

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 650 987**

51 Int. Cl.:

G11B 27/00 (2006.01)

G11B 27/10 (2006.01)

G11B 20/10 (2006.01)

G11B 27/32 (2006.01)

H04N 5/76 (2006.01)

H04N 9/82 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **19.08.2004 PCT/JP2004/012243**

87 Fecha y número de publicación internacional: **09.06.2005 WO05052940**

96 Fecha de presentación y número de la solicitud europea: **19.08.2004 E 04772199 (8)**

97 Fecha y número de publicación de la concesión europea: **18.10.2017 EP 1583098**

54 Título: **Dispositivo de reproducción, método de reproducción, programa de reproducción y medio de grabación**

30 Prioridad:

28.11.2003 JP 2003399546

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

23.01.2018

73 Titular/es:

**SONY CORPORATION (100.0%)
7-35, KITASHINAGAWA 6-CHOME
SHINAGAWA-KU
TOKYO 141-0001, JP**

72 Inventor/es:

**HAMADA, TOSHIYA y
KATO, MOTOKI**

74 Agente/Representante:

LEHMANN NOVO, María Isabel

ES 2 650 987 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

Dispositivo de reproducción, método de reproducción, programa de reproducción y medio de grabación.

5 Campo técnico

La presente invención se refiere a un aparato de reproducción, un método de reproducción, un programa de reproducción y un medio de grabación que permiten al usuario hacer funcionar, de forma interactiva, un programa grabado en un medio de grabación tipo gran capacidad como, por ejemplo, el disco Blu-ray.

Antecedentes de la técnica

10 En los años recientes, se ha propuesto el estándar de disco Blu-ray como un estándar para un medio de grabación tipo disco que es grabable y separable de un aparato de grabación y reproducción. El estándar de disco Blu-ray establece un disco que tiene un medio de grabación con un diámetro de 12 cm y una capa de cobertura que tiene un grosor de 0,1 mm. El estándar de disco Blu-ray usa como sistemas ópticos un láser púrpura azulado que tiene una longitud de onda de 405 nm y una lente del objetivo que tiene una apertura numérica de 0,85. El estándar de disco
15 Blu-ray logra una capacidad de grabación de 27 GB (gigabytes) como máximo. Como resultado, un programa de una transmisión de alta visión digital BS disponible en Japón puede grabarse durante dos horas o más sin deterioro en la calidad de imagen.

20 Como fuentes (fuentes de alimentación) de señales AV (Audio/Vídeo) grabadas en el disco óptico grabable, se espera usar una señal analógica de, por ejemplo, una transmisión de televisión analógica convencional y una señal digital de, por ejemplo, una transmisión de televisión digital como, por ejemplo, una transmisión digital BS. El estándar de disco Blu-ray ha establecido un subestándar para un método de grabación de señales AV de dichas transmisiones.

25 Como un estándar derivado del actual estándar de Disco Blu-ray, existe un movimiento para desarrollar medios de grabación de solo reproducción en los cuales las películas, música y similares se pregrababan. Como medios de grabación con forma de disco en los cuales las películas y música se pregrababan, se ha usado ampliamente el DVD (disco versátil digital). El disco óptico de solo reproducción según el estándar de disco Blu-ray es en gran parte diferente de y superior al DVD convencional en una gran capacidad de grabación y una alta velocidad de transferencia que permiten que las imágenes de alta visión se graben durante dos horas o más en alta calidad.

30 El estándar de vídeo DVD de solo reproducción logra una función interactiva con imágenes de botones y así sucesivamente en una pantalla de menú. Por ejemplo, mientras se reproduce una imagen en movimiento del vídeo DVD, se puede llevar a cabo un proceso que requiere una pantalla de menú con un comandante de mando a distancia o similares, selecciona una imagen de botón en la pantalla de menú y cambia la escena reproducida.

35 En el caso del vídeo DVD, una pantalla de menú consta de datos de subimagen fijos. Cuando se requiere la pantalla de menú, esta se despliega de tal manera que los datos de subimagen se combinan con los datos de la imagen en movimiento. Los datos de la imagen de botón se incluyen en los datos de la subimagen. El documento de la Patente "*Japanese Patent Laid-Open Publication No. HEI 10-308924*" describe una estructura cuyos datos de subimagen se combinan con los datos de la imagen en movimiento y los datos combinados se graban en un DVD grabable.

40 Sin embargo, dado que el estándar de vídeo DVD de solo reproducción establece funciones que usan comandos simples, es difícil lograr funciones interactivas complicadas. Por consiguiente, el creador necesita mucho trabajo para lograr las funciones interactivas complicadas con el estándar de vídeo DVD de solo reproducción.

45 Además, el estándar de vídeo DVD de solo reproducción establece funciones simples de programa con una región de comandos que se ejecutan uno después del otro. Por consiguiente, la programación de múltiples procesos y la programación accionada por evento no se pueden llevar a cabo en el estándar de vídeo DVD de solo reproducción. Aunque el estándar de vídeo DVD no tiene una función que accede a una red como, por ejemplo, Internet, cuando la función de acceso a la red se considera lograda, dado que los tiempos de respuesta de la red y del usuario tienden a ser largos, un proceso de múltiples procesos es muy efectivo como una estructura de sistema para mejorar el rendimiento.

50 Además, en el estándar de vídeo DVD, los comandos se incorporan a un flujo AV (Audio/Vídeo) cuyos datos de vídeo y datos de audio se multiplexan. Mientras el flujo AV se reproduce, los comandos incorporados se extraen de allí y se ejecutan. En otras palabras, se lleva a cabo un modelo de funcionamiento como el rastreo de una viña. Por consiguiente, solo cuando el usuario selecciona un botón y ejecuta un comando correspondiente al botón en una pantalla de menú de selección de título o de múltiples niveles que contiene derivaciones, se decide la siguiente función. Por consiguiente, en el estándar de vídeo DVD, las técnicas de procesos de alta velocidad que preleen comandos para la predicción de derivaciones y ejecución especulativa no pueden usarse.

Además, desde el punto de vista de autoría, que crea discos, el estándar de vídeo DVD de solo reproducción tiene desventajas. Ahora, se supone un botón diseñado para aparecer en un momento predeterminado en un flujo AV y la temporización se mueve hacia adelante o hacia atrás en el eje temporal. Por ejemplo, como se muestra en la Figura 1, los botones designados para aparecer después de la aparición del botón de hora de inicio t1 hasta la aparición del botón de hora de finalización t2 en el flujo AV se cambian de modo que los botones aparecen después de la hora t3 que es anterior a la hora t1.

Como se conoce, en el estándar de vídeo DVD de solo reproducción, un flujo AV se codifica y paquetiza según se establece en el estándar MPEG 2 (Grupo de Expertos de Imágenes en Movimiento 2). La gestión temporal de una salida de reproducción se lleva a cabo con pts (Indicación de Tiempo de Presentación). En el ejemplo que se muestra en la Figura 1, la aparición del botón de hora de inicio se cambia de pts1 correspondiente a la hora t1 a pts3 correspondiente a la hora t3.

En el estándar de vídeo DVD, la información de navegación, datos de vídeo, datos de audio y datos de subimágenes se paquetizan y multiplexan de manera individual como un flujo AV. Los comandos y así sucesivamente se incorporan al flujo AV. Los datos de vídeo, datos de audio y datos de subimágenes se designan para aparecer en sus tiempos multiplexados. Por ejemplo, como se muestra en la Figura 2A, dichos datos se multiplexan de modo que los datos predeterminados se reproducen en sus tiempos predeterminados en el eje temporal administrado en intervalos pts.

Dado que los datos se multiplexan de dicha manera, cuando el botón A designado para aparecer en el tiempo pts1 se cambia de modo que aparece en el tiempo pts3, el flujo AV que se muestra en la Figura 2A debe volver a multiplexarse de modo que los datos de subimágenes del botón 1 aparecen en el tiempo pts3. Cuando se cambian los contenidos de un comando que se ejecuta cuando se selecciona un botón, el flujo AV debe volver a multiplexarse. Cuando un flujo AV se vuelve a multiplexar, necesita reproducirse por un reproductor de DVD y almacenarse temporalmente en una memoria del reproductor.

Por consiguiente, de manera convencional, es difícil lograr funciones interactivas que tengan un grado más alto de libertad. Para lograrlas, el lado de creador necesita mucho trabajo.

El documento WO 2004/049710 describe un dispositivo de reproducción con una interfaz de usuario que tiene un alto grado de libertad para su uso con un medio de grabación en forma de disco de gran capacidad que contiene un contenido. El aparato comprende un reproductor virtual para reproducir datos del disco. Se provee un programa de control de reproducción. El programa de control de reproducción y el reproductor virtual intercambian comandos a través de una API.

El documento US 2003/161615 describe una nueva función de navegación añadida a un DVD-Vídeo heredado. La unidad de reproductor reproduce contenidos grabados que incluyen contenidos de vídeo o contenidos AV, y contenidos ENAV asociados a contenidos (menú, capítulo) de los contenidos de vídeo, de un disco DVD-Vídeo. Los contenidos de vídeo del disco DVD se reproducen por un motor de reproducción de vídeo. Los contenidos ENAV del disco DVD se reproducen por un motor ENAV. El motor ENAV controla la reproducción de los contenidos ENAV y la de los contenidos de vídeo en combinación, conexión y/o sincronismo entre sí según los contenidos reproducidos de los contenidos ENAV. El motor ENAV incluye un controlador de eventos de usuario para recibir eventos de usuario, y un procesador comando/propiedad y generación de evento.

El documento EP-A 1328114 describe un receptor/descodificador que comprende medios para recibir un tren de bits audiovisual en una resolución de entrada dada, y medios para seleccionar una resolución de salida que se aplicará a una representación de salida del tren de bits audiovisual. El receptor incluye una máquina virtual que incluye un manejador de eventos y un módulo de grabadora de vídeo para manejar las funciones de grabación y reproducción de un disco duro.

Descripción de la invención

La invención se define en las reivindicaciones anexas.

Un objeto de la presente invención es proveer un aparato de reproducción, un método de reproducción, un programa de reproducción y un medio de grabación que permiten lograr fácilmente funciones de interfaz con un alto grado de libertad para discos ópticos de solo reproducción con gran capacidad de almacenamiento.

Según se describe en la presente memoria descriptiva, un objeto de reproductor genera un evento correspondiente a un cambio predeterminado para controlar un proceso de reproducción para datos de contenido jerárquicamente grabados en el medio de grabación con forma de disco con un proceso correspondiente al evento descrito en un objeto de programa que se forma por encima del objeto de reproductor y que puede describir un proceso correspondiente al evento que tiene lugar. Por consiguiente, un objeto de reproductor puede llevar a cabo un control de reproducción para un proceso original para cada medio de grabación con forma de disco.

Además, al menos los datos de contenido y un objeto de programa que puede describir un proceso correspondiente a un evento que tiene lugar pueden grabarse en el medio de grabación. Un proceso de reproducción para los datos

de contenido se controla con un proceso correspondiente al evento descrito en el objeto de programa por un objeto de reproductor que genera un evento correspondiente a un cambio predeterminado y formado debajo del objeto de programa. Por consiguiente, un objeto de reproductor puede llevar a cabo un control de reproducción para un proceso original para cada medio de grabación.

5 Según se describe en la presente memoria descriptiva, se estructura un reproductor virtual que forma, de manera abstracta, una función de reproducción de flujo AV de un reproductor BD-ROM. Se designan los eventos que tienen lugar en un modelo de evento aplicado al reproductor virtual. Además, se definen los eventos de usuario correspondientes a las entradas de usuario. Por consiguiente, con un modelo de evento, pueden lograrse funciones de vídeo DVD.

10 Además, según se describe en la presente memoria descriptiva, se estructura un reproductor virtual que forma, de manera abstracta, una función de reproducción de flujo AV del reproductor BD-ROM. Se define una clase Java correspondiente al reproductor virtual. Se designan los eventos que tienen lugar en un modelo de evento aplicado al reproductor virtual. Además, se definen los eventos de usuario correspondientes a las entradas de usuario. Por consiguiente, como un método de control de reproducción para un disco de vídeo de solo reproducción, pueden aplicarse los modelos de evento de Java. Como resultado, con un programa Java, puede controlarse la reproducción de una lista de reproducción.

Por consiguiente, para implementar funciones interactivas complicadas, el lado de creador de disco no necesita mucho trabajo.

20 Además, el enfoque actual puede aplicarse no solo al lenguaje Java, sino a un lenguaje de guiones como, por ejemplo, el guión ECMA. El lado de creador de contenido puede describir contenidos para manejadores de eventos correspondientes a eventos que usan un lenguaje de guiones. Como resultado, cuando tiene lugar un evento, una función que el lado de creador de contenido desea puede lograrse fácilmente.

Breve descripción de los dibujos

25 La Figura 1 es un diagrama esquemático que describe un cambio de una aparición de hora de inicio de un botón según la técnica relacionada; la Figura 2A y Figura 2B son diagramas esquemáticos que describen la remultiplicación de un flujo; la Figura 3A y Figura 3B son diagramas esquemáticos que muestran un ejemplo de una estructura de capa de un modo de película HD del estándar BD-ROM; la Figura 4 es un diagrama esquemático que muestra un ejemplo de una estructura de capa de un reproductor BD-ROM; la Figura 5 es un diagrama esquemático que describe un reproductor básico BD 30; la Figura 6 es un diagrama esquemático que muestra ejemplos de eventos que tienen lugar mientras se reproduce una lista de reproducción; la Figura 7 es un diagrama esquemático que muestra una lista de ejemplos de métodos de clase BDBasicPlayer; la Figura 8A, Figura 8B, y Figura 8C son diagramas esquemáticos que muestran una lista de ejemplos de métodos de clase BDBasicPlayer; la Figura 9 es un diagrama esquemático que muestra una lista de ejemplos de métodos de clase BDBasicPlayer; la Figura 10 es un diagrama esquemático que muestra una lista de ejemplos de métodos de clase BDBasicPlayer; la Figura 11 es un diagrama esquemático que muestra una lista de ejemplos de métodos de clase BDBasicPlayer; la Figura 12 es un diagrama esquemático que muestra una lista de ejemplos de métodos de clase BDBasicPlayer; la Figura 13 es un diagrama esquemático que muestra una lista de ejemplos de métodos de clase BDBasicPlayer; la Figura 14A, Figura 14B, y Figura 14C son diagramas esquemáticos que muestran una lista de ejemplos de definiciones de eventos que tienen lugar en el objeto BDBasicPlayer del D-ROM; la Figura 15 es un diagrama esquemático que describe un evento y un oyente; la Figura 16A y Figura 16B son diagramas esquemáticos que muestran una lista de ejemplos de métodos con respecto al registro de oyentes de eventos; la Figura 17 es un diagrama de flujo que muestra un ejemplo de un proceso que crea un programa que usa el registro de oyentes de eventos; la Figura 18 es un diagrama de flujo que muestra un ejemplo de un proceso cuyo lado de reproducción implementa un programa que usa el registro de oyentes de eventos; la Figura 19A, Figura 19B, y Figura 19C son diagramas esquemáticos que muestran un archivo de flujo AV clip, un archivo que almacena datos de marca, y un archivo que almacena datos de un botón, respectivamente; la Figura 20 es un diagrama esquemático que describe un modelo de reproductor virtual BD-ROM basado en un modelo accionado por evento; la Figura 21A, Figura 21B, Figura 21C, y Figura 21D son diagramas esquemáticos que muestran una lista de ejemplos de definiciones de comandos de entrada de usuario que el modelo de reproductor virtual BD-ROM acepta; la Figura 22 es un diagrama esquemático que muestra un ejemplo de clase RmtKeyEvent de eventos de entrada de tecla de un comandante de mando a distancia; la Figura 23 es un diagrama esquemático que muestra una lista de ejemplos de métodos de objeto RmtKeyEvent; la Figura 24 es un diagrama esquemático que muestra una lista de ejemplos de teclas obtenidas con el método getRmtKeyCode; la Figura 25A y Figura 25B son diagramas esquemáticos que muestran una lista de ejemplos de teclas obtenidas con el método getRmtKeyCode; la Figura 26 es un diagrama de flujo que muestra un ejemplo de un proceso que ejecuta un programa accionado por un evento de entrada de tecla; la Figura 27 es un diagrama esquemático que muestra un ejemplo de una estructura de gestión de archivos aplicable a un perfil completo del BD-ROM; la Figura 28A, Figura 28B, y Figura 28C son diagramas de bloques funcionales que muestran un ejemplo de una estructura de un descodificador de reproductor 100 según una primera realización de la presente invención; la Figura 29 es un diagrama esquemático que muestra una estructura de capa del estándar de vídeo UMD; la Figura 30 es un diagrama esquemático que muestra un ejemplo de un modelo de reproductor según una segunda realización de la presente invención; la Figura 31 es un diagrama esquemático que muestra un ejemplo

de una estructura interna de un reproductor de películas; la Figura 32 es un diagrama esquemático que muestra tres estados del reproductor de películas; la Figura 33 es un diagrama esquemático que muestra un modelo de evento del reproductor de películas según la segunda realización de la presente invención; y la Figura 34A y Figura 34B son diagramas esquemáticos que muestran una lista de ejemplos de manejadores de eventos de un objeto de reproductor de películas.

5

Mejores modos de llevar a cabo la invención

A continuación, se describirá la presente invención correspondiente a la siguiente estructura.

1. Primera realización de la presente invención

1-1. Sobre las funciones de reproducción de flujo AV de BD-ROM

10 1-1a. Sobre el modo de película HD de BD-ROM

1-1b. Sobre el perfil completo de BD-ROM

1-2. Modelo de reproductor virtual BD basado en Java

1-3. Sobre eventos generados por el reproductor básico BD

1-3a. Categorías de eventos

15 1-3b. Sobre modelos de evento

1-4. Sobre entradas de usuario

1-5. Sobre la estructura de gestión de archivos

1-6. Sobre el modelo de descodificador

2. Segunda realización de la presente invención

20 2-1. Sobre el estándar de vídeo UMD

2-2. Sobre el modelo de reproductor basado en el estándar de vídeo UMD

2-3. Sobre el modelo de evento del reproductor de películas

2-4. Sobre el objeto de reproductor de películas y entrada de usuario

2-5. Sobre el manejo de eventos por el objeto de reproductor de películas

25 1. Primera realización de la presente invención

A continuación, se describirá una primera realización de la presente invención. Para una fácil comprensión de la presente invención, un estándar para un medio de grabación de solo reproducción se describirá brevemente como un derivado del estándar de disco Blu-ray actual. En la siguiente descripción, el medio de grabación de solo reproducción como un derivado del estándar de disco Blu-ray se llama BD-ROM (Disco Blu-ray con Memoria de Solo Lectura) para distinguirlo de un disco Blu-ray grabable. El estándar BD-ROM establece el modo de película HD (Alta Definición) que provee funciones interactivas similares a las del vídeo DVD.

30

1-1. Sobre las funciones del flujo AV de BD-ROM

1-1a. Sobre el modo de película HD de BD-ROM

La Figura 3A y Figura 3B muestran un ejemplo de una estructura de capa del modo de película HD del estándar BD-ROM. El modo de película HD del estándar BD-ROM consta de una capa de clip, un reproductor de lista de reproducción, una capa de objeto de película, y una capa de tabla de índice formadas en sucesión a partir de la capa más baja.

35

La capa de clip consta de uno o múltiples clips. Un clip se compone de un flujo AV clip e información de clip. La entidad de un flujo AV es un archivo de flujo AV clip. Un flujo AV se graba en un flujo de transporte MPEG (Grupo de Expertos de Imágenes en Movimiento 2) con una indicación de tiempo. Un archivo de flujo AV clip puede multiplexarse con objetos de botón para las imágenes de botones.

40

Cuando se graba un archivo de flujo AV clip, un archivo de información de clip correspondiente a aquel se crea en la relación de 1 a 1. Se hace referencia a un conjunto de un archivo de flujo AV clip y un archivo de información de clip como un clip.

Un clip es una unidad de grabación. El orden de reproducción de los clips se controla en una capa superior de clips. La capa de lista de reproducción es una capa en la cual se designa el trayecto de reproducción de clips. La capa de lista de reproducción contiene una o múltiples listas de reproducción. En el modo de película HD del estándar BD-ROM, solo existe un tipo de una lista de reproducción a la que se hace referencia como una lista de reproducción de películas. Una lista de reproducción de películas consta de un conjunto de elementos de reproducción. Un elemento de reproducción contiene un conjunto de un punto de ENTRADA (IN) y un punto de SALIDA (OUT) que representan el rango de reproducción de un clip. Con múltiples elementos de reproducción, los clips pueden reproducirse en cualquier orden. Los elementos de reproducción pueden designar el mismo clip. El punto de ENTRADA y el punto de SALIDA de un archivo de flujo AV clip se designan con posiciones de bytes.

El objeto de película existe por encima de la capa de lista de reproducción. La capa de objeto de película consta de uno o múltiples objetos de película. Un objeto de película consta de un comando de reproducción de lista de reproducción y un conjunto de comandos de establecimiento de reproductor. Con los comandos contenidos en un objeto de película, puede lograrse la reproducción de una lista de reproducción que reproduce un flujo en un lenguaje seleccionado o que tiene derivaciones condicionales que hacen que las listas de reproducción se seleccionen para variar según condiciones predeterminadas. Un ejemplo de una aplicación que lleva a cabo una reproducción de una lista de reproducción que tiene derivaciones condicionales sería una de varios niveles. Por consiguiente, los objetos de película proveen interactividad.

La capa de tabla de índice contiene una tabla de índice y existe por encima de la capa de lista de reproducción. La tabla de índice consta de un objeto de película que se ejecuta primero cuando el disco se carga en el reproductor, un objeto de película que se ejecuta cuando se presiona la tecla de menú, y una lista de títulos que el usuario puede ver. La tabla de índice provee una interfaz de usuario. Un título consta de uno o múltiples objetos de película. La tabla de índice enumera los objetos de película que se ejecutan primero en los títulos. El reproductor decide un objeto de película que ejecuta primero en cada función según la tabla de índice.

Formada en el disco en el modo de película HD de BD-ROM se encuentra la base de datos anterior. El reproductor lee la base de datos del disco y lleva a cabo la función designada.

A continuación, se describirá un modelo del lado de reproductor. En realidad, pueden considerarse varios modelos de implementación de reproductores. En el presente ejemplo, un modelo de reproductor se divide en un bloque funcional tipo hardware que reproduce un flujo y un bloque funcional de software que interpreta y ejecuta comandos y que existe por encima del bloque funcional tipo software. La Figura 4 muestra un ejemplo de una estructura de capa de un modelo de reproductor del BD-ROM. La porción izquierda de la Figura 4 muestra la estructura de capa, la cual se divide en dos bloques funcionales en el modo de película HD.

Un motor de presentación BD 200 es un bloque funcional que maneja listas de reproducción y clips, descodifica un flujo AV, y reproduce una imagen en movimiento y datos de sonido. El motor de presentación BD 200 también descodifica un flujo de subtítulo y un flujo de gráficos para imágenes de botones y menús.

El motor de presentación BD 200 reproduce una lista de reproducción designada correspondiente a un comando recibido de la capa superior.

En el modo de película HD del BD-ROM, un motor de navegación 201 existe por encima del motor de presentación BD 200. El motor de navegación 201 interpreta una tabla de índice y objetos de película e informa al motor de presentación BD 200 sobre una lista de reproducción que finalmente se reproduce.

1-1b. Sobre el perfil completo de BD-ROM

A continuación, se describirá un perfil completo del estándar BD-ROM. El perfil completo del estándar BD-ROM es un modo que tiene un grado más alto de libertad y expansibilidad que el modo de película HD del estándar BD-ROM. De aquí en adelante, se hace referencia al modo de perfil completo del estándar BD-ROM como el perfil completo.

Según la primera realización de la presente invención, el modelo de reproductor se provee con una máquina virtual Java (Java VM, por sus siglas en inglés) y ejecuta una aplicación Java para lograr el perfil completo para un grado más alto de libertad y expansibilidad. Este punto es en gran parte diferente del modo de película HD del BD-ROM. Java, JavaVM, y así sucesivamente son marcas comerciales registradas de Sun Microsystems, Inc.

El perfil completo necesita usar todas las funciones del reproductor. Una API (Interfaz de Programación de Aplicaciones) se define entre el motor de presentación BD 200 y una aplicación Java. Por consiguiente, el motor de presentación BD 200 puede funcionar a través de la API. Como el motor de navegación 201 en el modo de película HD del BD-ROM, puede pensarse que la aplicación Java existe por encima del motor de presentación BD 200. En otras palabras, como se ilustra en la porción derecha de la Figura 4, en el perfil completo, una aplicación Java 203 existe por encima del motor de presentación BD 200 a través de una Java VM 202.

Además, Java también tiene una función de red y una función de gráficos. Por consiguiente, existen una capa de red 204 y una capa GUI (Interfaz Gráfica de Usuario) 205 como capas inferiores de la aplicación Java 203 y la Java VM 202.

5 Cuando el motor de presentación BD 200 funciona con la aplicación Java 203, puede pensarse que el reproductor que reproduce una lista de reproducción es un reproductor virtual en el software, los comandos para la reproducción, parada, selección de flujo, y así sucesivamente se envían al reproductor, y el reproductor funciona con los comandos. El reproductor necesita informar al reproductor del estado actual del reproductor en cuanto a cuál lista de reproducción reproduce el reproductor y en qué tiempos varias funciones como, por ejemplo, selecciones de flujo y entradas de usuario, cambiarán. Según la primera realización de la presente invención, dichas funciones del reproductor se consideran objetos abstractivos en la aplicación Java 203. Se hace referencia a los objetos abstractivos de las funciones del reproductor como el reproductor básico BD.

10 A continuación, con referencia a la Figura 5, se describirá un reproductor básico BD. Un reproductor básico BD 30 reproduce datos de un medio de grabación con forma de disco definido en la primera realización de la presente invención. El reproductor básico BD 30 es un objeto como, por ejemplo, un ordenador personal en el entorno de ordenador. El entorno de ordenador no se limita a un ordenador personal de propósito general. Por ejemplo, el entorno de ordenador incluye un entorno de software implementado en un aparato de reproducción dedicado y/o un aparato de grabación y reproducción que reproduce datos de un medio de grabación con forma de disco definido en la primera realización de la presente invención. De aquí en adelante, el medio de grabación con forma de disco según la primera realización de la presente invención se abrevia a disco.

15 El reproductor básico BD 30 es un objeto abstractivo en el software. Debajo del reproductor básico BD 30 existen un SO (Sistema Operativo) y hardware 33. Como un programa superordinado del reproductor básico BD 30, existe un programa Java 31 a través de una API 32.

20 El reproductor básico BD 30 tiene dos estados A y B. En el estado A, puede reproducirse una lista de reproducción y gráficos. En el estado B, su reproducción se detiene. El estado A incluye múltiples funciones como, por ejemplo, reproducciones de velocidades variables, por ejemplo, reproducción de alta velocidad y reproducción en reversa y saltar la reproducción que reproduce datos de cualquier punto temporal en el disco. Además, el reproductor básico BD 30 contiene variables que mantienen estados del reproductor (parámetros comunes 34). Los parámetros comunes 34 constan de registros y así sucesivamente en el entorno de ordenador. Cuando una aplicación (programa Java 31) en la capa superior accede a los parámetros comunes 34 a través de un método de Java, la aplicación puede llevar a cabo un proceso correspondiente al estado del reproductor básico BD 30.

25 El reproductor básico BD 30 es un objeto abstractivo en el software. El reproductor básico BD 30 finalmente convierte el control por el programa Java 31 a través de la API 32 por el control por el SO y el hardware 33 que existen en el nivel inferior.

30 El reproductor básico BD 30 necesita controlarse por un programa superordinado (programa Java 31) a través de la API 32. Además, el reproductor básico BD 30 necesita informar al programa superordinado que el estado del reproductor básico BD 30 cambia. En otras palabras, cuando el reproductor básico BD 30 lleva a cabo múltiples funciones que sucesivamente reproducen una lista de reproducción y luego otra lista de reproducción, el programa superordinado necesita haber conocido el estado del reproductor básico BD 30. Según la primera realización de la presente invención, como un mecanismo que informa a la capa superior el estado del reproductor, se usan modelos de evento.

