

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 724 523**

51 Int. Cl.:

**G06F 9/4401** (2008.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **24.08.2011 PCT/CN2011/078825**

87 Fecha y número de publicación internacional: **01.03.2012 WO12025048**

96 Fecha de presentación y número de la solicitud europea: **24.08.2011 E 11819423 (2)**

97 Fecha y número de publicación de la concesión europea: **06.02.2019 EP 2598991**

54 Título: **Método de enumeración de dispositivo USB que incluye detectar un tipo de sistema operativo de anfitrión USB**

30 Prioridad:

**16.08.2011 US 201113211072  
24.08.2010 US 376413 P**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**11.09.2019**

73 Titular/es:

**MEDIATEK INC. (100.0%)  
No. 1, Dusing Rd. 1st., Science-Based Industrial  
Park  
Hsin-Chu 300, TW**

72 Inventor/es:

**CHEN, JIN-RU;  
HSU, YING-CHE;  
CHANG, TSAO-JIANG y  
HUANG, WEN-JUIN**

74 Agente/Representante:

**CARVAJAL Y URQUIJO, Isabel**

**ES 2 724 523 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

Método de enumeración de dispositivo USB que incluye detectar un tipo de sistema operativo de anfitrión USB

### Campo técnico

5 Esta invención se refiere a la enumeración de dispositivo USB, y en particular, se refiere a un procedimiento de enumeración de dispositivo USB en el que el dispositivo USB detecta en primer lugar el tipo de sistema operativo (OS) en el anfitrión USB y después realiza la enumeración de dispositivo USB usando un procedimiento específico para el tipo de OS detectado.

### Antecedentes

10 La enumeración de dispositivo USB es un procedimiento realizado por un dispositivo USB y un anfitrión USB tras conectarse el dispositivo al anfitrión. El procedimiento de enumeración de dispositivo se repite cuando el dispositivo USB se desconecta y vuelve a conectarse con el anfitrión, o cuando se reinicia el anfitrión, etc. Durante el procedimiento de enumeración de dispositivo, el anfitrión USB recopila información de características y tipo de dispositivo a partir del dispositivo USB, y asigna una dirección al dispositivo USB.

15 A medida que están diseñándose cada vez más dispositivos como dispositivos que cumplen con USB (por ejemplo, dispositivos de interfaz humana, dispositivos de almacenamiento masivo, dispositivos de audio, dispositivos de vídeo, dispositivos de comunicación, impresoras y muchas otras clases de dispositivos USB), a los anfitriones USB se les proporcionan con frecuencia capacidades correspondientes para comunicarse con estos diversos tipos de dispositivos USB. Por ejemplo, RNDIS (especificación de interfaz de controlador de red remoto) es un tipo de dispositivo de interfaz de red USB, que se soporta en Windows™ XP y sistemas operativos (OS) de Windows™  
20 posteriores; ISD (dispositivo de conexión compartida a Internet) es un tipo de RNDIS, que se soporta en Windows™ Vista™ y OS de Windows™ posteriores; y CDC/ECM (modelo de control de Ethernet CDC) es un tipo de dispositivo de interfaz de red USB, que se soporta por MAC™ OS y Linux™. Al mismo tiempo, dado que muchas plataformas informáticas soportan USB, es deseable que cualquier dispositivo USB dado pueda conectarse a, y usarse con,  
25 diversas plataformas informáticas diferentes. Sin embargo, con frecuencia diferentes OS soportan diferentes tipos de dispositivo USB, y el procedimiento de enumeración de dispositivo requerido puede ser diferente dependiendo de la clase de anfitrión al que se conecta el dispositivo. Para un dispositivo USB dado, con frecuencia es difícil usar un único flujo o secuencia de enumeración para satisfacer a todos los OS.

30 El documento EP1832982A1 da a conocer un dispositivo USB que es adecuado para determinar un sistema operativo de anfitrión basándose en un comando que se recibe en una posición específica en una secuencia de comando. El documento US2008/0005370A1 describe un método para identificar un sistema operativo en un sistema informático. WO2004/046942A1 describe un adaptador de red USB que proporciona soporte para múltiples configuraciones.

