

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 674 924**

51 Int. Cl.:

G10L 19/00 (2013.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **12.06.2014 PCT/US2014/042168**

87 Fecha y número de publicación internacional: **24.12.2014 WO14204783**

96 Fecha de presentación y número de la solicitud europea: **12.06.2014 E 14813862 (1)**

97 Fecha y número de publicación de la concesión europea: **09.05.2018 EP 2954515**

54 Título: **Decodificador y codificador de audio con información de programa o metadatos de estructura de flujo secundario**

30 Prioridad:

19.06.2013 US 201361836865 P

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

05.07.2018

73 Titular/es:

**DOLBY LABORATORIES LICENSING CORPORATION (100.0%)
100 Potrero Avenue
San Francisco, CA 94103-4813, US**

72 Inventor/es:

**RIEDMILLER, JEFFREY y
WARD, MICHAEL**

74 Agente/Representante:

LEHMANN NOVO, María Isabel

ES 2 674 924 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

Decodificador y codificador de audio con información de programa o metadatos de estructura de flujo secundario

5 REFERENCIA CRUZADA A APLICACIONES RELACIONADAS

Esta solicitud reivindica la prioridad de la Solicitud de Patente Provisional de los Estados Unidos Nº 61/836.865, presentada el 19 de junio de 2013.

10 CAMPO TÉCNICO

15 La invención se refiere al procesamiento de señales de audio, y más particularmente, a la codificación y decodificación de flujos binarios de datos de audio con metadatos indicativos de estructura de flujo secundario y/o información de programa con respecto al contenido de audio indicado por los flujos binarios. Algunas formas de realización de la invención generan o decodifican datos de audio en uno de los formatos conocidos como Dolby Digital (AC-3), Dolby Digital Plus (AC-3 Mejorado o E-AC-3) o Dolby E.

ANTECEDENTES DE LA INVENCIÓN

20 Dolby, Dolby Digital, Dolby Digital Plus y Dolby E son marcas registradas de Dolby Laboratories Licensing Corporation. Dolby Laboratories proporciona puestas en práctica propietarias AC-3 y E-AC-3, conocidas como Dolby Digital y Dolby Digital Plus, respectivamente.

25 Unidades de procesamiento de datos de audio suelen operar en una forma denominada 'a ciegas' y no prestan atención al historial de procesamiento de los datos de audio que se produce antes de que se reciban los datos. Esto puede funcionar en un marco de procesamiento en el que una sola entidad realiza todas las funciones de procesamiento y codificación de los datos de audio para una diversidad de dispositivos de representación multimedia objetivo mientras que un dispositivo de representación multimedia de destino pone en práctica toda la decodificación y representación de los datos de audio codificados. Sin embargo, este procesamiento 'a ciegas' no funciona bien (o no funciona) en situaciones en donde una pluralidad de unidades de procesamiento de audio están dispersadas a través de una red diversa o se colocan en tándem (es decir, en cadena) y está previsto que realicen, de forma óptima, sus respectivos tipos procesamiento de audio. A modo de ejemplo, algunos datos de audio pueden estar codificados para sistemas multimedia de alto rendimiento, y pueden tener que convertirse a una forma reducida adecuada para un dispositivo móvil a lo largo de una cadena de procesamiento multimedia. En consecuencia, una unidad de procesamiento de audio puede realizar, de forma innecesaria, un tipo de procesamiento en los datos de audio que ya se han realizado. A modo de ejemplo, una unidad de nivelación de sonoridad puede realizar el procesamiento sobre un clip de audio de entrada, independientemente de si se ha realizado anteriormente, o no, la misma nivelación de sonoridad en el clip de audio de entrada. En consecuencia, la unidad de nivelación de sonoridad puede realizar la nivelación incluso cuando no sea necesario. Este procesamiento innecesario puede causar, además, la degradación y/o la eliminación de características específicas mientras se representa el contenido de los datos de audio.

45 Es conocida, de conformidad con el documento de solicitud de patente internacional WO02/091361A1, una técnica para añadir datos a una trama de datos comprimida, utilizando un campo de omisión de un flujo binario AC-3. Estos bits de campo de omisión se sustituyen con bits que incluyen información. Los nuevos bits portadores de información deben ajustarse a un formato o sintaxis conocido o predeterminado, de modo que puedan recuperarse mediante un proceso de decodificación.

50 El documento ATSC Standard: Compresión de audio digital (AC-3, E-AC-3), doc. A/52: 2012, 17-12-2012, da a conocer las especificaciones de los flujos binarios AC-3 y E-AC-3.

55 El documento WO2006/113062A1 da a conocer un flujo binario digital que comprende bits de datos que representan audio, metadatos destinados a ser correctos para el audio, e información de verificación de metadatos, en particular una copia del parámetro DIALNORM. Estos metadatos de verificación se incluyen en un campo de omisión de un flujo binario AC-3.

60 El documento WO2014/113465A1, publicado el 24-07-2014, da a conocer una técnica para incluir metadatos de estado de procesamiento de sonoridad (LPSM) en un campo de omisión de una trama de un flujo binario, de conformidad con el formato AC-3 o E-AC-3. Este metadato LPSM no representa un metadato de información del programa.

BREVE DESCRIPCIÓN DE LA INVENCIÓN

65 La invención da a conocer un método para generar un flujo binario de audio codificado según la reivindicación 1, un método para decodificar un flujo binario de audio codificado según la reivindicación 2, un soporte de memorización legible por ordenador según la reivindicación 8, y una unidad de procesamiento de audio según la reivindicación 9.

En algunos ejemplos, se da a conocer una unidad de procesamiento de audio capaz de decodificar un flujo binario codificado que incluye metadatos de estructura de flujo secundario y/o metadatos de información sobre el programa (y, de modo opcional, también otros metadatos, p.ej., metadatos de estado de procesamiento de sonoridad) en al menos un segmento de al menos una trama del flujo binario y datos de audio en al menos uno de entre otros segmentos de la trama. En este caso, los metadatos de estructura flujo secundario (o "SSM") indican metadatos de un flujo binario codificado (o conjunto de flujos binarios codificados), que indican la estructura flujo secundario del contenido de audio de los flujos binarios codificados y "metadatos de información sobre el programa" (o "PIM") indican metadatos de un flujo binario de audio codificado, indicativo de al menos un programa de audio (p.ej., dos o más programas de audio), en donde los metadatos de información sobre el programa son indicativos de al menos una propiedad o característica del contenido de audio de al menos uno de dichos programas (ej., metadatos que indican un tipo o parámetro de procesamiento realizado en datos de audio del programa, o metadatos que indican qué canales del programa son canales activos).

En casos típicos (p.ej., en los que el flujo binario codificado es un flujo binario AC-3 o E-AC-3), los metadatos de información sobre el programa (PIM) son indicativos de información de programa que, prácticamente, no pueden transmitirse en otras partes del flujo binario. A modo de ejemplo, los PIM pueden ser indicativos del procesamiento aplicado al audio PCM antes de la codificación (p.ej., codificación AC-3 o E-AC-3), qué bandas de frecuencia del programa de audio se han codificado utilizando técnicas específicas de codificación de audio, y el perfil de compresión utilizado para crear datos de compresión de margen dinámico (DRC) en el flujo binario.

En otros ejemplos, un método incluye una etapa de multiplexación de datos de audio codificados con SSM y/o PIM en cada trama (o de cada de al menos algunas tramas) del flujo binario. En la decodificación típica, un decodificador extrae el SSM y/o PIM del flujo binario (incluyendo, mediante análisis y demultiplexación de SSM y/o PIM y los datos de audio) y procesa los datos de audio para generar un flujo de datos de audio decodificados (y, en algunos casos, realiza, además, un procesamiento adaptativo de los datos de audio). En algunas formas de realización, los datos de audio decodificados y metadatos SSM y/o PIM, se reenvían desde el decodificador a un post-procesador configurado para realizar un procesamiento adaptativo sobre los datos de audio decodificados utilizando los SSM y/o PIM.

En otros ejemplos, un método de codificación genera un flujo binario de audio codificado (p.ej., un flujo binario AC-3 o E-AC-3), que incluye segmentos de datos de audio (p.ej., los segmentos AB0-AB5 de la trama ilustrada en la Figura 4, o la totalidad o algunos de los segmentos AB0-AB5 de la trama ilustrada en la Figura 7), que incluye datos de audio codificados y segmentos de metadatos (incluyendo SSM y/o PIM y, opcionalmente, otros metadatos opcionales) multiplexados por división de tiempo con los segmentos de datos de audio. En algunos ejemplos, cada segmento de metadatos (a veces denominado en este documento como un "contenedor") tiene un formato que incluye una cabecera de segmento de metadatos (y, de forma opcional, también otros elementos obligatorios o "principales"), y una o más cargas útiles de metadatos después de la cabecera de segmento de metadatos. SIM, si está presente, se incluye en una de las cargas útiles de metadatos (que se identifica por una cabecera de carga útil y, por lo general, tiene un formato de un primer tipo). PIM, si está presente, se incluye en otra de las cargas útiles de metadatos (identificada por una cabecera de carga útil y que suele tener un formato de un segundo tipo). De modo similar, cada otro tipo de metadatos (si está presente) se incluye en una distinta de las cargas útiles de metadatos (que se identifica por una cabecera de carga y suele tener un formato específico para el tipo de metadatos). El formato, a modo de ejemplo, permite un acceso conveniente para metadatos SSM, PIM y otros metadatos en momentos diferentes a la decodificación (p.ej., por un post-procesador que sigue a la decodificación, o mediante un procesador configurado para reconocer los metadatos sin realizar una decodificación completa en el flujo binario codificado), y permite la detección y corrección de errores de forma conveniente y eficiente (p.ej., de identificación de flujo secundario) durante la decodificación del flujo binario. A modo de ejemplo, sin acceso a SSM en el formato de ejemplo, un decodificador podría identificar, incorrectamente, el número correcto de flujos secundarios asociados con un programa. Una carga útil de metadatos, en un segmento de metadatos, puede incluir SSM, otra carga útil de metadatos, en el segmento de metadatos, puede incluir PIM y, opcionalmente, al menos otra carga útil adicional de metadatos, en el segmento de metadatos, puede incluir otros metadatos (p.ej., metadatos de estado de procesamiento de sonoridad o "LPSM").

BREVE DESCRIPCIÓN DE LOS DIBUJOS

La Figura 1 es un diagrama de bloques de una forma de realización de un sistema que puede configurarse para realizar una forma de realización del método de la invención.

La Figura 2 es un diagrama de bloques de un codificador;

La Figura 3 es un diagrama de bloques de un decodificador, y un post-procesador acoplado al mismo.

La Figura 4 es un diagrama de una trama AC-3, que incluye los segmentos en los que está dividido.

La Figura 5 es un diagrama del segmento de Información de Sincronización (SI) de una trama AC-3, que incluye los segmentos en los que está dividido.

La Figura 6 es un diagrama del segmento de Información de Flujo binario (BSI) de una trama AC-3, incluidos los segmentos en los que está dividido.

5 La Figura 7 es un diagrama de una trama E-AC-3, incluidos los segmentos en los que está dividido.

La Figura 8 es un diagrama de un segmento de metadatos de un flujo binario codificado, generado de conformidad con una forma de realización de la invención, que incluye una cabecera de segmento de metadatos que comprende una palabra de sincronización de contenedor (identificada como "sincronización de contenedor" en la Figura 8) y valores de identificador ID de versión y clave, seguido de múltiples cargas útiles de metadatos y bits de protección.

10

Notación y terminología

15

A lo largo de esta descripción, las expresiones "forma de realización" o "formas de realización" cuando no se refieren, específicamente, como "forma de realización de la invención" o "formas de realización de la invención", deben entenderse como un ejemplo ilustrativo que no está necesariamente cubierto por las reivindicaciones.

20

A través de esta descripción, incluida en las reivindicaciones, la expresión que realiza una operación "sobre" una señal o datos (p.ej., filtrado, escalado, transformación o aplicación de ganancia a, la señal o datos) se usa en un sentido amplio para indicar la realización de la operación directamente en la señal o datos o, en una versión procesada, de la señal o datos (p.ej., en una versión de la señal que ha sido sometida a filtrado preliminar o pre-procesamiento antes de la ejecución de la propia operación).

25

A lo largo de esta descripción que se incluye en las reivindicaciones, la expresión "sistema" se utiliza en un sentido amplio para designar un dispositivo, sistema o subsistema. A modo de ejemplo, un subsistema que pone en práctica un decodificador se puede referir como un sistema decodificador, y un sistema que incluye dicho subsistema (p.ej., un sistema que genera X señales de salida en respuesta a múltiples entradas, en donde el subsistema genera M de las entradas y las otras X - M entradas se reciben desde una fuente externa) se puede denominar también como un sistema decodificador.

30

A través de esta descripción, que se incluye en las reivindicaciones, el término "procesador" se utiliza en un sentido amplio para designar un sistema o dispositivo programable o, de cualquier otro modo, configurable (p.ej., con software o firmware) para realizar operaciones en datos (p.ej., audio, o vídeo u otros datos de imagen). Ejemplos de procesadores incluyen una matriz de puertas programable in situ (u otro circuito o conjunto de circuitos integrados configurables), un procesador de señal digital programado y/o, de cualquier otro modo, configurado para realizar un procesamiento más eficiente en datos de audio u otros datos de sonido, un procesador u ordenador de uso general programable, y un conjunto de circuitos o circuitos de microprocesador programable.

35

40

A lo largo de esta descripción, que se incluye en las reivindicaciones, las expresiones "procesador de audio" y "unidad de procesamiento de audio" se utilizan indistintamente y, en un sentido amplio, para indicar un sistema configurado para procesar datos de audio. Ejemplos de unidades de procesamiento de audio incluyen, pero sin limitación, codificadores (p.ej., transcodificadores), decodificadores, códecs, sistemas de procesamiento previo, sistemas de procesamiento posterior y sistemas de procesamiento de flujo binario (a veces referidos como herramientas de procesamiento de flujo binario).

45

A través de esta descripción, que se incluye en las reivindicaciones, la expresión "metadatos" (de un flujo binario de audio codificado), se refiere a datos separados y diferentes de los datos de audio correspondientes del flujo binario.

50

A lo largo de esta descripción, que se incluye en las reivindicaciones, la expresión "metadatos de estructura de flujo secundario" (o "SSM"), indica metadatos de un flujo binario de audio codificado (o un conjunto de flujos binarios de audio codificados), que indican la estructura flujo secundario del contenido de audio de los flujos binarios codificados.

55

A través de esta descripción, que se incluye en las reivindicaciones, la expresión "metadatos de información sobre el programa" (o "PIM") indica metadatos de un flujo binario de audio codificado, indicativo de al menos un programa de audio (p.ej., dos o más programas de audio), en donde dicho los metadatos indican, al menos, una propiedad o característica del contenido de audio de al menos uno de dichos programas (p.ej., metadatos que indican un tipo o parámetro de procesamiento realizado sobre datos de audio del programa, o metadatos que indican qué canales del programa son canales activos).

60

65

A lo largo de esta descripción, que se incluye en las reivindicaciones, la expresión "metadatos de estado de procesamiento" (p.ej., como en la expresión "metadatos de estado de procesamiento de sonoridad"), se refiere a metadatos (de un flujo binario de audio codificado) asociados con datos de audio del flujo binario, que indican el estado de procesamiento de los datos de audio correspondientes (asociados) (p.ej., qué tipo de procesamiento ya se han realizado en los datos de audio) y, normalmente, indica, además, al menos una función o característica de los datos de audio. La asociación de los metadatos del estado de procesamiento con los datos de audio es sincronizada

en el tiempo. Por lo tanto, los metadatos de estado de procesamiento presentes (más recientemente recibidos o actualizados) indican que los correspondientes datos de audio comprenden, contemporáneamente, los resultados del tipo indicado de procesamiento de datos de audio. En algunos casos, metadatos de estado de procesamiento pueden incluir el historial de procesamiento y/o algunos, o la totalidad, de los parámetros que se utilizan y/o derivan de los tipos de procesamiento indicados. Además, los metadatos de estado de procesamiento pueden incluir al menos una función o característica de los correspondientes datos de audio, que se ha calculado o extraído a partir de los datos de audio. Metadatos de estado de procesamiento pueden incluir, además, otros metadatos que no están relacionados, ni derivados, de ningún procesamiento de los datos de audio correspondientes. A modo de ejemplo, datos de terceros, información de seguimiento, identificadores, información de propiedad o estándar, datos de anotación de usuario, datos de preferencia de usuario, etc., se pueden añadir por una unidad de procesamiento de audio particular para pasar a otras unidades de procesamiento de audio.

A través de esta descripción, que se incluye en las reivindicaciones, la expresión "metadatos de estado de procesamiento de sonoridad" (o "LPSM"), indica metadatos de estado de procesamiento, indicativos del estado de procesamiento de sonoridad de los correspondientes datos de audio (p.ej., qué tipo de procesamiento de sonoridad se han realizado en los datos de audio) y, normalmente, también al menos una función o característica (p.ej., sonoridad) de los correspondientes datos de audio. Los metadatos de estado de procesamiento de sonoridad pueden incluir datos (p.ej., otros metadatos) que no son (es decir, cuando se considera por sí solos) metadatos de estado de procesamiento de sonoridad.

A lo largo de esta descripción, que se incluye en las reivindicaciones, la expresión "canal" (o "canal de audio"), indica una señal de audio monofónica.

A través de esta descripción, que se incluye en las reivindicaciones, la expresión "programa de audio" indica un conjunto de uno o más canales de audio y, opcionalmente, también metadatos asociados (p.ej., metadatos que describen una presentación de audio espacial deseada, y/o PIM, y/o SSM, y/o LPSM y/o metadatos de límite de programa).

A lo largo de esta descripción, que se incluye en las reivindicaciones, la expresión "metadatos de límite de programa", indica metadatos de un flujo binario de audio codificado, en donde el flujo binario de audio codificado es indicativo de al menos un programa de audio (p.ej., dos o más programas de audio), y los metadatos de límite del programa indican la ubicación, en el flujo binario, de al menos un límite (comienzo y/o final) de al menos uno de dichos programas de audio. A modo de ejemplo, los metadatos de límite de programa (de un flujo binario de audio codificado, indicativo de un programa de audio), pueden incluir metadatos que indican la localización (p.ej., el inicio de la "N"-ésima trama del flujo binario, o la "M"-ésima localización de muestra de la "N"-ésima trama del flujo binario), desde el comienzo del programa, y metadatos adicionales indicativos de la ubicación (p.ej., la "J"-ésima trama del flujo binario, o la "K"-ésima localización de muestra de la "J"-ésima trama del flujo binario) del final de programa.

A través de esta descripción, que se incluye en las reivindicaciones, el término "acopla" o "acoplado" se utiliza para indicar una conexión directa o indirecta. En consecuencia, si un primer dispositivo se acopla a un segundo dispositivo, esa conexión puede ser a través de una conexión directa, o a través de una conexión indirecta, a través de otros dispositivos y conexiones.