35 La Figura 6 muestra ejemplos de eventos que pueden tener lugar mientras se reproduce una lista de reproducción. Cuando se inicia la reproducción, tiene lugar el evento PlayStarted. Luego, cuando se inicia la reproducción de una lista de reproducción, tiene lugar el evento PlaylistStarted. Dado que el comienzo de una lista de reproducción también es el comienzo de un elemento de reproducción, cuando el evento PlaylistStarted tiene lugar, el evento PlayItemStarted también tiene lugar. Si una marca de capítulo que presenta un capítulo está presente en el comienzo de la lista de reproducción, el evento MarkEncountered tiene lugar.

40 Asimismo, si una marca está presente en un elemento de reproducción, mientras se reproduce un elemento de reproducción, se detecta la marca. Por consiguiente, el evento MarkEncountered tiene lugar. Cuando se ha completado la reproducción del elemento de reproducción, el evento PlayItemEnded tiene lugar. Cuando también se ha completado la reproducción de la lista de reproducción, el evento PlaylistEnded tiene lugar. Cuando se ha completado la reproducción, el evento PlayEnded tiene lugar.

45 Por consiguiente, mientras el reproductor está funcionando, varios eventos tienen lugar. Cuando se informa al programa superordinado sobre la ocurrencia de un evento, el programa superordinado puede conocer el estado del reproductor. El lado de programa superordinado prepara programas que se ejecutan cuando se lo informa de la ocurrencia de eventos, y puede lidiar con la ocurrencia de los eventos.

50 Según la primera realización de la presente invención, se proveen un método de abstracción para un reproductor, modelos de método que permiten a un programa Java hacer funcionar el reproductor, y modelos de evento que informan al programa Java sobre el estado del reproductor. El lenguaje Java se implementa a un programa de control del BD-ROM. En el sistema según la primera realización de la presente invención, el creador de disco crea el programa Java 31 que hace funcionar el reproductor básico BD 30 a través de la API 32.

A continuación, se describirá un ejemplo práctico donde las funciones del reproductor se analizan en detalle, se abstraen y se representan como clases de Java.

1-2. Modelo de reproductor virtual BD basado en Java

5 Para designar el perfil completo del BD-ROM, es necesario definir un entorno de ejecución para el disco Blu-ray. Un modelo ideal del reproductor básico BD 30 que reproduce datos del BD-ROM se llama el modelo de reproductor virtual BD-ROM. El reproductor básico BD 30 corresponde a un bloque funcional que reproduce una lista de reproducción del modelo de reproductor virtual BD-ROM. En el modelo de reproductor virtual BD-ROM, se definen los estados y las condiciones para los cambios de estado. Además, se definen los atributos de flujos reproducidos por el reproductor básico BD 30, un método de descripción sin comandos (guión) que controla los flujos, y así sucesivamente. Los guiones y flujos grabados en el BD-ROM deben funcionar correctamente en el modelo de reproductor virtual BD-ROM.

Aunque un reproductor BD-ROM real puede instalarse de varias maneras, cuando se considera un modelo que no necesita instalarse, los recursos de hardware mínimamente requeridos pueden aclararse.

15 El modelo de reproductor virtual BD-ROM definido en la primera realización de la presente invención usa Java como un lenguaje de programación que controla el reproductor básico BD 30. Java es un lenguaje de programación creado por Sun Microsystems, Inc. y tiene múltiples ediciones como, por ejemplo, J2EE, J2SE, J2ME, y así sucesivamente que difieren en sus escalas de sistemas aplicables. En el presente ejemplo, se supone que J2ME, que es una edición diseñada para instalarse en un decodificador de salón y así sucesivamente, se usa para el modelo de reproductor virtual BD-ROM. En otras palabras, el modelo de reproductor virtual BD-ROM se extiende sobre la base de J2ME.

20 Como una característica del lenguaje Java, cuando se define una nueva clase para un programa, este puede extenderse fácilmente. Supongamos que un documento HTML (Lenguaje de Marcado de Hipertexto) no admite un proceso de evento. Para permitir que dicho documento HTML admita el proceso de evento, un guión para el proceso de evento, por ejemplo un guión ECMA (Asociación Europea de Fabricantes de Ordenadores) se describe en el documento HTML.

25 Cuando se usa el lenguaje Java, puede añadirse una función correspondiente a un método de extensión definido en el lenguaje Java sin usar múltiples técnicas que logran la función. Por consiguiente, un programa completado se describe en el lenguaje Java solamente. Además, Java ya ha provisto varios entornos de desarrollo de programa. Por consiguiente, puede decirse que el creador de disco puede crear fácilmente contenidos de disco.

30 Cuando el modelo de reproductor virtual BD-ROM se describe en el lenguaje Java, API, una clase que ha definido Sun Microsystems, Inc., puede usarse como está. Por otro lado, el reproductor básico BD 30 es una función original del BD-ROM. Por consiguiente, Java no provee el reproductor básico BD 30 como una clase y API existentes. Cuando dicha función original del BD-ROM se logra por Java, puede haber varios métodos. Sin embargo, según la primera realización de la presente invención, las funciones del reproductor básico BD 30, API, modelos de evento, y así sucesivamente se crean según el BD-ROM.

35 En primer lugar, el reproductor básico BD 30 se define como una clase de Java. El nombre de la clase se define como BDBasicPlayer. La Figura 7 a la Figura 13 muestra listas de ejemplos de métodos de clase BDBasicPlayer. Todos los comandos para el reproductor básico BD 30 se llevan a cabo con dichos métodos.

40 La Figura 7 muestra ejemplos de métodos usados para obtener información de establecimiento de reproductor. El método getPlayerSpeed() obtiene la velocidad de reproducción del reproductor. El método getPlayerDirection() obtiene la dirección de reproducción del reproductor. El método getDescriptionLanguage() obtiene el idioma de visualización del reproductor. El método getPlayListNumber() obtiene el número de lista de reproducción que se está reproduciendo actualmente. El método getChapterNumber() obtiene el número de capítulo que se está reproduciendo actualmente. El método getPlayerSupport() obtiene la versión y funciones del reproductor.

45 La Figura 8A, Figura 8B y Figura 8C muestran ejemplos de métodos usados para reproducir listas de reproducción. El método playPlayList(int playListNumber, long[playListTime]) inicia la reproducción de una lista de reproducción designada por "playListNumber". "playListNumber" no es esencial. "playListTime" designa un tiempo transcurrido de la lista de reproducción. La lista de reproducción se reproduce desde el tiempo transcurrido "playListTime". El método playPlayList(String playListURL, long[PlayListTime]) inicia la reproducción de una lista de reproducción designada por "playListURL". "playListURL" es una lista de reproducción designada por un URL (Localizador Uniforme de Recursos). Una lista de reproducción en Internet puede designarse por un URL. Cuando el reproductor tiene una función de conexión a Internet, una lista de reproducción puede descargarse de un URL designado por "playListURL" y reproducirse. "playListTime" no es esencial. "playListTime" designa un tiempo transcurrido de la lista de reproducción. La lista de reproducción se reproduce desde el tiempo transcurrido "playListTime".

55 El método playPlayItem(int playListNumber, int playItemNumber) inicia la reproducción de una lista de reproducción designada por "playListNumber" desde un elemento de reproducción designado por "playItemNumber". "playItemNumber" es "PlayItem_id" que identifica un elemento de reproducción y que es un valor numérico que

comienza en 0. Cuando la reproducción se inicia desde el comienzo de una lista de reproducción, el valor de "playItemNumber" es 0. El método playPlayItem (String playListURL, int playItemNumber) inicia la reproducción de una lista de reproducción designada por "playListURL" desde un elemento de reproducción designado por "playItemNumber". "playItemNumber" es "PlayItem_id" que es un valor numérico que comienza en 0. Cuando la reproducción se inicia desde el comienzo de una lista de reproducción, el valor de "playItemList" es 0.

El método playChapterMark(int playListNumber, int chapterNumber) reproduce una lista de reproducción designada por "playListNumber" desde un capítulo designado por "chapterNumber". "playListNumber" puede ser "playListURL". Una lista de reproducción siempre comienza con ChapterMark (marca de capítulo).

El método moveAndPlay(string position, object) posición = ("prev"|"next"|"top"|"tail") objeto = ("PlayItem"|"Chapter") designa un salto en una lista de reproducción. "next" designa un salto de la posición actualmente reproducida al siguiente elemento de reproducción o capítulo y comienza su reproducción. "top" designa un salto de la posición actualmente reproducida al final de la lista de reproducción. "prev" inicia la reproducción desde el comienzo de la lista de reproducción actualmente reproducida. El método stop() detiene la reproducción. En este punto, el valor de la resistencia estándar no se mantiene. El método playSoundEffect(int sound_id) reproduce un efecto de sonido seleccionado por "sound_id".

La Figura 9 muestra ejemplos de métodos usados para flujos de vídeo. El método getVideoStreamAvailability() obtiene un estado que representa si se contiene o no un flujo de vídeo designado. El método setVideoStreamNumber() designa un flujo de vídeo que se descodificará. El método getVideoStreamNumber() obtiene un número de flujo de vídeo que se ha seleccionado. El método getVideoStreamAttribute() obtiene atributos de un flujo de vídeo. Los atributos de un flujo de vídeo son, por ejemplo, sistema de codificación, resolución, relación de aspecto, modo de visualización en el caso de una relación de aspecto de 4 : 3, subtítulo codificado, y así sucesivamente. El método setAngleNumber() designa un número de ángulo. El método getAngleNumber() obtiene un número de ángulo que se ha seleccionado. El método getMaxVideoStreams() obtiene el número de flujos de vídeo que pueden seleccionarse.

La Figura 10 muestra ejemplos de métodos usados para flujos de audio. El método getAudioStreamAvailability() obtiene un estado que representa si se contiene o no un flujo de audio designado. El método getAudioStreamLanguage() obtiene información sobre el lenguaje usado en un flujo de audio designado. El método setAudioStreamNumber() designa un flujo de audio que se reproducirá. El método getAudioStreamNumber() obtiene un número de flujo de audio que se está reproduciendo. El método getAudioStreamAttribute() obtiene atributos de un flujo de audio. Los atributos de un flujo de audio son, por ejemplo, sistema de codificación, número de canales, número de bits de cuantificador, frecuencia de muestreo, y así sucesivamente. El método getMaxAudioStreams() obtiene el número de flujos de audio que pueden obtenerse.

La Figura 11 muestra ejemplos de métodos usados para flujos de subimágenes. El método getSPStreamAvailability() obtiene un estado que representa si se contiene o no un flujo de subimágenes designado. El método getSPStreamLanguage() obtiene el lenguaje usado en un flujo de subimágenes designado. El método getSPDisplayStatus() obtiene un estado de visualización de un flujo de subimágenes, a saber si se está visualizando o no actualmente. El método setSPDisplayStatus() establece un estado de visualización de un flujo de subimágenes, a saber si mostrarlo o no. El método getSPStreamAttribute() obtiene atributos de un flujo de subimágenes. Los atributos de un flujo de subimágenes son, por ejemplo, información sobre las resoluciones de los flujos de subimágenes, información sobre una relación de aspecto de 4 : 3 o tipo de ancho, y así sucesivamente.

La Figura 12 muestra ejemplos de métodos usados para temporizadores internos. El método sleep() detiene un proceso durante milisegundos designados. El método setTimeout() ejecuta una función o un proceso después de que transcurran los milisegundos designados. El método setInterval() ejecuta un proceso a intervalos de milisegundos predeterminados. El método clearTimer() detiene un proceso de un ID de temporizador de registro designado. El método pauseTimer() pausa un temporizador que tiene un ID de temporizador de registro designado. El método resumeTimer() reanuda un temporizador pausado que tiene un ID de temporizador de registro designado.

La Figura 13 muestra un ejemplo de un método usado para una entrada de tecla. El método getPressedKey() obtiene el tipo de una tecla que el usuario ha presionado.

1-3. Sobre eventos generados por el reproductor básico BD

Cuando se ejecuta un método para una función de reproductor, si la reacción de la función del reproductor se obtiene, por ejemplo, con respecto a si el funcionamiento del reproductor se ha iniciado de forma normal, puede usarse un valor de retorno del método. Sin embargo, si se designa un proceso que requiere un largo tiempo y es necesario esperar hasta que el proceso esté completo, no puede llevarse a cabo otro proceso. Además, existen procesos cuyas horas de inicio son impredecibles. En dichos casos, una técnica de eventos puede usarse para llevar a cabo los procesos de manera efectiva.

Mientras el reproductor está funcionando, varios eventos tienen lugar. Los eventos que tienen lugar se definen en el modelo de reproductor virtual BD-ROM. Un modelo en el cual cuando el funcionamiento del reproductor varía, un

evento tiene lugar y cuando existe un programa correspondiente a un evento, el programa se ejecuta se define por el modelo de reproductor virtual BD-ROM.

5 Cuando múltiples listas de reproducción se reproducen en un orden predeterminado se provee un programa que lleva a cabo un proceso que comienza la reproducción de una lista de reproducción después de recibir un evento que representa que la reproducción de la lista de reproducción precedente se ha completado. Además, se provee un programa que lleva a cabo un proceso de excepción cuando la siguiente lista de reproducción no se reproduce debido a cualquier causa.

10 Cuando no se describe dicho evento, una función (un manejador de eventos por defecto) que se construye en el reproductor se ejecuta o el evento se ignora. Cuando no es necesario llevar a cabo un proceso, el evento se ignora de forma activa. En este punto, no se registra un oyente de eventos (que se describirá más adelante). De manera alternativa, cuando no se describe un programa correspondiente al evento, el evento se ignora.

Se hace referencia al modelo en el cual un evento no periódico tiene lugar y un programa correspondiente al evento se ejecuta como el modelo accionado por evento.

1-3a. Categorías de eventos

15 Los eventos del modelo de reproductor virtual BD-ROM pueden clasificarse como tres tipos de (1) eventos que tienen lugar en un contenido que se está reproduciendo, (2) un evento que tiene lugar como una interrupción por el usuario, y (3) eventos que tienen lugar correspondientes a cambios de estado del reproductor.

20 (1) Los eventos que tienen lugar en un contenido que se está reproduciendo son eventos de interrupción de detección de marca que tienen lugar en momentos designados por las marcas. Cuando dichos eventos tienen lugar, los gráficos se visualizan o eliminan. Dichos eventos son interrupciones preestablecidas. Cuando se lleva a cabo la reproducción, dichos eventos tienen lugar en los mismos tiempos.

25 (2) Se supone que un evento que tiene lugar como una interrupción por el usuario es una entrada de tecla en un comandante de mando a distancia con el cual el usuario hace funcionar el reproductor. El tiempo de interrupción de dicho evento es impredecible dado que no está claro en qué momento el usuario llevará a cabo una entrada de tecla.

30 (3) Los eventos que tienen lugar correspondientes a los cambios de estado del reproductor son, por ejemplo, un evento que tiene lugar cuando el estado del reproductor cambia de los inicios de paradas al estado de reproducción y un evento que tiene lugar cuando un flujo que se está reproduciendo ha cambiado a otro flujo. Cuando un temporizador se establece en un programa, un evento de interrupción de temporizador tiene lugar en un tiempo designado. Los cambios de estado del reproductor pueden ser (1) eventos que tienen lugar en un contenido que se está reproduciendo y (2) un evento que tiene lugar como una interrupción por el usuario.

La Figura 14A, Figura 14B, y Figura 14C muestran una lista de ejemplos de definiciones de eventos que tienen lugar en el objeto BDBasicPlayer del BD-ROM. El objeto BDBasicPlayer es un objeto que tiene lugar en la clase BDBasicPlayer.

35 En primer lugar, se describirán (1) los eventos que tienen lugar en un contenido que se está reproduciendo. En la Figura 14A, Figura 14B, y Figura 14C, el evento MarkEncounteredEvent tiene lugar cuando se ha detectado una marca mientras se reproducía un contenido. Dicho evento se usa cuando, por ejemplo, se visualiza una pantalla de gráficos. El evento ValidPeriodStartedEvent tiene lugar cuando ha comenzado un período válido de selección de una entrada de usuario. El evento ValidPeriodEndedEvent tiene lugar cuando ha finalizado un período válido de una entrada de usuario. Dicho evento se usa cuando, por ejemplo, se ejecuta un enlace de manera forzada. El evento PlayRepeatedEvent tiene lugar cuando se detecta el comienzo de una región de reproducción de repetición. El evento VideoStreamChangedEvent tiene lugar cuando se cambia un flujo de vídeo que se estaba reproduciendo. El evento AudioStreamChangedEvent tiene lugar cuando se cambia un flujo de audio que se estaba reproduciendo. El evento SubpictureStreamChangedEvent tiene lugar cuando se cambia un flujo de subimágenes que se estaba reproduciendo. El evento StreamDisplayStatusChangedEvent tiene lugar cuando se cambia el estado de visualización/no visualización de un flujo. El evento PlaylistStartedEvent tiene lugar cuando se detecta el comienzo de una lista de reproducción. El evento PlaylistEndedEvent tiene lugar cuando se detecta el fin de una lista de reproducción. El evento PlayItemStartedEvent tiene lugar cuando se detecta el comienzo de un elemento de reproducción. El evento PlayItemEndedEvent tiene lugar cuando se detecta el fin de un elemento de reproducción.

50 A continuación, se describirá (2) un evento que tiene lugar como una interrupción por el usuario. El evento RmKeyEvent tiene lugar cuando se presiona una tecla del comandante de mando a distancia. Una tecla que se ha presionado se determina con un oyente de eventos correspondiente (que se describirá más adelante).

55 A continuación, se describirán (3) eventos que tienen lugar correspondientes a cambios de estado del reproductor. El evento PlayStartedEvent tiene lugar cuando el reproductor ha iniciado la reproducción. El evento PlayStoppedEvent tiene lugar cuando el reproductor ha detenido la reproducción. El evento PlayStilledEvent tiene lugar cuando el reproductor ha paralizado la reproducción. El evento StillReleaseEvent tiene lugar cuando el

reproductor ha liberado el estado paralizado de la reproducción. El evento `PlayPausedEvent` tiene lugar cuando la reproducción del reproductor se ha pausado por el usuario. El evento `PauseReleaseEvent` tiene lugar cuando el estado pausado se ha liberado. El evento `TimerFiredEvent` tiene lugar cuando un temporizador en cuenta regresiva se convierte en 0 o un temporizador en cuenta ascendente se convierte en un valor predeterminado.

5 1-3b Sobre modelos de evento

Se supone que los eventos descritos en la Figura 14A, Figura 14B, y Figura 14C tienen lugar en el objeto `BDBasicPlayer`. En otras palabras, se supone que una fuente de evento del lenguaje Java es el objeto `BDBasicPlayer`. Los eventos se definen como objetos de evento en el lenguaje Java. Un método que se ejecuta cuando un evento tiene lugar se llama un oyente. Para ejecutar un oyente, es necesario registrar un oyente de eventos en una fuente de eventos. Cuando un oyente se ha registrado en una fuente de eventos, el objeto `BDBasicPlayer` transmite un evento al oyente de evento. El objeto que ha recibido el evento ejecuta un método provisto en el objeto.

A continuación, con referencia a la Figura 15, se describirán oyentes y eventos. Los objetos 211, 211, ... y así sucesivamente tienen una interfaz de oyente que recibe un evento y ejecuta un oyente. El objeto 211 registra un oyente correspondiente a un evento que el objeto 211 desea recibir para el objeto `BDBasicPlayer 210` (etapa E10). El registro de un oyente correspondiente a un evento para el objeto `BDBasicPlayer 210` se llama el registro de oyente de eventos. Los múltiples objetos 211, 211, ..., y así sucesivamente pueden llevar a cabo el registro de oyente de eventos para el objeto `BDBasicPlayer 210`.

Un objeto que implementa una interfaz de oyente es un objeto que implementa un método que se ejecuta cuando un evento tiene lugar. En el ejemplo que se muestra en la Figura 15, el método `doAction()` que se ejecuta cuando un evento predeterminado tiene lugar se implementa para el objeto 211. Cuando un evento tiene lugar en el objeto `BDBasicPlayer 210` (en la etapa E11) y el evento se ha registrado en el objeto `BDBasicPlayer 210` como un oyente de eventos, el evento se transmite al objeto 211 que se ha registrado como un oyente de eventos (en la etapa E12). El objeto 211 que ha recibido el evento ejecuta el método `doAction()` (en la etapa E13).

La Figura 16A y Figura 16B muestran una lista de ejemplos de métodos usados para registrar oyentes de eventos. El método `addContentListener(ContentListener 1)` registra un oyente designado para el objeto `BDBasicPlayer` para recibir el evento `ContentsEvent` del objeto `BDBasicPlayer`. El oyente registrado se añade a una matriz `ContentListeners()`. El evento `ContentsEvent` es un término general de (1) eventos que tienen lugar en un contenido que se está reproduciendo. El método `addRmtKeyListener(RmtKeyListener 1)` registra un oyente designado para recibir el evento `RmtKeyEvent` ((2) un evento que tiene lugar como una interrupción por el usuario, descrito en la Figura 14A, Figura 14B y Figura 14C) del objeto `BDBasicPlayer`. El oyente registrado se añade a la matriz `RmtKeyListeners()`. El método `addPlayerStatusListener(PlayerStatusListener 1)` registra un oyente designado para recibir el evento `PlayerStatusEvent` del objeto `BDBasicPlayer`. El oyente registrado se añade a la matriz `PlayerStatusListeners()`. El evento `PlayerStatusEvent` es un término general de (3) eventos que tienen lugar correspondientes al cambio de estado del reproductor, según se describe en la Figura 14A, Figura 14B y Figura 14C.

El método `removeContentListener(ContentListener 1)` elimina un oyente designado para no recibir el evento `ContentsEvent` del objeto `BDBasicPlayer`. El método `removeRmtKeyListener(RmkKeyListener 1)` elimina un oyente designado para no recibir el evento `RmtKeyEvent` del objeto `BDBasicPlayer`. El método `removePlayerStatusListener(PlayerStatusListener 1)` elimina un oyente designado del objeto `BDBasicPlayer` para no recibir el evento `PlayerStatusEvent` del objeto `BDBasicPlayer`.

Cuando se ejecuta un método que comienza con "add" (añadir), se registra un oyente de eventos. Un oyente de eventos se registra antes de que un evento deseado tenga lugar. Cuando un evento no necesita recibirse, con un método que comienza con "remove" (eliminar), la transmisión del evento puede detenerse.

En la Figura 16A y la Figura 16B, el método `getContentListeners()` devuelve una matriz de todos los oyentes de contenido registrados en el objeto `BDBasicPlayer`. El método `getRmtkeyListeners()` devuelve una matriz de todos los `RmtkeyListeners` registrados en el objeto `BDBasicPlayer`. El método `getPlayerStatusListeners()` devuelve una matriz de todos los `PlayerStatusListeners` registrados en el objeto `BDBasicPlayer`.

A continuación, se describirá un ejemplo de un programa creado mediante el uso del registro de oyente de eventos. En el presente ejemplo, se crea un programa que hace que los botones aparezcan en una escena designada de una lista de reproducción que se está reproduciendo. Para determinar que una escena predeterminada se está reproduciendo en una lista de reproducción, puede usarse una función de marca. La función de marca es una función que puede establecer cualquier tiempo de una lista de reproducción. Cuando una escena marcada está por reproducirse, el evento `MarkEncounteredEvent` descrito en la Figura 14A, Figura 14B y Figura 14C tiene lugar. Un objeto de botón que implementa una interfaz de oyente que hace que un botón aparezca cuando el evento `MarkEncounteredEvent` tiene lugar puede crearse. En otras palabras, un programa que registra un oyente de eventos para el objeto `BDBasicPlayer` y hace que una imagen aparezca cuando el evento `MarkEncounteredEvent` se recibe puede implementarse en el objeto del botón.

De manera más práctica, como se ejemplifica en la Figura 17, se decide en qué escena de una lista de reproducción aparece un botón (en la etapa E20). Una marca se establece en el momento en que el botón aparece en la lista de reproducción (en la etapa E21). A continuación, un método que registra un oyente de eventos correspondiente a la marca que se ha establecido en la etapa E21 se describe en el programa (en la etapa E22). En otras palabras, en la etapa E22, se describe la función llevada a cabo en la etapa E10 que se muestra en la Figura 15. Un proceso que se ejecuta cuando un evento tiene lugar se describe en el objeto de botón. La interfaz de oyente correspondiente al evento de generación de marca se implementa en el objeto de botón (en la etapa E23). En otras palabras, en la etapa E23, se describe el contenido de la etapa E13. Dichos programas se describen en el lenguaje Java.

La Figura 18 muestra un ejemplo de un proceso que implementa una función de la programación que se muestra en la Figura 17 para el lado de reproducción. El objeto BDBasicPlayer comienza la reproducción de una lista de reproducción (en la etapa E30). Se supone que mientras se está reproduciendo una lista de reproducción, el objeto BDBasicPlayer ha detectado una marca en la lista de reproducción (en la etapa E31). El objeto BDBasicPlayer comprueba si un oyente de eventos correspondiente a la marca detectada se ha registrado o no (en la etapa E32). En otras palabras, en la etapa E32, se determina si el proceso en la etapa E10 descrita en la Figura 15 se ha llevado a cabo o no. Si el resultado determinado representa que el oyente de eventos correspondiente no se ha registrado, el flujo avanza a la etapa E36. Por consiguiente, el objeto BDBasicPlayer no transmite un evento.

Por otro lado, cuando el resultado determinado representa que el oyente de eventos correspondiente a la marca se ha registrado, el flujo avanza a la etapa E33. En la etapa E33, el objeto BDBasicPlayer transmite el evento MarkEncounteredEvent a un objeto que ha registrado el oyente de eventos. Dicho proceso es igual al proceso en la etapa E12 que se muestra en la Figura 15. Cuando el objeto ha recibido el evento MarkEncounteredEvent, el objeto ejecuta el método que se ejecutará cuando el evento MarkEncounteredEvent tenga lugar (en la etapa E34). En otras palabras, en la etapa E34, se ejecuta el método doAction() en la etapa E13 que se muestra en la Figura 15. En el presente ejemplo, el método doAction describe un proceso que hace que un botón aparezca en la pantalla. El botón aparece en la pantalla correspondiente a la descripción (en la etapa E35).

Según se describe en la sección de técnica relacionada que se muestra en la Figura 2A y Figura 2B, en el vídeo DVD convencional, dado que la información de los botones se ha multiplexado con un flujo AV, cuando la hora de inicio de la aparición del botón cambia, es necesario volver a multiplexar el flujo AV.

Por el contrario, en el modelo de reproductor virtual BD-ROM según la primera realización de la presente invención, mediante el cambio de la posición de establecimiento de una marca en el eje temporal solamente, la hora de inicio de aparición del botón puede cambiarse. Según la primera realización de la presente invención, como se muestra en la Figura 19A, Figura 19B y Figura 19C, un archivo de flujo AV clip (Figura 19A), un archivo de datos de marca (Figura 19B), y un archivo de datos de botón (Figura 19C) se estructuran de manera independiente (la estructura de archivo se describirá más adelante). El archivo de flujo AV clip es un archivo en el cual un paquete de vídeo y un paquete de audio se multiplexan.

En dicha manera, la información de una marca se almacena en un archivo de base de datos que es diferente de un archivo de flujo AV clip. Por consiguiente, como se muestra en la Figura 19A, Figura 19B y Figura 19C, cuando la hora de inicio de aparición de una imagen de botón cambia del tiempo pts1 al tiempo pts2, es necesario solamente cambiar el tiempo pts1 de la información de marca al tiempo pts2. Por consiguiente, incluso si la posición de una marca se cambia, no es necesario volver a multiplexar el flujo AV clip.

Además, según la primera realización, los datos que componen un botón se almacenan en un archivo que es diferente de un archivo de flujo AV clip. Por consiguiente, como un proceso ejecutado cuando se selecciona un botón, solo el método del objeto de botón independiente del archivo de flujo AV clip necesita cambiarse. Como se describe más arriba, en el vídeo DVD convencional, dado que un comando de botón que lleva a cabo un proceso que se ejecutará cuando el botón se selecciona se multiplexa con un flujo AV, el proceso ejecutado cuando el botón se selecciona no puede lograrse fácilmente.

En el perfil completo del BD-ROM, el anterior modelo de reproductor virtual BD-ROM puede proveer más flexibilidad y una autorización más fácil que el vídeo DVD convencional.

En la realización anterior, el modelo de reproductor virtual BD-ROM se estructura sobre la base de un modelo de oyente de eventos orientado por el objeto. De manera alternativa, un modelo de reproductor virtual BD-ROM puede estructurarse con un modelo accionado por evento más simple que dicho ejemplo.