### Sumario

35 Tal como se indicó anteriormente, el procedimiento de enumeración de dispositivo USB puede ser diferente dependiendo de la clase de anfitrión USB al que se conecta el dispositivo. Por tanto, una realización de la presente invención proporciona un método en el que un dispositivo USB puede ejecutar procedimientos de enumeración correspondientes a diferentes tipos de OS, y cuando el dispositivo USB se conecta a un anfitrión USB, el dispositivo USB identifica en primer lugar el tipo de OS del anfitrión USB, antes de comenzar el procedimiento de enumeración real. Entonces el dispositivo USB ejecuta etapas de enumeración adecuadas dependiendo del tipo de OS  
40 identificado.

Además, el comportamiento de BIOS de un anfitrión USB puede alterar el flujo de detección de tipo de OS por parte del dispositivo USB. Por tanto, según otra realización de la presente invención, cuando el dispositivo USB se conecta a un anfitrión USB, el dispositivo USB detecta si el anfitrión está en una fase de BIOS, y si es así, el dispositivo USB altera su comportamiento para impedir que se produzca el flujo de detección de OS en la fase de  
45 BIOS. Específicamente, se usa una ID específica de proveedor para lograr un objetivo.

Además, según aún otra realización de la presente invención, el flujo de enumeración de dispositivo puede configurarse por un proveedor de sistema, de modo que el dispositivo USB puede adaptarse para diferentes aplicaciones.

50 Por consiguiente, la presente invención se refiere a un método de enumeración de dispositivo USB que elimina sustancialmente uno o más de los problemas debidos a limitaciones y desventajas de la técnica relacionada.

Un objetivo de la presente invención es proporcionar un método de enumeración de USB que permita que un dispositivo USB realice de manera fiable la enumeración de dispositivo para diferentes tipos de OS de anfitrión.

Otro objetivo de la presente invención es proporcionar un método de enumeración de USB que elimine la influencia no deseada por la fase de BIOS.

Aún otro objetivo de la presente invención es proporcionar una enumeración de USB configurable por el cliente que pueda tratar de manera flexible con diferentes tipos de OS de anfitrión.

Características y ventajas adicionales de la invención se expondrán en las siguientes descripciones y en parte resultarán evidentes a partir de la descripción, o pueden aprenderse mediante la puesta en práctica de la invención. Los objetivos y otras ventajas de la invención se realizarán y obtendrán mediante la estructura particularmente indicada en la descripción escrita y las reivindicaciones de la misma así como en los dibujos adjuntos.

Para lograr estas y otras ventajas y según el propósito de la presente invención, tal como se implementa y se describe de manera amplia, la presente invención proporciona un método de enumeración de dispositivo USB implementado en un dispositivo USB, que incluye: (a) determinar un tipo de sistema operativo (OS) de un anfitrión USB conectado al dispositivo USB; y (b) realizar una enumeración de dispositivo USB con el anfitrión USB usando un procedimiento de enumeración de dispositivo USB seleccionado de una pluralidad de procedimientos de enumeración de dispositivo USB almacenados en el dispositivo USB basándose en el tipo de OS determinado del anfitrión USB.

La etapa (a) puede incluir: (a1) presentar el dispositivo USB al anfitrión USB como dispositivo USB soportado de manera habitual; (a2) recopilar factores de comportamiento de OS de anfitrión a partir de mensajes recibidos a partir del anfitrión USB; y (a3) determinar el tipo de OS del anfitrión USB basándose en los factores de comportamiento de OS de anfitrión recopilados.

Preferiblemente, el dispositivo USB soportado de manera habitual es un dispositivo de almacenamiento masivo USB o un dispositivo de interfaz humana USB. La pluralidad de tipos de OS de anfitrión que pueden determinarse incluyen Windows™ Vista, Win7, Windows™ XP, Linux y MAC OS.

Un procedimiento de aprendizaje para generar una base de datos de conocimiento de comportamiento de OS incluye: (c1) presentar el dispositivo USB a un anfitrión USB conectado al dispositivo USB como dispositivo USB soportado de manera habitual, teniendo el anfitrión USB un tipo de OS conocido; (c2) recopilar factores de comportamiento de OS de anfitrión a partir de mensajes recibidos a partir del anfitrión USB; (c3) repetir las etapas (c1) y (c2) para una pluralidad de anfitriones USB que tienen cada uno un tipo de OS conocido; y (c4) analizar los factores de comportamiento de OS de anfitrión recopilados en la etapa (c2) para descubrir relaciones entre los factores de comportamiento de OS y los tipos de OS.