DESCRIPCIÓN DETALLADA DE LAS FORMAS DE REALIZACIÓN DE LA INVENCION

Un flujo típico de datos de audio incluye tanto contenido de audio (p.ej., uno o más canales de contenido de audio) como metadatos indicativos de al menos una característica del contenido de audio. A modo de ejemplo, en un flujo binario AC-3 existen varios parámetros de metadatos de audio que están específicamente previstos para ser utilizados para cambiar el sonido del programa entregado a un entorno de escucha. Uno de los parámetros de metadatos es el parámetro DIALNORM, que está destinado a indicar el nivel medio de diálogo en un programa de audio, y se utiliza para determinar el nivel de señal de reproducción de audio.

Durante la reproducción de un flujo binario que comprende una secuencia de diferentes segmentos de programa de audio, (cada uno con un parámetro DIALNORM diferente), un decodificador AC-3 utiliza el parámetro DIALNORM de cada segmento para realizar un tipo de procesamiento de sonoridad, en el que modifica el nivel de reproducción o sonoridad de tal manera que la sonoridad percibida del diálogo de la secuencia de segmentos está en un nivel constante. Cada segmento de audio codificado (elemento), en una secuencia de elementos de audio codificados, tendría (en general) un parámetro DIALNORM diferente, y el decodificador establecería a escala el nivel de cada uno de los elementos de modo que el nivel de reproducción o la sonoridad del diálogo, para cada elemento, sea igual o muy similar, aunque esto podría requerir la aplicación de diferentes cantidades de ganancia a diferentes elementos durante la reproducción.

El parámetro DIALNORM se suele establecer por un usuario, y no se genera de forma automática, aunque existe un valor predeterminado del parámetro DIALNORM si el usuario no establece ningún valor. A modo de ejemplo, un creador de contenido puede realizar mediciones de sonoridad con un dispositivo externo a un codificador AC-3 y a

continuación, transmitir el resultado (indicativo de la intensidad del diálogo hablado de un programa de audio) al codificador para establecer el valor de DIALNORM. Por lo tanto, se depende del creador del contenido para establecer correctamente el parámetro DIALNORM.

5 Existen varias razones diferentes por las que el parámetro DIALNORM, en un flujo binario AC-3, puede ser incorrecto. En primer lugar, cada codificador AC-3 tiene un valor DIALNORM predeterminado que se utiliza durante la generación del flujo binario si el creador de contenido no establece un valor DIALNORM. Este valor predeterminado puede ser esencialmente distinto del nivel de sonoridad de diálogo real del audio. En segundo lugar, incluso si un creador de contenido mide la sonoridad y establece el valor de DIALNORM en consecuencia, es posible que se haya utilizado un algoritmo de medición de sonoridad, o medidor, que esté en conformidad con el método de medición de sonoridad AC-3 recomendado, lo que da como resultado un valor de DIALNORM incorrecto. En tercer lugar, incluso si se ha creado un flujo binario AC-3 con el valor DIALNORM medido y establecido correctamente por el creador del contenido, puede haberse cambiado a un valor incorrecto durante la transmisión y/o memorización del flujo binario. A modo de ejemplo, no es infrecuente que las aplicaciones de difusión de televisión para flujos binarios AC-3 se decodifiquen, modifiquen y a continuación, se vuelvan a codificar utilizando información de metadatos DIALNORM incorrecta. De este modo, un valor de DIALNORM, incluido en un flujo binario AC-3, puede ser incorrecto o impreciso y, por lo tanto, puede tener un impacto negativo en la calidad de la experiencia de escucha.

20 Además, el parámetro DIALNORM no indica el estado de procesamiento de la sonoridad de los correspondientes datos de audio (p.ej., qué tipo de procesamiento de sonoridad se ha realizado en los datos de audio). Metadatos de estado de procesamiento de sonoridad (en el formato en el que se proporcionan en algunas formas de realización de la presente invención), son útiles para facilitar el procesamiento de sonoridad adaptativo de un flujo binario de audio y/o la verificación de la validez del estado de procesamiento de sonoridad y la sonoridad del contenido de audio, de una manera particularmente eficiente.

Un flujo binario codificado AC-3 comprende metadatos, y uno a seis canales de contenido de audio. El contenido de audio son datos de audio que se ha comprimido utilizando una codificación de audio perceptual. Los metadatos incluyen varios parámetros de metadatos de audio que están destinados a ser utilizados para cambiar el sonido de un programa proporcionado a un entorno de escucha.

30 Cada trama de un flujo binario de audio codificado AC-3, incluye contenido de audio y metadatos para 1536 muestras de audio digital. Para una tasa de muestreo de 48 kHz, esto representa 32 milisegundos de audio digital o una tasa de 31.25 tramas por segundo de audio.

35 Cada trama de un flujo binario de audio codificado E-AC-3, incluye contenido de audio y metadatos para 256, 512, 768 o 1536 muestras de audio digital, dependiendo de si la trama contiene uno, dos, tres o seis bloques de datos de audio, respectivamente. Para una tasa de muestreo de 48 kHz, esto representa 5.333, 10.667, 16 o 32 milisegundos de audio digital, respectivamente, o una tasa de 189.9, 93.75, 62.5 o 31.25 tramas por segundo de audio, respectivamente.

40 Tal como se indica en la Figura 4, cada trama AC-3 está dividida en secciones (segmentos), que incluyen: una sección de Información de Sincronización (SI), que contiene (tal como se muestra en la Figura 5), una palabra de sincronización (SW) y la primera de dos palabras de corrección de errores (CRC1); una sección de Información de Flujo binario (BSI), que contiene la mayoría de los metadatos; seis Bloques de Audio (AB0 a AB5), que tienen contenidos de audio comprimidos de datos (y además, pueden incluir metadatos); segmentos de bits residuales (W) (también conocidos como "campos de omisión"), que contienen los bits no utilizados que quedan después de que se comprima el contenido de audio; una sección de información Auxiliar (AUX) que puede incluir más metadatos; y la segunda de dos palabras de corrección de error (CRC2).

50 Tal como se indica en la Figura 7, cada trama E-AC-3 está dividida en secciones (segmentos), que incluyen: una sección de Información de Sincronización (SI), que contiene (tal como se muestra en la Figura 5) una palabra de sincronización (SW); una sección de Información de Flujo binario (BSI), que contiene la mayoría de los metadatos; entre uno y seis Bloques de Audio (AB0 a AB5), que tienen contenidos de audio comprimidos de datos (y, además, pueden incluir metadatos); segmentos de bits residuales (W) (también conocidos como "campos de omisión"), que contienen cualesquiera bits no utilizados que quedan después de comprimir el contenido de audio (aunque solamente se ilustra un segmento de bit residual, un segmento de bit residual diferente, o segmento de campo de omisión, normalmente seguiría cada bloque de audio); una sección de información Auxiliar (AUX) que puede incluir más metadatos; y una palabra de corrección de error (CRC).

60 En un flujo binario AC-3 (o E-AC-3), existen varios parámetros de metadatos de audio que están previstos, específicamente, a ser utilizados para cambiar el sonido del programa que se proporciona a un entorno de escucha. Uno de los parámetros de metadatos es el parámetro DIALNORM, que está incluido en el segmento BSI.

65 Según se ilustra en la Figura 6, el segmento BSI de una trama AC-3 incluye un parámetro de cinco bits ("DIALNORM"), que indica el valor de DIALNORM para el programa. Se incluye un parámetro de cinco bits

("DIALNORM2"), que indica el valor de DIALNORM para un segundo programa de audio que se transmite en la misma trama AC-3 si el modo de codificación de audio ("acmod"), de la trama AC-3, es "0", lo que indica que se está utilizando una configuración de canal dual-mono o "1+1".

5 El segmento BSI incluye, además, un indicador ("addbsie"), que indica la presencia, (o ausencia), de información de flujo binario adicional después del bit "addbsie", un parámetro ("addbsil"), que indica la longitud de cualquier información adicional de flujo binario, sigue el valor "addbsil", y hasta 64 bits de información de flujo binario adicional ("addbsi") después del valor "addbsil".

10 El segmento BSI incluye otros valores de metadatos que no se ilustran, específicamente, en la Figura 6.

De conformidad con una clase de formas de realización, un flujo binario de audio codificado es indicativo de múltiples flujos secundarios de contenido de audio. En algunos casos, los flujos secundarios son indicativos del contenido de audio de un programa multicanal, y cada uno de los flujos secundarios indica uno o más de los canales del programa. En otros casos, múltiples flujos secundarios de un flujo binario de audio codificado, son indicativos de contenido de audio de varios programas de audio, normalmente un programa de audio "principal" (que puede ser un programa multicanal), y al menos otro programa de audio (p.ej., un programa que es un comentario sobre el programa de audio principal).

20 Un flujo binario de audio codificado, que es indicativo de al menos un programa de audio incluye, necesariamente, al menos un flujo secundario "independiente" de contenido de audio. El flujo secundario independiente es indicativo de al menos un canal de un programa de audio (p.ej., el flujo secundario independiente puede ser indicativo de los cinco canales de margen completo de un programa de audio de canal 5.1 convencional). En este caso, este programa de audio se refiere como un programa "principal".

25 En algunas clases de formas de realización, un flujo binario de audio codificado es indicativo de dos o más programas de audio (un programa "principal" y al menos otro programa de audio). En tales casos, el flujo binario incluye dos o más flujos secundarios independientes: un primer flujo secundario independiente, que indica al menos un canal del programa principal; y al menos otro flujo secundario independiente, indicativo de al menos un canal de otro programa de audio (un programa distinto del programa principal). Cada flujo binario independiente se puede decodificar, de forma independiente, y un decodificador podría funcionar para decodificar solamente un subconjunto (no la totalidad) de los flujos secundarios independientes de un flujo binario codificado.

30 En un ejemplo típico de un flujo binario de audio codificado, que es indicativo de dos flujos secundarios independientes, uno de los flujos secundarios independientes es indicativo de canales de altavoz de formato estándar de un programa principal multicanal (p.ej., Izquierda, Derecha, Centro, Izquierda Envoltente, Derecha Envoltente como canales de altavoz de gama completa de un programa principal de 5.1 canales), y el otro flujo secundario independiente es indicativo de un comentario de audio monofónico en el programa principal (p.ej., un comentario de un director sobre una película, en donde el programa principal es la banda sonora de la película). En otro ejemplo de un flujo binario de audio codificado, que indica múltiples flujos secundarios independientes, uno de los flujos secundarios independientes es indicativo de canales de altavoz de formato estándar de un programa principal multicanal (p.ej., un programa principal de 5.1 canales), que incluye diálogo en un primer idioma (p.ej., uno de los canales de altavoz del programa principal puede ser indicativo del diálogo), y cada uno de los otros flujos secundarios independientes es indicativo de una traducción monofónica (en un idioma diferente) del diálogo.

45 Opcionalmente, un flujo binario de audio codificado, que es indicativo de un programa principal (y, de forma opcional, también de al menos otro programa de audio) incluye al menos un flujo secundario "dependiente" de contenido de audio. Cada flujo secundario dependiente está asociado con un flujo secundario independiente del flujo binario, y es indicativo de al menos un canal adicional del programa (p.ej., el programa principal), cuyo contenido está indicado por el flujo secundario independiente asociado (es decir, el flujo secundario dependiente que indica al menos un canal de un programa que no está indicado por el flujo secundario independiente asociado, y el flujo secundario independiente asociado, que indica al menos un canal del programa).

50 En un ejemplo de un flujo binario codificado que incluye un flujo secundario independiente (indicativo de al menos un canal de un programa principal), el flujo binario incluye, además, un flujo secundario dependiente (asociado con el flujo binario independiente), que es indicativo de uno o más canales de altavoz del programa principal. Dichos canales de altavoz adicional, son adicionales a los canales del programa principal indicados por el flujo secundario independiente. A modo de ejemplo, si el flujo secundario independiente es indicativo de los formatos estándar izquierdo, derecho, central, izquierdo envolvente, derecho envolvente, como canales de altavoz de gama completa de un programa principal de 7.1 canales, pudiendo el flujo secundario dependiente ser indicativo de los otros dos canales de altavoz de gama completa del programa principal.

60 De conformidad con la norma E-AC-3, un flujo binario E-AC-3 debe ser indicativo de al menos un flujo secundario independiente (p.ej., un único flujo binario AC-3), y puede ser indicativo de hasta ocho flujos secundarios independientes. Cada flujo secundario independiente, de un flujo binario E-AC-3, puede asociarse con hasta ocho flujos secundarios dependientes.

Un flujo binario E-AC-3 incluye metadatos que indican la estructura de flujo secundario del flujo binario. A modo de ejemplo, un campo "chanmap", en la sección de Información de Flujo Binario (BSI), de un flujo binario E-AC-3, determina un mapa de canales para los canales de programa indicados por un flujo secundario dependiente del flujo binario. Sin embargo, los metadatos indicativos de la estructura de flujo secundario se incluyen, de modo convencional, en un flujo binario E-AC-3, en un formato tal que es conveniente para el acceso y uso (durante la decodificación del flujo binario codificado E-AC-3) solamente por un decodificador E-AC-3; no para el acceso y uso después de la decodificación (p.ej., por un post-procesador), o antes de la decodificación (p.ej., por un procesador configurado para reconocer los metadatos). Además, existe el riesgo de que un decodificador identifique, incorrectamente, los flujos secundarios de un flujo binario codificado E-AC-3 convencional, utilizando los metadatos incluidos convencionalmente, y hasta la presente invención no se ha tenido conocimiento de cómo incluir metadatos de estructura de flujo secundario en un flujo binario codificado (p.ej., un flujo binario E-AC-3 codificado) en un formato tal que permita la detección y corrección, conveniente y eficiente, de errores en la identificación del flujo secundario durante la decodificación del flujo binario.

Un flujo binario E-AC-3 puede incluir, además, metadatos con respecto al contenido de audio de un programa de audio. A modo de ejemplo, un flujo binario E-AC-3, indicativo de un programa de audio, incluye metadatos indicativos de frecuencias mínima y máxima a las que se ha utilizado el procesamiento de extensión espectral (y la codificación de acoplamiento de canal), con el fin de codificar el contenido del programa. Sin embargo, tales metadatos se suelen incluir en un flujo binario E-AC-3 en un formato tal que es conveniente para el acceso y uso (durante la decodificación del flujo binario E-AC-3 codificado) solamente por un decodificador E-AC-3; no para el acceso y uso después de la decodificación (p.ej., por un post-procesador), o antes de la decodificación (p.ej., por un procesador configurado para reconocer los metadatos). Además, dichos metadatos no se incluyen en un flujo binario E-AC-3 en un formato que permita la detección de error y la corrección de error conveniente y eficiente de dichos metadatos durante la decodificación del flujo binario.

De conformidad con las formas de realización típicas de la invención, metadatos PIM y/o SSM (y opcionalmente, también otros metadatos, a modo de ejemplo, metadatos de estado de procesamiento de sonoridad o "LPSM") están incluidos en uno o más campos reservados (o ranuras) de segmentos de metadatos de un flujo binario de audio que incluye, además, datos de audio en otros segmentos (segmentos de datos de audio). En condiciones normales, al menos un segmento de cada trama del flujo binario incluye PIM o SSM, y al menos otro segmento de la trama incluye datos de audio correspondientes (es decir, datos de audio cuya estructura de flujo secundario está indicada por SSM y/o tiene al menos una característica o propiedad indicada por los metadatos PIM).

En una clase de formas de realización, cada segmento de metadatos es una estructura de datos (a veces referida aquí como un contenedor) que puede incluir una o más cargas útiles de metadatos. Cada carga útil incluye una cabecera que comprende un identificador de carga útil específico (y datos de configuración de carga útil), para proporcionar una indicación inequívoca del tipo de metadatos presentes en la carga útil. El orden de las cargas dentro del contenedor no está definido, de modo que las cargas útiles se pueden memorizar en cualquier orden, y un analizador sintáctico debe poder analizar el contenedor completo con el fin de extraer cargas útiles pertinentes e ignorar las cargas útiles que no son pertinentes o son incompatibles. La Figura 8 (que se describirá a continuación) ilustra la estructura de dicho contenedor y cargas útiles dentro del contenedor.

La comunicación de metadatos (p.ej., SSM y/o PIM y/o LPSM) en una cadena de procesamiento de datos de audio, es particularmente útil cuando dos o más unidades de procesamiento de audio necesitan funcionar 'en tándem' entre sí a lo largo de la cadena de procesamiento (o ciclo de vida del contenido). Sin la inclusión de metadatos en un flujo binario de audio, pueden producirse graves problemas de procesamiento multimedia tales como calidad, nivel y degradaciones espaciales, a modo de ejemplo, cuando se utilizan dos o más códecs de audio en la cadena y la nivelación de volumen de terminación única se aplica más de una vez durante una ruta de flujo binario a un dispositivo que consume multimedia (o un punto de representación del contenido de audio del flujo binario).

Los metadatos de estado de procesamiento de sonoridad (LPSM), que se incluyen en un flujo binario de audio de conformidad con algunas formas de realización de la invención, se pueden autenticar y validar, a modo de ejemplo, para permitir a las entidades reguladoras de sonoridad la comprobación de si la sonoridad de un programa particular está ya dentro de un margen especificado, y que los propios datos de audio correspondientes no han sido modificados (con lo que se asegura el cumplimiento de la normativa aplicable). Un valor de sonoridad, incluido en un bloque de datos que comprende los metadatos de estado de procesamiento de sonoridad, puede ser objeto de lectura para la verificación de lo que antecede, en lugar de calcular la sonoridad de nuevo. En respuesta a LPSM, una agencia reguladora puede determinar que el contenido de audio correspondiente está en cumplimiento (según lo indicado por el LPSM) con requisitos reglamentarios y/o estatutarios de intensidad sonora (p.ej., las normativas promulgadas bajo la Ley de Mitigación de la Sonoridad de Anuncios Comerciales, también conocida como la Ley "CALM") sin la necesidad de calcular la sonoridad del contenido de audio.

La Figura 1 es un diagrama de bloques, a modo de ejemplo, de una cadena de procesamiento de audio (un sistema de procesamiento de datos de audio), en el que uno o más de los elementos del sistema se pueden configurar de conformidad con una forma de realización de la presente invención. El sistema incluye los siguientes elementos,

acoplados juntos, Según se ilustra: una unidad de procesamiento previo, un codificador, una unidad de corrección de metadatos y análisis de señal, un transcodificador, un decodificador y una unidad de procesamiento previo. En las variaciones del sistema ilustrado, se omiten uno o más de los elementos, o se incluyen unidades adicionales de procesamiento de datos de audio.

5 En algunas puestas en práctica, la unidad de procesamiento previo de la Figura 1 está configurada para aceptar muestras de PCM (dominio de tiempo), que comprenden contenido de audio como entrada, y para emitir muestras de PCM procesadas. El codificador se puede configurar para aceptar las muestras de PCM como entrada, y para proporcionar, a la salida, un flujo binario de audio codificado (p.ej., comprimido), que indica el contenido de audio.
10 Los datos del flujo binario que son indicativos del contenido de audio a veces se denominan aquí como "datos de audio". Si el codificador está configurado de conformidad con una forma de realización típica, la salida de flujo binario de audio, desde el codificador, incluye PIM y/o SSM (y opcionalmente, además, metadatos de estado de procesamiento de sonoridad y/o otros metadatos) así como datos de audio.