A continuación, con referencia a la Figura 20, se describirá un modelo de reproductor virtual BD-ROM basado en un modelo accionado por evento. Los eventos que tienen lugar en el objeto BDBasicPlayer son iguales a aquellos en un modelo orientado por objeto que se muestra en la Figura 15. En el caso del presente modelo accionado por evento, se supone que cuando un evento tiene lugar, solo se requiere el método processEvent() definido en el objeto BDBasicPlayer. En otras palabras, en el modelo, no importa qué evento tiene lugar, se requiere el mismo método. Por consiguiente, para cambiar un proceso correspondiente al tipo de un evento que tiene lugar, es necesario determinar qué evento ha tenido lugar en el método processEvent().

En el presente modelo, no es necesario registrar un oyente de eventos correspondiente a cada evento. Sin embargo, cuando un evento tiene lugar, dado que se requiere el método `processEvent()`, la eficiencia de dicho modelo es baja. Además, dado que el método `processEvent()` que se procesará se define de manera fija en el objeto `BDBasicPlayer`, dicho modelo no tiene expansibilidad que permita definir una nueva rutina de proceso en otro objeto y ejecutarla. Por consiguiente, según la primera realización de la presente invención, la cual enfatiza la alta expansibilidad, se prefiere estructurar el modelo de reproductor virtual BD-ROM con un modelo orientado por el objeto descrito en la Figura 15.

1-4. Sobre las entradas de usuario

La Figura 21A, Figura 21B, Figura 21C y Figura 21D muestran una lista de ejemplos de definiciones de comandos de entradas de usuario que el modelo de reproductor virtual BD-ROM puede recibir. Se supone que las entradas de usuario son entradas de teclas del comandante de mando a distancia con el cual el usuario hace funcionar el reproductor y entradas correspondientes a la pantalla de menú del reproductor. En el modelo de reproductor virtual BD-ROM, las entradas de usuario son un tipo de eventos de entrada de tecla.

El comando `timeSearch(playlistNumber, Time)` hace que un escenario que se está reproduciendo se reproduzca desde el tiempo designado por "Time" (Tiempo). El comando `play(palylistNumber, chapterNumber)` hace que un escenario designado se reproduzca desde un capítulo designado por "chapterNumber". El comando `stop()` hace que la reproducción se detenga. El comando `prev()` hace que la reproducción comience desde el inicio del capítulo previo al capítulo que se está reproduciendo. El comando `top()` hace que la reproducción comience desde el inicio de un capítulo que se está reproduciendo. El comando `next()` hace que la reproducción comience desde el inicio del capítulo siguiente al capítulo que se está reproduciendo. El comando `forwardScan(speed)` hace que la reproducción se lleve a cabo en la dirección hacia adelante a una velocidad designada por "speed" (velocidad). El comando `backwardScan(speed)` hace que la reproducción se lleve a cabo en la dirección hacia atrás a una velocidad designada por "speed" (velocidad). "speed" puede designarse en el rango de 0,1 a 360.

El comando `moveCursor(direction, [level])` hace que el cursor se mueva en las direcciones superior, inferior, izquierda y derecha. "direction" (dirección) es cualquiera de "upper" (superior), "lower" (inferior), "left" (izquierda) y "right" (derecha) que hacen que el cursor se mueva en las direcciones superior, inferior, izquierda y derecha, respectivamente. "level" (nivel) puede designarse con un valor en el rango de "0" a "255". "level" es opcional y normalmente no se usa o es "0". Cuando "level" es un valor intermedio ("1" a "255"), un *joystick* analógico puede representarse.

El comando `click(buttonNumber)` hace que se haga click en un botón en la pantalla. El comando `focus(buttonNumber)` hace que la atención se centre en un botón en la pantalla.

El comando `stillOff()` hace que la reproducción paralizada por el reproductor se reanude. El comando `stillOff()` puede integrarse con el comando `pauseOff`. El comando `pauseOn()` hace que la reproducción se pause. El comando `pauseOn()` puede funcionar solamente cuando se reproduce un escenario. El comando `pauseOff()` hace que la reproducción pausada se libere.

El comando `LanguageSelect(languageCode)` designa el idioma de visualización del menú. El comando `StreamChange(StreamNumber)` hace que un flujo que se está reproduciendo cambie. El comando `setEnabled(Boolean)` hace que un flujo se visualice/no se visualice. El comando `angleChange(angleNumber)` hace que un ángulo de visualización cambie. El comando `parentalCountrySelect(countryCode)` hace que un código de país para un nivel parental se establezca. El comando `videoPresentationModeChange(mode)` designa un formato de visualización de video. "mode" (modo) es cualquiera de "Ancho", "LB (Letter Box)" y "PanScan".

La Figura 22 muestra un ejemplo de clase `RmtKeyEvent` que representa un evento de entrada de tecla que tiene lugar en el comandante de mando a distancia. El parámetro "source" es un objeto de una fuente de un evento. Por consiguiente, "source" representa el objeto `BDBasicPlayer`. El parámetro "id" es un entero que representa el tipo de evento. El parámetro "when" es un entero en el formato largo que representa el tiempo en el cual un evento tiene lugar. El parámetro "mtKeyCode" es un código tipo entero correspondiente a una tecla virtual de entrada o "VK_UNDEFINED". "VK_UNDEFINED" se describirá más adelante.

Como se describe en la Figura 3A y Figura 3B, las entradas de usuario mediante teclas del comandante de mando a distancia se reciben por el hardware y se transfieren al objeto `BDBasicPlayer`. El objeto `BDBasicPlayer` genera el objeto `RmtKeyEvent` correspondiente a la entrada de usuario y clase `RmtKeyEvent`. Se informa a un objeto que ha registrado un oyente de eventos correspondiente al objeto `RmtKeyEvent` sobre un comando correspondiente a una entrada de usuario.

La Figura 23 muestra una lista de ejemplos de métodos del objeto `RmtKeyEvent`. El método `getRmtKeyCode` devuelve un código de tecla correspondiente a una tecla que ha ocasionado que un evento tenga lugar. El método `setRmtKeyCode` establece un código de tecla de una tecla que hace que un evento tenga lugar. Cuando se ejecuta el método `getRmtKeyCode`, puede determinarse qué objeto de tecla `RmtKeyEvent` tiene lugar.

La Figura 24, Figura 25A y Figura 25B muestran listas de ejemplos de teclas obtenidas por el método getRmtKeyCode descrito en la Figura 23. Las teclas que comienzan con "VK" que se muestran en la Figura 25A y Figura 25B son teclas virtuales en el objeto BDBasicPalyer. La tecla VK_POWER provee una función correspondiente a la tecla de encendido. La tecla VK_POWER_ON provee una función correspondiente a la tecla ON (ENCENDIDO). La tecla VK_POWER_OFF provee una función correspondiente a la tecla OFF (APAGADO). La tecla VK_MENU provee una función correspondiente a una tecla de menú que hace que un menú aparezca. La tecla VK_ENTER provee una función correspondiente a una tecla de ingreso que hace que se ingresen datos. La tecla VK_RETURN provee una función correspondiente a una tecla de retorno que hace que la etapa de un proceso se devuelva.

La tecla VK_PLAY provee una función correspondiente a una tecla de reproducción que hace que se lleve a cabo la reproducción. La tecla VK_STOP provee una función correspondiente a una tecla de parada que hace que se detenga la reproducción. La tecla VK_PAUSE provee una función correspondiente a una tecla de pausa que hace que se pause la reproducción. La tecla VK_FAST_FORWARD provee una función correspondiente a una tecla de avance rápido que hace que se lleve a cabo la reproducción hacia adelante de forma rápida. La tecla VK_FAST_REVERSE provee una función correspondiente a una tecla de marcha atrás que hace que se lleve a cabo la reproducción hacia atrás de forma rápida. La tecla VK_SLOW_FORWARD provee una función correspondiente a una tecla de avance lento que hace que se lleve a cabo la reproducción lenta hacia adelante. La tecla VK_SLOW_REVERSE provee una función correspondiente a una tecla de reproducción hacia atrás lenta que hace que se lleve a cabo la reproducción lenta hacia atrás. La tecla VK_STEP_FORWARD provee una función correspondiente a una tecla de avance de etapa que hace que se lleve a cabo la reproducción de etapa hacia adelante. La tecla VK_STEP_REVERSE provee una función correspondiente a una tecla de rebobinado de etapa que hace que se lleve a cabo la reproducción de etapa hacia atrás.

La tecla VK_NEXT provee una función correspondiente a una tecla de siguiente que hace que se ingrese un valor correspondiente a "next" (siguiente). La tecla VK_PREVIOUS provee una función correspondiente a una tecla de previo que hace que se ingrese un valor correspondiente a "previous" (previo). Con la tecla VK_NEXT y la tecla VK_PREVIOUS, se provoca que el capítulo actual salte a los capítulos siguiente y previo, respectivamente.

La tecla VK_UP provee una función correspondiente a la tecla de ascendente que hace que se ingrese un valor "up" (ascendente). La tecla VK_DOWN provee una función correspondiente a la tecla de descendente que hace que se ingrese un valor "down" (descendente). La tecla VK_RIGHT provee una función correspondiente a la tecla derecha que hace que se ingrese un valor "right" (derecho). La tecla VK_LEFT provee una función correspondiente a la tecla izquierda que hace que se ingrese un valor "left" (izquierdo). La tecla VK_UP_RIGHT provee una función correspondiente a la tecla derecha superior que hace que se ingrese un valor "upper right" (superior derecho). La tecla VK_UP_LEFT provee una función correspondiente a la tecla izquierda superior que hace que se ingrese un valor "upper left" (superior izquierdo). La tecla VK_DOWN_RIGHT provee una función correspondiente a la tecla derecha inferior que hace que se ingrese un valor "lower right" (inferior derecho). La tecla VK_DOWN_LEFT provee una función correspondiente a la tecla izquierda inferior que hace que se ingrese un valor "lower left" (inferior izquierdo). Dichas teclas direccionales hacen que el cursor se mueva en la pantalla.

La tecla VK_ANGLE provee una función correspondiente a la tecla de selección de ángulo que hace que se cambie el ángulo de un contenido de vídeo de múltiples ángulos. La tecla VK_PRESENTATION_GRAPHICS provee una función correspondiente a la tecla de selección de subtítulo que selecciona el subtítulo inglés/japonés, el modo de visualización/no visualización de subtítulo, y así sucesivamente. La tecla VK_AUDIO provee una función correspondiente a la tecla de selección de audio que selecciona un modo de audio como, por ejemplo, un modo envolvente o un modo bilingüe. La tecla VK_VIDEO_ASPECT provee una función correspondiente a la tecla de selección de relación de aspecto que selecciona una relación de aspecto de un contenido de vídeo. La tecla VK_COLORED_KEY_1 provee una función correspondiente a la tecla de función de color 1. La tecla VK_COLORED_KEY_2 provee una función correspondiente a la tecla de función de color 2. La tecla VK_COLORED_KEY_3 provee una función correspondiente a la tecla de función de color 3. La tecla VK_COLORED_KEY_4 provee una función correspondiente a la tecla de función de color 4. La tecla VK_COLORED_KEY_5 provee una función correspondiente a la tecla de función de color 5. La tecla VK_COLORED_KEY_6 provee una función correspondiente a la tecla de función de color 6. La tecla VK_UNDEFINED es una tecla indefinida.

Las teclas enumeradas en la Figura 24, Figura 25A y Figura 25B no siempre coinciden con las teclas físicamente dispuestas en un comandante de mando a distancia real. Además, dependiendo del tipo de comandante de mando a distancia, múltiples funciones pueden asignarse a una tecla física. De manera alternativa, el usuario puede seleccionar una de las imágenes de teclas de un controlador remoto que aparece en la pantalla. Por consiguiente, las teclas que se muestran en la Figura 24, Figura 25A y Figura 25B son teclas virtuales que, de forma abstracta, forman las funciones correspondientes. Dado que las entradas de usuario para el objeto BDBasicPlayer se representan con dichas teclas virtuales, las entradas de usuario que no dependen del aparato pueden definirse.

La Figura 26 muestra un ejemplo de un proceso que ejecuta un programa provisto cuando un evento de entrada de usuario tiene lugar. En el proceso que se muestra en la Figura 26, cuando el usuario presiona la tecla "next"

mientras un capítulo se reproduce normalmente, el capítulo actual salta al capítulo siguiente, el capítulo siguiente se reproduce y un mensaje preparado aparece en la pantalla.

La tecla "next" es una tecla que hace que el capítulo actual salte al capítulo siguiente y que el capítulo siguiente se reproduzca. La tecla "next" es, por ejemplo, una tecla dispuesta en el comandante de mando a distancia. El proceso del diagrama de flujo que se muestra en la Figura 26 comienza cuando el usuario presiona la tecla "next" mientras el reproductor reproduce un capítulo (en la etapa E41). Cuando se presiona la tecla "next", el objeto RmekeyEvent tiene lugar (en la etapa E42). La base de datos se busca para una marca de capítulo del capítulo siguiente (en la etapa E43). Según el resultado buscado, se determina si existe o no la marca de capítulo siguiente (en la etapa E44). Cuando no existe marca de capítulo, una serie de procesos se completan y la reproducción regular se reanuda. Cuando el resultado determinado en la etapa E44 representa que existe la marca de capítulo siguiente, el flujo avanza a la etapa E45.

En la etapa E45, el objeto BDBasicPlayer hace que la reproducción del capítulo actual que se está reproduciendo se detenga en respuesta a la tecla "next" (en la etapa E45). Cuando la reproducción del reproductor se detiene, el evento PlayStoppedEvent tiene lugar (en la etapa E46). El objeto BDBasicPlayer obtiene la posición de byte de la marca de capítulo siguiente en el archivo de flujo AV clip a partir de la información de punto de característica del archivo de información de clip. Se accede a la posición de byte obtenida del archivo y el flujo se lee y reproduce desde la posición de byte (en la etapa E47).

Después de la etapa E48, una serie de procesos que hacen que un mensaje aparezca en la pantalla se llevan a cabo para informar al usuario que el capítulo actual ha cambiado. Con el fin de hacer que un mensaje aparezca en una posición en la que el capítulo actual ha cambiado, un método que hace que el reproductor visualice un mensaje necesita implementarse como un oyente de eventos correspondiente al evento MarkEncounteredEvent. Cuando se inicia la reproducción del flujo, tiene lugar el evento PlayStartedEvent. Además, el evento MarkEncounteredEvent tiene lugar según la marca de capítulo (en la etapa E48).

Cuando el evento MarkEncounterEvent tiene lugar, en la etapa E49, se inicia la ejecución de un oyente correspondiente al evento MarkEncounteredEvent. En el oyente, se obtiene el tipo de marca detectada por el evento MarkEncounteredEvent (en la etapa E50). Se determina si la marca es o no una marca de capítulo (en la etapa E51). Cuando el resultado determinado representa que la marca no es una marca de capítulo, el flujo avanza a la etapa E53. En la etapa E53, se completa la ejecución del oyente. Cuando el resultado determinado representa que la marca detectada es un marca de capítulo, el flujo avanza a la etapa E52. En la etapa E52, aparece un mensaje que representa el comienzo de un capítulo.

En el presente ejemplo, cuando el usuario presiona una tecla del comandante de mando a distancia, el evento RmtKeyEvent tiene lugar. Cuando el evento RmtKeyEvent tiene lugar, múltiples cambios de estado tienen lugar. Como resultado, múltiples eventos, PlayStoppedEvent, PlayStartedEvent, y MarkEncounteredEvent tienen lugar.

Por consiguiente, en el modelo de reproductor virtual BD-ROM según la primera realización de la presente invención, un evento de entrada de usuario hace que el estado del reproductor cambie y que un nuevo evento tenga lugar. Con eventos nuevos, pueden llevarse a cabo varios procesos.

Según la primera realización de la presente invención, mediante la definición de un modelo de reproductor virtual BD-ROM, varios méritos que no están presentes en el vídeo DVD pueden obtenerse. Por ejemplo, según la primera realización de la presente invención, varias funciones del vídeo DVD convencional pueden lograrse con un modelo.

Como un primer ejemplo, una función de visualización de subtítulo forzada y una función de prohibición UOP (Función de Usuario) pueden usarse. La función de visualización de subtítulo forzada es una función que hace que un subtítulo predeterminado se visualice, de manera forzada, en una escena predeterminada incluso si el usuario ha apagado la función de visualización de subtítulo. La función de prohibición UOP es una función que prohíbe al usuario ingresar datos. En el vídeo DVD, se definen estructuras de datos y modelos de funcionamiento para dichas funciones.

En el modelo de reproductor virtual BD-ROM según la primera realización de la presente invención, con respecto a la función de visualización de subtítulo forzada, se marca una escena (tiempo) para la cual un subtítulo se visualiza de manera forzada. Un proceso que hace que un subtítulo se visualice de manera forzada en la pantalla independientemente de si el modo de visualización de subtítulo se ha encendido o apagado se provee como un oyente. Por consiguiente, con un modelo que procesa marcas regulares, la función de visualización de subtítulo forzada puede lograrse.

Con respecto a la función de prohibición UOP, a menos que un oyente de eventos para eventos de entrada de tecla del comandante de mando a distancia se hubiera registrado, incluso si un evento de entrada de tecla tiene lugar, no se lleva a cabo ningún proceso. Por consiguiente, la entrada de usuario se ignora y la UOP se prohíbe.

Por consiguiente, según la primera realización de la presente invención, dado que las funciones del DVD pueden lograrse con un modelo de evento, el lado de creador de disco puede crear, de manera no trabajosa, el DVD.

Además, en el vídeo DVD, las teclas del comandante de mando a distancia se preasignan para las funciones. Cuando se presiona la tecla de submenú del comandante de mando a distancia, como una función que se ha preasignado para la tecla de submenú, aparece una pantalla de submenú en la cual el modo de audio, modo de subtítulo y ángulo pueden cambiarse. Sin embargo, dado que el submenú no son datos esenciales del vídeo DVD, existe un disco que no tiene el submenú. En el presente caso, incluso si se presiona la tecla de submenú mientras el reproductor tiene dicho disco, ninguna función tiene lugar.

Por otro lado, en el modelo de reproductor virtual BD-ROM según la primera realización de la presente invención, según se describe con referencia a la Figura 21A, Figura 21B, Figura 21C, Figura 21D, Figura 22, Figura 23, Figura 24, Figura 25A, y Figura 25B, las relaciones de teclas y procesos llevados a cabo correspondientes a aquel pueden designarse por el lado de creador de disco. Por consiguiente, con un programa que hace que el reproductor de un mensaje aparezca en el caso en que el disco no tiene el submenú, cuando el usuario presiona la tecla de submenú, puede informarse al usuario sobre un mensaje que representa que el disco no tiene el submenú.

Por consiguiente, según la primera realización de la presente invención, dado que el lado de creador de disco tiene un grado más alto de libertad para crear un disco, el lado de creador de disco puede crear un disco que el usuario puede manejar fácilmente.

Además, según la primera realización de la presente invención, una clase Java se define correspondiente al modelo de reproductor virtual BD-ROM y la reproducción de una lista de reproducción se controla con un programa Java. Con el programa Java, un programa de múltiples procesos puede lograrse. Como resultado, los períodos de retardo debido a un acceso a la red y el tiempo de espera en respuesta a una entrada de usuario pueden manejarse fácilmente. Además, con el programa Java, una predicción de derivación, una ejecución especulativa, y así sucesivamente pueden lograrse fácilmente por los comandos de prelectura. Por consiguiente, los procesos pueden llevarse a cabo a una alta velocidad.

1-5. Sobre la estructura de gestión de archivos

A continuación, con referencia a la Figura 27, se describirá una estructura de gestión de archivos aplicable a un perfil completo del BD-ROM. Los archivos se gestionan de manera jerárquica. Un directorio (un directorio raíz en el ejemplo que se muestra en la Figura 27) se forma en un medio de grabación. Los archivos debajo del presente directorio se gestionan por un sistema de grabación y reproducción.

Debajo del directorio raíz, se ubica el directorio BDMV. Múltiples directorios BDMV pueden ubicarse debajo del directorio raíz.

Debajo del directorio BDMV, se ubican dos archivos "scenariodmvm" y "entrylist.data". Debajo del directorio BDMV, se ubican múltiples directorios "PLAYLIST", "CLIPINF", y "STREAM".

El archivo "scenariodmvm" es un archivo de escenario que almacena un escenario. Un escenario tiene una estructura en la cual se disponen múltiples listas de reproducción. Por ejemplo, un escenario controla el orden de reproducción de las listas de reproducción. Para permitir al usuario ver un escenario en sus unidades divididas, pueden proveerse capítulos. Los objetos de película en el modo de película HD del BD-ROM se almacenan en el archivo "scenariodmvm".

El archivo "entrylist.data" almacena información sobre un menú como, por ejemplo, puntos de búsqueda (entradas de título) de un escenario, listas de reproducción que componen un menú, y puntos de entrada para un grupo de listas de reproducción. Una tabla de índice en el modo de película HD del BD-ROM se almacena en el archivo "entrylist.data".

El directorio "PLAYLIST" almacena las listas de reproducción. Una lista de reproducción se almacena en un archivo "****.mpls". La información que representa las posiciones de marcas en un flujo AV clip se almacena en el archivo de lista de reproducción "****.mpls" que designa el rango de reproducción del flujo AV clip.

El directorio "CLIPINF" almacena archivos de información de clip "#####.clpi". El directorio "STREAM" almacena archivos de flujo AV clip "%%%%.m2ts". Cuando los nombres de archivos relacionados se asignan a archivos de flujo AV clip, ellos pueden correlacionarse fácilmente. Por ejemplo, los mismos nombres de archivo "#####" and "%%%%" excepto por las extensiones se usan para archivos de información de clip y archivos de flujo AV clip.

Un archivo de clase Java que compone un programa Java se almacena en un directorio predeterminado (no se muestra) ubicado debajo del directorio BDMV. De manera alternativa, el archivo de clase Java puede ubicarse en un directorio en el mismo nivel jerárquico que el directorio BDMV.

1-6. Sobre el modelo de descodificador

A continuación, se describirá un descodificador de reproductor 100 según la primera realización de la presente invención. La Figura 28A, Figura 28B y Figura 28C son un diagrama de bloques funcional que muestra un ejemplo de una estructura de un descodificador de reproductor 100 según la primera realización de la presente invención. El

descodificador de reproductor 100 interpreta los datos reproducidos desde un disco cargado en un dispositivo de unidad (no se muestra), produce un flujo AV (Audio/Vídeo) que compone un contenido, y permite al usuario hacer funcionar, de manera interactiva, el flujo AV de salida.

5 Todas las funciones del descodificador de reproductor 100 se controlan por una CPU (no se muestra). Los flujos y flujos de datos de porciones individuales del descodificador de reproductor 100 se monitorean y controlan por la CPU.

10 Cuando el disco se carga en el dispositivo de unidad (no se muestra), el archivo "scenariio.hdmv" que describe el orden de reproducción de listas de reproducción y el archivo "entrylist.data" que describe la lista de reproducción superior del grupo de listas de reproducción que componen un menú y un título se reproducen. Según las descripciones del archivo "scenariio.hdmv" y el archivo "entrylist.data", otros archivos necesarios se leen desde el disco y así se reproduce un contenido grabado en el disco.

15 Por ejemplo, según las descripciones del archivo "scenariio.hdmv" y el archivo "entrylist.data", los datos de imagen en movimiento visualizados en un plano de vídeo 134 y un segundo plano de vídeo 160, los datos de imágenes visualizados en un plano de gráficos A 132, un plano de gráficos B 133, y un segundo plano de vídeo 160, un archivo de lista de reproducción, y así sucesivamente se leen desde el disco. En el perfil completo, se lee y ejecuta un archivo que almacena un programa.

20 En la siguiente descripción, entre los datos que se leen desde el disco, se hace referencia a los flujos como, por ejemplo, datos de imágenes en movimiento, subimágenes (datos de subtítulo) y datos de sonido que deben procesarse de manera continua como flujos en tiempo real. Por el contrario, se hace referencia a los datos en tiempo no real como, por ejemplo, archivos de escenario, archivos de lista de reproducción, archivos de programa, parte de imágenes en movimiento, imágenes sin movimiento, y datos de sonido que no necesitan procesarse continuamente como objetos de almacenamiento. Los objetos de almacenamiento se almacenan en una memoria o similares y se expanden allí. Los objetos de almacenamiento almacenados en la memoria se procesan cuando fuera necesario.

25 El descodificador de reproductor 100 tiene dos sistemas de canales de entrada que son el canal (1) y el canal (2). Un objeto de almacenamiento se ingresa en una terminal de entrada 101 del canal de entrada (1). Un flujo en tiempo real se ingresa en una terminal de entrada 202 del canal de entrada (2). De manera alternativa, un objeto de almacenamiento puede ingresarse en la terminal de entrada 202. Según la primera y segunda realizaciones de la presente invención, un flujo en tiempo real y parte de los objetos de almacenamiento que se ingresan en la terminal de entrada 202 son, por ejemplo, MPEG2 TS (Flujos de Transporte del Grupo de Expertos de Imágenes en Movimiento 2).

35 Un flujo en tiempo real que se ingresa en la terminal de entrada 202 no se limita a un MPEG 2 TS. Siempre que un flujo en tiempo real pueda transmitirse en la unidad de un paquete y multiplexarse con datos de vídeo, datos de audio, datos de imágenes sin movimiento, o similares, puede ingresarse un flujo que tiene otro formato. En este punto, un filtro PID 110 que se describirá más adelante se usa como un demultiplexor que demultiplexa datos de vídeo, datos de audio, datos de imágenes sin movimiento, o similares.

Cuando la velocidad de rotación del disco en el dispositivo de unidad aumenta, por ejemplo, dos veces y la velocidad de transferencia de lectura del disco aumenta, las funciones de lectura para los dos sistemas de los canales (1) y (2) del disco pueden llevarse a cabo según la división de tiempo.

40 A continuación, se describirá el sistema del canal de entrada (1). Un objeto de almacenamiento que se ingresa en la terminal de entrada 101 se ingresa en un circuito de conmutación 102. Cuando se ingresa un código de programa de un archivo Java, un guión ECMA (Asociación Europea de Fabricantes de Ordenadores), un archivo HTML (Lenguaje de Marcado de Hipertexto) (o un archivo XHTML), o similares como un objeto de almacenamiento, el circuito de conmutación 102 selecciona una terminal de salida 102A. El código de programa de entrada se almacena en un búfer de código 104.

45 Por otro lado, cuando los datos de imágenes se ingresan como un objeto de almacenamiento, el circuito de conmutación 102 selecciona una terminal de salida 102B. Como resultado, los datos de imágenes de entrada se ingresan en un circuito de conmutación 103. Cuando un flujo en tiempo real que se ingresa en la terminal de entrada 202 no contiene datos de imágenes visualizados en el plano de gráficos A 132 y el plano de gráficos B 133, el circuito de conmutación 103 selecciona una terminal de entrada 103A. Los datos de imágenes que se ingresan desde el circuito de conmutación 102 se almacenan en un búfer de contenidos 105.

55 Asimismo, cuando los datos de imágenes visualizados en el plano de gráficos A 132 y el plano de gráficos B 133 se contienen en un flujo en tiempo real que se ingresa en la terminal de entrada 202, el circuito de conmutación 103 selecciona una terminal de entrada 103B. Como resultado, los datos de imágenes se almacenan en el búfer de contenidos 105. Los objetos de almacenamiento almacenados en el búfer de códigos 104 y el búfer de contenidos 105 se leen cuando es necesario y se proveen a un motor multimedia 106.

Los datos de imágenes del objeto de almacenamiento almacenado en el búfer de contenidos 105 se proveen también a un decodificador de gráficos A 116 y a un decodificador de gráficos B 117 a través de circuitos de conmutación 107 y 108, respectivamente.

5 El motor multimedia 106 incluye un analizador XML 106A, un intérprete de programa/guión 106B y un reproductor de gráficos 106C. El motor multimedia 106 también incluye un reproductor de sonido 106D que puede manejar datos de audio. El motor multimedia 106 puede constar de hardware independiente. De manera alternativa, el motor multimedia 106 puede lograrse por un proceso de un programa predeterminado que la anterior CPU (no se muestra) ejecuta.