La etapa de determinación puede incluir etapas que hacen que una fase de BIOS detenga el procedimiento de enumeración y después continuar la determinación de tipo de OS de anfitrión tras la fase de BIOS, incluyendo: (a1) presentar el dispositivo USB al anfitrión USB como dispositivo USB soportado de manera habitual; (a2) si se recibe una solicitud de descriptor de dispositivo a partir del anfitrión USB antes de recibir cualquier mensaje que establece una dirección de dispositivo, devolver un descriptor de dispositivo al anfitrión USB que indica un dispositivo específico de proveedor; (a3) si se recibe una solicitud de descriptor de dispositivo a partir del anfitrión USB tras recibir un mensaje que establece una dirección de dispositivo, devolver un descriptor de dispositivo que indica un dispositivo USB soportado de manera habitual; (a4) tras la etapa (a3), recopilar factores de comportamiento de OS de anfitrión a partir de mensajes recibidos a partir del anfitrión USB; y (a5) determinar el tipo de OS del anfitrión USB basándose en los factores de comportamiento de OS de anfitrión recopilados.

El procedimiento usado en la etapa de enumeración (b) puede ser configurable por el cliente, de modo que el dispositivo USB puede configurarse para actuar como diferentes dispositivos USB dependiendo del tipo de OS de anfitrión determinado.

En otro aspecto, la presente invención proporciona un dispositivo USB que tiene un controlador que ejecuta programas para realizar los métodos anteriores.

Debe entenderse que tanto la descripción general anterior como la siguiente descripción detallada son a modo de ejemplo y explicativas y se pretende que proporcionen explicación adicional de la invención tal como se reivindica.

#### **Breve descripción de dibujos**

La figura 1 ilustra un procedimiento de enumeración de dispositivo USB según una primera realización de la presente invención.

La figura 2 ilustra los mensajes de USB durante un procedimiento de enumeración de dispositivo en la primera realización.

La figura 3 ilustra una máquina de estados de un flujo de detección de OS a modo de ejemplo por el dispositivo USB en la primera realización.

La figura 4 ilustra un procedimiento para generar una base de datos de conocimiento de comportamiento de OS en la primera realización.

La figura 5 ilustra un procedimiento convencional en un anfitrión USB para configurar un dispositivo USB.

La figura 6 ilustra un procedimiento de enumeración de dispositivo USB comenzando a partir de la fase de BIOS según una segunda realización de la presente invención.

La figura 7 ilustra la transición de estados en un procedimiento de enumeración de dispositivo USB configurable por el cliente según una tercera realización de la presente invención.

5 Las figuras 8 y 9 ilustran dos ejemplos de un flujo de procedimiento de enumeración de dispositivo USB configurable por el cliente según la tercera realización.

La figura 10 es un diagrama de flujo de alto nivel que ilustra un procedimiento de enumeración de dispositivo USB que implementa las realizaciones primera, segunda y tercera de la presente invención.

10 La figura 11 ilustra un procedimiento mediante el cual un dispositivo USB detecta la presencia de un dispositivo accionador en el anfitrión y hace que se instale el accionador cuando sea necesario.

### Descripción detallada

Una primera realización de la presente invención proporciona un método en el que un dispositivo USB puede ejecutar diferentes procedimientos de enumeración correspondientes a diferentes tipos de sistema operativo (OS). Cuando el dispositivo USB se conecta a un anfitrión USB, el dispositivo USB identifica en primer lugar el tipo de OS del anfitrión USB, antes de comenzar el flujo de enumeración real. Entonces el dispositivo USB ejecuta etapas de enumeración adecuadas dependiendo del tipo de OS identificado.

15 Realizaciones de la presente invención pueden implementarse en un dispositivo USB que incluye un circuito de interfaz de USB, un circuito de control y una memoria. El circuito de control incluye un controlador de dispositivo USB que controla la interfaz de USB, y puede incluir otros circuitos o módulos. La memoria almacena programas ejecutados por el circuito de control, descriptores de USB y otra información.

