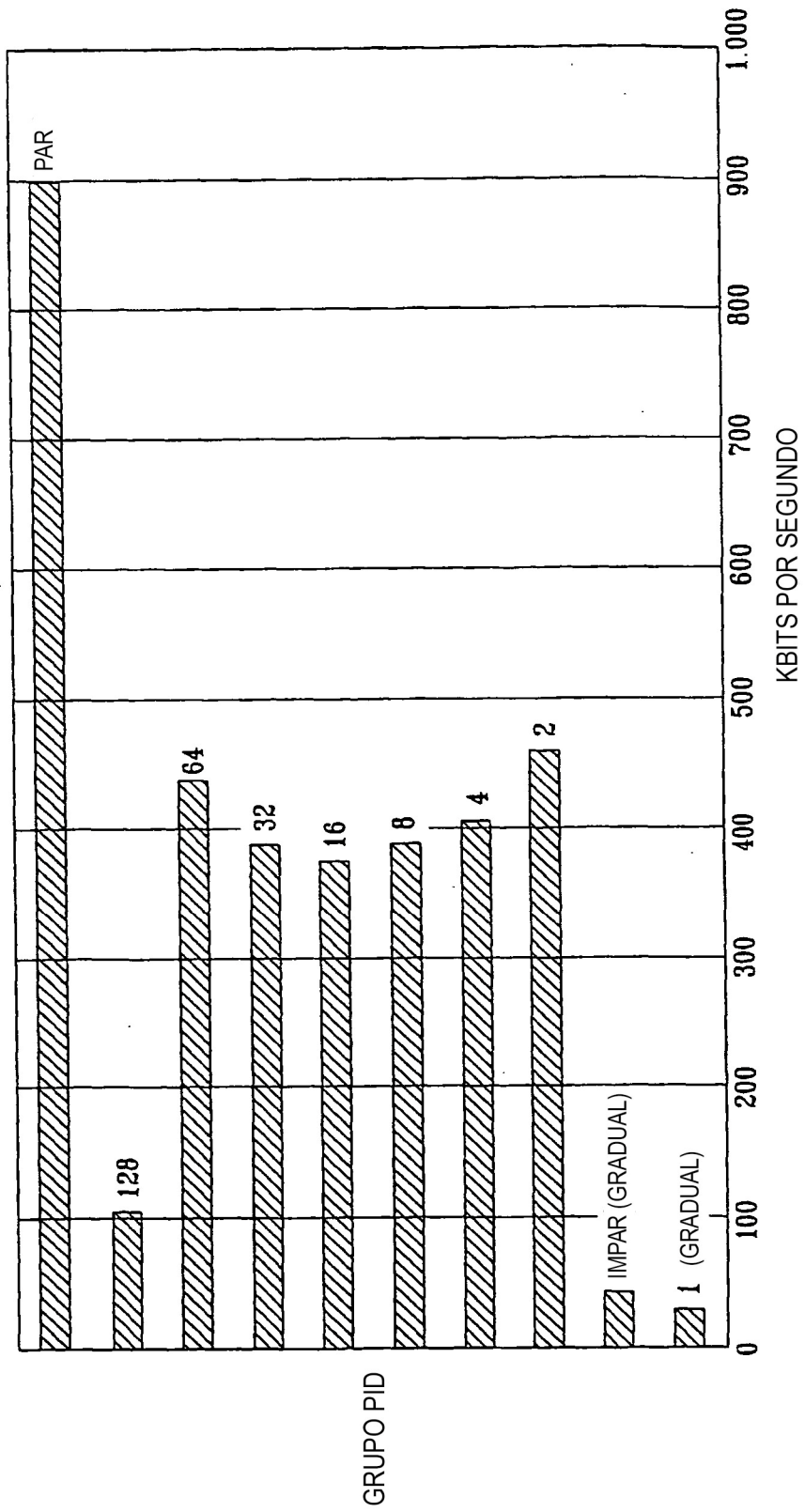


FIG. 5

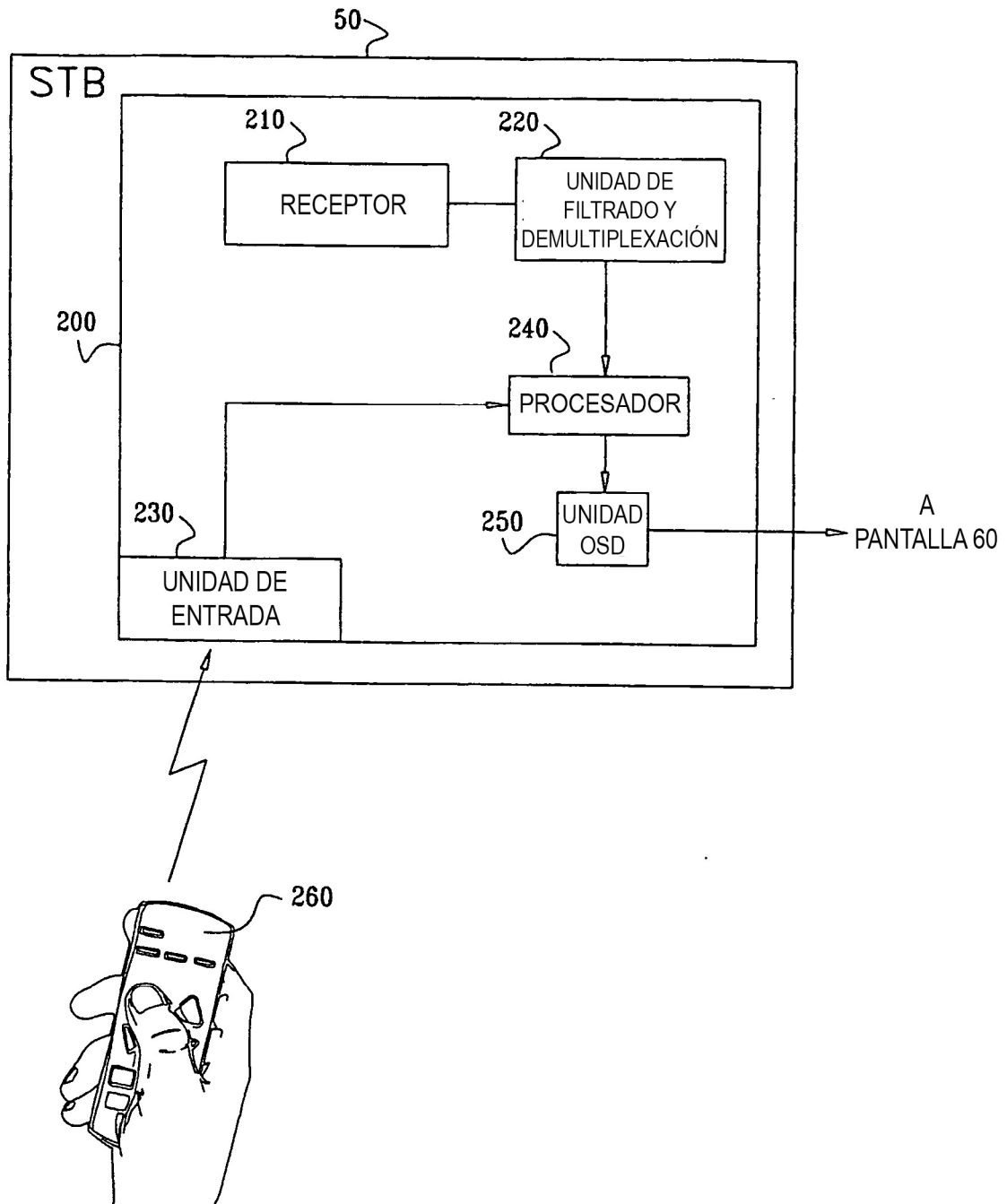