10 El analizador XML 106A tiene una función para analizar un documento XML (Lenguaje de Marcado Extensible). Además, el analizador XML 106A puede también analizar un documento HTML y un documento XHTML. Un documento HTML y un documento XHTML analizados por el analizador XML 106A se convierten en respectivos formatos que pueden ejecutarse por el decodificador de reproductor 100. El intérprete de programa/guión 106B analiza un programa Java (marca comercial registrada), un guión ECMA, y así sucesivamente y los convierte en respectivos formatos que pueden ejecutarse por el decodificador de reproductor 100. El reproductor de gráficos 15 106C descodifica datos de imágenes y obtiene un formato que puede expandirse en el plano de subtítulos 11 y el plano de gráficos 12.

20 El motor multimedia 106 lleva a cabo procesos para el analizador XML 106A, el intérprete de programa/guión 106B, y el reproductor de gráficos 106C con una memoria de trabajo de un búfer 109. Por ejemplo, el analizador XML 106A y el intérprete de programa/guión 106B usan un búfer de código 109a del búfer 109. El reproductor de gráficos 106C usa un búfer de gráficos 109D del búfer 109. El búfer 109 también incluye un búfer fuente 109B que almacena datos fuente usados para visualizar una cadena de caracteres, un búfer de árbol 109C que almacena el resultado analizado del documento HTML por el analizador XML 106A en una estructura de árbol jerárquica, un búfer de sonido 109E que almacena datos de audio usados en el reproductor de sonido 106D y así sucesivamente.

25 Cuando fuera necesario, el motor multimedia 106 lee un programa Java del búfer de códigos 104, datos de imágenes del búfer de contenidos 105 y así sucesivamente. Los datos que se almacenan en el búfer de códigos 104 y el búfer de contenidos 105 pueden almacenarse en el búfer de códigos 104 y el búfer de contenidos 105 hasta que los datos se conviertan en innecesarios. Por consiguiente, los datos almacenados en el búfer de códigos 104 y el búfer de contenidos 105 pueden leerse, de manera repetida, y usarse cuando fuera necesario.

30 Además de los datos anteriores, el motor multimedia 106 lee, por ejemplo, un guión ECMA almacenado en el búfer de códigos 104 y lo ejecuta. Cuando fuera necesario, el motor multimedia 106 puede leer otro guión ECMA y un documento HTML (o documento XHTML) del búfer de códigos 104 y ejecutarlos según la descripción del guión ECMA que se ha leído.

35 Además, el motor multimedia 106 recibe una entrada de usuario de un comandante de mando a distancia, un dispositivo de enfoque, o similares y lleva a cabo un proceso predeterminado para la entrada de usuario. La entrada de usuario se provee al decodificador de gráficos A 116, decodificador de gráficos B 117, decodificador de audio 118, decodificador de vídeo MPEG 120 y decodificador de sistema 121 que se describirán más adelante.

40 Los datos de imágenes procesados por el reproductor de gráficos 106C se proveen al plano de gráficos A 132 y al plano de gráficos B 133 a través de circuitos de conmutación 130 y 131, respectivamente. En el presente ejemplo, aunque los datos de imágenes provistos al plano de gráficos A 132 y al plano de gráficos B 133 pueden tener varios formatos, por ejemplo, el formato PNG, el formato de longitud de ejecución, y el formato JPEG, según la presente invención, los formatos de aquellos no se encuentran limitados. El tiempo en el cual los datos de imágenes se proveen a los planos 132 y 133 se controla por el motor multimedia 106.

45 Además, el motor multimedia 106 provee señales de control a un procesador de presentación 155 que se describirá más adelante para hacer que este seleccione uno del vídeo 134, el plano de subtítulo 134, el segundo plano de vídeo 160, el plano de gráficos A 132, y el plano de gráficos B 133 y lleve a cabo la mezcla alfa y así sucesivamente. Asimismo, el motor multimedia 106 provee una señal de control a un procesador de presentación 157 que se describirá más adelante para hacer que este controle una salida de un flujo de audio.

50 El motor multimedia 106 lee un programa Java de una ROM (no se muestra) o similares y genera el objeto BDBasicPlayer en el motor multimedia 106. El motor multimedia 106 lee un programa Java del búfer de códigos 104. Cuando fuera necesario, un objeto compuesto por un programa Java que se ha leído desde el búfer de códigos 104 registra un oyente de eventos para el objeto BDBasicPlayer.

55 Como se describe en la Figura 15, en el motor multimedia 106, el objeto BDBasicPlayer transmite un evento a un objeto compuesto por un programa Java que se ha leído del búfer de códigos 104 correspondiente a, por ejemplo, una entrada de usuario. El objeto ejecuta el método correspondiente. El objeto BDBasicPlayer detecta un cambio de estado del decodificador de reproductor 100. Cuando el objeto BDBasicPlayer ha detectado un cambio de estado predeterminado, el objeto genera un evento correspondiente. El motor multimedia 106 hace que un método correspondiente al evento controle un dispositivo de unidad (no se muestra) y cada porción del decodificador de reproductor 100 para llevar a cabo un control de reproducción para el disco.

- 5 A continuación, se describirá el sistema del canal de entrada (2). Un flujo en tiempo real que se ingresa como un MPEG 2 TS en la terminal de entrada 202 se provee al filtro PID 110. El filtro PID 110 extrae una PID (Identificación de Paquete) de un paquete de transporte del flujo de transporte MPEG 2 TS y detecta un atributo del flujo del paquete de transporte. El filtro PID 110 separa el flujo en tiempo real de entrada en sistemas correspondientes para cada paquete de transporte según el atributo del flujo.
- 10 Cuando un paquete de transporte es un paquete en el cual se contienen los datos de imágenes de un objeto de almacenamiento, el paquete de transporte se almacena de manera temporal en un búfer TBn 111A. El paquete de transporte se lee en un tiempo predeterminado y se ingresa en el circuito de conmutación 103 a través de la terminal de entrada 103B que se ha seleccionado. De allí en adelante, el paquete de transporte se almacena en el búfer de contenidos 105 a través del circuito de conmutación 103.
- 15 Cuando el filtro PID 110 ha determinado que el paquete de transporte contiene datos de subtítulos según la PID, el paquete de transporte se almacena de manera temporal en un búfer TBn 111B y un búfer Bn 112B. El paquete de transporte se lee en un tiempo predeterminado y se ingresa en el circuito de conmutación 107 a través de una terminal de entrada 107B que se ha seleccionado. El paquete de transporte se provee al descodificador de gráficos A 116 a través del circuito de conmutación 107.
- 20 El descodificador de gráficos A 116 elimina la información de encabezamiento del paquete de transporte provisto, descodifica los datos de subtítulos contenidos en el paquete de transporte, y obtiene datos de imágenes para subtítulos o similares. Los datos de imágenes se ingresan en una terminal de entrada 130B del circuito de conmutación 130 y se expanden al plano de gráficos A 132 a través del circuito de conmutación 130. Además, los datos de imágenes pueden expandirse al plano de gráficos B 133 a través del circuito de conmutación 131.
- 25 Cuando el filtro PID 110 ha determinado que el paquete de transporte contiene datos de gráficos según la PID, el paquete de transporte se almacena de manera temporal en un búfer TBn 111C y un búfer Bn 112C. El paquete de transporte se lee en un tiempo predeterminado y se ingresa en el circuito de conmutación 108 a través de una terminal de entrada 108B que se ha seleccionado. El paquete de transporte se provee al descodificador de gráficos B 117 a través del circuito de conmutación 108.
- 30 El descodificador de gráficos B 117 elimina la información de encabezamiento del paquete de transporte provisto, descodifica los datos de gráficos contenidos en el paquete de transporte, y obtiene datos de gráficos. Los datos de imágenes se ingresan en una terminal de entrada 131B del circuito de conmutación 131 en un tiempo predeterminado y se expanden al plano de gráficos B 133 a través del circuito de conmutación 131. Además, los datos de imágenes pueden expandirse al plano de gráficos A 132 a través del circuito de conmutación 131.
- 35 La función del descodificador de gráficos A 116 no es muy diferente de la función del descodificador de gráficos B 117. En otras palabras, como un modelo, existen dos sistemas de descodificadores de gráficos que funcionan de manera independiente. Por consiguiente, se considera que los datos de subtítulos y datos de gráficos pueden descodificarse de manera independiente. En una implementación real, un descodificador de gráficos de alta velocidad se usaría sobre la base de división del tiempo como si existieran dos sistemas virtuales de descodificadores de gráficos.
- 40 Cuando el filtro PID 110 ha determinado que el paquete de transporte contiene datos de audio según la PID, el paquete de transporte se almacena de manera temporal en un búfer TBn 111D y un búfer Bn 112D. El paquete de transporte se lee en un tiempo predeterminado y se provee al descodificador de audio 118. Los datos de audio contenidos en el paquete de transporte se codifican por compresión según un sistema basado en, por ejemplo, el MPEG.
- 45 El descodificador de audio 118 también tiene, por ejemplo, un descodificador de audio PCM (Modulación por Impulsos Codificados) lineal 119. El descodificador de audio 118 elimina la información de encabezamiento del flujo de transporte de entrada, descodifica los datos de audio codificados por compresión contenidos en el paquete de transporte, y obtiene datos de audio de PCM lineal.
- Los datos de audio de PCM lineal que se producen desde el descodificador de audio 118 se ingresan en el procesador de presentación 157 para el audio. El procesador de presentación 157 añade un sonido de efecto predeterminado y así sucesivamente a los datos de audio de PCM lineal bajo el control del motor multimedia 106. Los datos resultantes se obtienen de una terminal de salida 158.
- 50 Cuando el filtro PID 110 ha determinado que el paquete de transporte contiene datos de imágenes en movimiento según la PID, el paquete de transporte se almacena de manera temporal en un búfer TBn 111E, un búfer MBn 113, y un búfer EBn 114, leídos en un tiempo predeterminado, y provistos al descodificador de vídeo MPEG 120. Los datos de imágenes en movimiento contenidos en el paquete de transporte se han codificado por compresión según el sistema MPEG 2.
- 55 El descodificador de vídeo MPEG 120 elimina la información de encabezamiento del paquete de transporte provisto, descodifica datos de imágenes en movimiento que se han codificado por compresión según el sistema MPEG 2, y obtiene datos de imágenes en movimiento de banda base.

5 Los datos de imágenes en movimiento que se producen desde el descodificador de vídeo MPEG 120 se ingresan en una terminal de entrada 124A de un circuito de conmutación 124. En el circuito de conmutación 124, los datos de imágenes en movimiento que se producen desde el descodificador de vídeo MPEG 120 y los datos de imágenes en movimiento que se producen desde el motor multimedia 106 se seleccionan. Los datos de imágenes en movimiento seleccionados en un tiempo predeterminado se ingresan en una conmutación 123. La conmutación 123 selecciona un plano de vídeo al cual se expanden los datos de imágenes en movimiento. Los datos de imágenes en movimiento se desarrollan para el plano de vídeo 134 o el segundo plano de vídeo 160.

10 Cuando el filtro PID 110 ha determinado que el paquete de transporte contiene información de sistema según la PID, el paquete de transporte se provee al descodificador de sistema 121 a través de los búferes TBn 111F y Bsys 115. El descodificador de sistema 121 elimina la información de encabezamiento del paquete de transporte provisto y extrae de allí la información de sistema. La información de sistema se provee a, por ejemplo, una CPU (no se muestra).

15 Los datos de imágenes en el plano de gráficos A 132 se proveen a una paleta 150. La paleta tiene 256 colores. Se hace referencia a la paleta como un índice. Se producen los datos RGB. Además, se extraen los datos de intransparencia α_1 que son un parámetro para la mezcla alfa. Los datos RGB se convierten en datos YCbCr por un circuito de conversión RGB/YCbCr 151. Los datos YCbCr y los datos de intransparencia α_1 se proveen al procesador de presentación 155.

20 Los datos de imágenes en el plano de gráficos 133 se proveen a una paleta 152. La paleta tiene 256 colores. Se hace referencia a la paleta como un índice. Como resultado, se producen los datos RGB. Además, se extraen los datos de intransparencia α_2 . Los datos RGB se convierten en datos YCbCr por un circuito de conversión RGB/YCbCr 153. Los datos YCbCr y los datos de intransparencia α_2 se proveen al procesador de presentación 155.

Una salida del plano de vídeo 134 se provee al procesador de presentación 155 a través de un convertidor ascendente/descendente 154. Asimismo, una salida del segundo plano de vídeo 160 se provee al procesador de presentación 155 a través de un convertidor ascendente/descendente 161.

25 El convertidor ascendente/descendente 154 es un circuito que convierte la resolución de la imagen. El convertidor ascendente/descendente 154 convierte, por ejemplo, una imagen HD (Alta Definición) que tiene una alta resolución en una imagen SD (Definición Estándar) que tiene una resolución estándar.

30 El procesador de presentación 155 lleva a cabo un proceso de mezcla alfa mediante el uso de intransparencia α_1 de datos de imágenes del plano de gráficos A 132 e intransparencia α_2 del plano de gráficos B 133. Además, el procesador de presentación 155 selecciona, de manera alternativa, salidas del plano de vídeo 134 y del segundo plano de vídeo 160 para cada píxel para lograr la anterior función de imagen en imagen, la función de visualización de fondo de pantalla, y así sucesivamente.

35 En otras palabras, el procesador de presentación 155 selecciona los datos de imágenes del plano de vídeo 134 y del segundo plano de vídeo 160 con una conmutación (no se muestra) para generar datos de imágenes. Los datos de imágenes y los datos de imágenes del plano de gráficos A 132 se combinan según la intransparencia α_1 establecida para los datos de imágenes del plano de gráficos A 132. Además, los datos de imágenes del plano de gráficos B 133 se combinan con los datos de imágenes cuyo plano de vídeo y plano de gráficos A 132 se han combinado según la intransparencia α_2 que se ha establecido para los datos de imágenes del plano de gráficos B 133. Los datos de imágenes del plano de gráficos B 133, los datos de imágenes del plano de gráficos A 132 (datos de subtítulos), y los datos de imágenes cuyos datos de imágenes del plano de vídeo 134 y del segundo plano de vídeo 160 se han combinado se obtienen de una terminal de salida 156.

40 El procesador de presentación 155 puede llevar a cabo un proceso de efectos para los datos de imágenes en tiempo real.

45 En la descripción anterior, cada porción del descodificador de reproductor 100 consta de hardware. Sin embargo, la presente invención no se limita a dicho ejemplo. Por ejemplo, el descodificador de reproductor 100 puede lograrse por un proceso de software. En el presente caso, el descodificador de reproductor 100 puede funcionar en un dispositivo de ordenador. De manera alternativa, el descodificador de reproductor 100 puede lograrse por una combinación de hardware y software. Por ejemplo, el descodificador de audio 118 y el descodificador de vídeo MPEG 120 pueden constar de hardware. El resto del descodificador de reproductor 100 puede constar de software.

50 Un programa que hace que el dispositivo de ordenador ejecute el descodificador de reproductor 100 que se compone de solamente software o una combinación de hardware y software se graba en un medio de grabación, por ejemplo, un CD-ROM (Disco Compacto-Memoria de Solo Lectura) se provee con aquel. El CD-ROM se carga a una unidad de CD-ROM del dispositivo de ordenador. El programa grabado en el CD-ROM se instala en el dispositivo de ordenador. Como resultado, el proceso anterior puede ejecutarse en el dispositivo de ordenador. Dado que la estructura del dispositivo de ordenador es conocida, su descripción se omitirá.

55 2. Segunda realización de la presente invención

A continuación, se describirá una segunda realización de la presente invención. Según la segunda realización, un modelo de reproductor se describe con un lenguaje de guión llamado el guión ECMA. El guión ECMA es un lenguaje de guiones de plataforma cruzada establecido por ECMA (Asociación Europea de Fabricantes de Ordenadores). El guión ECMA se basa en JavaScript (marca comercial registrada). El guión ECMA tiene una cercana afinidad con un documento HTML. Además, el guión ECMA puede definir objetos originales. Por consiguiente, el guión ECMA puede usarse de manera apropiada para un modelo de reproductor según la presente invención.

Como se describe en la Figura 28A, Figura 28B y Figura 28C, dado que el motor multimedia 106 en el decodificador de reproductor 100 puede analizar y ejecutar el guión ECMA. El decodificador de reproductor 100 puede usarse en la segunda realización de la presente invención.

Según la segunda realización de la presente invención, el estándar que usa un lenguaje de guión basado en el guión ECMA se llama el estándar de vídeo UMD (Disco Universal de Medios, marca comercial registrada). La parte de guión del estándar de vídeo UMD se llama el estándar de guión de vídeo UMD.

2-1. Sobre el estándar de vídeo UMD

En primer lugar, para una fácil comprensión, se describirá el estándar de vídeo UMD aplicable a la segunda realización de la presente invención. La Figura 29 muestra una estructura de capa del estándar de vídeo UMD. La estructura de capa que se muestra en la Figura 29 corresponde a la estructura que se muestra en la Figura 1 según la primera realización. En el estándar de vídeo UMD, se definen tres capas, una capa de guión, una capa de lista de reproducción, y una capa de clip. Según la presente estructura, se gestionan los flujos.

En el estándar de vídeo UMD, los datos de vídeo, datos de audio, y datos de subtítulos que se han codificado de manera digital se tratan como un flujo MPEG 2 (Grupo de Expertos de Imágenes en Movimiento 2) en el cual se han multiplexado como flujos elementales MPEG 2. Un flujo MPEG 2 cuyos flujos elementales de vídeo, audio y subtítulos se han multiplexado se llama un flujo AV clip. Un flujo AV clip se almacena en un archivo de flujo AV clip. Cuando se graba un archivo de flujo AV clip, un archivo de información de clip se crea también en la relación de uno a uno. Un conjunto de un archivo de información de clip y un archivo de flujo AV clip correspondiente se llama un clip.

Un clip es una unidad de grabación de datos que se graba en el disco. El orden de reproducción de los clips se gestiona en la capa de lista de reproducción que es una capa superior de clips. La capa de lista de reproducción que es una capa que designa el trayecto de reproducción de clips, contiene una o múltiples listas de reproducción. Una lista de reproducción consta de un conjunto de elementos de reproducción. Un elemento de reproducción contiene un conjunto de un punto de entrada y un punto de salida que representan el rango de reproducción de un clip. Con elementos de reproducción, los clips pueden reproducirse en cualquier orden. Los elementos de reproducción pueden designar el mismo clip. El punto de entrada y el punto de salida de un archivo de flujo AV clip se designan con indicaciones de tiempo (tiempos en un clip). Una indicación de tiempo se convierte en una posición de byte del archivo de flujo AV clip con información del archivo de información de clip.

Una lista de reproducción tiene una estructura por la cual los elementos de reproducción que representan todos o una parte de los clips se reproducen en el orden. Con solamente una lista de reproducción, las derivaciones del orden de reproducción e interactividad con el usuario no pueden lograrse.

La capa de guión es una capa que consta del guión de vídeo UMD, el cual es una extensión del guión ECMA como un lenguaje. El guión de vídeo UMD es un guión basado en ECMA que se extiende para una función original del vídeo UMD.

La capa de guión es una capa superior de la capa de lista de reproducción. La capa de guión consta de comandos de reproducción de lista de reproducción y comandos de establecimiento de reproductor. Con comandos de la capa de guión, las reproducciones de la lista de reproducción, como, por ejemplo, una selección de idioma en la cual un flujo correspondiente a un idioma deseado se selecciona de varios flujos y derivaciones condicionales cuyas las listas de reproducción se reproducirán se seleccionan según condiciones predeterminadas pueden lograrse. Un ejemplo de una aplicación que lleva a cabo una reproducción de una lista de reproducción que tiene derivaciones condicionales es una de varios niveles. La capa de guión provee una función interactiva al usuario.

2-2. Sobre el modelo de reproductor basado en el estándar de vídeo UMD

A continuación, se describirá un modelo de un aparato de reproducción (reproductor) que reproduce datos sobre la base del estándar de vídeo UMD, a saber el modelo de reproductor. En primer lugar, el reproductor lee un programa de guión, una lista de reproducción, y un archivo de información de clip del disco. Luego, el reproductor lee un archivo de flujo AV clip del disco según el orden de reproducción descrito en el programa de guión, la lista de reproducción y el archivo de información de clip y reproduce los datos de vídeo, datos de audio y datos de subtítulos del archivo AV de clip.

En la especificación de lenguaje del programa de guión, un bloque funcional que reproduce una lista de reproducción se implementa como un objeto del programa de guión. En el estándar de vídeo UMD, se hace

referencia a un objeto que reproduce una lista de reproducción como el objeto de reproductor de películas. Los comandos de reproducción de lista de reproducción y comandos de establecimiento de reproductor son métodos del objeto de reproductor de películas. El objeto de reproductor de películas se controla con métodos de la capa de guión. En este punto, se requiere una función que permite al objeto de reproductor de películas informar la capa de guión sobre un cambio de estado, una posición de reproducción, y así sucesivamente. Ello significa que el objeto de reproductor de películas emite un evento al programa de guión. Un proceso correspondiente al evento se describe como un manejador de eventos.

Por consiguiente, con un modelo cuya información se transfiere del objeto de reproductor de películas al programa de guión usando eventos y el programa de guión controla el objeto de reproductor de películas usando métodos, la reproducción de un flujo AV clip puede controlarse por el programa de guión.

La Figura 30 muestra, de manera esquemática, un ejemplo del modelo de reproductor según la segunda realización de la presente invención. Un reproductor de películas 300 es un módulo que reproduce datos de vídeo, datos de audio, y datos de subtítulos según el estándar de vídeo UMD. El objeto de reproductor de películas es un objeto de un programa de guión de modo que el programa de guión hace funcionar un objeto de películas. En otras palabras, un objeto de reproductor de películas es un programa de guión que logra la función del reproductor de películas.

Dado que se considera que el reproductor de películas 300 y el objeto de reproductor de películas representan sustancialmente el mismo objeto, de aquí en adelante se los denota por el mismo numeral de referencia.

En la Figura 30, el reproductor de películas 300 lee, descodifica, y visualiza un archivo de flujo AV clip según un método de una capa inferior (una plataforma de implementación nativa 301) y un método de la capa de guión 302 como una capa superior sobre la base de una lista de reproducción y una base de datos de información de clip.

El interior del reproductor de películas 300 depende de la implementación de un reproductor de vídeo UMD que reproduce el vídeo UMD. Una capa de guión 302 provee una API (Interfaz de Programación de Aplicaciones) como, por ejemplo, un método y una propiedad como un objeto que es una caja negra. En el presente ejemplo, el reproductor de vídeo UMD representa un dispositivo real que implementa un reproductor de películas. Todos los reproductores de vídeo UMD implementan un reproductor de películas que cumple con el estándar de vídeo UMD y que tiene compatibilidad de reproducción.

Como se muestra en la Figura 30, el objeto de reproductor de películas 300 tiene tres trayectos de entrada/salida que son un trayecto que recibe un comando de control 311 de la plataforma de implementación nativa 301, un trayecto que informa a la capa de guión 302 sobre un evento 312, y un trayecto que recibe un método 313 de la capa de guión 302.

El comando de control 311 es un comando que controla el funcionamiento del objeto de reproductor de películas 300 recibido de la plataforma de implementación nativa 301. La plataforma de implementación nativa 301 es una porción específica para un dispositivo como un reproductor de vídeo UMD real y una interfaz con el reproductor de películas 300. El evento 312 es un evento de guión que el reproductor de películas 300 envía a la capa de guión 302. El método 313 es un método que el programa de guión de la capa de guión 302 envía al reproductor de películas 300.

El objeto de reproductor de películas 300 tiene una base de datos 320 para listas de reproducción e información de clip según el estándar de vídeo UMD. El objeto de reproductor de películas 300 oculta una entrada de usuario 310 y convierte una posición de reproducción de tiempo designada en una posición de byte de un flujo AV clip con la base de datos 320.

Un módulo de reproducción 321 del reproductor de películas 300 descodifica un flujo AV clip como un PS (Flujo de Programa) MPEG 2 en el cual los datos de vídeo, datos de audio y datos de subtítulos se han multiplexado. El módulo de reproducción 321 tiene tres estados que son reproducir, parar y pausar. Los cambios del módulo de reproducción 321 funcionan en uno de dichos tres estados según un comando de control y un método (es preciso ver la Figura 31).

La capa de guión 302 es una capa en la cual un programa de guión se ejecuta sobre la base del estándar de guión de vídeo UMD, el objeto de reproductor de películas 300 se controla y una pantalla se visualiza. La capa de guión 302 logra un escenario que el lado de proveedor de contenidos pretende crear. La capa de guión 302 emite el método 313 al objeto de reproductor de películas 300 y recibe el evento 312 del objeto de reproductor de películas 300. La capa de guión 302 intercambia con la plataforma de implementación nativa 301 un evento de tecla 314 correspondiente a una entrada de usuario 310 y un método 315 que hace que la plataforma de implementación nativa 301 muestre una pantalla.

Por ejemplo, los botones que aparecen en la pantalla de menú se dibujan por la plataforma de implementación nativa 301 correspondiente al método 315 que se transfiere del programa de guión de la capa de guión 302 a la plataforma de implementación nativa 301. Cuando el usuario selecciona un botón en la pantalla de menú, la plataforma de implementación nativa 301 informa a la capa de guión 302 sobre el evento de tecla 314 correspondiente a la entrada de usuario 310. El programa de guión de la capa de guión 302 lleva a cabo un proceso para la entrada de usuario 310 correspondiente al evento de tecla 314.

El programa de guión y datos como, por ejemplo, datos de imágenes y datos de sonido usados en el archivo de guión se graban como un archivo para el disco. Dicho archivo corresponde al archivo escenario.hdmv que se muestra en la Figura 25A y Figura 25B.

5 Por consiguiente, el reproductor de películas 300 controla los procesos de decodificación y visualización para los datos de vídeo, datos de audio y datos de subtítulos. Por otro lado, la capa de guión 302 controla los procesos de asignación y visualización para imágenes de partes como, por ejemplo, botones que componen la GUI (Interfaz Gráfica de Usuario) y lleva a cabo un proceso contra un proceso de selección para partes GUI.

10 La plataforma de implementación nativa 301 es una plataforma en la cual funcionan el objeto de reproductor de películas 300 y un programa de guión. Cuando el reproductor de vídeo UMD real consta de hardware, la plataforma de implementación nativa 301 se implementa, específicamente, como hardware de modo que la plataforma de implementación nativa 301 media entre el hardware y el modelo de reproductor.

15 Por ejemplo, la plataforma de implementación nativa 301 recibe la entrada de usuario 310 del usuario y determina si la entrada de usuario 310 recibida es un comando para el reproductor de películas 300 o un comando para un botón que la capa de guión 302 extrae y muestra. Cuando la plataforma de implementación nativa 301 ha determinado que la entrada de usuario 310 es un comando para el reproductor de películas 300, la plataforma de implementación nativa 301 convierte la entrada de usuario 310 en el comando de control 311 que es un comando de control interno del reproductor de películas 300 y emite un comando de control al reproductor de películas 300. Por otro lado, cuando la plataforma de implementación nativa 301 ha determinado que la entrada de usuario 310 es un comando para una parte GUI que la capa de guión 302 extrae y muestra, la plataforma de implementación nativa 301 informa a la capa de guión 302 sobre el evento de tecla 314 correspondiente a la entrada de usuario 310.

20 A continuación, se describirá en detalle el reproductor de películas 300. La Figura 31 muestra una estructura interna de un ejemplo del reproductor de películas 300. Según se describe más arriba, el reproductor de películas 300 consta de la base de datos 320 y del módulo de reproducción 321. La base de datos 320 es una área para la información de las listas de reproducción que se leen desde el disco e información de clips, a saber información de clip.

25 El módulo de reproducción 321 consta de un motor de decodificador 322 y una propiedad 323 que es un valor que representa el estado del módulo de reproducción 321. Como un código de lenguaje, la propiedad 323 consta de dos tipos de una propiedad 323A (parámetro de solo lectura) cuyo valor depende del establecimiento inicial del reproductor de películas 300 y una propiedad 323B (estado de reproductor) cuyo valor depende del estado del módulo de reproducción 321.

30 La propiedad 323A cuyo valor depende del establecimiento inicial se establece por un sistema nativo, por ejemplo un dispositivo real. La propiedad 323A no varía con una lista de reproducción, información de clip, o programa de guión. La propiedad 323A puede leerse desde un programa de guión. Por otro lado, la propiedad 323B que representa el estado del módulo de reproducción 321 puede leerse desde un programa de guión. La propiedad 323B puede escribirse desde una parte de los programas de guión.

35 En el modelo de funcionamiento, se supone que una lista de reproducción y la información de clip se precargan antes de que se reproduzca un flujo AV clip. Otro modelo de implementación puede usarse siempre que las funciones para el modelo de reproductor de películas puedan lograrse.