20 La figura 1 ilustra el flujo de un procedimiento de enumeración de dispositivo USB según la primera realización. Las etapas mostradas en la figura 1 se realizan por el dispositivo USB; las etapas de enumeración realizadas por el anfitrión USB durante la enumeración de dispositivo son etapas normales realizadas por anfitriones convencionales y no se alteran por esta realización. Dentro del dispositivo USB, el procedimiento mostrado en la figura 1 puede implementarse en un controlador de dispositivo USB que es un componente que controla la comunicación con un controlador de anfitrión USB del anfitrión USB, o implementarse en un módulo de detección de OS de anfitrión independiente que se comunica con el controlador de dispositivo. La implementación puede ser en forma de circuitos de hardware, o firmware o software almacenados en una memoria del dispositivo USB y ejecutados por el controlador de dispositivo USB o módulo de detección de OS de anfitrión.

30 Tal como se muestra en la figura 1, tras conectarse el dispositivo USB a un anfitrión USB (etapa S11), el dispositivo USB se presenta al anfitrión USB como dispositivo de almacenamiento masivo USB (tal como CD-ROM, unidad flash, etc.), o como otro dispositivo USB soportado de manera habitual (es decir un dispositivo que se usa ampliamente y se soporta por prácticamente todas las plataformas informáticas; incluyendo los ejemplos dispositivos de clase de HID (dispositivo de interfaz humana) tales como ratón y teclado) (etapa S12). En esta etapa, el dispositivo USB se presenta como dispositivo habitualmente soportado (devolviendo el descriptor de dispositivo apropiado) independientemente de qué dispositivo es realmente. Después, el dispositivo USB lleva a cabo un procedimiento de pseudoenumeración con el anfitrión USB (etapa S13).

40 Durante este procedimiento de pseudoenumeración, el dispositivo USB recibe mensajes (solicitudes) a partir del anfitrión, que el anfitrión envía como parte normal de su procedimiento de enumeración. El dispositivo USB puede responder a las solicitudes del anfitrión según sea apropiado, aunque el contenido de la respuesta puede no reflejar las características auténticas del dispositivo real. Los mensajes a partir del anfitrión USB contienen diversa información, parte de la cual es característica del tipo de OS del anfitrión. Por ejemplo, la solicitud a partir del anfitrión puede indicar si el anfitrión soporta determinadas clases de dispositivos USB; los valores de determinados campos de algunos mensajes pueden ser diferentes para diferentes OS, etc. El dispositivo USB recopila tal información a partir de los mensajes y la usa para intentar determinar el tipo de OS del anfitrión (etapa S14). En la etapa de determinación, el dispositivo USB puede usar diversos algoritmos y lógicas, incluyendo lógica difusa, para determinar el tipo de OS del anfitrión, usando conocimiento previamente almacenado (por ejemplo una base 16 de datos de conocimiento de comportamiento de OS previamente almacenada en el dispositivo USB) sobre los comportamientos esperados de los diferentes tipos de OS.

50 Por ejemplo, la información (algunas veces denominada factores de comportamiento de OS en esta divulgación) recopilada por el dispositivo USB en la etapa de pseudoenumeración puede proceder de mensajes relacionados con MOD (descriptor de OS de Microsoft, un conjunto de descriptores de cadena y solicitud de proveedor de USB usado para permitir que el dispositivo USB subyacente reivindique su clase de dispositivo compatible o cambie configuraciones en el anfitrión), wLength de descriptor de obtener configuración, mensajes relacionados con IAD (descriptor de asociación de interfaz, un descriptor de USB para informar al anfitrión de la relación de interfaces de un dispositivo USB), identidad de la segunda solicitud tras SET\_ADDRESS (establecer dirección), etc.:

## ES 2 724 523 T3

- MOD-ID compat. = S o N,
  - MOD-Propiedad ext. = S o N,
  - Soporte de IAD = S o N
  - wLength del primer descriptor de configuración, wLength = 4, wLength = 255, wLength = otro valor
- 5 - etc.

La tabla I a continuación es un resumen de algunos factores de comportamiento a modo de ejemplo basados en la observación.