15 La unidad de corrección de metadatos y análisis de señal de la Figura 1 puede aceptar uno o más flujos binarios de audio codificados como entrada, y determinar (p.ej., validar), si los metadatos (p.ej., metadatos de estado de procesamiento), en cada flujo binario de audio codificado son correctos, realizando análisis de señal (p.ej., utilizando metadatos de límite de programa en un flujo binario de audio codificado). Si la unidad de corrección de metadatos y análisis de señal encuentra que los metadatos incluidos no son válidos, en condiciones normales, sustituye los valores incorrectos con los valores correctos, obtenidos a partir del análisis de señal. De este modo, cada flujo binario de audio codificado, proporcionado, a la salida, desde la unidad de corrección de metadatos y análisis de señal, puede incluir metadatos de estado de procesamiento corregidos, (o no corregidos), así como datos de audio codificados.

25 El transcodificador de la Figura 1 puede aceptar flujos binarios de audio codificados como entrada, y a la salida, responder con flujos binarios de audio modificados (p.ej., codificados de modo distinto) (p.ej., decodificando un flujo de entrada y recodificando el flujo decodificado en un formato de codificación diferente). Si el transcodificador está configurado de conformidad con una forma de realización típica, el flujo binario de audio, proporcionado a la salida, procedente del transcodificador, incluye SSM y/o PIM (y además, normalmente otros metadatos) así como datos de audio codificados. Los metadatos pueden haberse incluido en el flujo binario de entrada.

35 El decodificador de la Figura 1 puede aceptar flujos binarios de audio codificados (p.ej., comprimidos) como entrada y, a la salida (en respuesta) flujos de muestras de audio PCM decodificadas. Si el decodificador está configurado de conformidad con una forma de realización típica, la salida del decodificador, en funcionamiento típico, es o incluye cualquiera de lo que sigue:

un flujo de muestras de audio, y al menos un flujo correspondiente de metadatos SSM y/o PIM (y en condiciones normales, también otros metadatos), extraídos de un flujo binario codificado de entrada; o

40 un flujo de muestras de audio, y un flujo correspondiente de bits de control determinados a partir de SSM y/o PIM (y, normalmente, también otros metadatos, a modo de ejemplo, LPSM), extraídos de un flujo binario codificado de entrada; o

45 un flujo de muestras de audio, sin un flujo correspondiente de metadatos o bits de control, determinados a partir de metadatos. En este último caso, el decodificador puede extraer metadatos a partir del flujo binario codificado de entrada y realizar al menos una operación en los metadatos extraídos (p.ej., validación), aun cuando no proporciona a la salida los metadatos extraídos o bits de control determinados a partir de los mismos.

50 Mediante la configuración de la unidad de procesamiento posterior de la Figura 1 de conformidad con una forma de realización típica, la unidad de procesamiento posterior está configurada para aceptar un flujo de muestras de audio PCM decodificadas, y para realizar un procesamiento posterior sobre el mismo (p.ej., nivelación de volumen del contenido de audio) utilizando SSM y/o PIM (y, normalmente, también otros metadatos, p.ej., LPSM), recibidos con las muestras, o bits de control determinados por el decodificador a partir de los metadatos recibidos con las muestras. La unidad de post-procesamiento está configurada normalmente, además, para procesar el contenido de audio procesado posteriormente para su reproducción por uno o más altavoces.

60 Las formas de realización típicas dan a conocer una cadena de procesamiento de audio mejorada, en donde unidades de procesamiento de audio (p.ej., codificadores, decodificadores, transcodificadores y unidades de procesamiento previo y posterior), adaptan su respectivo procesamiento para aplicarlo a datos de audio de conformidad con un estado contemporáneo de los datos multimedia, según lo indicado por los metadatos recibidos, respectivamente, por las unidades de procesamiento de audio.

65 La entrada de datos de audio a cualquier unidad de procesamiento de audio del sistema de la Figura 1 (p.ej., el codificador o transcodificador de la Figura 1) puede incluir metadatos SSM y/o PIM (y, opcionalmente, también otros metadatos) así como datos de audio (p.ej., datos de audio codificados). Estos metadatos pueden haber sido incluidos en el audio de entrada por otro elemento del sistema de la Figura 1 (u otra fuente, no ilustrada en la Figura

1), de conformidad con una forma de realización. La unidad de procesamiento que recibe el audio de entrada (con metadatos) se puede configurar para realizar al menos una operación en los metadatos (p.ej., validación), o en respuesta a los metadatos (p.ej., procesamiento adaptativo del audio de entrada) y, normalmente, incluir, además, en su audio de salida los metadatos, una versión procesada de los metadatos, o bits de control determinados a partir de los metadatos.

Una forma de realización típica de la unidad de procesamiento de audio (o procesador de audio) está configurada para realizar un procesamiento adaptativo de datos de audio, sobre la base del estado de los datos de audio, tal como se indica por los metadatos correspondientes a los datos de audio. En algunas formas de realización, el procesamiento adaptativo es, (o incluye), procesamiento de sonoridad (si los metadatos indican que el procesamiento de sonoridad, o procesamiento similar al mismo, aún no se ha realizado en los datos de audio, pero no es (y no incluye) procesamiento de sonoridad (si los metadatos indican que dicho procesamiento de sonoridad, o procesamiento similar al mismo, ya se ha realizado en los datos de audio). En algunas formas de realización, el procesamiento adaptativo es, o incluye, la validación de metadatos (p.ej., realizada en una sub-unidad de validación de metadatos) con el fin de garantizar que la unidad de procesamiento de audio realice otro procesamiento adaptativo de los datos de audio sobre la base del estado de los datos de audio, según lo indicado por los metadatos. En algunas formas de realización, la validación determina la fiabilidad de los metadatos asociados (p.ej., incluidos en un flujo binario con) los datos de audio. A modo de ejemplo, si los metadatos se validan para ser fiables, en ese caso, los resultados de un tipo de procesamiento de audio previamente realizado, se pueden reutilizar y se puede evitar una nueva ejecución del mismo tipo de procesamiento de audio. Por otro lado, si se descubre que los metadatos han sido manipulados (o de otro modo, no fiables) entonces, se puede repetir el tipo de procesamiento multimedia supuestamente realizado con anterioridad por la unidad de procesamiento de audio sobre los metadatos y/o los datos de audio. La unidad de procesamiento de audio puede, además, estar configurada para señalar a otras unidades de procesamiento de audio, en flujo descendente en una cadena de procesamiento multimedia mejorada, que los metadatos (p.ej., presentes en un flujo binario multimedia) son válidos, si la unidad determina que los metadatos son válidos (p.ej., sobre la base de una coincidencia de un valor criptográfico extraído y un valor criptográfico de referencia).

La Figura 2 es un diagrama de bloques de un codificador (100) que es una forma de realización de la unidad de procesamiento de audio. Cualquiera de los componentes o elementos del codificador 100 se puede poner en práctica como uno o más procesos y/o uno o más circuitos (p.ej., ASICs, FPGAs u otros circuitos integrados) en hardware, software o una combinación de hardware y software. El codificador 100 comprende una memoria intermedia de trama 110, un analizador sintáctico 111, un decodificador 101, un validador de estado de audio 102, una etapa de procesamiento de sonoridad 103, una etapa de selección de flujo de audio 104, un codificador 105, una etapa de rellenador/formateador 107, una etapa de generación de metadatos 106, un subsistema de medición de sonoridad de diálogo 108 y una memoria intermedia de trama 109, conectados según se ilustra. En condiciones normales, el codificador 100 incluye, además, otros elementos de procesamiento (no ilustrados).

El codificador 100 (que es un transcodificador) está configurado para convertir un flujo binario de audio de entrada (que, a modo de ejemplo, puede ser uno de entre un flujo binario AC-3, un flujo binario E-AC-3 o un flujo binario Dolby E), en un flujo binario de audio de salida codificado (que, a modo de ejemplo, puede ser otro de entre un flujo binario AC-3, un flujo binario E-AC-3 o un flujo binario Dolby E) que incluye la realización de un procesamiento de sonoridad automatizado y adaptativo, utilizando metadatos de estado de procesamiento de sonoridad incluidos en el flujo binario de entrada. A modo de ejemplo, el codificador 100 se puede configurar para convertir un flujo binario Dolby E de entrada (un formato normalmente utilizado en instalaciones de producción y difusión, pero no en dispositivos de consumo que reciben programas de audio que han sido difundidos) a un flujo binario de audio codificado de salida (adecuado para la difusión a dispositivos de consumo) en formato AC-3 o E-AC-3.

El sistema de la Figura 2 incluye, además, el subsistema de entrega de audio codificado 150 (que memoriza y/o entrega, los flujos binarios codificados procedentes del codificador 100) y el decodificador 152. El subsistema 150 puede memorizar un flujo binario de audio codificado, de salida, procedente del codificador 100 (p.ej., en la forma de un DVD o disco de Blu-ray), o transmitido por el subsistema 150 (que puede poner en práctica un enlace de transmisión o red), o puede ser, a la vez, memorizado y transmitido por el subsistema 150. El decodificador 152 está configurado para decodificar un flujo binario de audio codificado (generado por el codificador 100), que se recibe a través del subsistema 150, incluyendo mediante la extracción de metadatos (PIM y/o SSM y, opcionalmente, también metadatos de estado de procesamiento de sonoridad u otros metadatos) desde cada trama del flujo binario (y, como opción, extrayendo, además, metadatos de límite de programa del flujo binario), y generando datos de audio decodificados. En condiciones normales, el decodificador 152 está configurado para realizar un procesamiento adaptativo sobre los datos de audio decodificados utilizando PIM y/o SSM, y/o LPSM (y, de modo opcional, también metadatos de límite de programa), y/o para reenviar los datos de audio decodificados y los metadatos a un post-procesador configurado para realizar un procesamiento adaptativo en los datos de audio decodificados utilizando los metadatos. Normalmente, el decodificador 152 incluye una memoria intermedia que memoriza (p.ej., de manera no transitoria) el flujo binario de audio codificado, que se recibe desde el subsistema 150.

Diversas puestas en práctica del codificador 100, y el decodificador 152, están configuradas para realizar diferentes formas de realización.

La memoria intermedia de trama 110 es una memoria intermedia acoplada para recibir un flujo binario de audio de entrada codificado. En funcionamiento, la memoria intermedia 110 memoriza (p.ej., de forma no transitoria), al menos una trama del flujo binario de audio codificado, y una secuencia de las tramas del flujo binario de audio codificada se reafirma desde la memoria intermedia 110 al analizador 111.

El analizador 111 está acoplado y configurado para extraer metadatos PIM y/o SSM, y metadatos de estado de procesamiento de sonoridad (LPSM) y, opcionalmente, también metadatos de límite de programa (y/u otros metadatos) desde cada trama del audio de entrada codificado en donde se incluyen dichos metadatos, para establecer al menos el LPSM (y, opcionalmente, también metadatos de límite de programa y/u otros metadatos) al validador de estado de audio 102, etapa de procesamiento de sonoridad 103, etapa 106 y subsistema 108, para extraer datos de audio del audio de entrada codificado y para establecer los datos de audio al decodificador 101. El decodificador 101, del codificador 100, está configurado para decodificar los datos de audio para generar datos de audio decodificados, y para establecer los datos de audio decodificados a la etapa de procesamiento de sonoridad 103, etapa de selección de flujo de audio 104, subsistema 108, y normalmente, también para el validador de estado 102.

El validador de estado 102 está configurado para autenticar y validar el LPSM (y, opcionalmente, otros metadatos) establecido en el mismo. En algunas formas de realización, el LPSM es, (o está incluido en), un bloque de datos que se ha incluido en el flujo binario de entrada (p.ej., de conformidad con una forma de realización de la presente invención). El bloque puede incluir un denominado *hash* criptográfico (un código de autenticación de mensaje basado en *hash* o "HMAC") para el procesamiento del LPSM (y, opcionalmente, también otros metadatos) y/o los datos de audio subyacentes (que se proporcionan a partir del decodificador 101, al validador 102). El bloque de datos puede estar firmado digitalmente en estas formas de realización, de modo que una unidad de procesamiento de audio de flujo del pueda autenticar y validar, con relativa facilidad, los metadatos del estado de procesamiento.

A modo de ejemplo, el HMAC se usa para generar un *digest*, y los valores de protección incluidos en el flujo binario de la invención pueden incluir el *digest*. El *digest* se puede generar del modo siguiente para una trama AC-3:

1. Después de que se codifiquen los datos de AC-3 y LPSM, bytes de datos de trama (concatenadas *frame_date* #1 y *frame_data* #2) y los bytes de datos de LPSM se utilizan como entrada para la función HMAC basada en la función denominada *hash*. Otros datos, que pueden estar presentes dentro de un campo *auxdata*, no se tienen en cuenta para calcular el *digest*. Dichos otros datos pueden ser bytes que no pertenecen a los datos AC-3, ni a los datos de LPSM. Bits de protección, incluidos en LPSM, pueden no ser considerados para el cálculo del *digest* de HMAC.

2. Después de calcular el *digest*, se escribe en el flujo binario en un campo reservado para bits de protección.

3. La última etapa de la generación de la trama AC-3 completa es el cálculo de la verificación CRC. Esto se escribe el extremo final de la trama, y se tienen en cuenta todos los datos que pertenecen a esta trama, incluidos los bits de LPSM.

Otros métodos criptográficos que incluyen, pero no se limitan a, uno o más métodos criptográficos no de HMAC, se pueden utilizar para la validación de LPSM y/u otros metadatos (p.ej., en el validador 102) con el fin de asegurar la transmisión y recepción segura de los metadatos y/o los datos de audio subyacentes. A modo de ejemplo, la validación (utilizando dicho método criptográfico) se puede realizar en cada unidad de procesamiento de audio que recibe, en una forma de realización de la invención, el flujo binario de audio para determinar si los metadatos y los datos de audio correspondientes, incluidos en el flujo binario se han sometido (y/o son el resultado de) al procesamiento específico (como lo indican los metadatos) y no se han modificado después la realización de dicho procesamiento específico.

El validador de estado 102 establece datos de control a la etapa de selección de flujo de audio 104, el generador de metadatos 106 y el subsistema de medición de sonoridad de diálogo 108, para indicar los resultados de la operación de validación. En respuesta a los datos de control, la etapa 104 puede seleccionar (y pasar a través del codificador 105) ya sea:

la salida procesada adaptativamente de la etapa de procesamiento de sonoridad 103 (p.ej., cuando LPSM indica que la salida de datos de audio desde el decodificador 101 no ha realizado un tipo específico de procesamiento de sonoridad, y los bits de control procedentes del validador 102 indican que los LPSM son válidos); o

los datos de audio de salida del decodificador 101 (p.ej., cuando LPSM indica que los datos de audio, a la salida del decodificador 101, ya ha experimentado el tipo específico de procesamiento de sonoridad que se realizaría en la etapa 103, y los bits de control del validador 102 indican que los LPSM son válidos).

La etapa 103 del codificador 100 está configurada para poner en práctica un procesamiento de sonoridad adaptativo, sobre los datos de audio decodificados, a la salida del decodificador 101, sobre la base de una o más características de datos de audio que se indican por LPSM, extraídas por el decodificador 101. La etapa 103 puede ser un dominio-

transformación adaptativa de sonoridad en tiempo real y un procesador de control de margen dinámico. La etapa 103 puede recibir una entrada del usuario (p.ej., valores de sonoridad objetivo del usuario/margen dinámico o valores del parámetro dialnorm) u otra entrada de metadatos (p.ej., uno o más tipos de datos de terceros, información de seguimiento, identificadores, información de propiedad o estándar, datos de anotación del usuario, datos de preferencia del usuario, etc.) y/u otra entrada (p.ej., desde un proceso de huella dactilar), y utilizar dicha entrada para procesar los datos de audio decodificados, a la salida del decodificador 101. La etapa 103 puede realizar un procesamiento de sonoridad adaptativo sobre datos de audio decodificados (a la salida del decodificador 101), indicativo de un único programa de audio (como se indica por los metadatos de límite de programa extraídos por el analizador sintáctico 111) y puede reiniciar el procesamiento de sonoridad en respuesta a la recepción de datos de audio decodificados (salida del decodificador 101), que indican un programa de audio diferente, tal como se indica por metadatos de límite de programa extraídos por el analizador sintáctico 111.

El subsistema de medición de sonoridad de diálogo 108 puede funcionar para determinar la sonoridad de segmentos del audio decodificado (procedente del decodificador 101) que son indicativos de diálogo (u otra expresión vocal), a modo de ejemplo, utilizando LPSM (y/u otros metadatos) extraídos por el decodificador 101, cuando los bits de control del validador 102 indican que los LPSM no son válidos. La operación del subsistema de medición de sonoridad de diálogo 108 puede desactivarse cuando los LPSM indican, previamente, una determinada sonoridad de segmentos de diálogo (u otra expresión vocal) del audio decodificado (desde el decodificador 101) cuando los bits de control del validador 102 indican que los LPSM son válidos. El subsistema 108 puede realizar una medición de sonoridad en datos de audio decodificados, indicativos de un único programa de audio (según lo indicado por los metadatos de límites del programa extraídos por el analizador 111), y puede reiniciar la medición en respuesta a la recepción de datos de audio decodificados indicativos de un programa de audio diferente, tal como se indica por tales metadatos de límites del programa.

Existen herramientas útiles (p.ej., el medidor de sonoridad Dolby LM100) para medir el nivel del diálogo en contenido de audio, de manera conveniente y fácil. Algunas formas de realización de la APU (p.ej., etapa 108 del codificador 100) se ponen en práctica para incluir (o para realizar las funciones de) dicha herramienta para medir la sonoridad de diálogo media del contenido de audio de un flujo binario de audio (p.ej., un flujo binario AC-3 decodificado establecido en la etapa 108 del decodificador 101, del codificador 100).

Si la etapa 108 se pone en práctica para medir la sonoridad de diálogo media real de los datos de audio, la medición puede incluir una etapa de aislamiento de segmentos del contenido de audio que contienen, de forma predominante, voz. Los segmentos de audio predominantemente hablados se procesan, entonces, de conformidad con un algoritmo de medición de sonoridad. Para datos de audio decodificados a partir de un flujo binario AC-3, este algoritmo puede ser una medida de sonoridad ponderada K estándar (de conformidad con la norma internacional ITU-R BS.1770). Como alternativa, se pueden utilizar otras medidas de sonoridad (p.ej., aquellas basadas en modelos psico-acústicos de sonoridad).

El aislamiento de los segmentos de voz no es esencial para medir la sonoridad de diálogo media de los datos de audio. Sin embargo, mejora la precisión de la medida y, por lo general, proporciona resultados más satisfactorios desde la perspectiva del oyente. Puesto que no todo el contenido de audio incluye diálogo (expresión vocal), la medición de sonoridad del contenido de audio completo puede proporcionar una aproximación suficiente del nivel de diálogo del audio, si la expresión vocal hubiera estado presente.