40 El objeto de reproductor de películas 300 reproduce una lista de reproducción designada correspondiente a un comando recibido de la capa de guión 302 o plataforma de implementación nativa 301. Por ejemplo, el reproductor de películas 300 hace referencia a la base de datos 320 y obtiene la posición de reproducción de un flujo AV clip correspondiente a la lista de reproducción designada como una posición de byte. El motor de decodificador 322 del módulo de reproducción 321 controla el proceso de decodificación para el flujo AV clip correspondiente a la información de posición de reproducción.

45 Como se muestra en la Figura 32, el reproductor de películas 300 tiene tres estados, los cuales son reproducir, parar y pausar, correspondientes a los estados de reproducción de una lista de reproducción. El estado de reproducción representa que el reproductor de películas 300 está reproduciendo una lista de reproducción y que un tiempo está transcurriendo. Además, en la reproducción normal, el estado de reproducción incluye reproducciones de velocidad variable como, por ejemplo, reproducción a una velocidad de 2x y reproducción a una velocidad de 1/2x, y avance y rebobinado rápidos. El estado de pausa representa que el eje temporal se detiene mientras el reproductor de películas 300 está reproduciendo una lista de reproducción. El así llamado estado de reproducción de etapas representa que el reproductor de películas 300 reproduce, hacia adelante o hacia atrás, una lista de reproducción para cada fotograma. El estado de detención representa que el reproductor de vídeo 300 no reproduce ninguna lista de reproducción.

55 Los estados del reproductor de películas 300 cambian de manera correspondiente a los cambios de estado entre reproducción, pausa y detención del motor de decodificador 322 del reproductor de películas 300. El valor de la propiedad 323B se actualiza según el cambio de estado del motor de decodificador 322.

La información de reanudación representa el estado inmediatamente anterior a que tenga lugar el estado de detención. Cuando el reproductor de películas 300 descodifica una lista de reproducción y se convierte en el estado de reproducción, si el reproductor de películas 300 se convierte en el estado de detención, la información de reanudación representa el estado inmediatamente anterior a que tenga lugar el estado de detención. La información de reanudación puede almacenarse en una memoria permanente que el reproductor tiene como hardware de modo que cada título del disco puede identificarse.

2-3. Sobre el modelo de evento del reproductor de películas

A continuación, se describirá un modelo de evento del reproductor de películas 300. El reproductor de películas 300 genera varios eventos en el estado de reproducción en el cual el reproductor de películas 300 reproduce una lista de reproducción. Dicho evento hace que un programa de proceso descrito en un guión se ejecute. Dicho programa de proceso se llama un manejador de eventos. El manejador de eventos es un método requerido cuando un evento tiene lugar. Cuando dicho evento tiene lugar, un modelo de ejecución de programa inicia la ejecución de un programa de proceso. El presente modelo de ejecución de programa se llama un modelo accionado por evento. Según la segunda realización de la presente invención, el programa de guión controla el funcionamiento del objeto de reproductor de películas 300 con un grupo de manejadores de eventos.

La Figura 33 muestra, de manera esquemática, un modelo de evento del reproductor de películas 300 según la segunda realización de la presente invención. En la Figura 33, los manejadores de eventos `onEventA()`, `onEventB()`, y `onEventC()` son interfaces. Los contenidos de dichos manejadores de eventos se describen en el guión. Los contenidos de los manejadores de eventos se crean e implementan por el lado de creador de contenido. En el estándar de guión de vídeo UMD, cada evento cuyo objeto de reproductor de películas 300 informa al programa de guión se provee con un manejador de eventos. En el ejemplo que se muestra en la Figura 33, un programa de proceso que se ejecuta cuando el evento A tiene lugar es el manejador de eventos `onEventA()`. Asimismo, con respecto al evento B y evento C, cuando el evento B tiene lugar, se ejecuta el manejador de eventos `onEventB()`. Cuando el evento C tiene lugar, se ejecuta el manejador de eventos `onEventC()`.

Dado que un manejador de eventos requerido cuando un evento tiene lugar se selecciona en el lado de sistema, el lado de creador de contenidos no necesita describir en el programa de guión un proceso que determina un evento que tiene lugar.

El modelo de evento según la segunda realización es más simple que el modelo de oyente de eventos según la primera realización descrita en la Figura 15 porque este último necesita procesos que registren eventos y borren eventos registrados. Por otro lado, en un modelo que requiere un método `processEvent()` independientemente de qué evento tiene lugar (es preciso ver la Figura 20), es necesario describir en el método `processEvent()` preprocesos que conozcan qué evento tiene lugar y seleccionen una rutina de procesos para un evento que tiene lugar. Dado que el método `processEvent()` debe implementarse por el lado de creador de contenidos, incluso si el modelo es simple, el lado de creador de contenidos necesita crear, de manera trabajosa, el disco. Además, cuando un evento tiene lugar, se requiere un programa de proceso (método) grande. Por consiguiente, se usaría un gran espacio de memoria y la velocidad de ejecución se convertiría en lenta. Por consiguiente, dado que el modelo según la segunda realización de la presente invención provee a cada evento un programa de proceso (manejador de eventos), con respecto a este punto, el modelo de la segunda realización es superior al de la primera realización.

2-4. Sobre el objeto de reproductor de películas 300

A continuación, se describirán las especificaciones externas del objeto de reproductor de películas 300. En general, un objeto definido en un lenguaje correspondiente a la especificación de lenguaje de guión ECMA tiene una propiedad y un método. El objeto de reproductor de películas 300 según la segunda realización de la presente invención tiene una propiedad y un método como se describe en la Figura 31 y Figura 32. Una propiedad puede leerse y escribirse directamente designando el nombre de objeto predeterminado y el nombre de propiedad. Además, con el método `setXXX()` (donde "XXX" representa un nombre de propiedad designado) que establece un valor de propiedad y el método `getXXX()` que lee un valor de propiedad, una propiedad de otro objeto puede leerse y escribirse por dichos métodos.

Para las propiedades y métodos del reproductor de películas 300 según la segunda realización de la presente invención, pueden aplicarse las mismas especificaciones que aquellas según la primera realización descrita en la Figura 7 a la Figura 12. En otras palabras, según la segunda realización de la presente invención, puede pensarse que las especificaciones (API) entre el programa de guión de la capa de guión 302 y el objeto de reproductor de películas 300, a saber el método 313 descrito en la Figura 30, son la misma clase definida en Java de la primera realización de la presente invención.

Como se describe en la Figura 30, la entrada de usuario 310 se recibe primero por la plataforma de implementación nativa 301. En otras palabras, la plataforma de implementación nativa 301 recibe la entrada de tecla como la entrada de usuario 310 del usuario. La plataforma de implementación nativa 301 determina si la entrada de usuario 310 es un comando para el reproductor de películas 300 o un evento para el programa de guión de la capa de guión 302. Según el resultado determinado, la plataforma de implementación nativa 301 genera el comando de control 311 o el

evento de tecla 314 e informa a la capa superior correspondiente (el reproductor de películas 300 o la capa de guión 302) sobre ello.

5 La entrada de usuario 310 es una entrada de tecla que se describe en la Figura 24, Figura 25A y Figura 25B. Los roles de las funciones correspondientes a las entradas de tecla que se muestran en la Figura 24 son diferentes de las que se muestran en la Figura 25A y Figura 25B. Por consiguiente, la plataforma de implementación nativa 301 necesita seleccionar un destino al que se informa de la entrada de usuario.

10 Las entradas de tecla que se muestran en la Figura 24 hacen que el reproductor de películas 300 reproduzca datos de vídeo, datos de audio y datos de subtítulos. Cuando la plataforma de implementación nativa 301 recibe una entrada de tecla que se muestra en la Figura 24 como la entrada de usuario 310, la plataforma de implementación nativa 301 convierte la entrada de tecla recibida en un comando descrito en la Figura 21A, Figura 21B, Figura 21C y Figura 21D. Se informa al reproductor de películas 300 sobre el comando convertido como el comando de control 311. En otras palabras, los comandos de entrada de usuario descritos en la Figura 21A, Figura 21B, Figura 21C y Figura 21D corresponden al comando de control 311 que se muestra en la Figura 30.

15 Por otro lado, las entradas de tecla que se muestran en la Figura 25A y Figura 25B son la entrada de usuario 310 para la GUI. Cuando la plataforma de implementación nativa 301 recibe una entrada de tecla que se muestra en la Figura 25A y Figura 25B como la entrada de usuario 310, la plataforma de implementación nativa 301 convierte la entrada de tecla recibida en un evento de tecla 314 e informa a la capa de guión 302 sobre el evento de tecla 314.

20 La Figura 25A y Figura 25B también muestran entradas de tecla con respecto a selecciones de flujo que son la tecla VK_ANGLE, tecla VK_PRESENTATION_GRAPHICS, y tecla VK_AUDIO. Dichas entradas de tecla se logran por métodos de un programa de guión para el reproductor de películas 300. Por consiguiente, son entradas de tecla sobre las cuales debe informarse a la capa de guión 302.

2-5. Sobre el manejo de eventos por el objeto de reproductor de películas 300

25 Según la primera realización que usa el lenguaje Java, el objeto BDBasicPlayer tiene los eventos que se muestran en la Figura 14A, Figura 14B, y Figura 14C. Sobre la base del modelo de oyente de eventos que se muestra en la Figura 15, los oyentes se registran y eliminan del registro con los métodos que se muestran en la Figura 16A y Figura 16B. En el modelo de oyente de eventos según la primera realización, mientras se está ejecutando un programa, un oyente puede registrarse y eliminarse del registro de manera dinámica. Un proceso de respuesta a un evento puede cambiarse mientras se está ejecutando un programa. Por consiguiente, la programación que tiene un grado más alto de libertad y una función avanzada superior puede lograrse en comparación con la técnica relacionada.

30 Por otro lado, en el lenguaje de guión basado en el guión ECMA según la segunda realización de la presente invención, un programa de proceso que se ha preparado correspondiente a un evento que tiene lugar se requiere sin necesidad de registrar y eliminar del registro a los oyentes. Una interfaz correspondiente a un oyente según la primera realización son los manejadores de eventos del reproductor de películas 300. En un modelo de manejador de eventos, los métodos con respecto al registro de oyentes que se muestra en la Figura 16A y Figura 16B no son necesarios.

40 Según la segunda realización de la presente invención, los ejemplos de manejadores de eventos en la Figura 34A y Figura 34B se proveen según los eventos. Los nombres de los manejadores de eventos tienen el prefijo "on". Los ejemplos de los manejadores de eventos que se muestran en la Figura 34A y Figura 34B corresponden a eventos que se muestran en la Figura 14A, Figura 14B y Figura 14C de la primera realización.

El manejador de eventos onTimerFired() corresponde a un evento que tiene lugar cuando el valor del temporizador en cuenta regresiva del reproductor de películas 300 se convierte en "0" o cuando el valor del temporizador en cuenta ascendente se convierte en un valor predeterminado. Por consiguiente, un proceso puede lograrse en un tiempo predeterminado.

45 El manejador de eventos onPlayStopped() corresponde a un evento que tiene lugar cuando la reproducción se detiene en el reproductor de películas 300. El manejador de eventos onPlayStilled() corresponde a un evento que tiene lugar cuando la reproducción se pausa en el reproductor de películas 300. El manejador de eventos onPlayStarted() corresponde a un evento que tiene lugar cuando la reproducción se inicia en el reproductor de películas 300. El manejador de eventos onPlayRepeated() corresponde a un evento que tiene lugar cuando el inicio de datos predeterminados se detecta en la reproducción de repetición del reproductor de películas 300. Por consiguiente, un proceso correspondiente al estado del motor de descodificador 322 puede lograrse.

55 El manejador de eventos onSPDisplayStatusChanged() corresponde a un evento que tiene lugar cuando el estado de visualización o no visualización de un flujo de datos de subtítulos (subimagen) cambia en el reproductor de películas 300. El manejador de eventos onSelectedAudioChanged() corresponde a un evento que tiene lugar cuando un flujo de audio que se está reproduciendo cambia en el reproductor de películas 300. El manejador de eventos onVideoStopped() corresponde a un evento que tiene lugar cuando un flujo de vídeo que se está reproduciendo

cambia en el reproductor de películas 300. Por consiguiente, un proceso correspondiente a un cambio de un flujo que se está reproduciendo puede lograrse.

5 El manejador de eventos `onPlayListStarted()` corresponde a un evento que tiene lugar cuando el inicio de una lista de reproducción se detecta en el reproductor de películas 300. El manejador de eventos `onPlayListEnded()` corresponde a un evento que tiene lugar cuando el fin de una lista de reproducción se detecta en el reproductor de películas 300. El manejador de eventos `onPlayItemStarted()` corresponde a un evento que tiene lugar cuando el inicio de un elemento de reproducción se detecta en el reproductor de películas 300. El manejador de eventos `onPlayItemEnded()` corresponde a un evento que tiene lugar cuando el fin de un elemento de reproducción se detecta en el reproductor de películas 300. Por consiguiente, los procesos correspondientes al comienzo y fin de la reproducción para listas de reproducción y elementos de reproducción como, por ejemplo, una reproducción continua para listas de reproducción y elementos de reproducción, pueden lograrse.

15 El manejador de eventos `onMarkEncountered()` corresponde a un evento que tiene lugar cuando una marca se detecta desde una lista de reproducción mientras la lista de reproducción se está reproduciendo en el reproductor de películas 300. Cuando una marca que representa un tiempo se preestablece para una lista de reproducción, un proceso de derivación para una lista de reproducción que se está reproduciendo puede lograrse.

Los contenidos de dichos manejadores de eventos se describen mediante el uso de un lenguaje descriptivo por el lado de creador de contenido. Por consiguiente, cuando un evento tiene lugar, una función que el lado de proveedor de contenido pretende puede lograrse.

REIVINDICACIONES

1. Un aparato de reproducción para reproducir datos de contenido jerárquicamente grabados en un medio de grabación con forma de disco, que comprende:
- 5 un objeto de reproductor (30) definido por una clase java para controlar un proceso de reproducción para datos de contenido jerárquicamente grabados en el medio de grabación con forma de disco y para generar un evento (312) correspondiente a un cambio predeterminado;
- medios de entrada de usuario para recibir una entrada de usuario e informar a dicho objeto de reproductor sobre la entrada de usuario, en donde el cambio predeterminado se basa en una entrada para dichos medios de entrada de usuario; y
- 10 un objeto de programa java (31) formado por encima de dicho objeto de reproductor, en donde el objeto de programa java es un programa superordinado con respecto al objeto de reproductor y controla el objeto de reproductor a través de una interfaz de programación de aplicaciones (32),
- en donde el objeto de programa java se configura para preparar programas que se ejecutan cuando se informa al objeto de programa java sobre la ocurrencia de eventos;
- 15 en donde un programa lleva a cabo un proceso correspondiente al evento que tiene lugar; y
- en donde dicho objeto de reproductor se configura para informar al objeto de programa java sobre la ocurrencia de dicho evento, y para controlar el proceso de reproducción para datos de contenido jerárquicamente grabados en el medio de grabación con forma de disco con dicho proceso correspondiente al evento.
2. El aparato de reproducción según se establece en la reivindicación 1,
- 20 en donde dicho objeto de programa java se reproduce desde el medio de grabación.
3. El aparato de reproducción según se establece en la reivindicación 1,
- en donde cuando dicho objeto de programa java registra un proceso correspondiente al evento que tiene lugar para dicho objeto de reproductor, dicho objeto de reproductor ejecuta el proceso correspondiente al evento que tiene lugar.
- 25 4. El aparato de reproducción según se establece en la reivindicación 3,
- en donde mientras los datos de contenido se están reproduciendo, el proceso puede registrarse.
5. El aparato de reproducción según se establece en la reivindicación 3,
- en donde mientras los datos de contenido se están reproduciendo, el proceso que se ha registrado puede eliminarse del registro.
- 30 6. El aparato de reproducción según se establece en la reivindicación 1,
- en donde el cambio predeterminado es un cambio que tiene lugar mientras los datos de contenido se están reproduciendo.
7. El aparato de reproducción según se establece en la reivindicación 6,
- en donde el cambio predeterminado es la detección de una marca que representa un tiempo que se establece en los datos de contenido.
- 35 8. El aparato de reproducción según se establece en la reivindicación 1,
- en donde una tecla virtual se asigna a una tecla física de dichos medios de entrada de usuario por dicho objeto de programa Java, el cambio predeterminado siendo una entrada para la tecla virtual.
9. El aparato de reproducción según se establece en la reivindicación 1,
- 40 en donde el cambio predeterminado es un cambio de estado de un reproductor.
10. El aparato de reproducción según se establece en la reivindicación 1,
- en donde uno o múltiples eventos tienen lugar según el cambio basado en el evento.
11. El aparato de reproducción según se establece en la reivindicación 1,

en donde el proceso de reproducción incluye un control de reproducción que reproduce los datos de contenido del medio de grabación con forma de disco y un proceso para los datos de contenido reproducidos desde el medio de grabación con forma de disco.

- 5 12. Un método de reproducción para reproducir datos de contenido jerárquicamente grabados en un medio de grabación con forma de disco, que comprende las etapas de:

medios de entrada de usuario que reciben una entrada de usuario e informan a un objeto de reproductor sobre la entrada de usuario;

- 10 el objeto de reproductor definido por una clase java y que controla un proceso de reproducción para datos de contenido jerárquicamente grabados en el medio de grabación con forma de disco y genera un evento correspondiente a un cambio predeterminado, en donde el cambio predeterminado se basa en dicha entrada de usuario para los medios de entrada de usuario;

- 15 dicho objeto de reproductor informa a un objeto de programa java formado por encima de dicho objeto de reproductor sobre la ocurrencia de dicho evento, en donde el objeto de programa java es un programa superordinado con respecto al objeto de reproductor y controla el objeto de reproductor a través de una interfaz de programación de aplicaciones;

el objeto de programa java que prepara programas que se ejecutan cuando se informa al objeto de programa java sobre la ocurrencia de eventos;

en donde un programa lleva a cabo un proceso correspondiente al evento; y

- 20 dicho objeto de reproductor informa al objeto de programa java sobre la ocurrencia de dicho evento, y controla el proceso de reproducción para datos de contenido jerárquicamente grabados en el medio de grabación con forma de disco con dicho proceso correspondiente al evento.

13. Un programa de reproducción que hace que un dispositivo de ordenador ejecute el método de reproducción de la reivindicación 12 para reproducir los datos de contenido jerárquicamente grabados en un medio de grabación con forma de disco.

- 25 14. Un medio de grabación legible por ordenador en el cual el programa de reproducción de la reivindicación 13 se ha grabado.