Tabla I

OS	MOD - ID compat.	MOD - Propiedad ext.	Soporte de IAD	wLength de 2º descriptor de configuración	2ª acción tras establecer dirección	wLength de 1º descriptor de configuración
Win7 X64	S	S	S	N/A	Descriptor de obtener configuración	255
Win7 X86	S	S	S	N/A	Descriptor de obtener configuración	255
Vista SP2 X64	S	S	S	N/A	Descriptor de obtener configuración	255
Vista SP2 X86	S	S	S	N/A	Descriptor de obtener configuración	255
Vista X86	S	S	S	N/A	Descriptor de obtener configuración	255
XP X86 SP3	S	S	S	255	Descriptor de obtener configuración	9
XP X86 SP2	S	S	S	255	Descriptor de obtener configuración	9
XP X86 SP1	S	N	N	255	Descriptor de obtener configuración	9
XP X86	N	N	N	255	Descriptor de obtener configuración	9
Linux x86 (2.6.31)	N	N	N	ajuste	Descriptor de obtener configuración	9
MAC-OS 10.5 x86	N	N	N	ajuste	Descriptor de obtener configuración	4
MAC-OS 10.6 x86	N	N	N	ajuste	Obtener cadena de iProduct	4
PC0910191487 arrancar a Win7 X64	N	N	N	ajuste	Descriptor de obtener configuración	9
PC09701019335 arrancar XP SP2	N	N	N	N/A	Establecer configuración	255
PC0701014439 arrancar XP	N	N	N	N/A	Obtener bRequest=0xFE	511
Arrancar MAC-OS 10.5 x86	N	N	N	ajuste	Descriptor de obtener configuración	4
Arrancar MAC-OS 10.6 x86	N	N	N	ajuste	Obtener cadena	2

En una realización particular, en la etapa S14, los factores de comportamiento de OS se usan para determinar el tipo de OS usando un sistema de puntuación. La asignación de puntuación descrita en la tabla II a continuación se usa como ejemplo:

Tabla II

Factor de comportamiento de OS		puntuación
Con MOD		+1000
primer Obtener configuración	wLength = 4	-100
	wLength = 9	-50
	wLength = 255	+1000
2º descriptor de obtener configuración	wLength = 255	+500
SET_ADDRESS		Poner a cero

- 5 Después de que el dispositivo USB recopile los diversos factores de comportamiento (etapa S13), se calcula el total de las puntuaciones, y se determina el tipo de OS de anfitrión a partir de la puntuación total tal como se muestra en la tabla III a continuación:

Tabla III

Tipo de OS de anfitrión	puntuación total
Windows™ Vista, Win7	+2000
Windows™ XP SP2, SP3	+1450
Windows™ XP SP1	+1450
Windows™ XP	+450
Linux	-50
MAC OS	-100

- 10 La asignación de puntuación en la tabla II y la relación de puntuación frente a OS en la tabla III pueden almacenarse en la base 16 de datos de conocimiento de comportamiento de OS.

A partir de lo anterior, puede observarse que este sistema de puntuación puede detectar diversos OS incluyendo Windows™ Vista y Win7, Windows™ XP, Linux y MAC OS. Este sistema de puntuación es un algoritmo de detección difuso y puede determinar el tipo de OS más probable del anfitrión. Evidentemente, el sistema de puntuación mostrado anteriormente es simplemente un ejemplo; pueden asignarse otras puntuaciones, siempre que las puntuaciones sirvan para diferenciar las características de los diversos OS. También pueden usarse otros tipos de algoritmos de detección de OS. Puede usarse cualquier algoritmo de detección de OS adecuado para implementar realizaciones de la presente invención. Además, los factores de comportamiento de OS para diversas plataformas informáticas pueden cambiar en el futuro, y puede necesitarse cambiar el algoritmo de detección de OS en consecuencia.

- 20 Tras detectarse el tipo de OS de anfitrión, el dispositivo USB puede comenzar un procedimiento de enumeración real para el dispositivo USB objetivo usando un flujo de procedimiento según el tipo de OS detectado (etapa S15). En este caso, el término "dispositivo USB objetivo" se refiere al tipo de dispositivo como el que funcionará realmente el dispositivo USB, en vez del tipo de dispositivo que se presenta inicialmente al anfitrión USB en las etapas de pseudoenumeración de S12 a S14. El dispositivo USB se carga previamente con información relevante que permite que el dispositivo lleve a cabo la enumeración de manera correcta para diversos tipos de OS; en la etapa S15, la información para el tipo de OS determinado en la etapa S14 se usa para llevar a cabo el procedimiento de enumeración real.