El generador de metadatos 106 genera (y/o pasa a la etapa 107) metadatos para ser incluidos por la etapa 107 en el flujo binario codificado, a proporcionarse, a la salida, desde el codificador 100. El generador de metadatos 106 puede pasar a la etapa 107 los LPSM (y además, opcionalmente LIM y/o PIM y/o metadatos de límite de programa y/u otros metadatos) extraídos por el codificador 101 y/o el analizador sintáctico 111 (p.ej., cuando los bits de control procedentes del validador 102 indican que los LPSM y/u otros metadatos son válidos), o generan nuevos LIM y/o PIM y/o LPSM y/o metadatos de límite de programa y/u otros metadatos y establecer los nuevos metadatos a la etapa 107 (p.ej., cuando los bits de control del validador 102 indican que los metadatos extraídos por el decodificador 101 no son válidos) o puede establecer, en la etapa 107, una combinación de metadatos extraídos por el decodificador 101 y/o el analizador 111, y los metadatos recientemente generados. El generador de metadatos 106 puede incluir datos de sonoridad generados por el subsistema 108, y al menos un valor indicativo del tipo de procesamiento de sonoridad realizado por el subsistema 108, en LPSM que son establecidos en la etapa 107 para su inclusión en el flujo binario codificado que se enviará desde el codificador 100.

El generador de metadatos 106 puede generar bits de protección (que pueden consistir o incluir un código de autenticación de mensaje basado en el denominado *hash*, o "HMAC") útil para al menos una de entre las funciones de descryptación, autenticación o validación de los LPSM (y, opcionalmente, también otros metadatos) que han de incluirse en el flujo binario codificado, y/o los datos de audio subyacentes que han de incluirse en el flujo binario codificado. El generador de metadatos 106 puede proporcionar dichos bits de protección a la etapa 107 para su inclusión en el flujo binario codificado.

En una operación típica, el subsistema de medición de sonoridad de diálogo 108 procesa los datos de audio, a la salida del decodificador 101, para generar en respuesta a sus valores de sonoridad (p.ej., valores de sonoridad de

diálogo bloqueados y no vinculados) y valores de margen dinámico. En respuesta a estos valores, el generador de metadatos 106 puede generar metadatos de estado de procesamiento de sonoridad (LPSM) para su inclusión (por el relleno/formateador 107) en el flujo binario codificado que se enviará desde el codificador 100.

5 De forma adicional, opcionalmente o como alternativa, los subsistemas de 106 y/o 108 del codificador 100 pueden realizar un análisis adicional de los datos de audio para generar metadatos indicativos de al menos una característica de los datos de audio, para su inclusión en el flujo binario codificado que se proporciona, a la salida, desde la etapa 107.

10 El codificador 105 codifica (p.ej., realizando una compresión) los datos de audio emitidos desde la etapa de selección 104, y establece el audio codificado a la etapa 107 para su inclusión en el flujo binario codificado que se proporcionará, a la salida, desde la etapa 107.

15 La etapa 107 realiza la multiplexación del audio codificado procedente del codificador 105, y los metadatos (incluyendo PIM y/o SSM) del generador 106, para generar el flujo binario codificado que se emitirá desde la etapa 107, preferiblemente de modo que el flujo binario codificado tenga el formato que se especifica mediante una forma de realización preferida de la presente invención.

20 La memoria intermedia de trama 109 es una memoria intermedia que memoriza, (p.ej., de forma no transitoria), al menos una trama del flujo binario de audio codificado a la salida de la etapa 107 y, a continuación, una secuencia de las tramas del flujo binario de audio codificado se establecido desde la memoria intermedia 109 como salida del codificador 100 al sistema de entrega 150.

25 Los metadatos LPSM que se generan por el generador de metadatos 106, y se incluyen en el flujo binario codificado por la etapa 107 suelen ser indicativos del estado de procesamiento de sonoridad de los datos de audio correspondientes (p.ej., qué tipos de procesamientos de sonoridad se han realizado sobre los datos de audio) y sonoridad (p.ej., sonoridad de diálogo medida, sonoridad bloqueada y/o no bloqueada, y/o margen dinámico) de los correspondientes datos de audio.

30 En este caso, "bloqueo" de sonoridad y/o mediciones de nivel realizadas en datos de audio, se refiere a un nivel específico o umbral de sonoridad, en donde los valores calculados que superen el umbral se incluyen en la medición final (p.ej., ignorando los valores de sonoridad a corto plazo por debajo de -60 dBFS en los valores de medición final). Dicha función *gating* aplicada sobre un valor absoluto se refiere a un nivel fijo o sonoridad, mientras que dicha misma función aplicada a un valor relativo se refiere a un valor que depende de un valor de medición "no bloqueado"
35 actual.

En algunas puestas en práctica del codificador 100, el flujo binario codificado que se memoriza en memoria intermedia 109 (y se proporciona, al a salida, al sistema de distribución 150), es un flujo binario AC-3 o un flujo binario E-AC-3, y comprende segmentos de datos de audio (p.ej., los segmentos AB0 -AB5 de la trama ilustrada en la Figura 4) y segmentos de metadatos, en donde los segmentos de datos de audio son indicativos de datos de audio, y cada uno de al menos algunos de los segmentos de metadatos incluye PIM y/o SSM (y, opcionalmente, también otros metadatos). La etapa 107 establece segmentos de metadatos (incluidos los metadatos) en el flujo binario en el siguiente formato. Cada uno de los segmentos de metadatos que incluye PIM y/o SSM se incluye en un segmento de bits residuales del flujo binario (p.ej., un segmento de bits residuales "W", tal como se muestra en la Figura 4 o Figura 7) o un campo "addbsi" del segmento de Información de Flujo Binario ("BSI") de una trama del flujo binario, o en un campo auxdata (p.ej., el segmento AUX, ilustrado en la Figura 4 o Figura 7) al final de una trama del flujo binario. Una trama del flujo binario puede incluir uno o dos segmentos de metadatos, cada uno de los cuales incluye metadatos, y si la trama incluye dos segmentos de metadatos, uno puede estar presente en el campo addbsi de la trama y el otro en el campo AUX de la trama. De conformidad con la forma de realización de la invención, al menos un segmento de metadatos, que incluye los metadatos PIM, se incluye en un segmento de bits residuales (campo de omisión) del flujo binario.
40
45
50

En algunas formas de realización, cada segmento de metadatos (a veces referido aquí como un "contenedor") establecidos por la etapa 107 tiene un formato que incluye una cabecera de segmento de metadatos (y, opcionalmente, también otros elementos obligatorios o "principales"), y una o más cargas útiles de metadatos, siguiendo la cabecera del segmento de metadatos. SIM, si está presente, se incluye en una de las cargas útiles de metadatos (identificada por una cabecera de carga útil y, por lo general, tiene un formato de un primer tipo). PIM, de conformidad con una forma de realización de la invención, está incluido en otra de las cargas útiles de metadatos (identificada por una cabecera de carga útil y que, típicamente, tiene un formato de un segundo tipo). De modo similar, cada otro tipo de metadatos (si está presente) se incluye en otra de las cargas útiles de metadatos (que se identifica por una cabecera de carga y, normalmente, tiene un formato específico para el tipo de metadatos). El formato, a modo de ejemplo, permite un acceso conveniente a los metadatos SSM, PIM y otros metadatos en momentos distintos a la duración de la decodificación (p.ej., mediante un post-procesador que sigue la decodificación, o mediante un procesador configurado para reconocer los metadatos sin realizar una decodificación completa en el flujo binario codificado), y permite la detección y corrección de errores, conveniente y eficiente, (p.ej., de identificación del flujo secundario) durante la decodificación del flujo binario. A modo de ejemplo, sin acceso a
55
60
65

SSM en el formato a modo de ejemplo, un decodificador podría identificar, de forma incorrecta, el número correcto de flujos secundarios asociados con un programa. Una carga útil de metadatos, en un segmento de metadatos, puede incluir SSM, otra carga útil de metadatos, en el segmento de metadatos, puede incluir PIM y, opcionalmente, además al menos otra carga útil de metadatos en el segmento de metadatos puede incluir otros metadatos (p.ej., metadatos de estado de procesamiento de sonoridad o "LPSM").

En algunas formas de realización, una carga útil de metadatos de estructura de flujo secundario (SSM) incluida (por la etapa 107), en una trama de un flujo binario codificado (p.ej., un flujo binario E-AC-3 indicativo de al menos un programa de audio), incluye SSM en el siguiente formato:

una cabecera de carga útil, que suele incluir al menos un valor de identificación (p.ej., un valor de 2 bits indicativo de la versión de formato SSM y, de forma opcional, valores adicionales de asociación de longitud, período, conteo y flujo secundario); y después de la cabecera:

metadatos de flujo secundario independientes, que indican el número de flujos secundarios independientes del programa, que se indica por el flujo binario; y

metadatos de flujo secundario dependientes, que indican si cada flujo secundario independiente del programa tiene al menos un flujo secundario dependiente asociado (es decir, si al menos un flujo secundario dependiente está asociado con dicho flujo secundario independiente), y de ser así, el número de flujos secundarios dependientes asociados con cada flujo secundario independiente del programa.

Conviene señalar que un flujo secundario independiente, de un flujo binario codificado, puede ser indicativo de un conjunto de canales de altavoz de un programa de audio (p.ej., los canales de altavoz de un programa de audio de canal de altavoz 5.1), y que cada uno de entre los uno o más flujos secundarios dependientes (asociados con el flujo secundario independiente, tal como se indica por los metadatos de flujo secundario dependiente) pueden ser indicativas de un canal de objeto del programa. En condiciones normales, sin embargo, un flujo secundario independiente de un flujo binario codificado, es indicativo de un conjunto de canales de altavoz de un programa, y cada flujo secundario dependiente, asociado con el flujo secundario independiente (tal como lo indican los metadatos de flujo secundario dependiente) es indicativo de al menos un canal de altavoz adicional del programa.

En algunas formas de realización de la invención, una carga útil de metadatos de información sobre el programa (PIM), incluida (por la etapa 107) en una trama de un flujo binario codificado, (p.ej., un flujo binario E-AC-3 indicativo de al menos un programa de audio) tiene el siguiente formato:

una cabecera de carga útil, que suele incluir, al menos un valor de identificación (p.ej., un valor indicativo de la versión de formato PIM y, de forma opcional, también valores de asociación de longitud, período, conteo y flujo secundario); y

después de la cabecera, metadatos PIM en el siguiente formato:

metadatos de canales activos, indicativos de cada canal silencioso, y cada canal no silencioso, de un programa de audio (es decir, qué canales del programa contienen información de audio y cuáles (si los hay) contienen solamente silencio (normalmente para la duración de la trama)). En formas de realización en las que el flujo binario codificado es un flujo binario AC-3 o E-AC-3, los metadatos de canal activo, en una trama del flujo binario, se pueden utilizar junto con metadatos adicionales del flujo binario (p.ej., el campo de modo de codificación de audio ("acmod") de la trama y, si está presente, el campo 'chanmap' en la trama o las tramas de flujo secundario dependiente asociadas) para determinar qué canales del programa contienen información de audio y cuáles contienen silencio. El campo "acmod" de una trama AC-3 o E-AC-3 indica la cantidad de canales de margen completo de un programa de audio, indicado por el contenido de audio de la trama (p.ej., si el programa es un programa monofónico de canal 1.0, un programa estéreo de canal 2.0, o un programa que comprende canales de margen completo L, R, C, Ls, Rs), o que la trama es indicativa de dos programas monofónicos de canal 1.0 independientes. Un campo "chanmap" de un flujo binario E-AC-3 indica un mapa de canales para un flujo secundario dependiente, indicado por el flujo binario. Los metadatos de canal activo pueden ser útiles para poner en práctica la mezcla ascendente (en un post-procesador) en flujo descendente de un decodificador, a modo de ejemplo, para añadir audio a canales que contienen silencio en la salida del decodificador;

los metadatos de estado de procesamiento de mezcla descendente indican si el programa fue objeto de mezcla descendente (antes o durante la codificación), y si es así, el tipo de operación de mezcla descendente que se aplicó. Los metadatos de estado de procesamiento de mezcla descendente pueden ser útiles para poner en práctica una mezcla ascendente (en un post-procesador) en flujo descendente de un decodificador, a modo de ejemplo para la mezcla ascendente del contenido de audio del programa utilizando los parámetros que más se asemejan a un tipo de operación de mezcla descendente que se aplicó. En formas de realización en las que el flujo binario codificado es un flujo binario AC-3 o E-AC-3, los metadatos de estado de procesamiento de mezcla descendente pueden utilizarse junto con el campo de modo de codificación de audio ("acmod") de la trama para determinar el tipo de operación de mezcla descendente (si corresponde) aplicado a los canales del programa;

5 metadatos de estado de procesamiento de mezcla ascendente, indicativos de si el programa fue objeto de mezcla ascendente (p.ej., de un número menor de canales) antes, o durante, la codificación y, de ser así, el tipo de mezcla ascendente que se aplicó. Los metadatos de estado de procesamiento de mezcla ascendente pueden ser útiles para
 10 poner en práctica una operación de mezcla descendente (en un post-procesador) en flujo descendente de un decodificador, a modo de ejemplo para la mezcla descendente del contenido de audio del programa de un modo que sea compatible con un tipo de operación de mezcla ascendente (p.ej., Dolby Pro Logic, o Dolby Pro Logic II Movie Mode, o Dolby Pro Logic II Music Mode, o Dolby Professional Upmixer) que se aplicó al programa. En formas de realización en las que el flujo binario codificado es un flujo binario E-AC-3, los metadatos de estado de
 15 procesamiento de mezcla ascendente se pueden utilizar junto con otros metadatos (p.ej., el valor de un campo "strmtyp" de la trama), para determinar el tipo de operación de mezcla ascendente (si corresponde) fue aplicado a los canales del programa. El valor del campo "strmtyp" (en el segmento BSI de una trama de un flujo binario E-AC-3) indica si el contenido de audio de la trama pertenece a un flujo independiente (que determina un programa) o un flujo secundario independiente (de un programa que incluye, o está asociado con, múltiples flujos secundarios) y por lo tanto, puede decodificarse independientemente de cualquier otro flujo secundario indicado por el flujo binario E-AC-3, o si el contenido de audio de la trama pertenece a un flujo secundario dependiente (de un programa que incluye, o está asociado con, múltiples flujos secundarios) y, en consecuencia, debe decodificarse junto con un flujo secundario independiente con el que está asociado; y

20 metadatos de estado de pre-procesamiento, que indican si el pre-procesamiento se realizó en el contenido de audio de la trama (antes de la codificación del contenido de audio para generar el flujo binario codificado) y, en ese caso, el tipo de pre-procesamiento que se realizó.

25 En algunas realizaciones, los metadatos del estado de pre-procesamiento son indicativos de:

si se aplicó una atenuación envolvente (p.ej., si los canales envolventes del programa de audio fueron atenuados en 3 dB antes de la codificación),

30 si se aplicó un cambio de fase de 90 grados (p.ej., para canales Ls envolventes y canales Rs del programa de audio antes de la codificación),

si se aplicó un filtro de paso bajo a un canal LFE del programa de audio antes de la codificación,

35 si el nivel de un canal LFE del programa fue supervisado durante la producción y, de ser así, el nivel supervisado del canal LFE en relación con el nivel de los canales de audio de margen completo del programa,

40 si se debe realizar una compresión de margen dinámico (p.ej., en el decodificador) en cada bloque de contenido de audio decodificado del programa y, si es así, el tipo (y/o parámetros) de compresión de margen dinámico que ha de realizarse (p.ej., este tipo de estado de pre-procesamiento los metadatos pueden ser indicativos de cuál de los siguientes tipos de perfil de compresión fue asumido por el codificador para generar valores de control de compresión de margen dinámico que están incluidos en el flujo binario codificado: Película estándar, Luz de película, Música estándar, Luz musical o Voz. Como alternativa, este tipo de metadatos de estado de pre-procesamiento puede indicar que ha de realizarse la compresión de margen dinámico intensa (compresión "compr") en cada trama de contenido de audio decodificado del programa, de un modo que se determina por los valores de control de
 45 compresión de margen dinámico que se incluyen en el flujo binario codificado).

50 si el procesamiento de extensión espectral y/o codificación de acoplamiento de canal se empleó para codificar márgenes de frecuencia específicos del contenido del programa y, en caso afirmativo, las frecuencias mínima y máxima de los componentes de frecuencia del contenido en que se realizó la codificación de extensión espectral, y las frecuencias mínima y máxima de componentes de frecuencia del contenido en el que se realizó la codificación de acoplamiento de canal. Este tipo de información de metadatos de estado de pre-procesamiento puede ser útil para realizar la ecualización (en un post-procesador) en flujo descendente de un decodificador. Tanto el acoplamiento de canales como la información de extensión espectral son útiles, además, para optimizar la calidad durante las operaciones de trans-codificación y aplicaciones. A modo de ejemplo, un codificador puede optimizar su
 55 comportamiento (incluida la adaptación de etapas de pre-procesado, como la virtualización de auriculares, operación de mezcla ascendente, etc.) sobre la base del estado de los parámetros, tal como la extensión espectral y la información de acoplamiento de canal. Además, el codificador podría adaptar sus parámetros de acoplamiento y extensión espectral de forma dinámica para que coincidan y/o para valores óptimos, en función del estado de los metadatos entrantes (y autenticados), y

60 si los datos de margen de ajuste de mejora de diálogo se incluyen en el flujo binario codificado y, si es así, el margen de ajuste disponible durante la realización del procesamiento de mejora de diálogo (p.ej., en un post-procesador de flujo descendente de un decodificador) para ajustar el nivel de contenido de diálogo en relación al nivel de contenido sin diálogo en el programa de audio.

65 En algunas puestas en práctica, metadatos de estado de pre-procesamiento adicionales (p.ej., metadatos indicativos

de parámetros relacionados con auriculares) se incluyen (por la etapa 107) en una carga útil de PIM de un flujo binario codificado para proporcionarse, a la salida, desde el codificador 100.

5 En algunas formas de realización, una carga útil LPSM incluida (por la etapa 107) en una trama de un flujo binario codificado (p.ej., un flujo binario E-AC-3, indicativo de al menos un programa de audio) incluye LPSM en el siguiente formato:

10 una cabecera (que suele incluir, una palabra de sincronización que identifica el inicio de la carga útil de LPSM, seguida por al menos un valor de identificación, p.ej., la versión de formato LPSM, longitud, período, conteo y valores de asociación flujo secundario indicados en la Tabla 2 siguiente); y

después de la cabecera,

15 al menos un valor de indicación de diálogo (p.ej., parámetro "Canales de Diálogo" de la Tabla 2), que indica si los correspondientes datos de audio indican diálogo, o no indican diálogo, (p.ej., qué canales de los correspondientes datos de audio indican diálogo);

20 al menos un valor de cumplimiento de normativa de sonoridad (p.ej., parámetro "Tipo de Normativa de Sonoridad" de la Tabla 2), que indica si los correspondientes datos de audio cumplen con un conjunto indicado de normativas de sonoridad;

25 al menos un valor de procesamiento de sonoridad (p.ej., uno o más de los parámetros "Indicador de corrección de sonoridad bloqueada por diálogo", "Tipo de Corrección de Sonoridad" de la Tabla 2), que indica al menos un tipo de procesamiento de sonoridad que se ha realizado en los correspondientes datos de audio; y

30 al menos un valor de sonoridad (p. ej., uno o más de los parámetros "Sonoridad bloqueada relativa de ITU", "Sonoridad bloqueada de expresión vocal de ITU", "Sonoridad de corto plazo de 3s de ITU (EBU 3341)" y "Pico verdadero" de la Tabla 2), que indican al menos una característica de sonoridad (p.ej., pico o promedio de sonoridad) de los datos de audio correspondientes.