Fig. 1

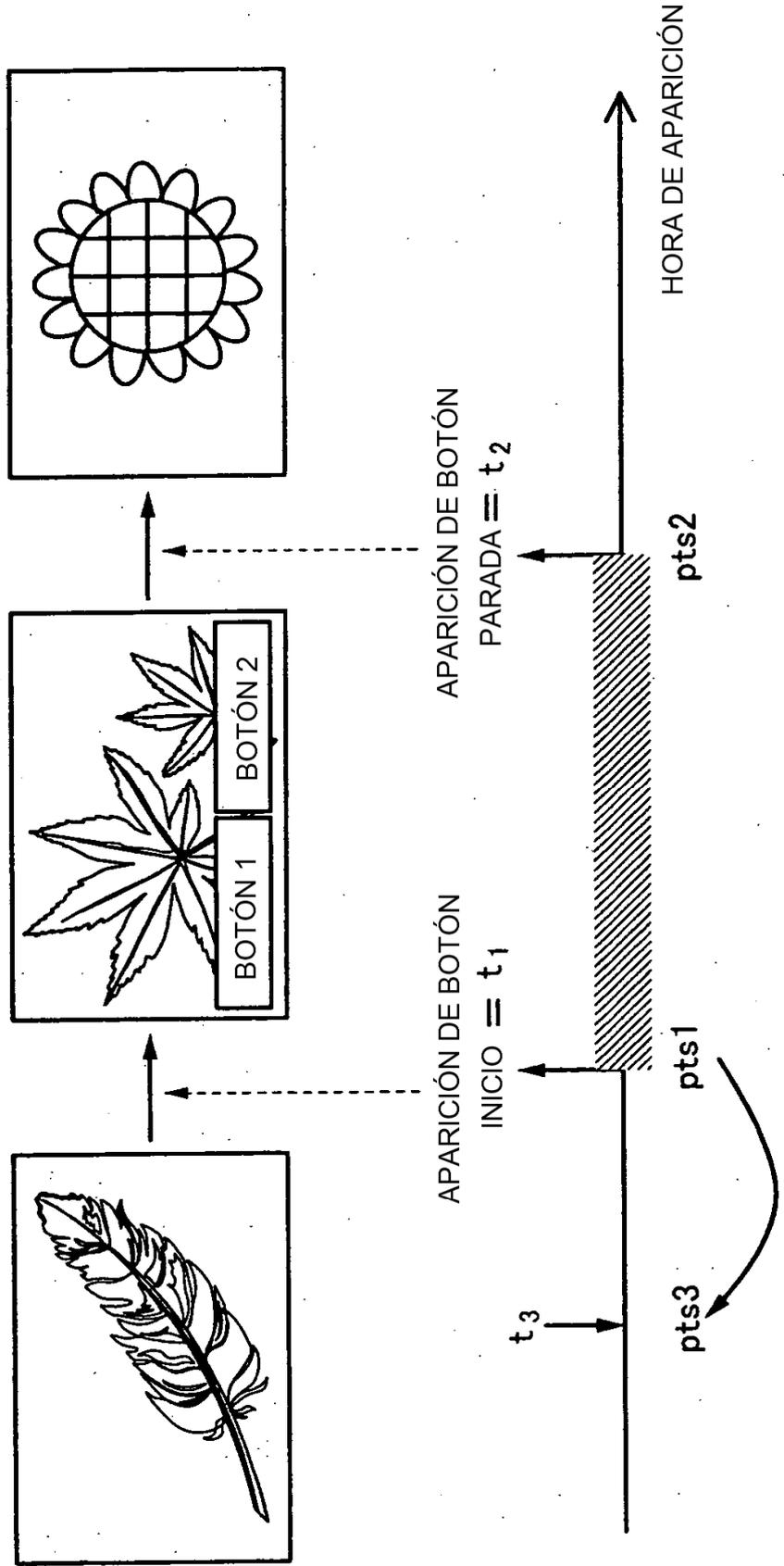


Fig. 3A

Fig. 3

Fig. 3A
Fig. 3B

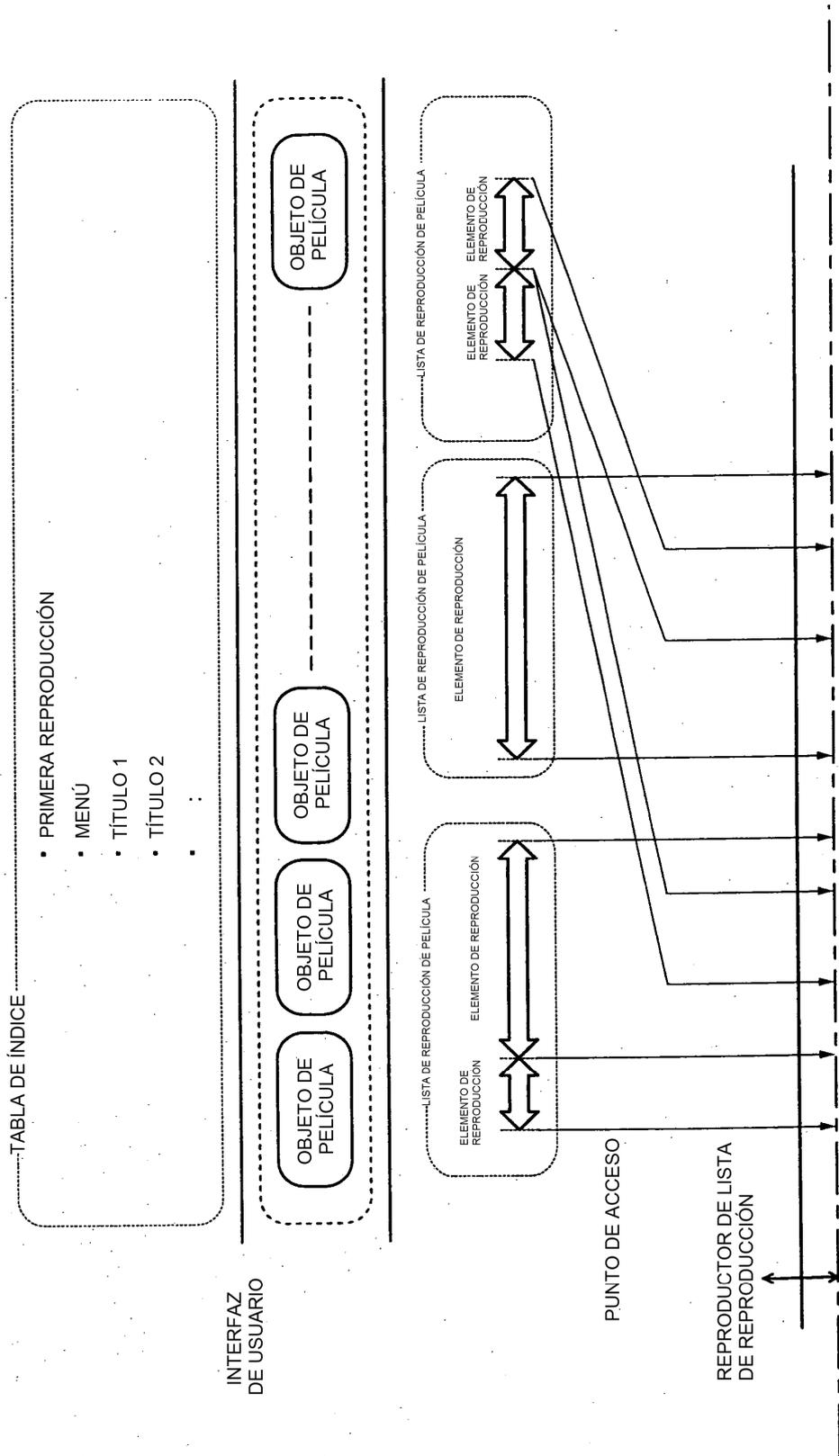


Fig. 3B

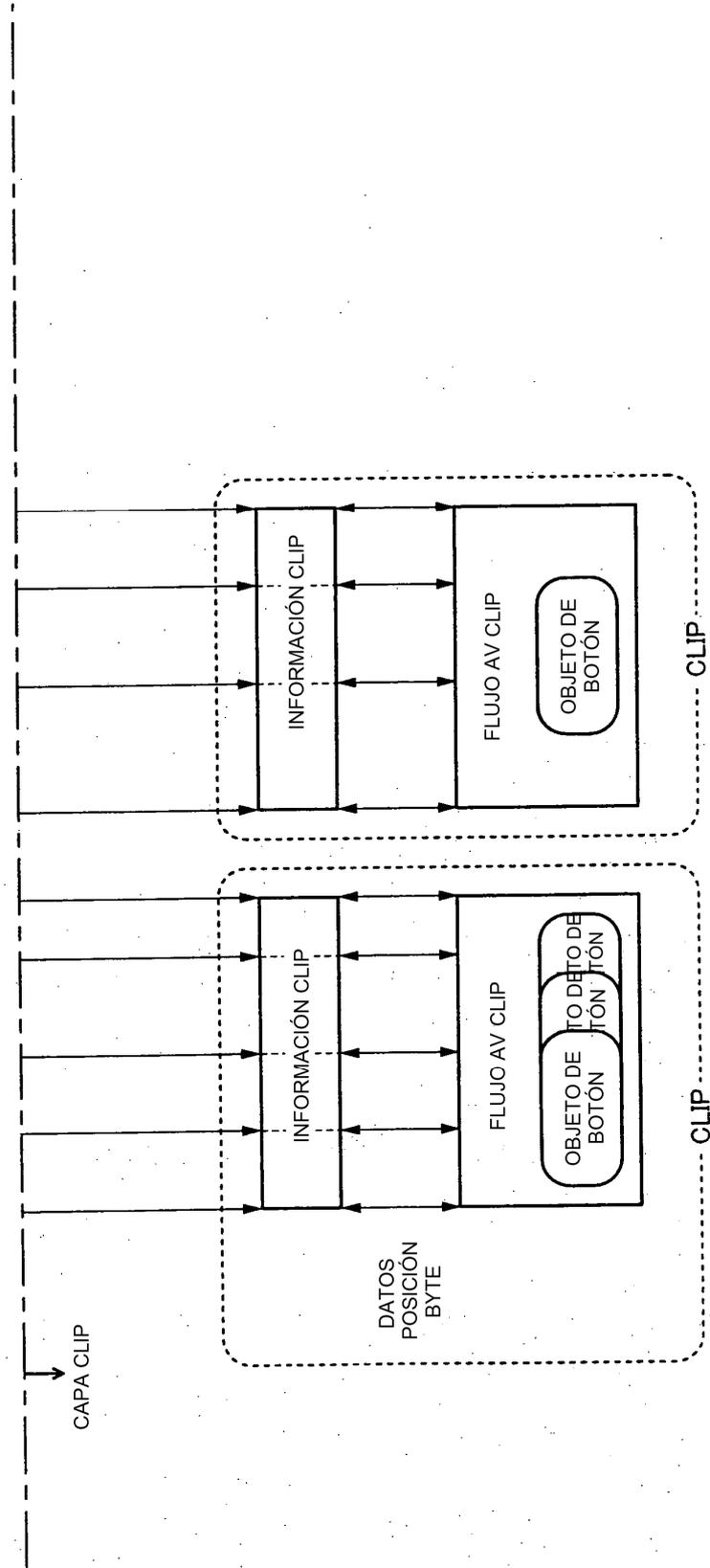


Fig. 4

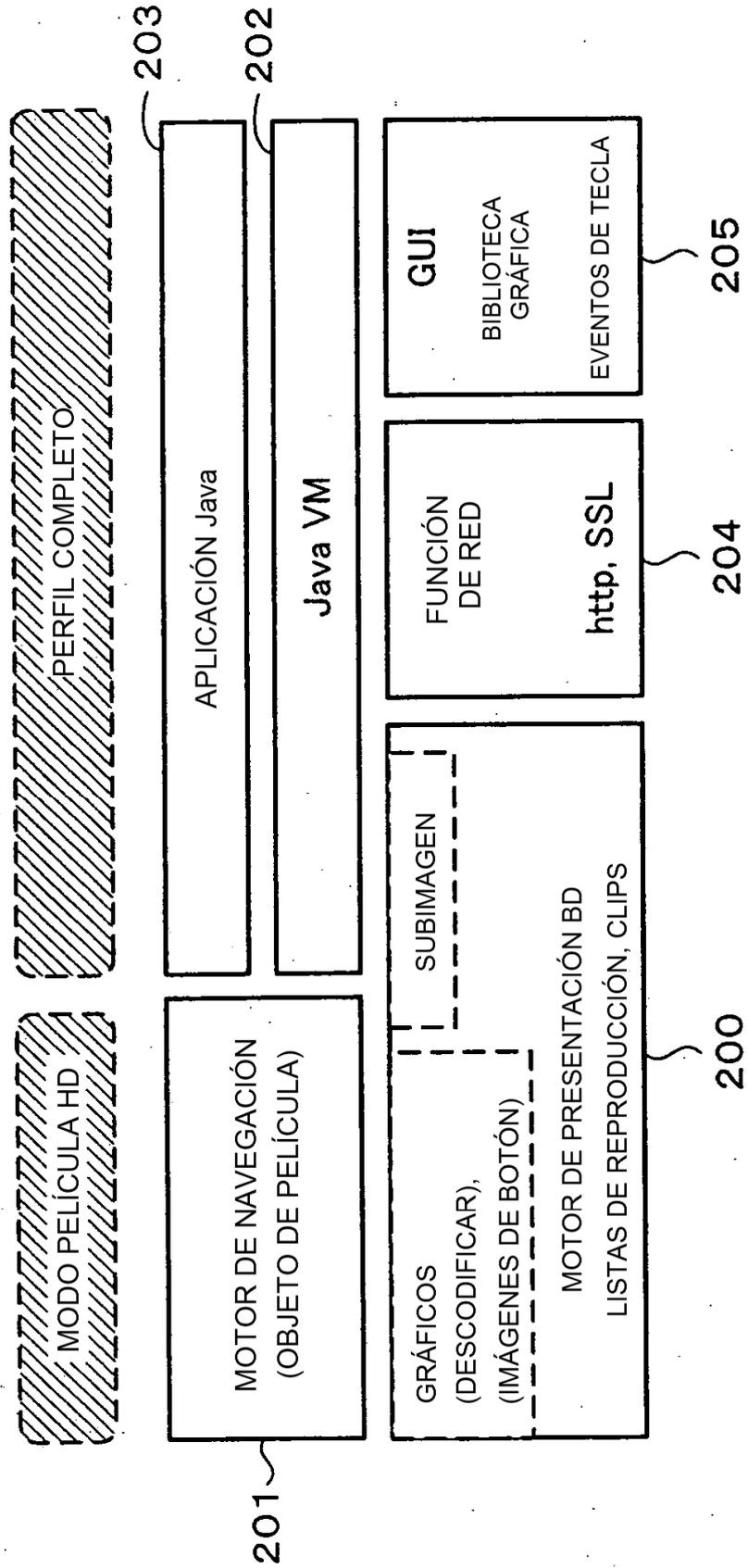


Fig. 5

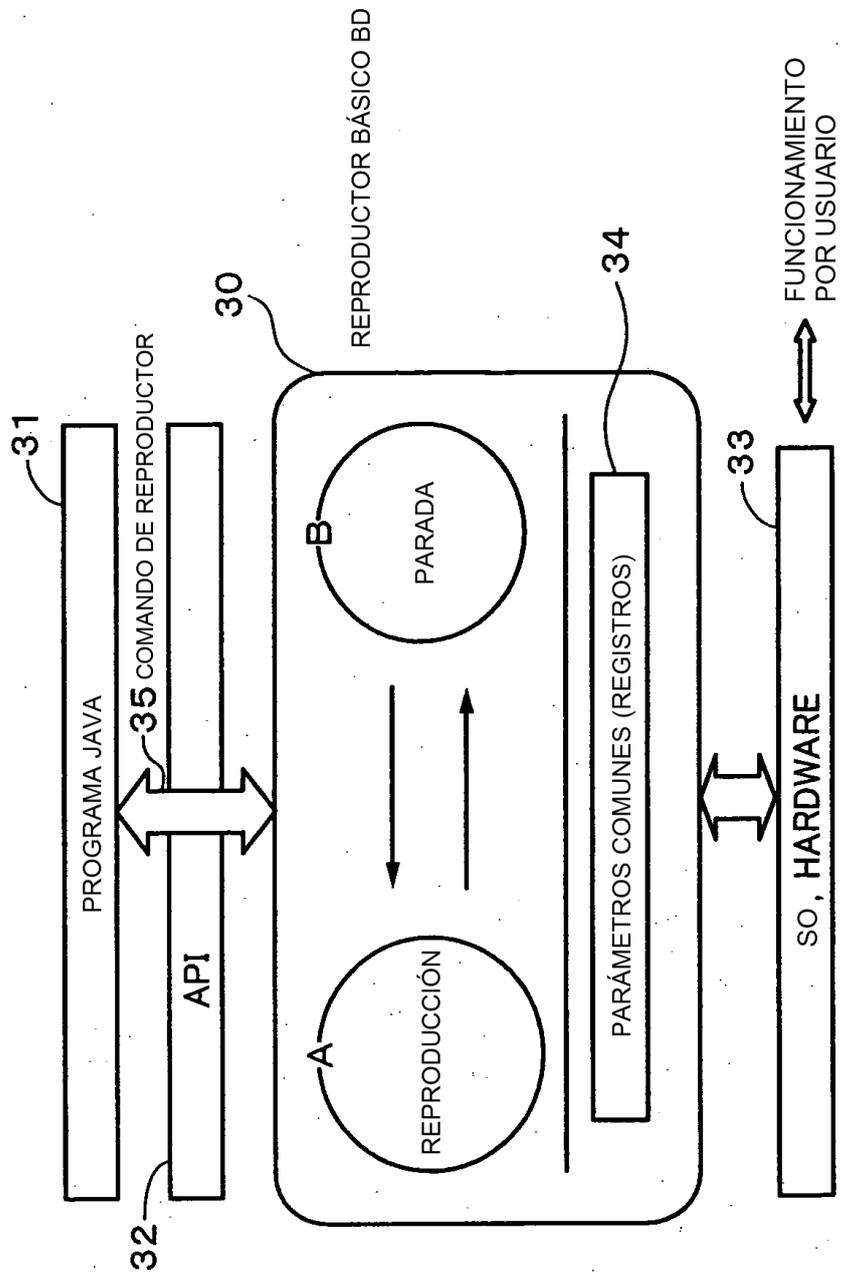


Fig. 6

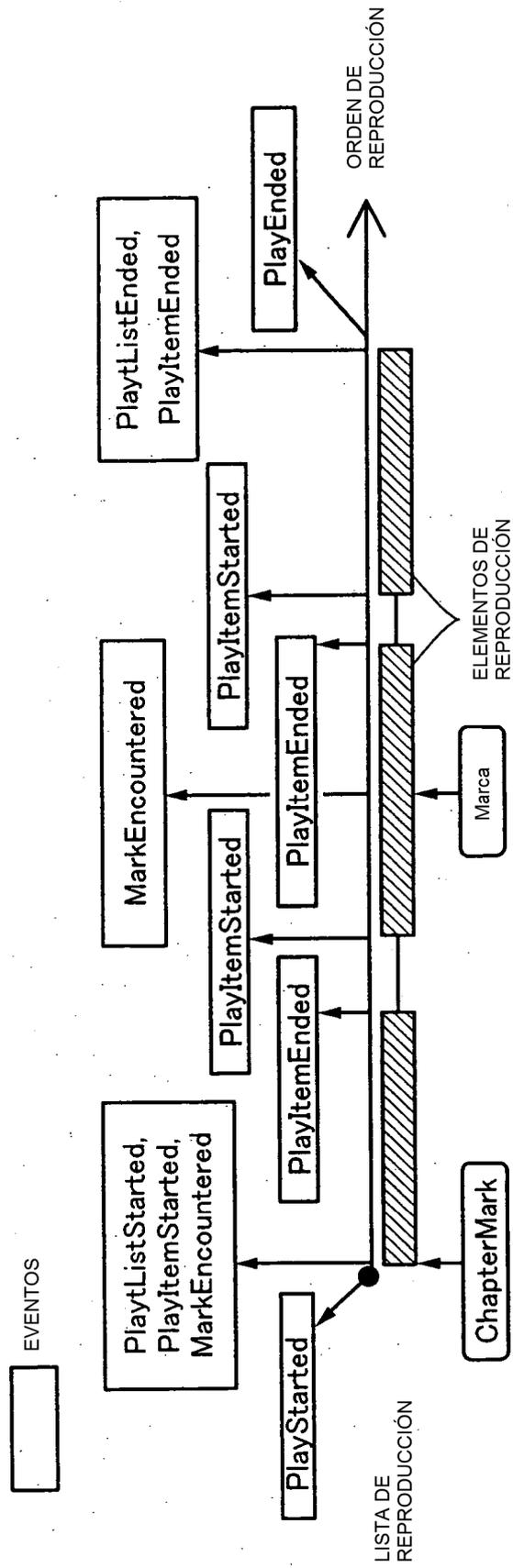


Fig. 7

MÉTODOS USADOS PARA OBTENER INFORMACIÓN DE ESTABLECIMIENTO DE REPRODUCTOR	
<code>getPlayerSpeed()</code>	OBTIENE VELOCIDAD DE REPRODUCCIÓN.
<code>getPlayerDirection()</code>	OBTIENE DIRECCIÓN DE REPRODUCCIÓN.
<code>getDescriptionLanguage()</code>	OBTIENE IDIOMA DE VISUALIZACIÓN DE REPRODUCTOR.
<code>getPlayListNumber()</code>	OBTIENE NÚMERO DE LISTA DE REPRODUCCIÓN QUE SE ESTÁ REPRODUCIENDO ACTUALMENTE.
<code>getChapterNumber()</code>	OBTIENE NÚMERO DE CAPÍTULO QUE SE ESTÁ REPRODUCIENDO ACTUALMENTE.
<code>getPlayerSupport()</code>	OBTIENE VERSIÓN Y FUNCIONES DEL REPRODUCTOR.

Fig. 8A

Fig. 8

Fig. 8A

Fig. 8B

Fig. 8C

MÉTODOS USADOS PARA REPRODUCIR LISTAS DE REPRODUCCIÓN	
<p><code>playPlaylist(int playListNumber, long [playListTime])</code></p>	<p>INICIA REPRODUCCIÓN DE LISTA DE REPRODUCCIÓN DESIGNADA POR <code>playListNumber</code>. <code>playListTime</code> NO ES ESENCIAL. DESIGNA TIEMPO TRANSCURRIDO EN LISTA DE REPRODUCCIÓN. INICIA LA REPRODUCCIÓN DESDE <code>playListTime</code>.</p>
<p><code>playPlaylist(String playListURL, long [playListTime])</code></p>	<p>INICIA REPRODUCCIÓN DE LISTA DE REPRODUCCIÓN DESIGNADA POR <code>playListURL</code>. <code>playListTime</code> NO ES ESENCIAL. DESIGNA TIEMPO TRANSCURRIDO EN LISTA DE REPRODUCCIÓN. INICIA REPRODUCCIÓN DE LISTA DE REPRODUCCIÓN DESDE <code>playListTime</code>.</p>

Fig. 8B

<p><code>playPlayItem(int playListNumber, int playItemNumber)</code></p>	<p>INICIA REPRODUCCIÓN DE LISTA DE REPRODUCCIÓN DESIGNADA DESDE ELEMENTO DE REPRODUCCIÓN DESIGNADO. playItemNumber ES PlayItem_id COMIENZA DESDE 0. CUANDO REPRODUCCIÓN COMIENZA DESDE INICIO DE LISTA DE PREPRODUCCIÓN, playItemNumber ES 0.</p>
<p><code>playPlayItem(String playListURL, int playItemNumber)</code></p>	<p>INICIA REPRODUCCIÓN DE LISTA DE REPRODUCCIÓN DESIGNADA DESDE ELEMENTO DE REPRODUCCIÓN DESIGNADO. playItemNumber ES PlayItem_id COMIENZA DESDE 0. CUANDO REPRODUCCIÓN COMIENZA DESDE INICIO DE LISTA DE REPRODUCCIÓN, playItemNumber ES 0.</p>
<p><code>playChapterMark (int playListNumber, int chapterNumber)</code></p>	<p>REPRODUCE LISTA DE REPRODUCCIÓN DESIGNADA POR playListNumber FROM CAPITULO DESIGNADO POR chapterNumber. playListNumber PUEDE SER playListURL. LISTA DE REPRODUCCIÓN SIEMPRE COMIENZA CON MARCA DE CAPÍTULO.</p>

Fig. 8C

<p>6.</p> <p><code>moveAndPlay(string position, object)</code></p> <p>posición = ("prev" "next" "top" "tail")</p> <p>objeto = ("PlayItem" "Chapter")</p>	<p>DESIGNA SALTO EN LISTA DE REPRODUCCIÓN.</p> <p>next DESIGNA SALTO DE POSICIÓN ACTUALMENTE REPRODUCIDA A SIGUIENTE ELEMENTO DE REPRODUCCIÓN O CAPITULO E INICIA SU REPRODUCCIÓN.</p> <p>top DESIGNA SALTO AL COMIENZO DE LA LISTA DE REPRODUCCIÓN.</p> <p>tail DESIGNA SALTO AL FINAL DE LA LISTA DE REPRODUCCIÓN.</p> <p>prev INICIA REPRODUCCIÓN DESDE COMIENZO DE ELEMENTO DE REPRODUCCIÓN ACTUALMENTE REPRODUCIDO.</p>
<p><code>stop()</code></p>	<p>PARA LA REPRODUCCIÓN.</p> <p>NO SE MANTIENE VALOR DE REGISTRO ESTÁNDAR.</p>
<p><code>playSoundEffect(int sound_id)</code></p>	<p>REPRODUCE EFECTO DE SONIDO SELECCIONADO.</p>

Fig. 9

MÉTODOS USADOS PARA FLUJOS DE VIDEO	
<code>getVideoStreamAvailability()</code>	OBTIENE ESTADO QUE REPRESENTA SI SE CONTIENE O NO FLUJO DE VIDEO DESIGNADO.
<code>setVideoStreamNumber()</code>	DESIGNA FLUJO DE VIDEO A DESCODIFICARSE.
<code>getVideoStreamNumber()</code>	OBTIENE NÚMERO DE FLUJO DE VIDEO QUE SE HA SELECCIONADO.
<code>getVideoStreamAttribute()</code>	OBTIENE ATRIBUTOS DE FLUJO DE VIDEO (SISTEMA DE CODIFICACIÓN, RESOLUCIÓN, RELACIÓN DE ASPECTO, MODO DE VISUALIZACIÓN EN CASO DE RELACIÓN DE ASPECTO DE 4 : 3, Y SUBTÍTULO CODIFICADO).
<code>setAngleNumber()</code>	DESIGNA NÚMERO DE ÁNGULO.
<code>getAngleNumber()</code>	OBTIENE NÚMERO DE ÁNGULO QUE SE HA SELECCIONADO.
<code>getMaxVideoStreams()</code>	OBTIENE NÚMERO DE FLUJOS DE VIDEO QUE PUEDEN SELECCIONARSE.

Fig. 10

MÉTODOS USADOS PARA FLUJOS DE AUDIO	
<code>getAudioStreamAvailability()</code>	OBTIENE ESTADO QUE REPRESENTA SI SE CONTIENE O NO EL FLUJO DESIGNADO.
<code>getAudioStreamLanguage()</code>	OBTIENE INFORMACIÓN SOBRE IDIOMA USADO EN FLUJO DE AUDIO DESIGNADO.
<code>setAudioStreamNumber()</code>	DESIGNA FLUJO DE AUDIO A REPRODUCIRSE.
<code>getAudioStreamNumber()</code>	OBTIENE NÚMERO DE FLUJO DE AUDIO QUE SE ESTÁ REPRODUCIENDO.
<code>getAudioStreamAttribute()</code>	OBTIENE ATRIBUTOS DE FLUJO DE AUDIO (SISTEMA DE CODIFICACIÓN, NÚMERO DE chs, Q, Y fs).
<code>getMaxAudioStreams()</code>	OBTIENE NÚMERO DE FLUJOS DE AUDIO QUE PUEDEN SELECCIONARSE.

Fig. 11

MÉTODOS USADOS PARA FLUJOS DE SUBIMÁGENES (SP)	
getSPStreamAvailability()	OBTIENE ESTADO QUE REPRESENTA SI SE CONTIENE O NO EL FLUJO SP DESIGNADO.
getSPStreamLanguage()	OBTIENE IDIOMA USADO EN FLUJO SP DESIGNADO.
getSPDisplayStatus()	OBTIENE ESTADO DE VISUALIZACIÓN DE SP (VISUALIZADO O NO VISUALIZADO).
setSPDisplayStatus()	ESTABLECE ESTADO DE VISUALIZACIÓN DE SP (VISUALIZARSE O NO VISUALIZARSE).
getSPStreamAttribute()	OBTIENE ATRIBUTOS DE SP. ATRIBUTOS DE SP SON, POR EJEMPLO, INFORMACIÓN SOBRE RESOLUCIONES, RELACIÓN DE ASPECTO DE TIPO 4:3, TIPO DE ANCHO, ETC.

Fig. 12

MÉTODOS USADOS PARA TEMPORIZADORES INTERNOS	
<code>sleep()</code>	PARA EL PROCESO DURANTE MILLISEGUNDOS DESIGNADOS.
<code>setTimeout()</code>	EJECUTA FUNCIÓN O PROCESO DESPUÉS DE QUE HUBIERAN TRANSCURRIDO LOS MILLISEGUNDOS DESIGNADOS.
<code>setInterval()</code>	EJECUTA PROCESO A INTERVALOS DE MILLISEGUNDOS DESIGNADOS.
<code>clearTimer()</code>	PARA EL PROCESO DEL TEMPORIZADOR QUE TIENE ID DE TEMPORIZADOR REGISTRADO DESIGNADO.
<code>pauseTimer()</code>	PAUSA EL TEMPORIZADOR QUE TIENE ID DE TEMPORIZADOR REGISTRADO DESIGNADO.
<code>resumeTimer()</code>	REANUDA EL TEMPORIZADOR PAUSADO QUE TIENE ID DE TEMPORIZADOR REGISTRADO DESIGNADO.

Fig. 13

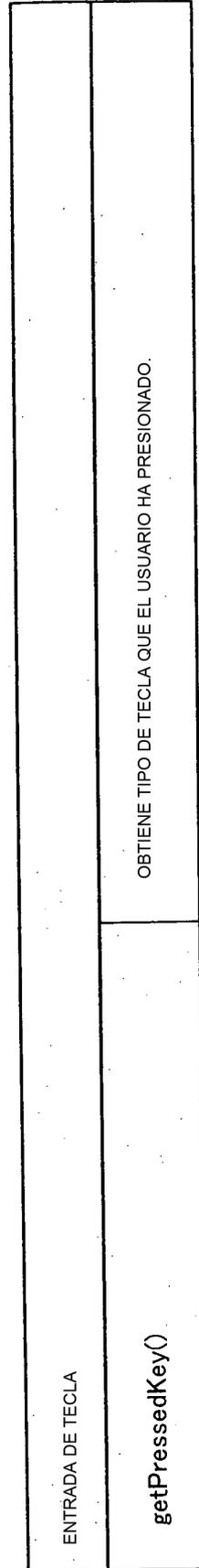
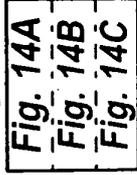


Fig. 14A

Fig. 14



NOMBRE DE EVENTO	DESCRIPCIÓN
EVENTOS QUE TIENEN LUGAR EN EL CONTENIDO QUE SE ESTÁ REPRODUCIENDO	
MarkEncounteredEvent	SE HA DETECTADO UNA MARCA MIENTRAS EL CONTENIDO SE REPRODUCÍA. DICHO EVENTO SE USA CUANDO SE VISUALIZA LA PANTALLA DE GRÁFICOS.
ValidPeriodStartedEvent	PERÍODO VÁLIDO DE SELECCIÓN DE ENTRADA DE USUARIO INICIADO.
ValidPeriodEndedEvent	PERÍODO VÁLIDO DE ENTRADA DE USUARIO FINALIZADO. DICHO EVENTO SE USA CUANDO SE EJECUTA UN ENLACE DE MANERA FORZADA.

Fig. 14B

PlayRepeatedEvent	SE HA DETECTADO EL COMIENZO DE REGIÓN DE REPRODUCCIÓN DE REPETICIÓN.
VideoStreamChangedEvent	SE HA CAMBIADO EL FLUJO DE VÍDEO QUE SE ESTABA REPRODUCIENDO.
AudioStreamChangedEvent	SE HA CAMBIADO EL FLUJO DE AUDIO QUE SE ESTABA REPRODUCIENDO.
SubpictureStreamChangedEvent	SE HA CAMBIADO EL FLUJO DE SUBIMÁGENES QUE SE ESTABA REPRODUCIENDO.
StreamDisplayStatusChangedEvent	SE HA CAMBIADO EL ESTADO DE VISUALIZACIÓN/NO VISUALIZACIÓN DEL FLUJO.
PlaylistStartedEvent	SE HA DETECTADO EL INICIO DE LA LISTA DE REPRODUCCIÓN.
PlaylistEndedEvent	SE HA DETECTADO EL FINAL DE LA LISTA DE REPRODUCCIÓN.
PlayItemStartedEvent	SE HA DETECTADO EL INICIO DEL ELEMENTO DE REPRODUCCIÓN.
PlayItemEndedEvent	SE HA DETECTADO EL FINAL DEL ELEMENTO DE REPRODUCCIÓN.

Fig. 14C

EVENTO QUE TIENE LUGAR CORRESPONDIENTE A LA ENTRADA DE TECLA	
RmKeyEvent	SE HA PRESIONADO LA TECLA DE CONTROLADOR REMOTO. LA TECLA QUE SE HA PRESIONADO SE DETERMINA EN EL OYENTE DE EVENTOS CORRESPONDIENTE.
EVENTOS QUE TIENEN LUGAR CORRESPONDIENTES A CAMBIOS DE ESTADO DEL REPRODUCTOR	
PlayStartedEvent	SE HA INICIADO LA REPRODUCCIÓN.
PlayStoppedEvent	SE HA PARADO LA REPRODUCCIÓN.
PlayStilledEvent	SE HA PARALIZADO LA REPRODUCCIÓN.
StillReleasedEvent	EL ESTADO PARALIZADO SE HA LIBERADO.
PlayPausedEvent	EL USUARIO HA PAUSADO LA REPRODUCCIÓN.
PauseReleasedEvent	EL ESTADO PAUSADO SE HA LIBERADO.
	EL TEMPORIZADOR EN CUENTA REGRESIVA SE HA CONVERTIDO EN 0.
TimerFiredEvent	EL TEMPORIZADOR EN CUENTA ASCENDENTE SE HA CONVERTIDO EN UN VALOR PREDETERMINADO.

Fig. 15

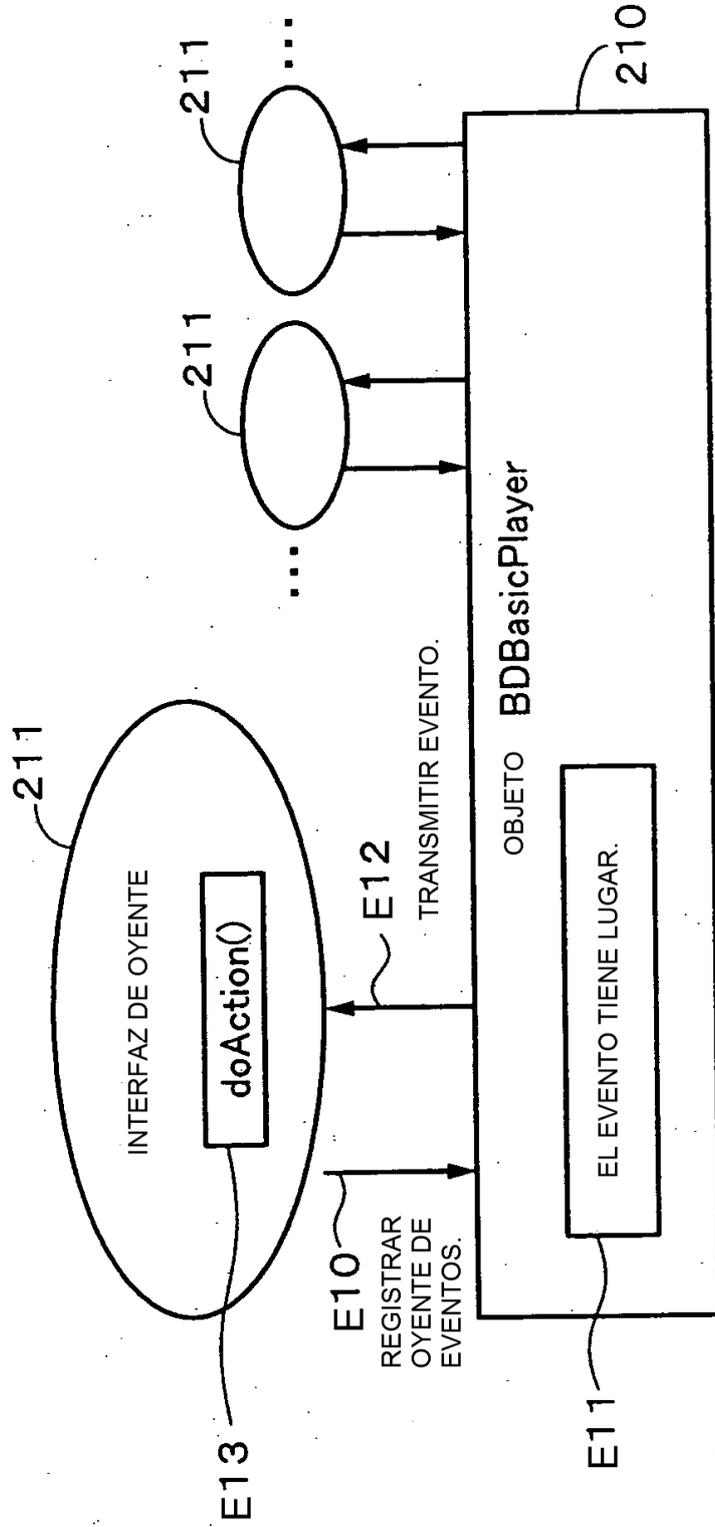


Fig. 16A

Fig. 16

Fig. 16A

Fig. 16B

MÉTODOS USADOS PARA REGISTRAR OYENTES	
addContentsListener(ContentsListener l)	REGISTROS HAN DESIGNADO OYENTE PARA BDBasicPlayer PARA QUE RECIBA ContentsEvent DE BDBasicPlayer .
addRmtKeyListener(RmtKeyListener l)	REGISTROS HAN DESIGNADO OYENTE PARA QUE RECIBA RmtKeyEvent DE BDBasicPlayer .
addPlayerStatusListener (PlayerStatusListener l)	REGISTRO HA DESIGNADO OYENTE PARA QUE RECIBA PlayerStatusEvent DE BDBasicPlayer .