FIG. 6

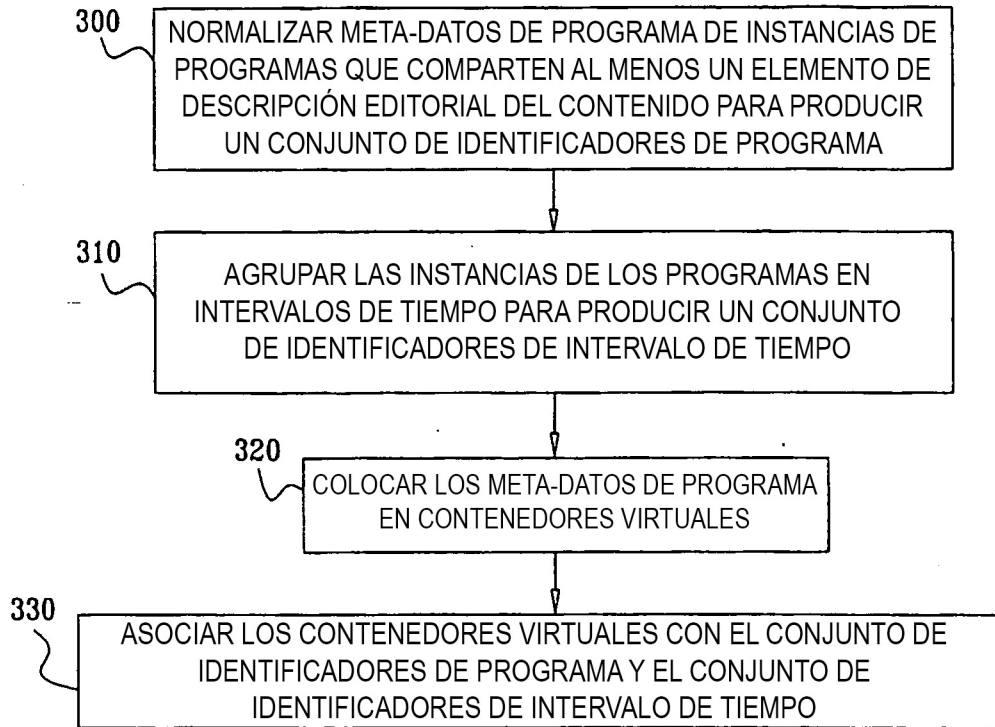


FIG. 7