35 En algunas formas de realización, cada segmento de metadatos que contiene PIM y/o SSM (y opcionalmente, también otros metadatos) contiene una cabecera de segmento de metadatos (y, como opción, elementos principales adicionales) y después de la cabecera del segmento de metadatos (o la cabecera del segmento de metadatos y otros elementos principales), al menos un segmento de carga útil de metadatos que tiene el siguiente formato:

una cabecera de carga útil, que suele incluir, al menos un valor de identificación (p.ej., versión de formato SSM o PIM, longitud, período, conteo y valores de asociación de flujo secundario), y

40 después de la cabecera de la carga útil, los SSM o PIM (o metadatos de otro tipo).

45 En algunas realizaciones, cada uno de los segmentos de metadatos (a veces denominados en este documento como "contenedores de metadatos" o "contenedores") insertados en la etapa 107 en un segmento de campo de omisión/bit residual (o un campo "addbsi", o un campo auxdata) de una trama del flujo binario que tiene el siguiente formato:

una cabecera de segmento de metadatos (que suele incluir, una palabra de sincronización que identifica el inicio del segmento de metadatos, seguida por valores de identificación, a modo de ejemplo, versión, longitud, período, conteo de elementos expandidos, y valores de asociación de flujo secundario, tal como se indica en la Tabla 1 siguiente); y

50 después de la cabecera del segmento de metadatos, al menos un valor de protección (p.ej., los valores HMAC digest y huella dactilar de audio de la Tabla 1), útiles para al menos una función de entre desencriptación, autenticación o validación de al menos uno de los metadatos, del segmento de metadatos, o los correspondientes datos de audio); y

55 también después de la cabecera de segmento de metadatos, identificación de carga útil de metadatos ("ID") y valores de configuración de carga útil, que identifican el tipo de metadatos en cada carga útil de metadatos, e indican al menos un aspecto de la configuración (p.ej., tamaño) de cada carga útil.

60 Cada carga útil de metadatos sigue el identificador ID de carga útil correspondiente y los valores de configuración de carga útil.

En algunas formas de realización, cada uno de los segmentos de metadatos, en el segmento de bits residuales (o campo auxdata, o campo "addbsi") de una trama, tiene tres niveles de estructura:

65 una estructura de alto nivel (p.ej., una cabecera de segmento de metadatos), que incluye un indicador que indica si el campo bit residuales (o auxdata o addbsi) incluye metadatos, al menos un valor de ID que indica qué tipos de

metadatos están presentes y, en condiciones normales, además, un valor que indica cuántos bits de metadatos (p.ej., de cada tipo) están presentes (si los metadatos están presentes). Un tipo de metadatos que podría estar presente es PIM, otro tipo de metadatos que podría estar presente es SSM, y otros tipos de metadatos que podrían estar presentes son LPSM, y/o metadatos de límite de programa y/o metadatos de investigación de soportes;

5 una estructura de nivel intermedio, que comprende datos asociados con cada tipo de metadatos identificado (p.ej., cabecera de carga útil de metadatos, valores de protección e ID de carga útil y valores de configuración de carga útil, para cada tipo de metadatos identificado); y

10 una estructura de nivel bajo, que comprende una carga útil de metadatos para cada tipo de metadatos identificado (p.ej., una secuencia de valores de PIM, si se identifica PIM como estando presente, y/o valores de metadatos de otro tipo (p.ej., SSM o LPSM), si otro tipo de metadatos se identifica como estando presente).

15 Los valores de datos en dicha estructura de tres niveles pueden estar anidados. A modo de ejemplo, los valores de protección para cada carga útil (p.ej., cada PIM o SSM u otra carga útil de metadatos), que se identifican por las estructuras de nivel alto e intermedio, pueden incluirse después de la carga útil (y, por lo tanto, después de la cabecera de carga útil de metadatos de la carga útil), o los valores de protección para todas las cargas útiles de metadatos, identificados por las estructuras de nivel alto e intermedio pueden incluirse después de la carga útil de metadatos final en el segmento de metadatos (y, por lo tanto, después de las cabeceras de carga útil de metadatos de todas las cargas útiles del segmento de metadatos).

20 En un ejemplo (que se describirá con referencia al segmento de metadatos o "contenedor" de la Figura 8), una cabecera de segmento de metadatos identifica cuatro cargas útiles de metadatos. Según se ilustra en la Figura 8, la cabecera del segmento de metadatos incluye una palabra de sincronización de contenedor (identificada como "container sync") y valores de identificador ID de versión y clave. La cabecera del segmento de metadatos es seguida por las cuatro cargas útiles de metadatos y bits de protección. Los valores de ID y configuración de carga útil (p.ej., tamaño de carga útil), para la primera carga útil (p.ej., una carga útil de PIM), siguen a la cabecera de segmento de metadatos, la propia primera carga útil sigue los valores de ID y configuración, valores de ID de carga útil y configuración de carga útil (p.ej., tamaño de carga útil) para la segunda carga útil (p.ej., una carga útil de SSM) siguen a la primera carga útil, la propia segunda carga útil sigue a estos valores de ID y configuración, los valores de ID de carga útil y configuración de carga útil (p.ej., tamaño de carga útil) para la tercera carga útil (p.ej., una carga útil de LPSM) siguen a la segunda carga útil, la propia tercera carga útil sigue a estos valores de ID y configuración, los valores de ID de carga útil y configuración de carga útil (p.ej., tamaño de carga útil) para la cuarta carga útil, siguen a la tercera carga útil, la propia cuarta carga útil sigue estos valores de ID y configuración, y los valores de protección (identificados como "Datos de Protección" en la Figura 8) para la totalidad o algunas de las cargas útiles (o para la estructura de nivel alto e intermedio, y la totalidad o algunas de las cargas útiles), siguen a la última carga útil.

40 En algunas formas de realización, si el decodificador 101 recibe un flujo binario de audio generado de conformidad con una forma de realización con un denominado *hash* criptográfico, el decodificador está configurado para analizar y recuperar el *hash* criptográfico desde un bloque de datos determinado a partir del flujo binario, en donde dicho bloque incluye metadatos. El Validador 102 puede utilizar el *hash* criptográfico para validar el flujo binario recibido y/o los metadatos asociados. A modo de ejemplo, si el validador 102 encuentra que los metadatos son válidos sobre la base de una coincidencia entre un *hash* criptográfico de referencia y el *hash* criptográfico recuperado del bloque de datos, entonces, se puede inhabilitar la operación del procesador 103 en los datos de audio correspondientes y hacer que la etapa de selección 104 pase a través de (sin cambiar) los datos de audio. De modo adicional, como opción, o de forma alternativa, se pueden utilizar otros tipos de técnicas criptográficas en lugar de un método basado en un denominado *hash* criptográfico.

50 El codificador 100 de la Figura 2 puede determinar (en respuesta a LPSM y opcionalmente, de modo adicional, metadatos de límite de programa, extraídos por el decodificador 101) que una unidad de post/pre-procesamiento ha realizado un tipo de procesamiento de sonoridad sobre los datos de audio que han de codificarse (en los elementos 105, 106, y 107) y, por lo tanto, pueden crear (en el generador 106) metadatos de estado de procesamiento de sonoridad que incluyen los parámetros específicos utilizados y/o derivados del procesamiento de sonoridad realizado con anterioridad. En algunas puestas en práctica, el codificador 100 puede crear (e incluir en el flujo binario codificado, a la salida del mismo) metadatos indicativos del historial de procesamiento sobre el contenido de audio siempre que el codificador conozca los tipos de procesamiento que se han realizado sobre el contenido de audio.

60 La Figura 3 es un diagrama de bloques de un decodificador (200) que es una forma de realización de la unidad de procesamiento de audio, y de un post-procesador (300) acoplado al mismo. El post-procesador (300) es, también, una forma de realización de la unidad de procesamiento de audio. Cualquiera de los componentes o elementos del decodificador 200, y del post-procesador 300, se puede poner en práctica como uno o más procesos y/o uno o más circuitos (p.ej., ASICs, FPGAs u otros circuitos integrados), en hardware, software o una combinación de hardware y software. El decodificador 200 incluye la memoria intermedia de trama 201, el analizador sintáctico 205, el decodificador de audio 202, la etapa de validación de estado de audio (validador) 203, y la etapa de generación de bits de control 204, conectados Según se ilustra. Además, el decodificador 200 suele incluir otros elementos de

procesamiento (no ilustrados).

La memoria intermedia de trama 201 (una memoria intermedia) memoriza (p.ej., de manera no transitoria) al menos una trama del flujo binario de audio codificado, recibido por el decodificador 200. Una secuencia de las tramas del flujo binario de audio codificada se establecido desde la memoria intermedia 201 al analizador 205.

El analizador sintáctico 205 está acoplado y configurado para extraer metadatos PIM y/o SSM (y, además, como opción, otros metadatos, a modo de ejemplo, LPSM) a partir de cada trama del audio de entrada codificado, para establecer al menos algunos de los metadatos (p.ej., LPSM y metadatos de límite de programa, si se extrae alguno, y/o PIM y/o SSM) al validador de estado de audio 203 y la etapa 204, con el fin de establecer los metadatos extraídos como salida (p. ej., al post-procesador 300) para extraer datos de audio procedentes del audio de entrada codificado, y para establecer los datos de audio extraídos al decodificador 202.

La entrada del flujo binario de audio codificado al decodificador 200 puede ser uno de entre un flujo binario AC-3, un flujo binario E-AC-3, o un flujo binario Dolby E.

El sistema de la Figura 3 incluye, además, el post-procesador 300. El post-procesador 300 comprende la memoria intermedia de trama 301 y otros elementos de procesamiento (no ilustrados), que incluyen al menos un elemento de procesamiento acoplado a la memoria intermedia 301. La memoria intermedia de trama 301 realiza la memorización (p.ej., de forma no transitoria), en al menos una trama del flujo binario de audio decodificado, recibido por el post-procesador 300 desde el decodificador 200. Los elementos de procesamiento del post-procesador 300 están acoplados y configurados para recibir y procesar, de forma adaptativa, una secuencia de las tramas de la salida de flujo binario de audio decodificado procedente de la memoria intermedia 301, utilizando metadatos proporcionados, a la salida, del decodificador 200 y/o bits de control a la salida desde la etapa 204 del decodificador 200. En condiciones normales, el post-procesador 300 está configurado para realizar un procesamiento adaptativo sobre los datos de audio decodificados utilizando metadatos procedentes del decodificador 200 (p.ej., procesamiento de sonoridad adaptativo en los datos de audio decodificados que utilizan valores LPSM y, además, de forma opcional, metadatos de límite de programa, en donde el procesamiento adaptativo puede estar basado en el estado de procesamiento de sonoridad y/o una o más características de datos de audio, que se indican por LPSM para datos de audio indicativos de un único programa de audio).

Varias puestas en práctica del decodificador 200, y el post-procesador 300, están configuradas para realizar diferentes formas de realización del método de la invención.

El decodificador de audio 202, del decodificador 200, está configurado para decodificar los datos de audio extraídos por el analizador sintáctico 205, con el fin de generar datos de audio decodificados, y para establecer los datos de audio decodificados como salida (p.ej., al post-procesador 300).

El validador de estado 203 está configurado para autenticar y validar los metadatos establecido en el mismo. En algunas formas de realización, los metadatos son, (o están incluidos en), un bloque de datos que se ha incluido en el flujo binario de entrada (p.ej., de conformidad con una forma de realización de la presente invención). El bloque puede incluir un denominado *hash* criptográfico (un código de autenticación de mensaje basado en un denominado *hash*, o "HMAC") para el procesamiento de los metadatos y/o los datos de audio subyacentes (proporcionados desde el analizador sintáctico 205 y/o el decodificador 202, al validador 203). El bloque de datos puede estar firmado, de forma digital, en estas formas de realización, de modo que una unidad de procesamiento de audio de flujo descendente pueda autenticar y validar, con relativa facilidad, los metadatos de estado de procesamiento.

Otros métodos criptográficos que incluyen, pero no se limitan a, uno o más métodos criptográficos no de HMAC, se pueden utilizar para la validación de metadatos (p.ej., en el validador 203) con el fin de garantizar la transmisión y recepción segura de los metadatos y/o los datos de audio subyacentes. A modo de ejemplo, la validación (utilizando dicho método criptográfico) se puede realizar en cada unidad de procesamiento de audio que recibe una forma de realización del flujo binario de audio, para determinar si los metadatos de estado de procesamiento de sonoridad y los correspondientes datos de audio, incluidos en el flujo binario, se han sometido a, (y/o tienen un resultado de), un procesamiento de sonoridad específico (tal como se indica por los metadatos) y no se han modificado después de la realización de dicho procesamiento de sonoridad específico.

El validador de estado 203 establece datos de control para controlar el generador de bits 204, y/o establecer los datos de control como salida (p.ej., al post-procesador 300), para indicar los resultados de la operación de validación. En respuesta a los datos de control (y opcionalmente, también otros metadatos extraídos del flujo binario de entrada), la etapa 204 puede generar (y establecer al post-procesador 300) cualquiera de lo que sigue:

bits de control que indican que la salida de datos de audio decodificados, procedentes del decodificador 202, ha experimentado un tipo específico de procesamiento de sonoridad (cuando LPSM indica que la salida de datos de audio del decodificador 202 ha experimentado el tipo específico de procesamiento de sonoridad, y los bits de control, procedentes del validador 203, indican que los LPSM son válidos); o

bits de control que indican que los datos de audio decodificados, proporcionados, a la salida, desde el decodificador 202, deben someterse a un tipo específico de procesamiento de sonoridad (p.ej., cuando LPSM indica que los datos de audio proporcionados, a la salida, desde el decodificador 202, no se han sometido al tipo específico de procesamiento de sonoridad, o cuando el LPSM indica que la salida de datos de audio del decodificador 202 ha experimentado el tipo específico de procesamiento de sonoridad, pero los bits de control del validador 203 indican que los LPSM no son válidos).

Como alternativa, el decodificador 200 establece metadatos extraídos por el decodificador 202 desde el flujo binario de entrada, y metadatos extraídos por el analizador sintáctico 205 desde el flujo binario de entrada hasta el post-procesador 300, y el post-procesador 300 realiza un procesamiento adaptativo sobre los datos de audio decodificados utilizando los metadatos, o realiza la validación de los metadatos y luego, realiza un procesamiento adaptativo en los datos de audio decodificados utilizando los metadatos, si la validación indica que los metadatos son válidos.

En algunas formas de realización, si el decodificador 200 recibe un flujo binario de audio generado de conformidad con una forma de realización de la invención, con un *hash* criptográfico, el decodificador está configurado para analizar y recuperar el *hash* criptográfico desde un bloque de datos determinado a partir del flujo binario, comprendiendo dicho bloque metadatos de estado de procesamiento de sonoridad (LPSM). El Validador 203 puede utilizar el *hash* criptográfico para validar el flujo binario recibido y/o metadatos asociados. A modo de ejemplo, si el validador 203 encuentra que los metadatos LPSM son válidos, sobre la base de una coincidencia entre un *hash* criptográfico de referencia y el *hash* criptográfico recuperado del bloque de datos, entonces, puede indicar a una unidad de procesamiento de audio de flujo descendente (p.ej., post-procesador 300, que puede ser, o incluir una unidad de nivelación de volumen) para pasar a través (sin cambio), los datos de audio del flujo binario. Además, de modo opcional, o como alternativa, se pueden utilizar otros tipos de técnicas criptográficas en lugar de un método basado en un *hash* criptográfico.

En algunas puestas en práctica del decodificador 200, el flujo binario codificado recibido (y memorizado, temporalmente, en la memoria 201), es un flujo binario AC-3 o un flujo binario E-AC-3, e incluye segmentos de datos de audio (p.ej., los segmentos AB0-AB5 de la trama ilustrada en la Figura 4), y segmentos de metadatos, en donde los segmentos de datos de audio son indicativos de datos de audio, y cada uno de al menos algunos de los segmentos de metadatos incluye PIM o SSM (u otros metadatos). La etapa del decodificador 202 (y/o el analizador sintáctico 205) está configurada para extraer los metadatos del flujo binario. Cada uno de los segmentos de metadatos que incluye PIM y/o SSM (y opcionalmente, también otros metadatos), se incluye en un segmento de bits residuales de una trama del flujo binario, o un campo "addbsi" del segmento de Información de Flujo Binario ("BSI") de una trama del flujo binario, o en un campo auxdata (p.ej., el segmento AUX, que se ilustra en la Figura 4) al final de una trama del flujo binario. Una trama del flujo binario puede incluir uno o dos segmentos de metadatos, cada uno de los cuales incluye metadatos, y si la trama incluye dos segmentos de metadatos, uno puede estar presente en el campo addbsi de la trama, y el otro en el campo AUX de la trama.

En algunas formas de realización, cada segmento de metadatos (a veces referido aquí como un "contenedor") del flujo binario memorizado en la memoria intermedia 201, tiene un formato que incluye una cabecera de segmento de metadatos (y además, de forma opcional, otros elementos obligatorios o "principales"), y una o más cargas útiles de metadatos que siguen la cabecera del segmento de metadatos. SIM, si está presente, se incluye en una de las cargas útiles de metadatos (identificada por una cabecera de carga útil y, normalmente, tiene un formato de un primer tipo). PIM, si está presente, se incluye en otra de las cargas útiles de metadatos (identificada por una cabecera de carga útil y que suele tener un formato de un segundo tipo). De modo similar, cada otro tipo de metadatos (si está presente), se incluye en otra de las cargas útiles de metadatos (que se identifica por una cabecera de carga útil y, en condiciones normales, tiene un formato específico para el tipo de metadatos). El formato, a modo de ejemplo, permite un acceso conveniente a los SSM, PIM y otros metadatos en momentos distintos a cuando se realiza la decodificación (p.ej., por del post-procesador 300 después de la decodificación, o por un procesador configurado para reconocer los metadatos sin realizar una decodificación completa en el flujo binario codificado), y permite la detección y corrección de errores, de forma conveniente y eficiente (p.ej., de identificación de flujo secundario) durante la decodificación del flujo binario. A modo de ejemplo, sin acceso a SSM en el formato a modo de ejemplo, el decodificador 200 podría identificar, incorrectamente, el número correcto de flujo secundarios asociados con un programa. Una carga útil de metadatos, en un segmento de metadatos, puede incluir SSM, otra carga útil de metadatos en el segmento de metadatos puede incluir PIM y, opcionalmente, también al menos otra carga útil de metadatos en el segmento de metadatos, puede incluir otros metadatos (p.ej., metadatos de estado de procesamiento de sonoridad o "LPSM").