Fig. 16B

removeContentsListener(ContentsListener l)	ELIMINAR OYENTE DESIGNADO PARA NO RECIBIR ContentsEvent DE BDBasicPlayer.
removeRmtKeyListener(RmtKeyListener l)	ELIMINAR OYENTE DESIGNADO PARA NO RECIBIR RMTKEYEVENT DE BDBasicPlayer.
removePlayerStatusListener (PlayerStatusListener l)	ELIMINAR OYENTE DESIGNADO PARA NO RECIBIR PlyerStatusVvent DE BDBasicPlayer.
getContentsListeners()	DEVUELVE MATRIZ DE TODOS LOS CountsListeners REGISTRADOS EN BDBasicPlayer.
getRmtkeyListeners()	DEVUELVE MATRIZ DE TODOS LOS RmtkeyListeners REGISTRADOS EN BDBasicPlayer.
getPlayerStatusListeners()	DEVUELVE MATRIZ DE TODOS LOS PlayerStausListeners REGISTRADOS EN BDBasicPlayer.

Fig. 17

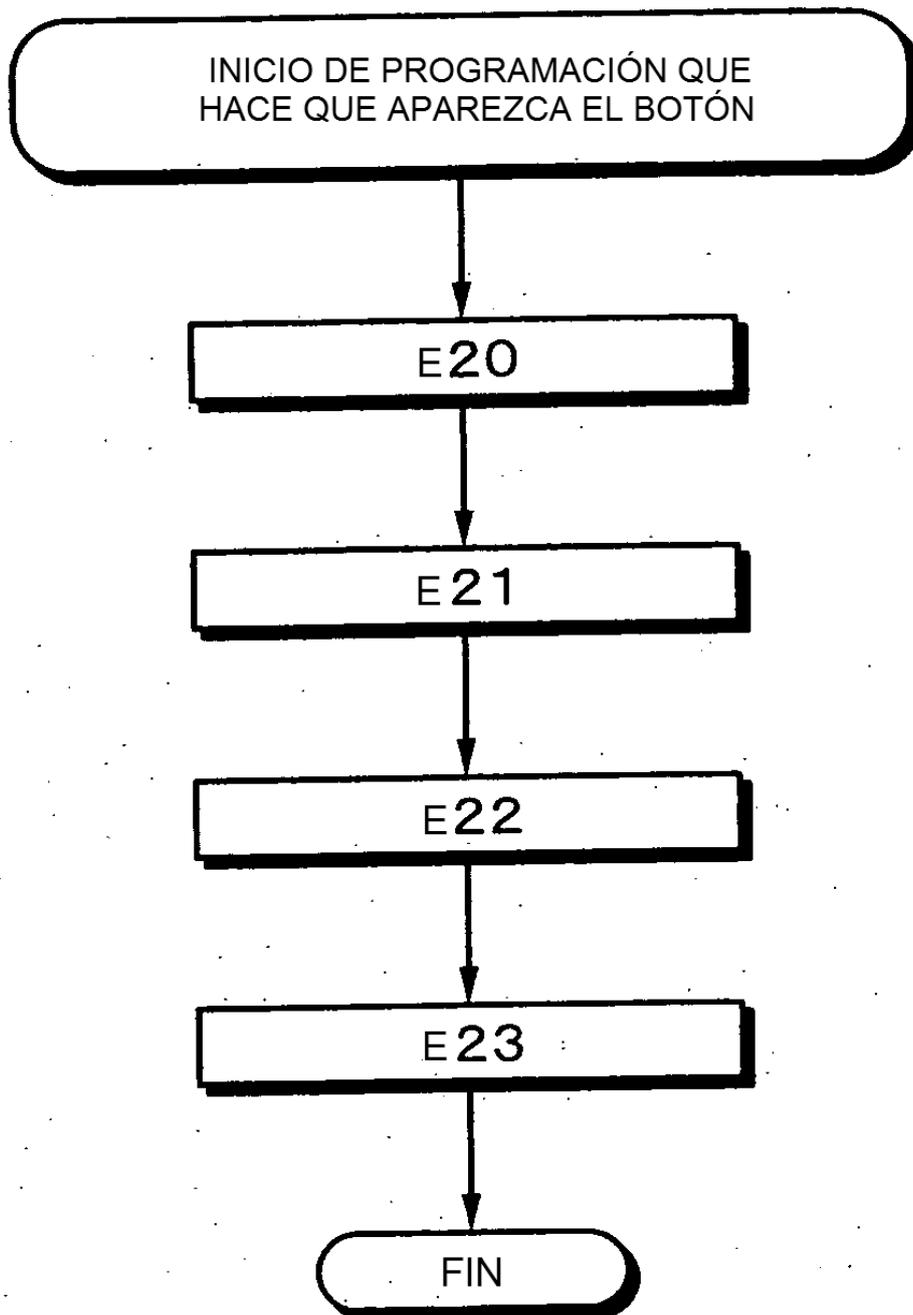
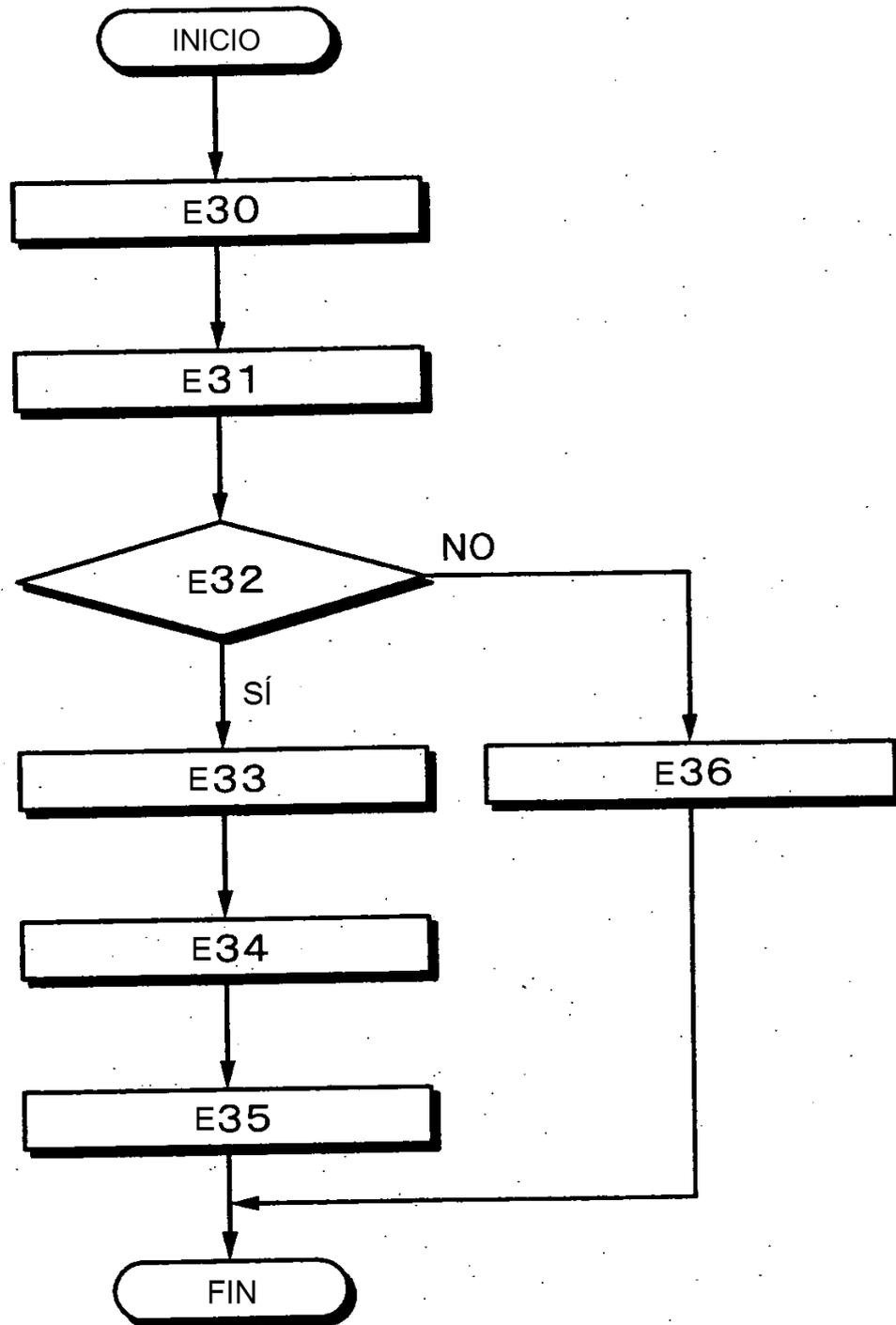


Fig. 18



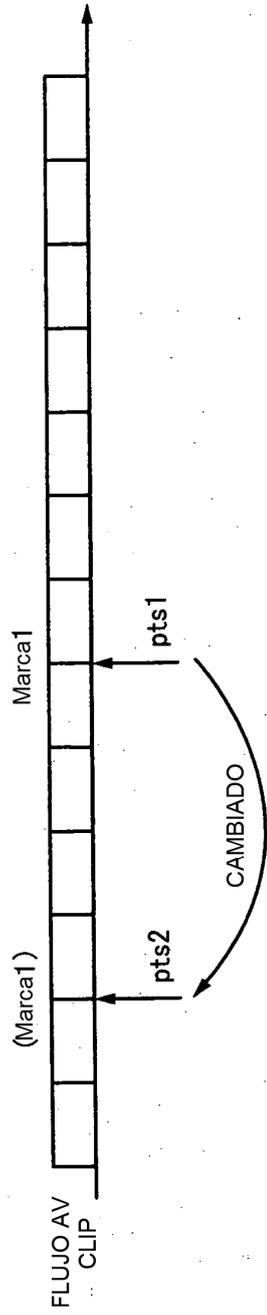


Fig. 19A

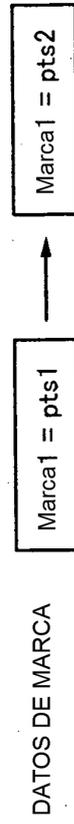


Fig. 19B

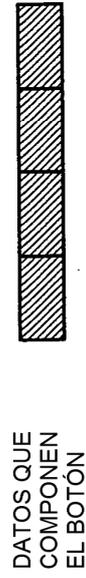


Fig. 19C

Fig. 20

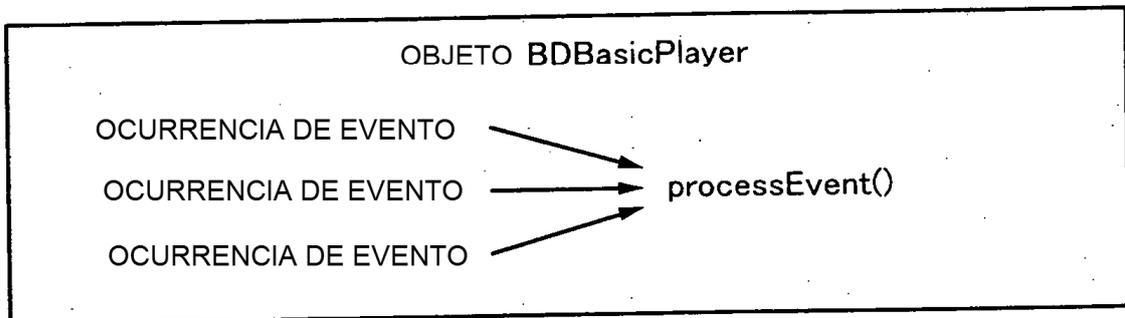


Fig. 21A

Fig. 21

Fig. 21A
Fig. 21B
Fig. 21C
Fig. 21D

COMANDOS DE ENTRADA DE USUARIO	DESCRIPCIÓN
<code>timeSearch(playlistNumber, Time)</code>	HACE QUE EL ESCENARIO QUE SE ESTÁ REPRODUCIENDO SE REPRODUZCA DESDE UN TIEMPO DESIGNADO.
<code>play(playlistNumber, chapterNumber)</code>	HACE QUE EL ESCENARIO DESIGNADO SE REPRODUZCA DESDE UN CAPÍTULO DESIGNADO.
<code>stop()</code>	HACE QUE LA REPRODUCCIÓN SE DETENGA.
<code>prev()</code>	HACE QUE LA REPRODUCCIÓN COMIENZE DESDE EL INICIO DEL CAPÍTULO PREVIO.
<code>top()</code>	HACE QUE LA REPRODUCCIÓN COMIENZE DESDE EL INICIO DEL CAPÍTULO ACTUAL.

Fig. 21B

<p><code>next()</code></p>	<p>HACE QUE LA REPRODUCCIÓN COMIENZE DESDE EL INICIO DEL CAPÍTULO SIGUIENTE.</p>
<p><code>forwardScan(speed)</code></p>	<p>HACE QUE LA REPRODUCCIÓN SE LLEVE A CABO EN LA DIRECCIÓN HACIA ADELANTE A LA VELOCIDAD DESIGNADA POR speed. speed OSCILA ENTRE 0,1 Y 360.</p>
<p><code>backwardScan(speed)</code></p>	<p>HACE QUE LA REPRODUCCIÓN SE LLEVE A CABO EN LA DIRECCIÓN HACIA ATRÁS A LA VELOCIDAD DESIGNADA POR speed.</p>
<p><code>moveCursor(direction, [level])</code> <code>direction=("upper" "lower" "left" "right")</code> <code>level="0"~"255"</code></p>	<p>HACE QUE EL CURSOR SE MUEVA EN LAS DIRECCIONES SUPERIOR, INFERIOR, IZQUIERDA Y DERECHA. level ES OPCIONAL. NORMALMENTE level NO SE USA O ES 0. CON UN VALOR INTERMEDIO (1 A 255), PUEDE REPRESENTARSE UN JOYSTICK ANALÓGICO O SIMILARES.</p>
<p><code>click(buttonNumber)</code></p>	<p>HACE QUE SE HAGA CLICK EN EL BOTÓN EN LA PANTALLA.</p>

Fig. 21C

focus(buttonNumber)	HACE QUE LA ATENCIÓN SE CENTRE EN EL BOTÓN EN LA PANTALLA.
stillOff()	HACE QUE LA REPRODUCCIÓN PARALIZADA POR EL REPRODUCTOR SE REANUDE. INTEGRADA CON pauseOFF .
pauseOn()	PAUSA. SOLO FUNCIONA CUANDO EL ESCENARIO SE REPRODUCE.
pauseOff()	HACE QUE LA REPRODUCCIÓN PAUSADA SE LIBERE.
LanguageSelect(languageCode)	DESIGNA IDIOMA DE VISUALIZACIÓN DEL MENÚ.

Fig. 21D

StreamChange(StreamNumber)	HACE QUE EL FLUJO QUE SE ESTÁ REPRODUCIENDO CAMBIE.
setEnabled(boolean)	HACE QUE EL FLUJO SE VISUALICE/NO SE VISUALICE.
angleChange(angleNumber)	HACE QUE EL ÁNGULO DE VISUALIZACIÓN CAMBIE.
parentalCountrySelect(countryCode)	HACE QUE EL CÓDIGO DE PAÍS DE NIVEL PARENTAL SE ESTABLEZCA.
videoPresentationModeChange(mode)	modo =(Ancho LB PanScan)

Fig. 22

RmtKeyEvent Clase	
RmtKeyEvent publico	(Fuente de objeto , int id, long when, int rmtKeyCode)
source	OBJETO DE FUENTE DE EVENTO. REPRESENTA BDBasicPlayer .
id	ENTERO QUE REPRESENTA TIPO DE EVENTO
when	ENTERO TIPO LARGO QUE DESIGNA TIEMPO EN QUE TIEMPO EL EVENTO TIENE LUGAR
rmtKeyCode	CÓDIGO TIPO ENTERO CORRESPONDIENTE A LA TECLA VIRTUAL DE ENTRADA O VK_UNDEFINED

Fig. 23

RESUMEN DE MÉTODOS	
getRmtKeyCode	DEVUELVE CÓDIGO DE TECLA CORRESPONDIENTE A LA TECLA QUE HA PROVOCADO QUE EL EVENTO TENGA LUGAR. público sin retorno
setRmtKeyCode	ESTABLECE CÓDIGO DE TECLA CORRESPONDIENTE A LA TECLA QUE HA PROVOCADO QUE EL EVENTO TENGA LUGAR. público sin retorno

Fig. 24

RESUMEN DE CAMPOS		
VK_POWER	TECLA DE ENCENDIDO /APAGADO	public static final int
VK_POWER_ON	TECLA DE ENCENDIDO	public static final int
VK_POWER_OFF	TECLA DE APAGADO	public static final int
VK_MENU	TECLA DE MENU	public static final int
VK_ENTER	INGRESAR	public static final int
VK_RETURN	RETORNAR	public static final int
VK_PLAY	REPRODUCIR	public static final int
VK_STOP	DETERNER	public static final int
VK_PAUSE	PAUSAR	public static final int
VK_FAST_FORWARD	AVANCE RÁPIDO	public static final int
VK_FAST_REVERSE	RETROCESO RÁPIDO	public static final int
VK_SLOW_FORWARD	AVANCE LENTO	public static final int
VK_SLOW_REVERSE	RETROCESO LENTO	public static final int
VK_STEP_FORWARD	AVANCE DE ETAPA	public static final int
VK_STEP_REVERSE	RETROCESO DE ETAPA	public static final int

Fig. 25A

Fig. 25

Fig. 25A
Fig. 25B

RESUMEN DE CAMPOS		
	SIGUIENTE	public static final int
VK_NEXT		
VK_PREVIOUS	PREVIO	public static final int
VK_UP	SUPERIOR	public static final int
VK_DOWN	INFERIOR	public static final int
VK_RIGHT	DERECHA	public static final int
VK_LEFT	IZQUIERDA	public static final int
VK_UP_RIGHT	SUPERIOR DERECHA	public static final int
VK_UP_LEFT	SUPERIOR IZQUIERDA	public static final int
VK_DOWN_RIGHT	INFERIOR DERECHA	public static final int
VK_DOWN_LEFT	INFERIOR IZQUIERDA	public static final int

Fig. 25B

VK_ANGLE	SELECCIÓN DE ÁNGULO	public static final int
VK_PRESENTATION_GRAPHICS	SELECCIÓN DE SUBTÍTULO	public static final int
VK_AUDIO	SELECCIÓN DE AUDIO	public static final int
VK_VIDEO_ASPECT	SELECCIÓN DE RELACIÓN DE ASPECTO DE VIDEO	public static final int
VK_COLORED_KEY_1	TECLA DE FUNCIÓN DE COLORES 1	public static final int
VK_COLORED_KEY_2	TECLA DE FUNCIÓN DE COLORES 2	public static final int
VK_COLORED_KEY_3	TECLA DE FUNCIÓN DE COLORES 3	public static final int
VK_COLORED_KEY_4	TECLA DE FUNCIÓN DE COLORES 4	public static final int
VK_COLORED_KEY_5	TECLA DE FUNCIÓN DE COLORES 5	public static final int
VK_COLORED_KEY_6	TECLA DE FUNCIÓN DE COLORES 6	public static final int
VK_UNDEFINED		public static final int