En algunas formas de realización, una carga útil de metadatos de estructura flujo secundario (SSM), incluida en una trama de un flujo binario codificado, (p.ej., un flujo binario E-AC-3 indicativo de al menos un programa de audio), que se memoriza en la memoria intermedia 201, incluye metadatos SSM en el formato siguiente:

una cabecera de carga útil, que suele incluir, al menos un valor de identificación (p.ej., un valor de 2 bits indicativo de la versión de formato de SSM y, además como opción, valores de longitud, período, conteo y asociación de flujo secundario); y

después de la cabecera:

5 metadatos de flujo secundario independiente, que indican el número de flujos secundarios independientes del programa indicados por el flujo binario; y

10 metadatos de flujo secundario dependiente, indicativos de si cada flujo secundario independiente del programa tiene al menos un flujo secundario dependiente asociado con el mismo, y si es así, el número de flujos secundarios dependientes, asociados con cada flujo secundario independiente del programa.

10 En algunas formas de realización, una carga útil de metadatos de información sobre el programa (PIM), incluida en una trama de un flujo binario codificado, (p.ej., un flujo binario E-AC-3, indicativo de al menos un programa de audio), que se memoriza, de forma temporal, en la memoria intermedia 201, tiene el siguiente formato:

15 una cabecera de carga útil, que suele incluir, al menos un valor de identificación (p.ej., un valor indicativo de la versión de formato de PIM y, además, de forma opcional, valores de longitud, período, conteo y asociación de flujo secundario); y

20 después de la cabecera, metadatos PIM en el siguiente formato:

20 metadatos de canales activos de cada canal silencioso, y cada canal no silencioso, de un programa de audio (es decir, qué canales del programa contienen información de audio y cuáles, (si los hay), contienen solamente silencio (normalmente, para la duración de la trama)). En formas de realización en las que el flujo binario codificado es un flujo binario AC-3 o un flujo binario E-AC-3, los metadatos de canal activo en una trama del flujo binario, se pueden utilizar junto con metadatos adicionales del flujo binario (p.ej., el campo de modo de codificación de audio ("acmod") de la trama y, si está presente, el campo de mapa de canales en la trama, o tramas de flujo secundario dependiente asociado, para determinar qué canales del programa contienen información de audio y cuáles contienen silencio;

30 los metadatos de estado de procesamiento de mezcla descendente, indican si el programa fue objeto de mezcla descendente (antes o durante la codificación), y si es así, el tipo de operación de mezcla descendente que se aplicó. Los metadatos de estado de procesamiento de mezcla descendente pueden ser útiles para poner en práctica la mezcla ascendente (p.ej., en el post-procesador 300) de flujo descendente de un decodificador, a modo de ejemplo, para la mezcla ascendente del contenido de audio del programa, utilizando parámetros que coincidan más estrechamente con un tipo de operación de mezcla descendente que fue aplicado. En formas de realización en las que el flujo binario codificado es un flujo binario AC-3 o E-AC-3, los metadatos de estado de procesamiento de mezcla descendente se pueden utilizar junto con el campo de modo de codificación de audio ("acmod") de la trama para determinar el tipo de operación de mezcla descendente (si corresponde), aplicado a los canales del programa;

40 metadatos de estado de procesamiento de mezcla ascendente, indicativos de si el programa fue objeto de mezcla ascendente, (p.ej., a partir de una pequeña cantidad de canales), antes o durante la codificación y, de ser así, el tipo de operación de mezcla ascendente que se aplicó. Los metadatos de estado de procesamiento de mezcla ascendente pueden ser útiles para poner en práctica la operación de mezcla descendente (en un post-procesador) en flujo descendente de un decodificador, a modo de ejemplo para la mezcla descendente del contenido de audio del programa, en un modo que sea compatible con un tipo de mezcla ascendente (p.ej., Dolby Pro Logic, o Dolby Pro Logic II Movie Mode, o Dolby Pro Logic II Music Mode, o Dolby Professional Upmixer) que se aplicó al programa. En formas de realización en las que el flujo binario codificado es un flujo binario E-AC-3, los metadatos de estado de procesamiento de mezcla ascendente se pueden utilizar junto con otros metadatos (p.ej., el valor de un campo "strmtyp" de la trama), con el fin de determinar el tipo de operación de mezcla ascendente, (si corresponde), aplicado a los canales del programa. El valor del campo "strmtyp" (en el segmento BSI de una trama de un flujo binario E-AC-3) indica si el contenido de audio de la trama pertenece a un flujo independiente (que determina un programa) o un flujo secundario independiente (de un programa que incluye, o está asociado con, múltiples flujos secundarios) y, por lo tanto, se puede decodificar, con independencia de cualquier otro flujo secundario indicado por el flujo binario E-AC-3, o si el contenido de audio de la trama pertenece a un flujo secundario dependiente (de un programa que incluye, o está asociado con, múltiples flujos secundarios) y, por lo tanto, se debe decodificar junto con un flujo secundario independiente con el que está asociado; y

60 metadatos de estado de pre-procesamiento, que indican si el pre-procesamiento se realizó en el contenido de audio de la trama (antes de la codificación del contenido de audio para generar el flujo binario codificado), y en este caso, el tipo de pre-procesamiento que fue realizado.

60 En algunas realizaciones, los metadatos de estado de pre-procesamiento son indicativos de:

65 si se aplicó una atenuación envolvente (p.ej., si los canales envolventes del programa de audio se atenuaron en 3 dB antes de la codificación),

si se aplicó un cambio de fase de 90 grados (p.ej., para canales envolventes Ls y canales Rs del programa de audio

antes de la codificación),

si se aplicó un filtro de paso bajo a un canal LFE del programa de audio antes de la codificación,

5 si el nivel de un canal LFE del programa fue supervisado, o no, durante la producción y, de ser así, el nivel supervisado del canal LFE en relación con el nivel de los canales de audio de margen completo del programa,

10 si se debe realizar una compresión de margen dinámico (p.ej., en el decodificador), en cada bloque de contenido de audio decodificado del programa y, si es así, el tipo (y/o parámetros) de compresión de margen dinámico que ha de realizarse (p.ej., este tipo de metadatos de estado de pre-procesamiento puede ser indicativo de cuál de los siguientes tipos de perfil de compresión fue asumido por el codificador para generar valores de control de compresión de margen dinámico que están incluidos en el flujo binario codificado: Película estándar, Luz de película, Estándar musical, Luz musical o Voz. Como alternativa, este tipo de metadatos de estado de pre-procesamiento puede indicar que debe realizarse la compresión de margen dinámico intensa (compresión "compr") en cada trama de contenido de audio decodificado del programa, de un modo que se determina por los valores de control de compresión de margen dinámico que se incluyen en el flujo binario codificado),

20 si el procesamiento de extensión espectral y/o codificación de acoplamiento de canal se empleó para codificar márgenes de frecuencia específicos del contenido del programa y, en caso afirmativo, las frecuencias mínima y máxima de los componentes de frecuencia del contenido en que se realizó la codificación de extensión espectral, y las frecuencias mínima y máxima de componentes de frecuencia del contenido en el que se realizó la codificación de acoplamiento de canal. Este tipo de información de metadatos de estado de pre-procesamiento puede ser útil para realizar la ecualización (en un post-procesador) en flujo descendente de un decodificador. Tanto el acoplamiento de canales como la información de extensión espectral son útiles, además, para optimizar la calidad durante las operaciones de trans-codificación y aplicaciones. A modo de ejemplo, un codificador puede optimizar su comportamiento (incluida la adaptación de etapas de pre-procesamiento, tal como la virtualización de auriculares, operación de mezcla ascendente, etc.), sobre la base del estado de los parámetros, como la extensión espectral y la información de acoplamiento de canal. Además, el codificador podría adaptar sus parámetros de acoplamiento y extensión espectral, de forma dinámica, para que coincidan y/o para los valores óptimos basados en el estado de los metadatos entrantes (y autenticados), y

30 si los datos de margen de ajuste de mejora de diálogo están incluidos en el flujo binario codificado y, si es así, el margen de ajuste disponible durante la realización del procesamiento de mejora de diálogo (p.ej., en un post-procesador de flujo descendente de un decodificador) para ajustar el nivel de contenido de diálogo en relación con el nivel de contenido sin diálogo en el programa de audio.

35 En algunas formas de realización, una carga útil de LPSM, incluida en una trama de un flujo binario codificado (p.ej., un flujo binario E-AC-3 indicativo de al menos un programa de audio), que se memoriza, temporalmente, en la memoria intermedia 201 incluye metadatos LPSM en el siguiente formato:

40 Una cabecera (que incluye típicamente una palabra de sincronización que identifica el inicio de la carga útil de LPSM, seguido de al menos un valor de identificación, por ejemplo, la versión de formato LPSM, longitud, período, recuento y valores de asociación flujo secundario indicados en la Tabla 2 siguiente); y

45 después de la cabecera,

50 al menos un valor de indicación de diálogo (p.ej., parámetro "Canales de Diálogo" de la Tabla 2), que indica si los correspondientes datos de audio indican diálogo, o no indican diálogo, (p.ej., qué canales de los correspondientes datos de audio indican diálogo);

al menos un valor de cumplimiento de normativa de sonoridad (p.ej., parámetro "Tipo de Normativa de Sonoridad" de la Tabla 2), que indica si los correspondientes datos de audio cumplen con un conjunto indicado de normativas de sonoridad;

55 al menos un valor de procesamiento de sonoridad (p.ej., uno o más de los parámetros "Indicador de corrección de sonoridad bloqueada por diálogo", "Tipo de corrección de sonoridad" de la Tabla 2), que indica al menos un tipo de procesamiento de sonoridad que se ha realizado en los correspondientes datos de audio; y

60 al menos un valor de sonoridad (p. ej., uno o más de los parámetros "Sonoridad bloqueada relativa de ITU", " Sonoridad bloqueada de expresión vocal de ITU", "Sonoridad de corto plazo de 3s de ITU (EBU 3341)" y "Pico verdadero" de la Tabla 2), que indican al menos una característica de sonoridad (p.ej., pico o promedio de sonoridad) de los datos de audio correspondientes.

65 En algunas puestas en práctica, el analizador sintáctico 205 (y/o la etapa de decodificador 202) se configuran para extraer, a partir de un segmento de bits residuales, o un campo "addbsi", o un campo de datos auxiliares, de una trama del flujo binario, cada segmento de metadatos que tiene el siguiente formato:

una cabecera de segmento de metadatos (que suele incluir, una palabra de sincronización que identifica el inicio del segmento de metadatos, seguido de al menos un valor de identificación, p.ej., valores de versión, longitud y período, conteo de elementos expandidos y de asociación de flujo secundario); y

5 después de la cabecera del segmento de metadatos, al menos un valor de protección (p.ej., los valores HMAC digest y huella dactilar de audio de la Tabla 1) útiles para al menos una función de entre descriptación, autenticación o validación de al menos uno de los metadatos del segmento de metadatos, o el correspondiente dato de audio); y

10 además, después de la cabecera del segmento de metadatos, valores de identificación de carga útil de metadatos ("ID") y de configuración de carga útil, que identifican el tipo y al menos un aspecto de la configuración (p.ej., tamaño) de cada carga útil de metadatos siguiente.

15 Cada segmento de carga útil de metadatos (que tiene, preferentemente, el formato especificado con anterioridad) sigue los correspondientes valores de ID de carga útil de metadatos y de configuración de carga útil.

Más generalmente, el flujo binario de audio codificado, generado por formas de realización preferidas, tiene una estructura que proporciona un mecanismo para etiquetar elementos de metadatos y subelementos como elementos o sub-elementos principales (obligatorios) o expandidos (de forma opcional). Lo que antecede permite que la tasa de datos del flujo binario (incluidos sus metadatos) sea objeto de escalada en numerosas aplicaciones. Los elementos principales (obligatorios) de la sintaxis del flujo binario preferido deberían ser capaces, además, de señalar que los elementos expandidos (opcionales), asociados con el contenido de audio están presentes (en banda) y/o en una localización distante (fuera de banda).

25 Se requiere que los elementos principales estén presentes en cada trama del flujo binario. Algunos sub-elementos, de los elementos principales, son opcionales y pueden estar presentes en cualquier combinación. No se requiere que elementos expandidos estén presentes en cada trama (para limitar la sobrecarga de la tasa de bits). En consecuencia, los elementos expandidos pueden estar presentes en algunas tramas y no en otras. Algunos sub-elementos, de un elemento expandido, son opcionales y pueden estar presentes en cualquier combinación, mientras que algunos sub-elementos de un elemento expandido pueden ser obligatorios (es decir, si el elemento expandido está presente en una trama del flujo binario).

35 En una clase de formas de realización, se genera un flujo binario de audio codificado que comprende una secuencia de segmentos de datos de audio y segmentos de metadatos (p.ej., mediante una unidad de procesamiento de audio que es un ejemplo ilustrativo de la invención). Los segmentos de datos de audio son indicativos de datos de audio, cada uno de al menos alguno de los segmentos de metadatos incluye PIM y/o SSM (y además, de modo opcional, metadatos de al menos otro tipo), y los segmentos de datos de audio son objeto de multiplexación por división de tiempo con los segmentos de metadatos. En formas de realización preferidas de esta clase, cada uno de los

40 segmentos de metadatos tiene un formato preferido que se describe en este documento.

En un formato preferido, el flujo binario codificado es un flujo binario AC-3 o un flujo binario E-AC-3, y cada uno de los segmentos de metadatos que incluye metadatos SSM y/o PIM está incluido (p.ej., por la etapa 107 de una puesta en práctica preferida del codificador 100), como información de flujo binario adicional en el campo "addbsi" (ilustrado en la Figura 6) del segmento de Información de Flujo Binario ("BSI") de una trama del flujo binario, o en un campo auxdata de una trama del flujo binario, o en un segmento de bits residuales de una trama del flujo binario.

45 En el formato preferido, cada una de las tramas incluye un segmento de metadatos (a veces aquí referido como un contenedor de metadatos, o contenedor), en un segmento de bits residuales (o campo addbsi) de la trama. El segmento de metadatos tiene los elementos obligatorios (referidos, de forma colectiva, como el "elemento principal") que se ilustra en la Tabla 1 siguiente (y pueden incluir los elementos opcionales que se ilustran en la Tabla 1). Al menos alguno de los elementos requeridos que se ilustran en la Tabla 1, están incluidos en la cabecera del segmento de metadatos, del segmento de metadatos, pero algunos se pueden incluir en cualquier parte del

55 segmento de metadatos:

Tabla 1

Parámetro	Descripción	Obligatorio/ Opcional
SYNC [ID]		Obligatorio
Versión de elemento principal		Obligatorio
Longitud de elemento principal		Obligatorio
Periodo de elemento principal (xxx)		Obligatorio
Conteo de elementos expandidos	Indica la cantidad de elementos de metadatos expandidos que se asocian con el elemento principal. Este valor puede aumentarse/disminuirse a medida que el flujo binario pasa de producción a distribución y emisión final.	Obligatorio
Asociación de flujo secundario	Describe a qué flujo secundario está asociado el elemento principal.	Obligatorio
Firma (HMAC digest)	HMAC digest de 256 bits (utilizando el algoritmo SHA-2) calculado sobre los datos de audio, el elemento principal y todos los elementos expandidos de la trama completa.	Obligatorio
Cuenta regresiva del límite de PGM	El campo solamente aparece para alguna cantidad de tramas en la parte superior o posterior de un fichero/flujo de programas de audio. Por lo tanto, un cambio en la versión del elemento principal se podría utilizar para señalar la inclusión de este parámetro.	Opcional
Huella dactilar de audio	Huella dactilar de audio tomada sobre varias muestras de audio PCM representadas por el campo de período de elemento principal.	Opcional
Huella dactilar de video	Huella dactilar de video tomada sobre varias muestras de video comprimido (si las hay) representadas por el campo del período de elemento principal.	Opcional
URL/UUID	Este campo está definido para transmitir un URL y/o un UUID (puede ser redundante para la huella dactilar) que hace referencia a una localización externa de contenido de programa adicional (esencia) y/o metadatos asociados con el flujo binario.	Opcional

5 En el formato preferido, cada segmento de metadatos (en un segmento de bit residual, o campo adbsi o auxdata de una trama de un flujo binario codificado) que contiene metadatos SSM, PIM o LPSM contiene una cabecera de segmento de metadatos (y, además, de modo opcional, elementos principales adicionales), y después de la cabecera del segmento de metadatos (o la cabecera del segmento de metadatos y otros elementos principales), una o más cargas útiles de metadatos. Cada carga útil de metadatos incluye una cabecera de carga útil de metadatos (que indica un tipo específico de metadatos (p.ej., SSM, PIM o LPSM)) incluido en la carga útil, seguido de
10 metadatos del tipo específico. Normalmente, la cabecera de carga útil de metadatos incluye los valores siguientes (parámetros):

15 un identificador ID de carga útil (que identifica el tipo de metadatos, p.ej., SSM, PIM o LPSM) que sigue a la cabecera del segmento de metadatos (que puede incluir los valores especificados en la Tabla 1);

un valor de configuración de carga útil (que generalmente indica el tamaño de la carga útil) que sigue al identificador ID de carga útil;

20 y además, de modo opcional, valores de configuración adicionales de carga útil (p.ej., un valor de compensación que indica el número de muestras de audio desde el inicio de la trama a la primera muestra de audio a la que pertenece la carga útil y el valor de prioridad de la carga útil, p.ej., que indica una condición en la que la carga útil puede descartarse).

25 En condiciones normales, los metadatos de la carga útil tienen uno de los siguientes formatos:

los metadatos de la carga útil son SSM, que incluyen metadatos de flujo secundario independiente, que indican el número de flujos secundarios independientes del programa indicado por el flujo binario; y metadatos de flujo secundario dependiente, indicativos de si cada flujo secundario independiente del programa tiene al menos un flujo

secundario dependiente asociada con él, y de ser así, el número de flujos secundarios dependientes asociados con cada flujo secundario independiente del programa;

5 los metadatos de la carga útil son PIM, que incluyen metadatos de canal activo, indicativos de qué canales de un programa de audio contienen información de audio, y cuáles (si los hay) contienen solamente silencio (generalmente para la duración de la trama); metadatos de estado de procesamiento de mezcla descendente, indicativos de si el programa fue objeto de mezcla descendente (antes o durante la codificación), y si es así, el tipo de operación de mezcla descendente que se aplicó, metadatos de estado de procesamiento de mezcla ascendente, indicativos de si el programa fue objeto de mezcla ascendente (p.ej., a partir de una menor cantidad de canales) antes o durante la codificación, y si es así, el tipo de operación de mezcla ascendente que se aplicó, y metadatos de estado de pre-procesamiento, indicativos de si se realizó el pre-procesamiento en el contenido de audio de la trama (antes de codificar el contenido de audio para generar el flujo binario codificado), y si es así, el tipo de pre-procesamiento que se realizó; o

15 los metadatos de la carga útil son LPSM que tienen el formato que se indica en la siguiente tabla (Tabla 2):

Tabla 2

Parámetro LPSM [Sonoridad inteligente]	Descripción	Número de estados únicos	Obligatorio/Opcional	Tasa de inserción (Periodo de actualización del parámetro)
Versión LPSM			O	
Período LPSM (xxx)	Aplicable solamente a campos xxx		Obligatorio	
Conteo de LPSM			Obligatorio	
Asociación flujo secundario LPSM			Obligatorio	
Canales de diálogo	Indica qué combinación de canales de audio L, C y R contienen voz durante los 0.5 segundos anteriores. Cuando la voz no está presente en ninguna combinación de L, C o R, este parámetro deberá indicar "sin diálogo"	8	Obligatorio	-0.5 segundos (típico)
Tipo de normativa de sonoridad	indica que el flujo de datos de audio asociado cumple con un conjunto específico de normativas (p.ej., ATSC A/85 o EBU R128)	8	Obligatorio	Trama
Indicador de corrección de sonoridad bloqueada por diálogo	Indica si el flujo de audio asociado se ha corregido según el control de diálogo	2	Obligatorio (solamente está presente si Loudness_Regulation_Type indica que el audio correspondiente NO ESTÁ CORREGIDO)	Trama
Tipo de corrección de sonoridad	Indica si el flujo de audio asociado se ha corregido con una búsqueda infinita anticipada (basada en archivos) o con controlador de margen dinámico y sonoridad en tiempo real (RT)	2	Obligatorio (solamente está presente si Loudness_Regulation_Type indica que el audio correspondiente NO ESTÁ CORREGIDO)	Trama
Sonoridad bloqueada relativa de ITU (INF)	Indica la norma ITU-R BS.1770-3 de sonoridad integrada del flujo de audio asociado sin metadatos aplicados (p.ej., 7 bits: -58 -> +5.5 LKFS en pasos de 0.5 LKFS)	128	Obligatorio	1 seg.
Sonoridad bloqueada de expresión vocal de ITU (INF)	Indica la norma ITU-R BS.1770-1/3 de sonoridad integrada de voz/diálogo del flujo de audio asociado sin metadatos aplicados	128	Obligatorio	1 seg.

Parámetro LPSM [Sonoridad inteligente]	Descripción	Número de estados únicos	Obligatorio/Opcional	Tasa de inserción (Periodo de actualización del parámetro)
	(p.ej., 7 bits: -58 -> +5.5 LKFS en pasos de 0.5 LKFS)			
ITU (EBU 3341) Sonoridad a corto plazo de 3s	Indica la norma ITU (ITU-BS.1771-1) de 3 segundos sin control de sonoridad del flujo de audio asociado sin metadatos aplicados (ventana deslizante) @ tasa de inserción de ~ 10 Hz (p. Ej., 8 bits: 116 -> +11.5 LKFS en pasos de 0.5 LKFS)	256	Obligatorio	0.1 seg.
Valor de pico verdadero	Indica la norma ITU-R BS.1770-3 Anexo 2, de Valor de Pico Verdadero (dB TP) del flujo de audio asociado sin metadatos aplicados (es decir, el valor más alto durante el período de trama señalizado en el campo de período de elemento) 116 -> +11.5 LKFS en pasos de 0.5 LKFS	256	Obligatorio	0.5 seg.
Compensación de mezcla descendente	Indica la compensación de sonoridad de mezcla descendente			
Límite del programa	Indica, en tramas, cuándo habrá o haya ocurrido un límite de programa. Cuando el límite del programa no está en el límite de la trama, el desplazamiento de muestra opcional indicará en qué momento ocurre el límite de programa real de la trama.			

- 5 En otro formato preferido de un flujo binario codificado, generado de conformidad con la invención, el flujo binario es un flujo binario AC-3, o un flujo binario E-AC-3, y se incluye cada uno de los segmentos de metadatos que incluyen PIM y/o SSM (y opcionalmente, también metadatos de al menos otro tipo) (p.ej., mediante la etapa 107 de una puesta en práctica preferida del codificador 100) en al menos un segmento de bit residuales de una trama del flujo binario; (ilustrado en la Figura 6) del segmento Información de Flujo Binario ("BSI") de una trama del flujo binario; o un campo auxdata (p.ej., el segmento AUX ilustrado en la Figura 4) al final de una trama del flujo binario. Una trama puede incluir uno o dos segmentos de metadatos, cada uno de los cuales incluye al menos PIM y, opcionalmente, SSM, y (en algunas formas de realización) si la trama incluye dos segmentos de metadatos, uno puede estar presente en el campo addbsi de la trama, y el otro en el campo AUX de la trama. Cada segmento de metadatos tiene, preferentemente, el formato especificado anteriormente con referencia a la Tabla 1 anterior (es decir, incluye los elementos principales especificados en la Tabla 1, seguidos por ID de carga útil (que identifica el tipo de metadatos en cada carga útil del segmento de metadatos) y valores de configuración de carga útil, y cada carga útil de metadatos). Cada segmento de metadatos que incluye metadatos LPSM tiene, preferentemente, el formato especificado anteriormente con referencia a las Tablas 1 y 2 anteriores (es decir, incluye los elementos principales especificados en la Tabla 1, seguidos por ID de carga útil (que identifica los metadatos como LPSM) y valores de configuración de carga útil, seguidos por la carga útil (datos LPSM que tiene un formato según se indica en la Tabla 2)).
- 10
- 15
- 20 En otro formato preferido, el flujo binario codificado es un flujo binario Dolby E, y cada uno de los segmentos de metadatos que incluye PIM y/o SSM (y opcionalmente, también otros metadatos) son las primeras N ubicaciones de muestra del intervalo de banda de guarda Dolby E. Un flujo binario Dolby E, incluyendo dicho segmento de metadatos que incluye LPSM, incluye, preferentemente, un valor indicativo de la longitud de carga útil de LPSM señalada en la palabra Pd del preámbulo SMPTE 337M (la tasa de repetición de la palabra SMPTE 337M Pa permanece, preferentemente, idéntica a la tasa de trama de vídeo asociada).
- 25

En un formato preferido, en el que el flujo binario codificado es un flujo binario E-AC-3, se incluye cada uno de los segmentos de metadatos que incluye PIM y/o SSM (y opcionalmente, también LPSM y/u otros metadatos) (p.ej., en

la etapa 107 de una puesta en práctica preferida del codificador 100) como información de flujo binario adicional en un segmento de bit residuales, o en el campo "addbsi" del segmento de Información de Flujo Binario ("BSI"), de una trama del flujo binario. A continuación, se describen aspectos adicionales de la codificación de un flujo binario E-AC-3 con metadatos LPSM en este formato preferido:

5 1. Durante la generación de un flujo binario E-AC-3, mientras que el codificador E-AC-3 (que inserta los valores de LPSM en el flujo binario) está "activo", para cada trama (trama de sincronización) generada, el flujo binario debe incluir un bloque de metadatos (incluyendo LPSM) transmitido en el campo addbsi (o segmento de bit residuales) de la trama. Los bits necesarios para transmitir el bloque de metadatos no deberían aumentar la tasa de bits del
10 codificador (longitud de trama);

2. Cada bloque de metadatos (que contiene LPSM) debe contener la siguiente información:

15 loudness_correction_type_flag: en donde '1' indica que la sonoridad de los datos de audio correspondientes se corrigió en flujo ascendente desde el codificador, y '0' indica que la sonoridad fue corregida por un corrector de sonoridad integrado en el codificador (p.ej., el procesador de sonoridad 103 del codificador 100 de la Figura 2);

20 speech_channel: indica qué canal origen contiene voz (en los últimos 0.5 segundos). Si no se detecta la voz, esto se indicará como tal;

speech_loudness: indica la sonoridad de voz integrada de cada canal de audio correspondiente que contiene voz (durante los 0,5 segundos previos);

25 ITU_loudness: indica la sonoridad integrada de ITU BS.1770-3 de cada canal de audio correspondiente; y

gain: ganancia compuesta de sonoridad para el retorno en un decodificador (para demostrar la reversibilidad);

30 3. Mientras que el codificador E-AC-3 (que inserta los valores LPSM en el flujo binario) está "activo" y recibe una trama AC-3 con un indicador de "confianza", el controlador de sonoridad en el codificador (p.ej., procesador de sonoridad 103 del codificador 100 de la Figura 2) debe ser objeto de bypass. Los valores de los parámetros dialnorm y DRC origen "de confianza" deben hacerse pasar a través de (p.ej., por el generador 106 del codificador 100) al componente del codificador E-AC-3 (p.ej., etapa 107 del codificador 100). La generación del bloque LPSM continúa y loudness_correction_type_flag se pone a '1'. La secuencia de derivación del controlador de sonoridad debe estar sincronizada con el inicio de la trama decodificada AC-3, en donde aparece el indicador 'trust'. La secuencia de derivación del controlador de sonoridad debe ponerse en práctica como sigue: el control leveller_amount se disminuye desde un valor de 9 a un valor de 0 sobre 10 periodos de bloque de audio (es decir, 53.3 mseg) y el control leveler_back_end_meter se coloca en modo de bypass (esta operación debería resultar en una transición sin problemas). El término de bypass "de confianza" del nivelador implica que el valor dialnorm del flujo binario origen se reutiliza, además, a la salida del codificador, (p.ej., si el flujo binario origen 'de confianza' tiene un valor de dialnorm de -30, entonces la salida del codificador debe utilizar -30 para el valor del parámetro dialnorm de salida);
40

45 4. Mientras que el codificador E-AC-3 (que inserta los valores LPSM en el flujo binario) está "activo" y está recibiendo una trama AC-3 sin el indicador de "confianza", el controlador de sonoridad incluido en el codificador (p.ej., procesador de sonoridad 103 del codificador 100 de la Figura 2) debe estar activo. La generación del bloque LPSM continúa y el loudness_correction_type_flag se establece a '0'. La secuencia de activación del controlador de sonoridad debe sincronizarse con el inicio de la trama AC-3 decodificada, en donde desaparece el indicador de 'trust'. La secuencia de activación del controlador de sonoridad debe ponerse en práctica como sigue: el control leveller_amount se incrementa desde un valor de 0 a un valor de 9 en 1 período de bloque de audio, (es decir, 5.3 mseg) y el control del limitador_der_end_meter se coloca en el modo "activo" (esta operación debería dar como resultado una transición sin interrupciones e incluir un reinicio de integración back_end_meter); y
50

5. Durante la codificación, una interfaz gráfica de usuario (GUI) debe indicar a un usuario los siguientes parámetros: "Programa de Audio de Entrada:

55 [Fiable/no fiable]": el estado de este parámetro se basa en la presencia del indicador de "confianza" dentro de la señal de entrada y "Corrección de sonoridad en tiempo real: [Activado/Desactivado]": el estado de este parámetro se basa en el hecho de si este controlador de sonoridad, integrado en el codificador, está activo.

60 Cuando se decodifica un flujo binario AC-3 o E-AC-3 que tiene LPSM (en el formato preferido), incluido en un segmento de bit residuales o campo de omisión, o el campo "addbsi" del segmento de Información de Flujo Binario ("BSI"), de cada trama del flujo binario, el decodificador debe analizar los datos del bloque LPSM (en el segmento de bit residual o campo addbsi) y pasar todos los valores LPSM extraídos a una interfaz gráfica de usuario (GUI). El conjunto de valores de LPSM extraídos se actualiza cada trama.

65 En otro formato preferido de un flujo binario codificado, generado de conformidad con la invención, el flujo binario codificado es un flujo binario AC-3 o un flujo binario E-AC-3, y se incluye cada uno de los segmentos de metadatos,

- que incluye al menos PIM y, opcionalmente, SSM (y opcionalmente, también LPSM y/u otros metadatos) (p.ej., en la etapa 107 de una puesta en práctica preferida del codificador 100) en al menos un segmento de bits residuales, opcionalmente en un segmento Aux, de modo opcional como información de flujo binario adicional en el campo "addbsi" (ilustrado en la Figura 6) del segmento Información de Flujo Binario ("BSI"), de una trama del flujo binario.
- 5 En este formato (que es una variación del formato descrito anteriormente con referencias a las Tablas 1 y 2), cada uno de los campos addbsi (o Aux o bit residuales), que contiene LPSM, contiene los siguientes valores de LPSM:
- los elementos principales especificados en la Tabla 1, seguidos por el identificador ID de carga útil (identificando los metadatos como LPSM) y valores de configuración de carga útil, seguidos por la carga útil (datos LPSM) que tiene el siguiente formato (similar a los elementos obligatorios indicados en la Tabla 2 anterior):
- 10 versión de la carga útil de LPSM: un campo de 2 bits que indica la versión de la carga útil de LPSM;
- dialchan: un campo de 3 bits que indica si los canales izquierdo, derecho y/o central de los datos de audio correspondientes contienen diálogo hablado. La asignación de bits del campo dialchan puede ser como sigue: bit 0, que indica la presencia de diálogo en el canal izquierdo, se memoriza en el bit más importante del campo dialchan; y el bit 2, que indica la presencia de diálogo en el canal central, se memoriza en el bit menos importante del campo dialchan. Cada bit del campo dialchan se establece a '1' si el canal correspondiente contiene diálogo hablado durante los 0.5 segundos anteriores del programa;
- 15 loudregtyp: un campo de 4 bits que indica la norma de normativa de la intensidad que cumple la sonoridad del programa. Al establecer el campo "loudregtyp" en "000", se indica que los metadatos LPSM no indican el cumplimiento de la normativa de intensidad. A modo de ejemplo, un valor de este campo (p.ej., 0000) puede indicar que no está indicado el cumplimiento con una norma de normativa de la intensidad, otro valor de este campo (p.ej., 0001) puede indicar que los datos de audio del programa cumplen con la norma ATSC A/85, y otro valor de este campo (p.ej., 0010) puede indicar que los datos de audio del programa cumplen con la norma EBU R128. En el ejemplo, si el campo se pone a cualquier valor distinto de '0000', los campos loudcorrldialgat y loudcorrtyp deberían seguir en la carga útil;
- 20 loudcorrldialgat: un campo de un bit que indica si se ha aplicado la corrección de la sonoridad bloqueada por diálogo. Si la sonoridad del programa se ha corregido mediante utilizando el bloqueo de diálogo, el valor del campo loudcorrldialgat se pone a '1'. En caso contrario, se pone a '0';
- loudcorrtyp: un campo de un bit que indica el tipo de corrección de intensidad aplicado al programa. Si la sonoridad del programa se ha corregido con un proceso de corrección de sonoridad infinita anticipada (basado en fichero), el valor del campo loudcorrtyp se establece a '0'. Si la intensidad del programa se ha corregido utilizando una combinación de medición de sonoridad en tiempo real y control de margen dinámico, el valor de este campo se establece a '1';
- 35 loudrelgate: un campo de un bit que indica si existen datos de sonoridad bloqueados relativos (ITU). Si el campo loudrelgate se establece a '1', debe seguir un campo ituloudrelgat de 7 bits en la carga útil;
- loudrelgat: un campo de 7 bits que indica la sonoridad relativa del programa bloqueado (ITU). Este campo indica la sonoridad integrada del programa de audio, medida de conformidad con la norma ITU-R BS.1770-3 sin ningún ajuste de ganancia debido a la aplicación de la compresión de margen dinámico y dialnorm (DRC). Los valores de 0 a 127 se interpretan como -58 LKFS a +5.5 LKFS, en pasos de 0.5 LKFS;
- 45 loudspchgate: un campo de un bit que indica si existen datos de sonoridad bloqueados por la voz (ITU). Si el campo loudspchgate está establecido en '1', un campo loudspchgat de 7 bits debe seguir en la carga útil;
- 50 loudspchgat: un campo de 7 bits que indica la sonoridad del programa bloqueado por voz. Este campo indica la sonoridad integrada del correspondiente programa de audio completo, medido de conformidad con la fórmula (2) de la ITU-R BS.1770-3 y sin ningún ajuste de ganancia debido a la aplicación de compresión de margen dinámico y dialnorm. Los valores de 0 a 127 se interpretan como -58 a +5.5 LKFS, en pasos de 0.5 LKFS;
- 55 loudstrm3se: un campo de un bit que indica si existen datos de sonoridad a corto plazo (3 segundos). Si el campo está establecido en '1', un campo loudstrm3s de 7 bits debe permanecer en la carga útil;
- loudstrm3s: un campo de 7 bits que indica la sonoridad no bloqueada de los 3 segundos anteriores del programa de audio correspondiente, medido de conformidad con la norma ITU-R BS.1771-1 y sin ningún ajuste de ganancia debido a la aplicación de la compresión de margen dinámico y dialnorm. Los valores de 0 a 256 se interpretan como -116 LKFS a +11.5 LKFS en pasos de 0.5 LKFS;
- 60 truepke: un campo de un bit que indica si existen datos de sonoridad de pico verdadero. Si el campo truepke está establecido en '1', un campo truepk de 8 bits debe seguir en la carga útil; y
- 65

truepk: un campo de 8 bits que indica el valor de muestra de pico verdadero del programa, medido de conformidad con el Anexo 2 de la norma ITU-R BS.1770-3 y sin ningún ajuste de ganancia debido a la aplicación de la compresión de margen dinámico y dialnorm. Los valores de 0 a 256 se interpretan como -116 LKFS a +11.5 LKFS en pasos de 0.5 LKFS.

5 En algunas formas de realización, el elemento principal de un segmento de metadatos, en un segmento de bits residuales o en un campo auxdata (o "addbsi") de una trama de un flujo binario AC-3, o un flujo binario E-AC-3, comprende una cabecera de segmento de metadatos (normalmente incluye valores de identificación, p.ej., versión) y después la cabecera del segmento de metadatos; valores indicativos de si los datos de huellas dactilares se incluyen (u otros valores de protección) para los metadatos del segmento de metadatos, valores indicativos de si los datos
10 externos existen (relacionados a los datos de audio correspondientes a los metadatos del segmento de metadatos), ID de carga útil y valores de configuración de carga útil para cada tipo de metadatos (p.ej., PIM y/o SSM y/o LPSM y/o metadatos de un tipo) identificados por el elemento principal, y valores de protección para al menos un tipo de metadatos identificados por la cabecera del segmento de metadatos (u otros elementos principales del segmento de metadatos). Las cargas útiles de metadatos, del segmento de metadatos, siguen a la cabecera del segmento de metadatos y están (en algunos casos), anidadas dentro de elementos principales del segmento de metadatos.

Formas de realización de la presente invención se pueden poner en práctica en hardware, firmware o software, o una combinación de ambos (p.ej., como una matriz lógica programable). A no ser que se especifique de otro modo, los algoritmos o procesos incluidos como parte de la invención no están intrínsecamente relacionados con ningún ordenador particular u otro aparato. En particular, se pueden utilizar varias máquinas de uso general con programas escritos de conformidad con las enseñanzas aquí dadas a conocer, o puede ser más conveniente construir aparatos más especializados (p.ej., circuitos integrados) para realizar las etapas requeridas del método. Por lo tanto, la invención se puede poner en práctica en uno o más programas informáticos que se ejecutan en uno o más sistemas
20 informáticos programables (p.ej., una realización de cualquiera de los elementos de la Figura 1, o el codificador 100 de la Figura 2 (o un elemento del mismo), o decodificador 200 de la Figura 3 (o un elemento del mismo), o post-procesador 300 de la Figura 3 (o un elemento del mismo)) que comprende cada uno al menos un procesador, al menos un sistema de memorización de datos (incluyendo memoria volátil y no volátil/o elementos de memorización), al menos un dispositivo o puerto de entrada, y al menos un dispositivo o puerto de salida. El código de programa se aplica a datos de entrada para realizar las funciones descritas en este documento y para generar información de salida. La información de salida se aplica a uno o más dispositivos de salida, de manera conocida.