Fig. 26

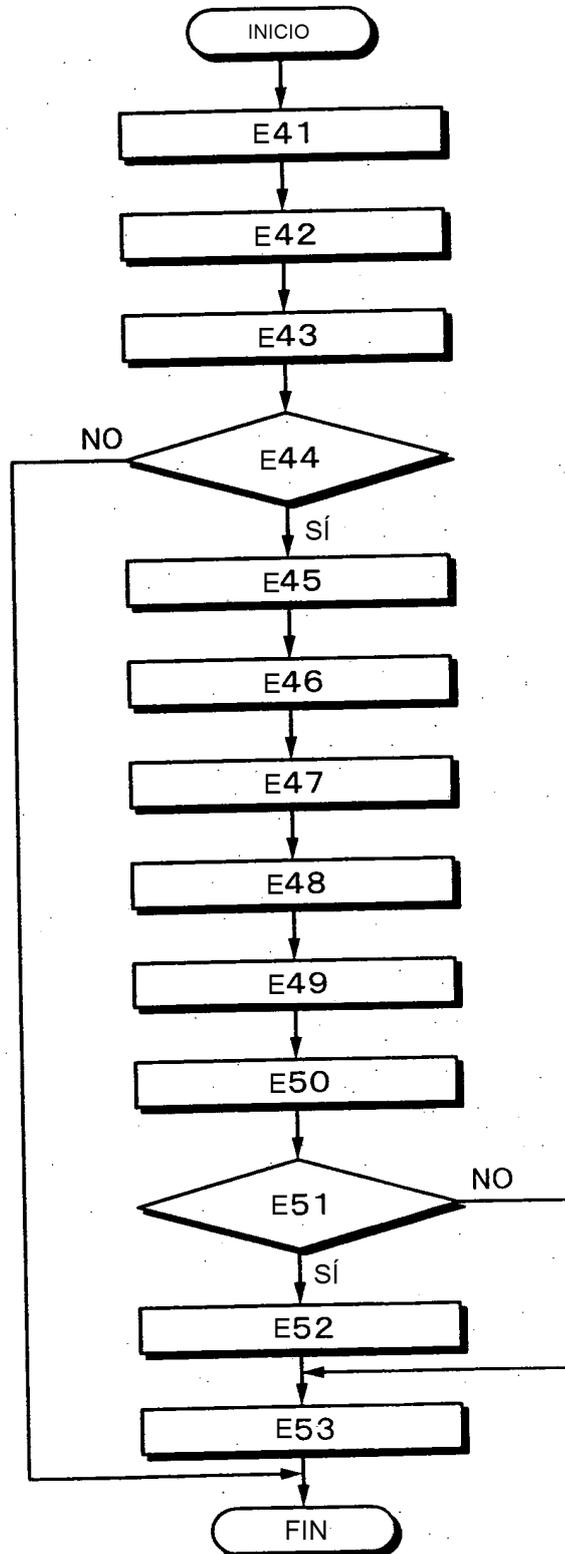


Fig. 27

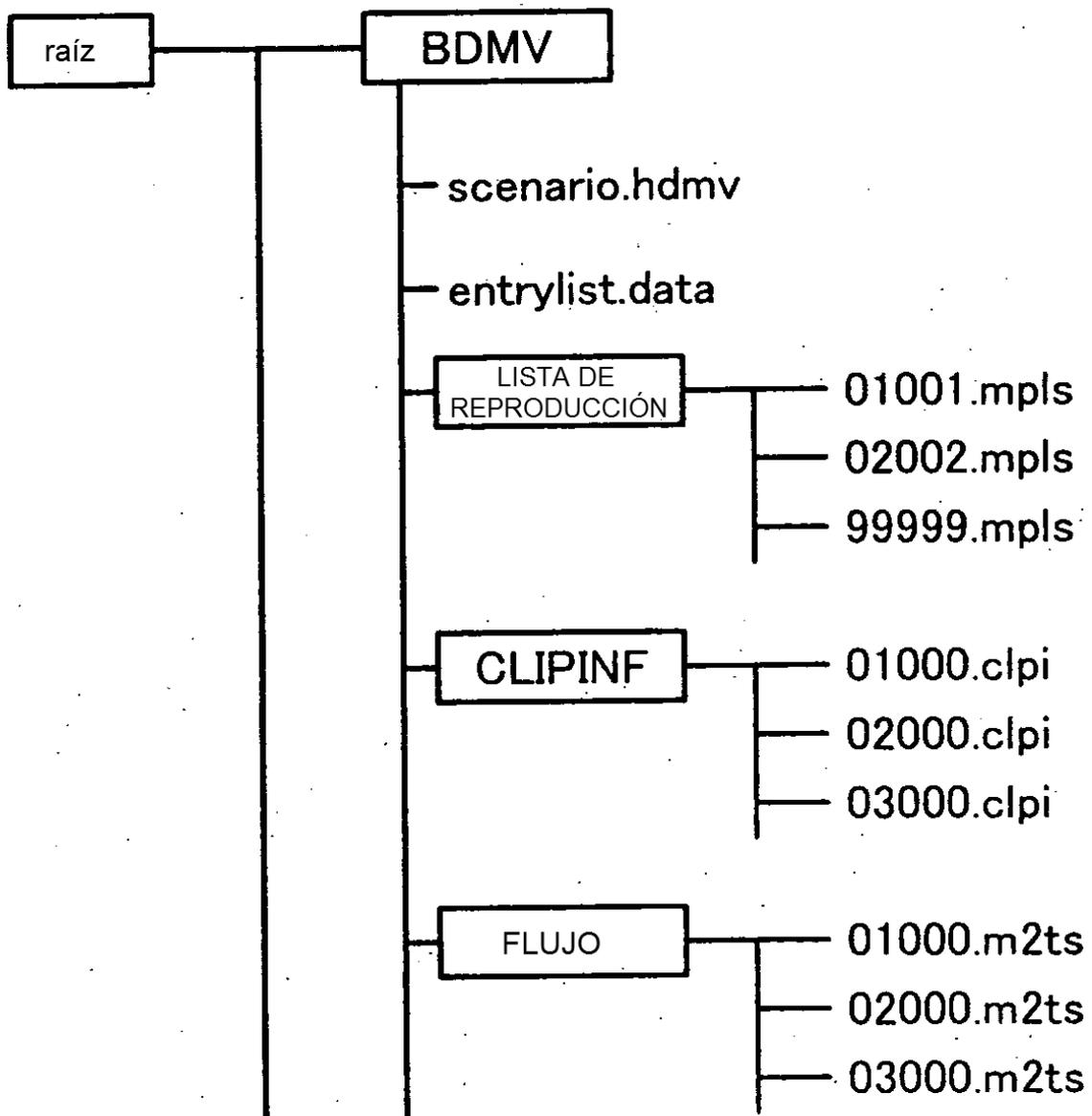


Fig. 28A

Fig. 28

Fig. 28A | Fig. 28B | Fig. 28C

100

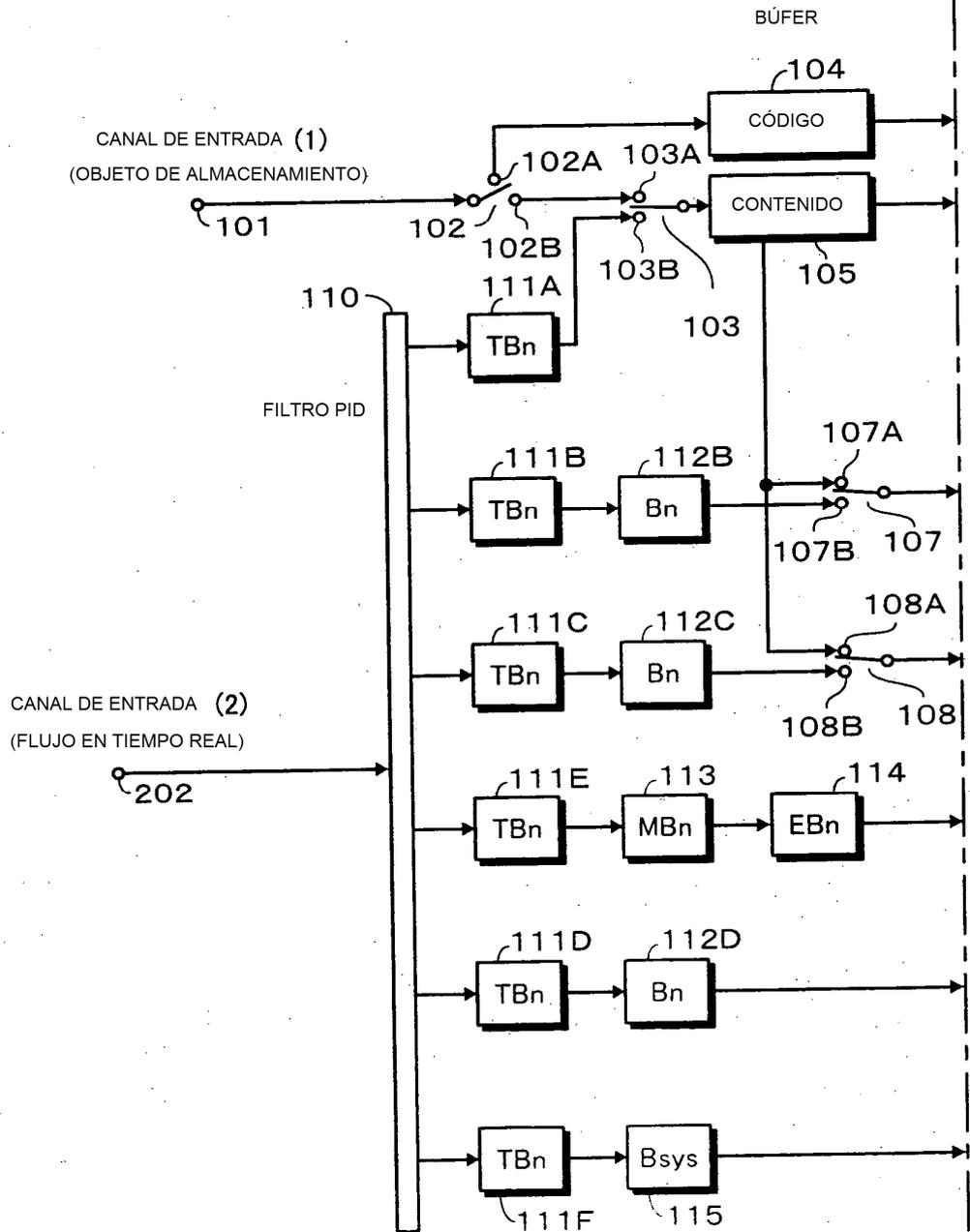


Fig. 28B

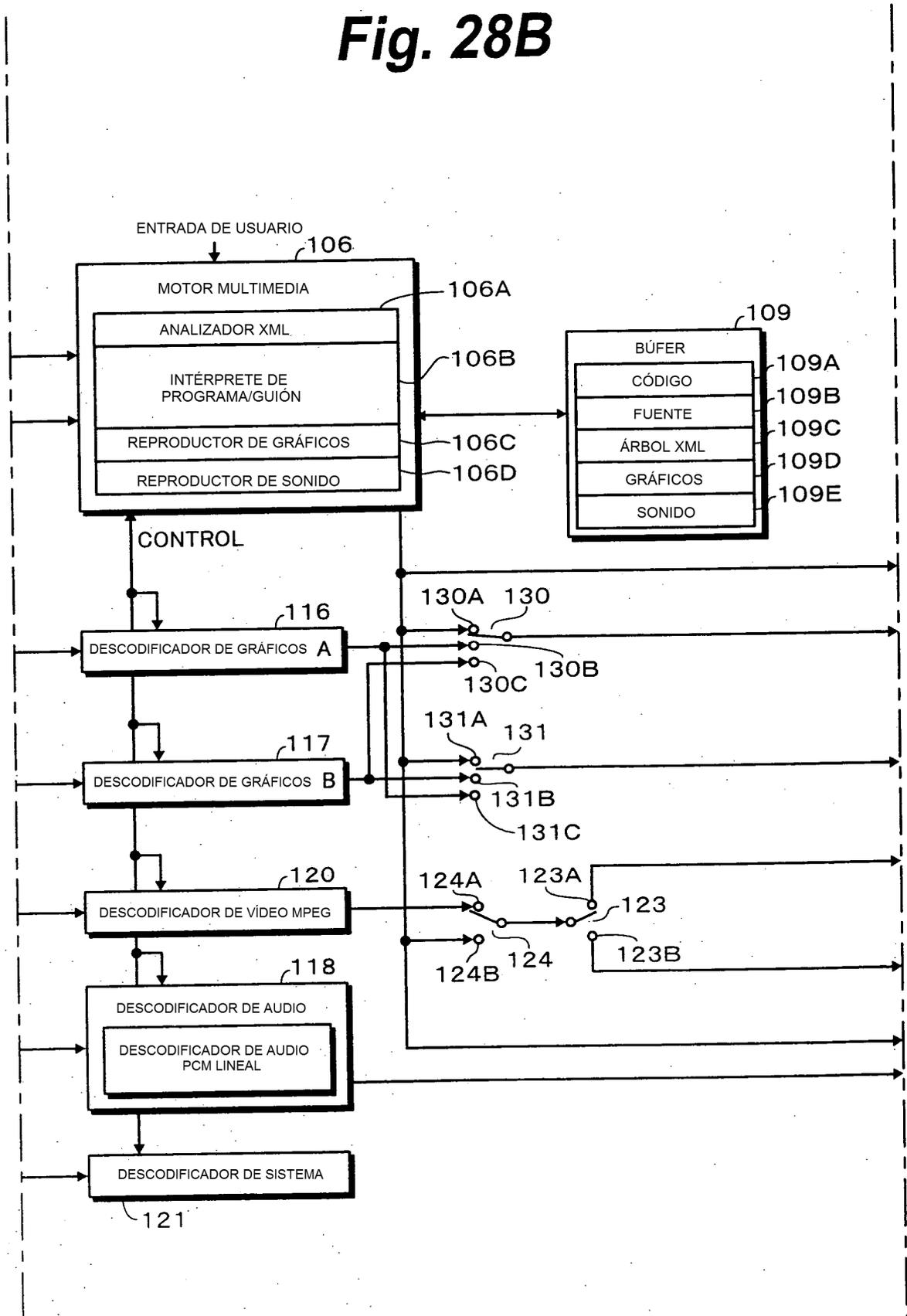


Fig. 28C

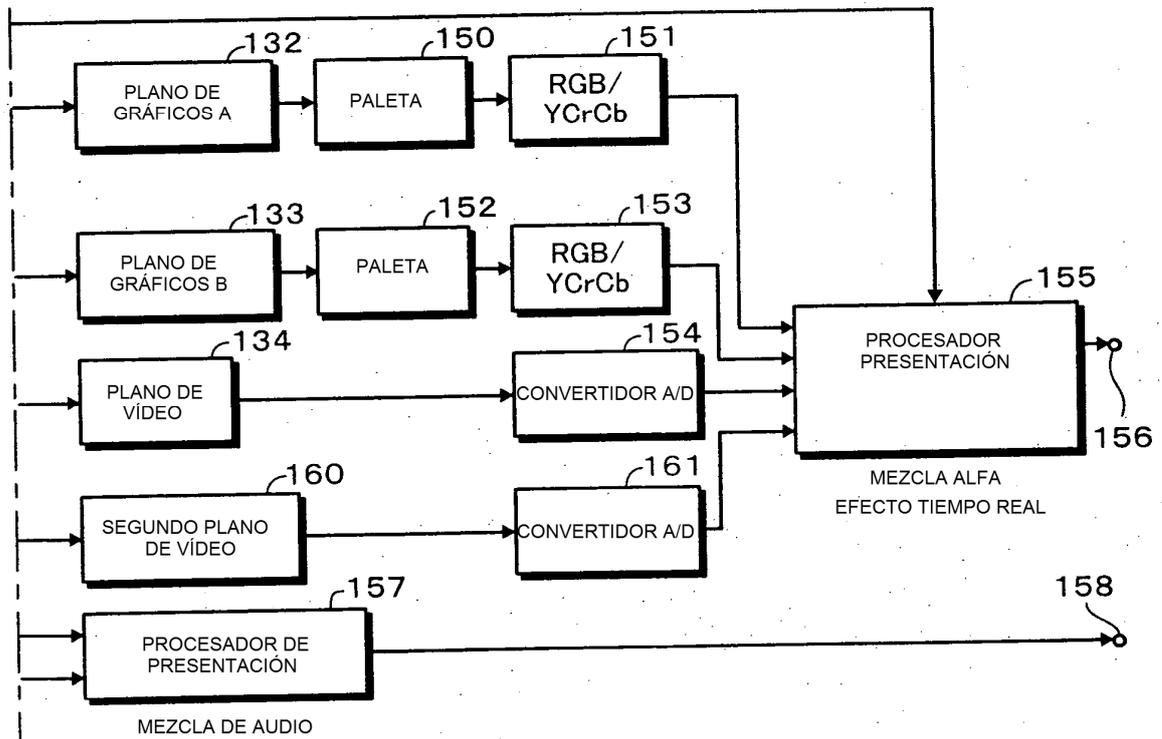


Fig. 29

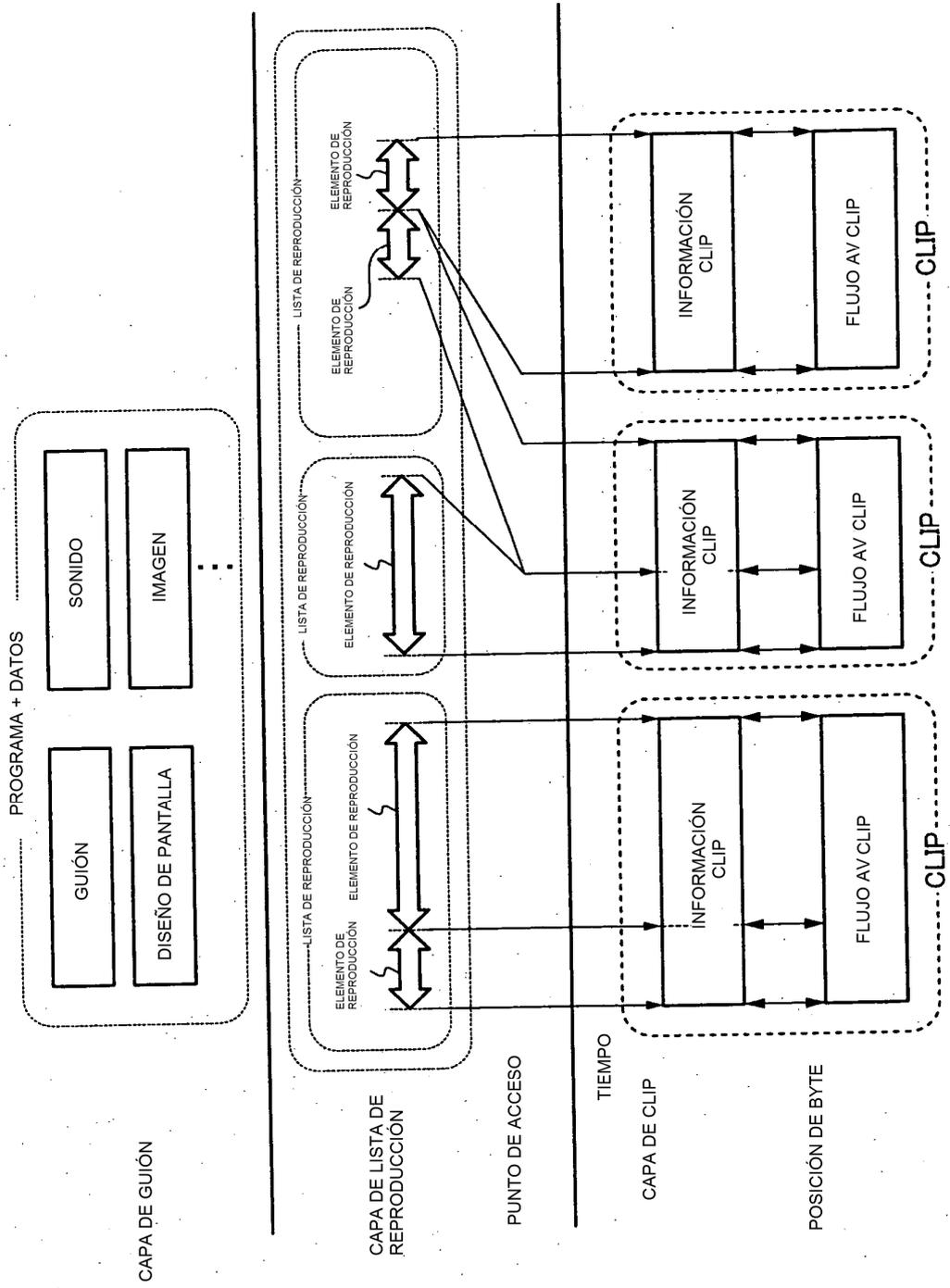


Fig. 30

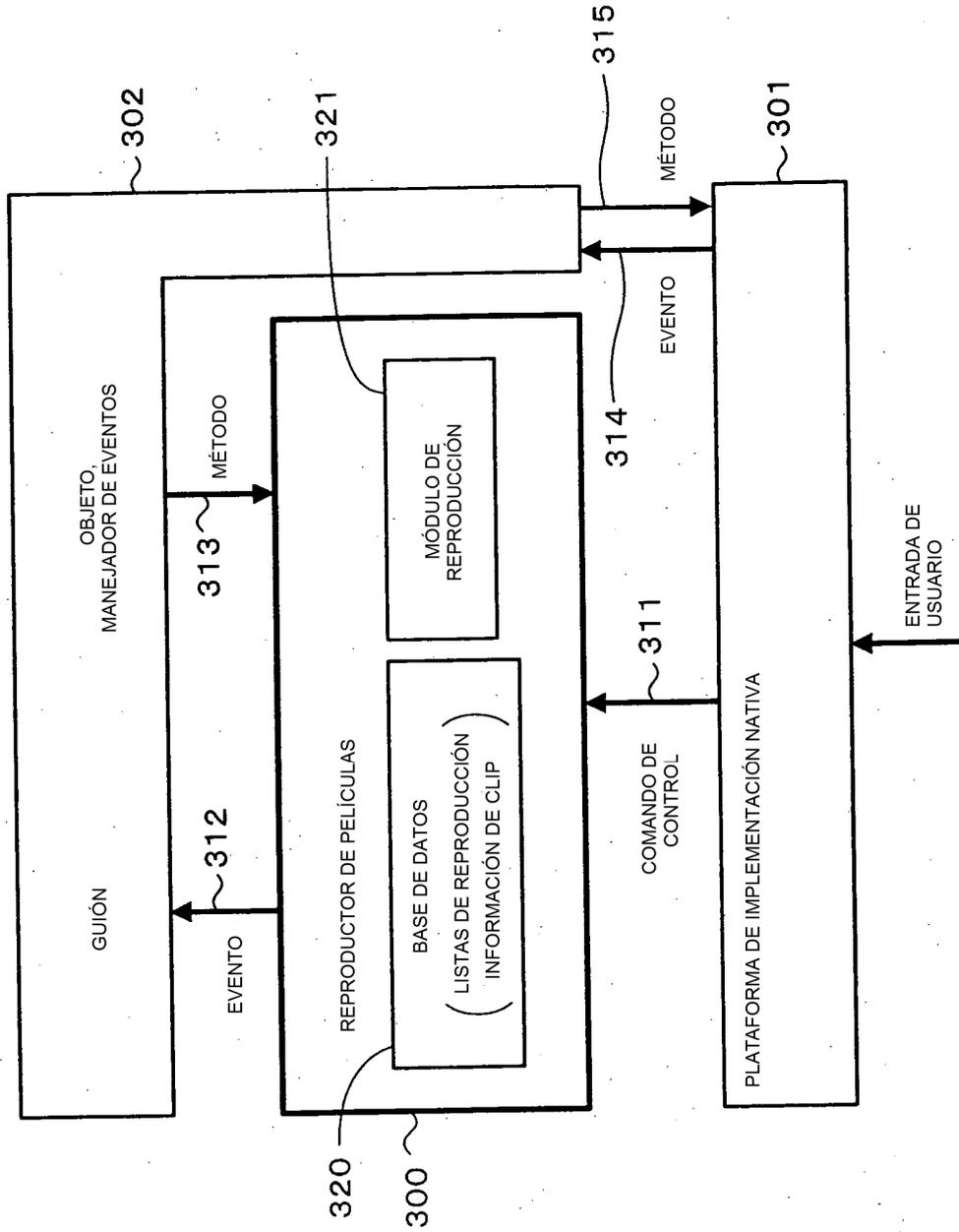


Fig. 31

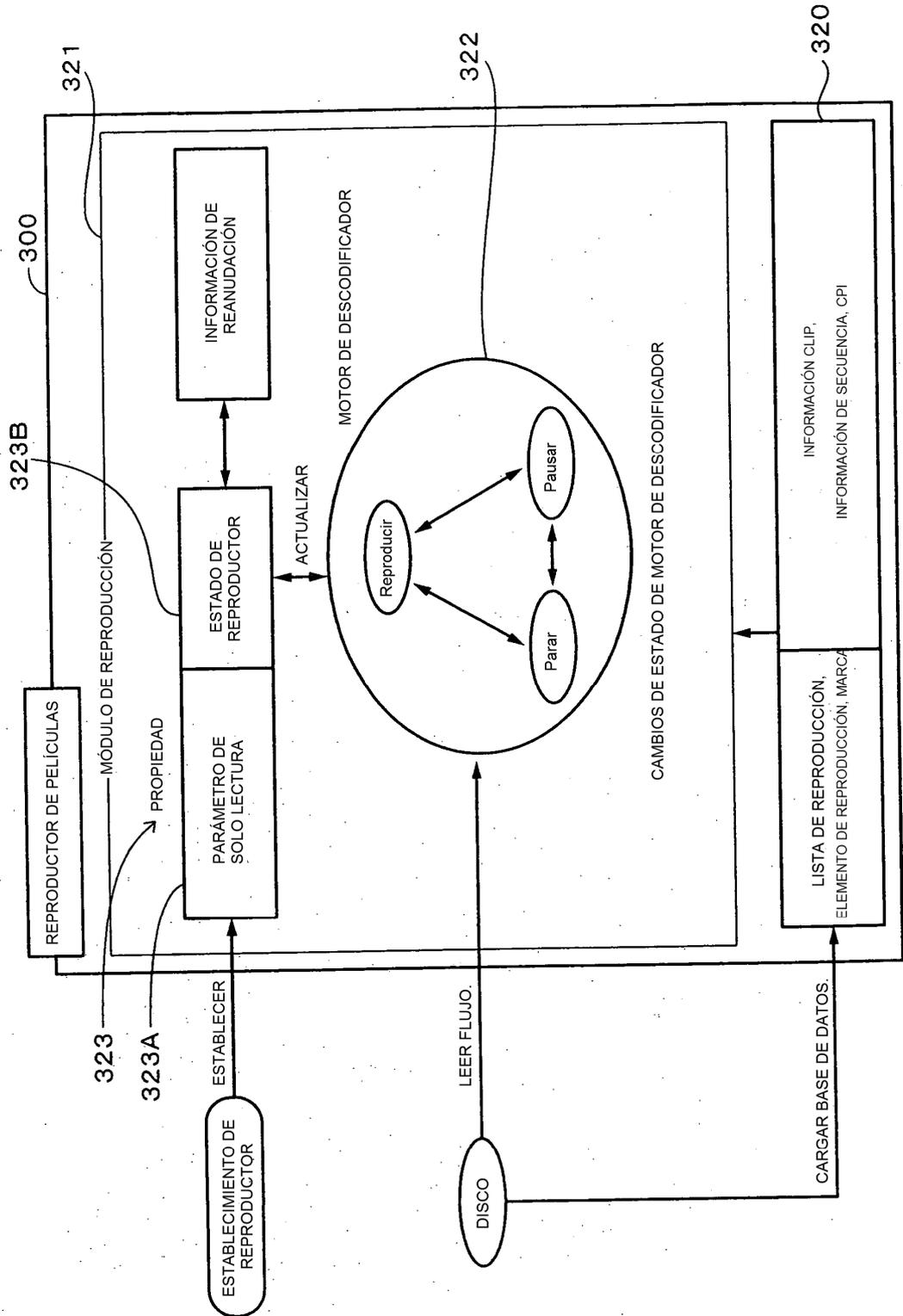


Fig. 32

ESTADO	DESCRIPCIÓN
reproducción	<p>REPRESENTA QUE EL REPRODUCTOR DE VÍDEOS ESTÁ REPRODUCIENDO LA LISTA DE REPRODUCCIÓN Y EL TIEMPO ESTÁ TRANSCURRIENDO. ADEMÁS DE LA REPRODUCCIÓN REGULAR, ESTE ESTADO INCLUYE REPRODUCCIÓN CON VELOCIDAD VARIABLE, AVANCE RÁPIDO Y REBINADO. LA REPRODUCCIÓN DE AVANCE DE ETAPA Y LA REPRODUCCIÓN DE REBINADO DE ETAPA REPRESENTAN QUE EL ESTADO DE PAUSA Y EL ESTADO DE REPRODUCCIÓN SE LLEVAN A CABO DE MANERA REPETIDA.</p>
pausa.	<p>REPRESENTA QUE EL EJE TEMPORAL SE DETIENE MIENTRAS LA LISTA DE REPRODUCCIÓN DE REPRODUCTOR DE VÍDEOS ESTÁ REPRODUCIENDO LA LISTA DE REPRODUCCIÓN.</p>
parado	<p>REPRESENTA QUE EL REPRODUCTOR DE VÍDEOS NO REPRODUCE LA LISTA DE REPRODUCCIÓN.</p>

Fig. 33

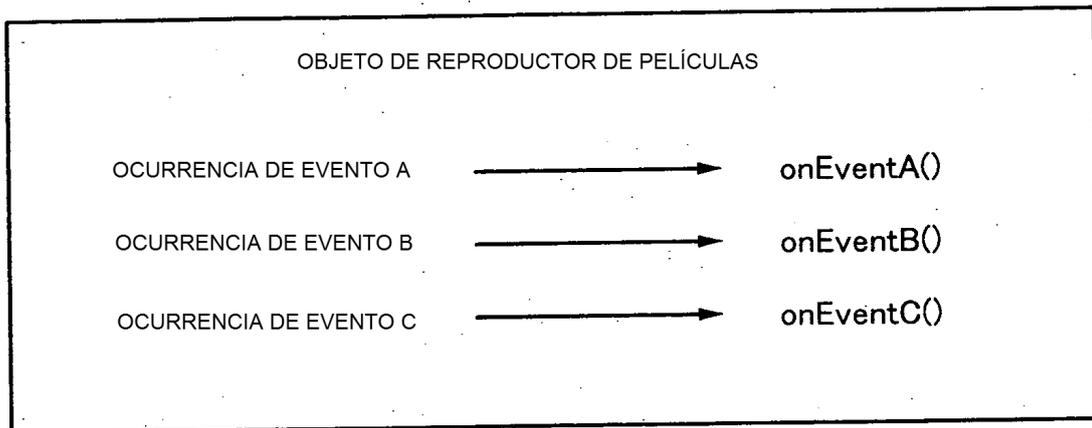


Fig. 34A

Fig. 34

Fig. 34A

Fig. 34B

NOMBRE DE MANEJADOR DE EVENTOS	DESCRIPCIÓN
onTimerFired()	<p>EL TEMPORIZADOR EN CUENTA REGRESIVA SE CONVIERTE EN 0. EL TEMPORIZADOR EN CUENTA ASCENDENTE SE CONVIERTE EN UN VALOR DESIGNADO.</p>
onPlayStopped()	<p>LA REPRODUCCIÓN SE DETIENE.</p>
onPlayStilled()	<p>LA REPRODUCCIÓN SE PAUSA.</p>
onPlayStarted()	<p>COMIENZA LA REPRODUCCIÓN.</p>
onPlayRepeated()	<p>SE DETECTA EL COMIENZO DE DATOS PREDETERMINADOS EN LA REPRODUCCIÓN DE REPETICIÓN.</p>
onSPDisplayStatusChanged()	<p>SE CAMBIA EL ESTADO DE VISUALIZACIÓN/NO VISUALIZACIÓN DEL FLUJO DE SUBTÍTULOS (SUBIMÁGENES).</p>

Fig. 34B

onSelectedAudioChanged()	SE CAMBIA EL FLUJO DE AUDIO QUE SE ESTÁ REPRODUCIENDO.
onVideoStopped()	SE CAMBIA EL FLUJO DE VIDEO QUE SE ESTÁ REPRODUCIENDO.
onPlayListStarted()	SE DETECTA EL COMIENZO DE LA LISTA DE REPRODUCCIÓN.
onPlayListEnded()	SE DETECTA EL FINAL DE LA LISTA DE REPRODUCCIÓN.
onPlayItemStarted()	SE DETECTA EL COMIENZO DEL ELEMENTO DE REPRODUCCIÓN.
onPlayItemEnded()	SE DETECTA EL FINAL DEL ELEMENTO DE REPRODUCCIÓN.
onMarkEncountered()	SE DETECTA UNA MARCA DURANTE LA REPRODUCCIÓN.

DESCRIPCIÓN DE NUMERALES DE REFERENCIA

30	REPRODUCTOR BÁSICO BD
31	PROGRAMA Java
32	API
34	PARÁMETRO COMÚN
35	COMANDO DE REPRODUCTOR
104	BÚFER DE CÓDIGO
106	MOTOR MULTIMEDIA
200	MOTOR DE PRESENTACIÓN BD
201	MOTOR DE NAVEGACIÓN
202	Java VM
203	APLICACIÓN Java
210	OBJETO BDBasicPlayer
211	INTERFAZ DE OYENTE
300	REPRODUCTOR DE PELÍCULAS
301	PLATAFORMA DE IMPLEMENTACIÓN NATIVA
302	CAPA DE GUIÓN
310	ENTRADA DE USUARIO
311	COMANDO DE CONTROL
312, 314	EVENTO
313, 315	MÉTODO
320	BASE DE DATOS
321	MÓDULO DE REPRODUCCIÓN
322	MOTOR DE DESCODIFICADOR
323	PROPIEDAD

- E20 DECIDIR EN QUÉ ESCENA DE LA LISTA DE REPRODUCCIÓN APARECE EL BOTÓN.
- E21 ESTABLECER MARCA EN TIEMPO DESIGNADO.
- E22 DESCRIBIR MÉTODO QUE REGISTRA EL OYENTE DE EVENTOS CORRESPONDIENTE A LA MARCA PARA EL OBJETO DE BOTÓN EN EL PROGRAMA.
- E23 IMPLEMENTAR INTERFAZ DE OYENTE CORRESPONDIENTE AL EVENTO DE GENERACIÓN DE MARCA (DESCRIBIR PROCESO EJECUTADO CUANDO EL EVENTO TIENE LUGAR).
- E30 COMENZAR REPRODUCCIÓN DE LISTA DE REPRODUCCIÓN.
- E31 **BDBasicPlayer** DETECTA QUE EL TIEMPO MARCADO HA TRANSCURRIDO MIENTRAS SE REPRODUCÍA LA LISTA DE REPRODUCCIÓN.
- E32 ¿SE HA REGISTRADO EL OYENTE DE EVENTOS CORRESPONDIENTE AL EVENTO DE MARCA?
- E33 TRANSMITIR EL EVENTO DE MARCA AL OBJETO QUE HA REGISTRADO EL OYENTE DE EVENTOS.
- E34 EL OBJETO INFORMADO SOBRE EL EVENTO EJECUTA EL MÉTODO A EJECUTARSE CUANDO EL EVENTO DE MARCA TIENE LUGAR.
- E35 HACE QUE EL BOTÓN APAREZCA.
- E36 NO INFORMADO SOBRE EL EVENTO DE DETECCIÓN DE MARCA.
- E41 EL USUARIO HA PRESIONADO LA TECLA "NEXT" DURANTE LA REPRODUCCIÓN.
- E42 **RmtKeyEvent** TIENE LUGAR.
- E43 OBTENER POSICIÓN DE MARCA DE CAPÍTULO SIGUIENTE DE BASE DE DATOS.
- E44 ¿HAY MARCA DE CAPÍTULO?
- E45 PARAR REPRODUCCIÓN.
- E46 **PlayStoppedEvent** TIENE LUGAR.
- E47 ACCEDER A POSICIÓN DE TIPO DE MARCA DE CAPÍTULO SIGUIENTE E INICIAR REPRODUCCIÓN DE DATOS DE VÍDEO.

- E48 **PlayStartedEvent** Y **MarkEncounteredEvent** TIENEN LUGAR.
- E49 INICIAR EJECUCIÓN DE OYENTE CORRESPONDIENTE A
MarkEncounteredEvent.
- E50 OBTENER TIPO DE Marca.
- E51 ¿MARCA DE CAPÍTULO?
- E52 VISUALIZAR MENSAJE QUE REPRESENTA COMIENZO DEL CAPÍTULO.
- E53 COMPLETAR EJECUCIÓN DE OYENTE.