Cada uno de dichos programas se puede poner en práctica en cualquier lenguaje informático deseado (incluyendo lenguajes de máquina, montaje o procesales de alto nivel, lógicos, o de programación orientados al objeto) para comunicarse con un sistema informático. En cualquier caso, el idioma puede ser un lenguaje compilado o interpretado.

A modo de ejemplo, cuando se pone en práctica mediante secuencias de instrucciones de software, varias funciones y etapas de formas de realización de la invención pueden realizarse mediante secuencias de instrucciones de software multiproceso que funcionan en hardware de procesamiento de señal digital adecuado, en cuyo caso los diversos dispositivos, pasos y funciones, de las formas de realización, pueden corresponder a partes de las instrucciones del software.

Cada uno de dichos programas informáticos preferentemente se memoriza en, o descarga en, un soporte de almacenamiento o dispositivo (p.ej., memoria o soporte de estado sólido, o soporte magnético u óptico) legible por un ordenador programable de finalidad general o especial, para la configuración y funcionamiento del ordenador cuando el sistema informático lee el dispositivo o soporte de memorización, para realizar los procedimientos aquí descritos. El sistema inventivo puede ponerse en práctica, además, como un soporte de memorización legible por ordenador, configurado con (es decir, memorizando) un programa informático, en donde el soporte de memorización así configurado hace que un sistema informático opere de manera específica y predefinida para realizar las funciones aquí descritas.

Se han descrito varias formas de realización de la invención. Sin embargo, ha de entenderse que pueden realizarse diversas modificaciones sin desviarse del alcance de la invención. Numerosas modificaciones y variaciones de la presente invención son posibles a la luz de las enseñanzas anteriores. Conviene señalar que, dentro del alcance de las reivindicaciones adjuntas, la invención se puede poner en práctica de otro modo distinto al que concretamente aquí se describe.

60

REIVINDICACIONES

1. Un método para generar un flujo binario de audio codificado, comprendiendo el método:

5 la generación de una secuencia de tramas de un flujo binario de audio codificado, en donde el flujo binario de audio codificado es un flujo binario AC-3 o un flujo binario E-AC-3, siendo indicativo el flujo binario de audio codificado de al menos un programa de audio, cada trama de al menos un subconjunto de dichas tramas que incluyen i) metadatos de información sobre el programa, en al menos un segmento de metadatos de al menos un campo de omisión de la trama y ii) datos de audio en al menos otro segmento de la trama, estando el método caracterizado por
10 cuanto que:

el segmento de metadatos incluye al menos una carga útil de metadatos, comprendiendo dicha carga útil de metadatos: una cabecera; y después de la cabecera, al menos algunos de los metadatos de información sobre el programa,
15

en donde los metadatos de información sobre el programa son indicativos de al menos una propiedad o característica del contenido de audio del al menos un programa de audio,

20 en donde los metadatos de información sobre el programa son indicativos de información sobre el al menos un programa de audio que no se transmite en otras partes del flujo binario de audio codificado,

y los metadatos de información sobre el programa no incluyen metadatos de estado de procesamiento de sonoridad, en donde metadatos de estado de procesamiento de sonoridad incluyen al menos uno de entre: un valor de indicación de diálogo, que indica si el contenido de audio correspondiente indica diálogo, un valor de cumplimiento de normativa de sonoridad, que indica si los datos de audio correspondientes cumplen con un conjunto de reglamentos de sonoridad indicados, un valor de procesamiento de sonoridad, que indica al menos un tipo de procesamiento de sonoridad que se ha realizado sobre los datos de audio correspondientes, y un valor de sonoridad que indica al menos una característica de sonoridad de los datos de audio correspondientes.
25

30 2. Un método para decodificar un flujo binario de audio codificado,

incluyendo dicho método las etapas de:

35 la recepción de un flujo binario de audio codificado,

en donde el flujo binario de audio codificado es un flujo binario AC-3 o un flujo binario E-AC-3,

40 en donde el flujo binario de audio codificado comprende una secuencia de tramas y es indicativo de al menos un programa de audio, incluyendo cada una de las tramas al menos un segmento de datos de audio, e incluyendo cada segmento de datos de audio, datos de audio,

caracterizado por cuanto que

45 cada trama de al menos un subconjunto de las tramas, incluye al menos un campo de omisión que comprende al menos un segmento de metadatos, incluyendo el segmento de metadatos al menos una carga útil de metadatos, y comprendiendo dicha carga útil de metadatos: una cabecera; y después de la cabecera, metadatos de información sobre el programa,

50 en donde los metadatos de información sobre el programa son indicativos de al menos una propiedad o característica del contenido de audio del programa de audio; y

la extracción de los datos de audio y los metadatos de información sobre el programa de flujo binario de audio codificado

55 en donde los metadatos de información sobre el programa son indicativos de información sobre el al menos un programa de audio que no se transmite en otras partes del flujo binario de audio codificado,

60 y los metadatos de información sobre el programa no incluyen metadatos de estado de procesamiento de sonoridad, en donde los metadatos de estado de procesamiento de sonoridad incluyen al menos uno de entre: un valor de indicación de diálogo, que indica si el contenido de audio correspondiente indica diálogo, un valor de cumplimiento de normativa de sonoridad, que indica si los datos de audio correspondientes cumplen con un conjunto indicado de reglamentos de sonoridad, un valor de procesamiento de sonoridad, que indica al menos un tipo de procesamiento de sonoridad que se ha realizado en los datos de audio correspondientes, y un valor de sonoridad que indica al menos una característica de sonoridad de los datos de audio correspondientes.
65

3. El método según la reivindicación 1, o el método según la reivindicación 2, en donde el segmento de metadatos

incluye una carga útil de metadatos de información sobre el programa, comprendiendo dicha carga útil de metadatos de información sobre el programa:

una cabecera de metadatos de información sobre el programa; y

después de la cabecera de metadatos de información sobre el programa, dichos metadatos de información sobre el programa, cuyos dichos metadatos de información sobre el programa incluyen metadatos del canal activo indicativos de cada canal no silencioso y cada canal silencioso del programa.

4. El método según la reivindicación 1, o el método según la reivindicación 2, en donde los metadatos de información sobre el programa incluyen, además, al menos uno de entre:

metadatos del estado de procesamiento de mezcla descendente, que indican si el programa fue objeto de mezcla descendente, y si es así, un tipo de mezcla descendente que se aplicó al programa;

metadatos del estado de procesamiento de mezcla ascendente, que indican si el programa fue objeto de mezcla ascendente y, de ser así, un tipo de mezcla ascendente que se aplicó al programa;

metadatos de estado de pre-procesamiento, indicativos de si el pre-procesamiento se realizó sobre el contenido de audio de la trama, y si es así, un tipo de pre-procesamiento que se realizó sobre dicho contenido de audio; o

el procesamiento de extensión espectral, o metadatos de acoplamiento de canal, indicativos de si el procesamiento de extensión espectral o acoplamiento de canal se aplicó al programa, y si es así, un margen de frecuencia que se aplicó a la extensión espectral o acoplamiento de canal.

5. El método según la reivindicación 1, o el método según la reivindicación 2, en donde el al menos un programa de audio tiene al menos un flujo secundario independiente de contenido de audio, y el segmento de metadatos incluye una carga útil de metadatos de estructura de flujo secundario, comprendiendo dicha carga útil de metadatos de estructura de flujo secundario:

una cabecera de carga útil de metadatos de estructura de flujo secundario; y

después de la cabecera de carga útil de los metadatos de la estructura de flujo secundario, metadatos de flujo secundario independientes, indicativos del número de flujos secundarios independientes del programa, y los metadatos de flujo secundario dependientes, que indican si cada flujo secundario independiente del programa tiene por lo menos un flujo secundario dependiente asociado.

6. El método según la reivindicación 1, o el método según la reivindicación 2, en donde el segmento de metadatos incluye:

una cabecera de segmento de metadatos;

después de la cabecera de segmento de metadatos, al menos un valor de protección útil para al menos una de entre las funciones de desencriptación, autenticación o validación de los metadatos de información sobre el programa, o los datos de audio que corresponden a dichos metadatos de información sobre el programa; y

después de la cabecera del segmento de metadatos, la identificación de carga útil de metadatos y los valores de configuración de carga útil, en donde la carga útil de metadatos sigue la identificación de carga útil de metadatos y los valores de configuración de carga útil.

7. El método según la reivindicación 6, en donde la cabecera del segmento de metadatos incluye una palabra de sincronización que identifica el inicio del segmento de metadatos, y al menos un valor de identificación que sigue a la palabra de sincronización, y la cabecera de la carga útil de metadatos incluye al menos un valor de identificación.

8. Un soporte de memorización legible por ordenador, en el que se memoriza un programa informático configurado para hacer que un sistema informático realice el método de conformidad con cualquier reivindicación precedente.

9. Una unidad de procesamiento de audio, que comprende:

una memoria intermedia (109, 110, 201, 301); y

al menos un subsistema de procesamiento, acoplado a la memoria intermedia, y configurado para realizar el método según cualquiera de las reivindicaciones 1 a 7.

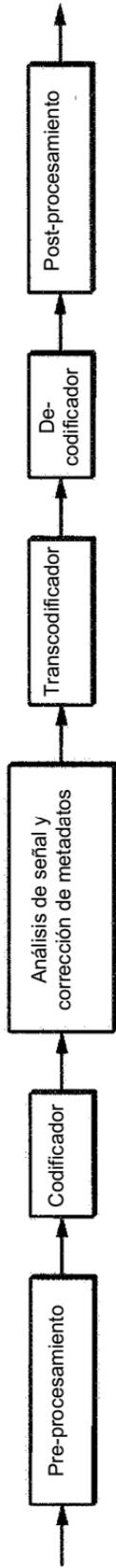


FIG. 1

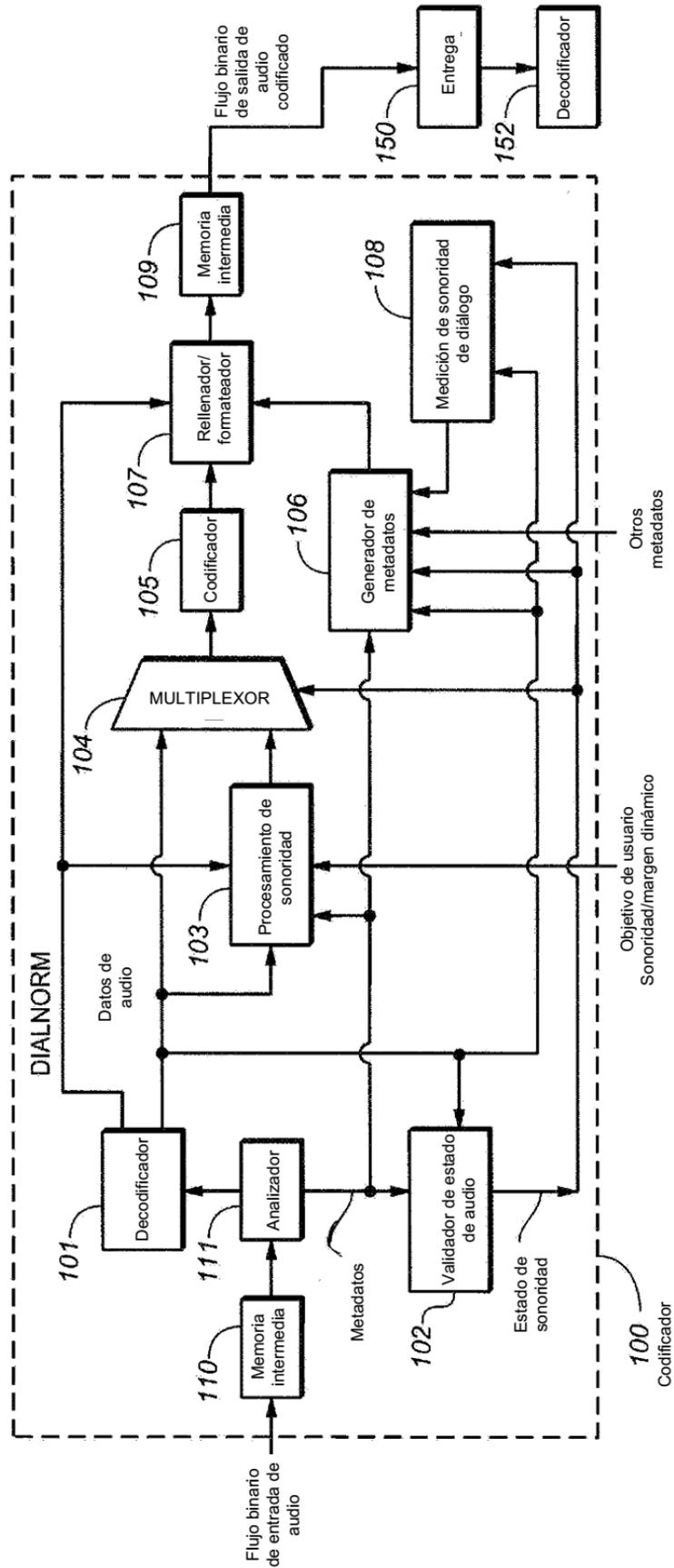


FIG. 2

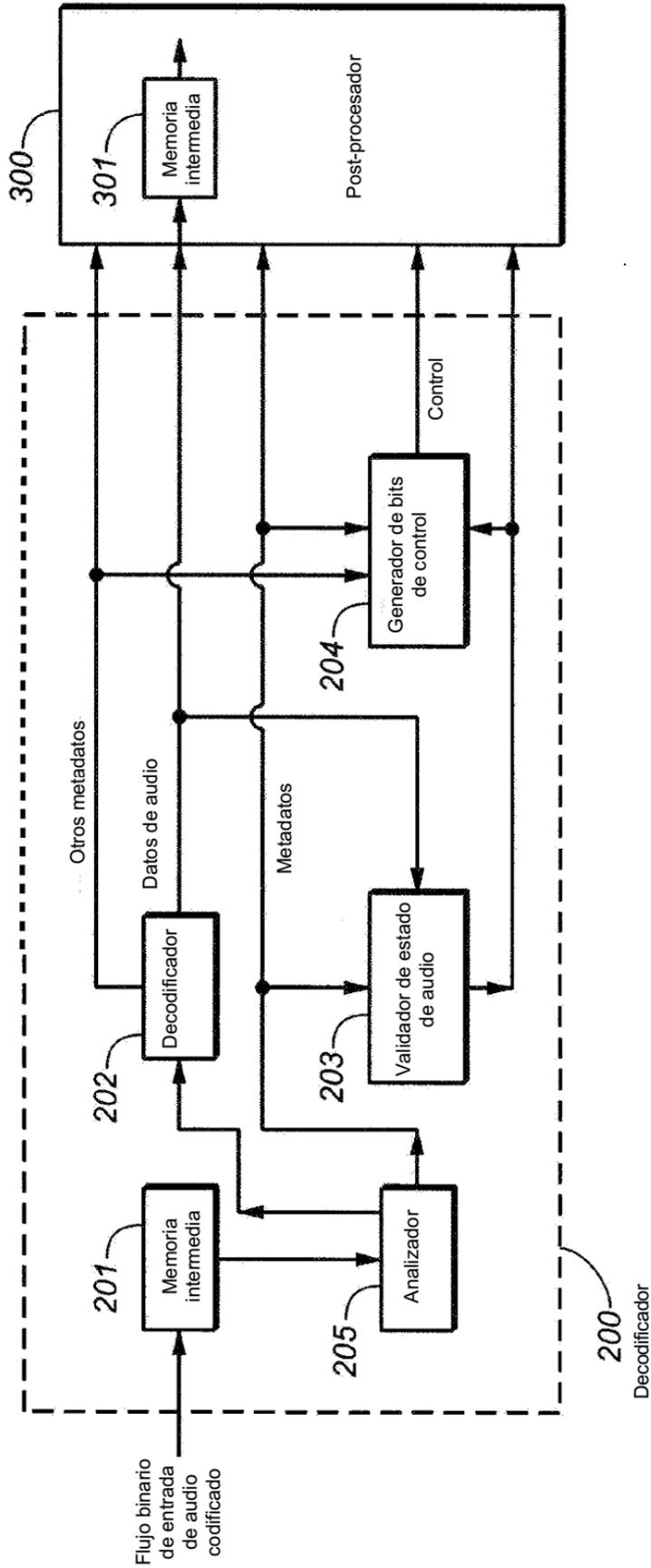


FIG. 3



FIG. 4

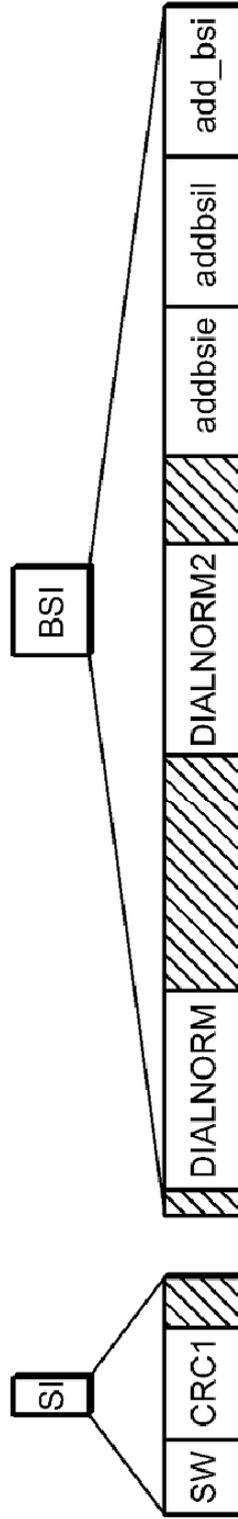


FIG. 5

FIG. 6



FIG. 7

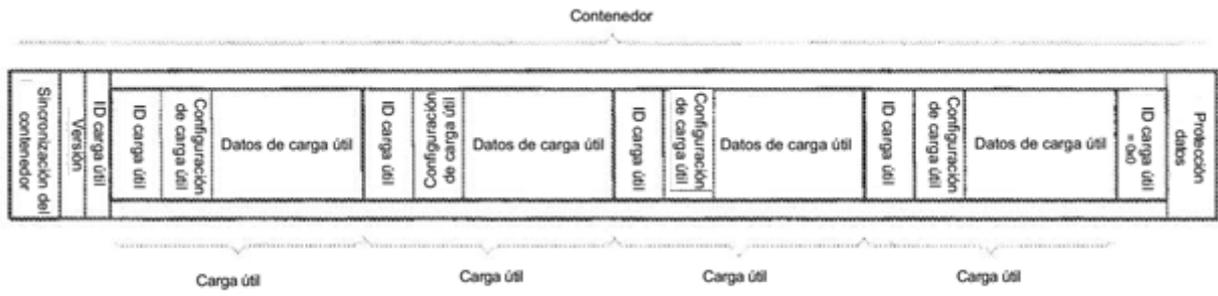


Fig. 8 (Estructura de Contenedor (segmento de metadatos))