30 Para mejorar la precisión de detección de OS, las etapas S12, S13 y S14 de la figura 1 pueden repetirse para iteraciones adicionales. Cada iteración adicional puede iniciarse simulando un desacoplamiento de USB y después un acoplamiento de USB. En diferentes iteraciones, el dispositivo USB puede presentarse como diferentes tipos de dispositivos USB para recopilar diferentes factores de comportamiento del OS. Por ejemplo, en una primera iteración, los dispositivos USB pueden presentarse como dispositivo de almacenamiento masivo USB, y recopilar factores de comportamiento de OS suficientes para determinar si el anfitrión ejecuta un OS de Windows o un OS distinto de Windows. Si se determina que el OS es un OS distinto de Windows, se ejecuta una segunda iteración en la que el dispositivo USB se presenta como un dispositivo de CDC/ECM, y se recopila el factor de comportamiento de OS para determinar si el anfitrión es un MAC OS/Linux o algo distinto (por ejemplo fase de BIOS, que se considera un tipo de OS particular en esta realización).

- 40 En el procedimiento de detección de OS de anfitrión de la figura 1 (etapas S12-S14), usar un dispositivo USB soportado de manera habitual (por ejemplo un dispositivo de almacenamiento masivo) para recopilar la información de comportamiento de OS puede impedir que el procedimiento de inicialización de dispositivo USB se bloquee en algún OS. Por ejemplo, en OS de la serie Windows™ XP, si se usa inicialmente un dispositivo USB desconocido, el OS visualizará un cuadro de diálogo emergente para pedir al usuario que instale un accionador para el dispositivo

antes de acceder a otros nuevos dispositivos USB, y el usuario debe cancelar el cuadro de diálogo emergente con el fin de seguir avanzando. Esto resultaría inconveniente para el usuario.

La figura 2 ilustra esquemáticamente el flujo de mensajes durante el procedimiento de enumeración de la figura 1, usando un ejemplo particular. En este ejemplo, el dispositivo USB incluye un controlador 22 de dispositivo USB y un módulo 21 de detección de anfitrión independiente. El módulo 21 de detección de anfitrión le indica al controlador 22 de dispositivo que inicialice la detección de anfitrión. El controlador de dispositivo envía señales de desacoplamiento de USB y acoplamiento de USB al anfitrión (al controlador 23 de anfitrión), y el anfitrión envía varios mensajes al controlador de dispositivo, tales como SET\_ADDRESS, GET\_DESCRIPTOR (obtener descriptor) que es una solicitud de descriptores a partir del dispositivo, descriptores MOD, SET\_CONFIGURATION (establecer configuración), etc., tal como se representa en la figura 2. Los mensajes a partir del anfitrión se reenvían por el controlador 22 de dispositivo al módulo 21 de detección de anfitrión que determina el tipo de OS de anfitrión.

Dado que el dispositivo USB se presenta inicialmente al anfitrión USB como dispositivo de almacenamiento masivo, el anfitrión USB enviará mensajes relevantes para un dispositivo de almacenamiento masivo. En un determinado punto, después de que el módulo 21 de detección de anfitrión haya recopilado suficiente información de comportamiento de OS y determinado el tipo de OS, el módulo 21 de detección de anfitrión hace que el controlador 22 de dispositivo USB se desacople del anfitrión USB. Esto puede realizarse, por ejemplo, enviando un comando de apagado al controlador 22 de dispositivo USB tal como se ilustra en la figura 2. Posteriormente, el módulo 21 de detección de anfitrión hace que el controlador 22 de dispositivo USB se acople de nuevo al anfitrión USB, esta vez presentándose como el dispositivo USB objetivo. El momento de las señales de desacoplamiento y acoplamiento de USB debe elegirse de tal manera como para evitar errores perceptibles por el usuario.

Aunque en la figura 2 el módulo 21 de detección de anfitrión y el controlador 22 de dispositivo USB son dos módulos independientes, alternativamente pueden ser un único módulo (por ejemplo un controlador de dispositivo USB) que realiza las funciones tanto de comunicación con el anfitrión USB como de detección de OS.

La figura 3 ilustra una máquina de estados de un flujo de detección de OS a modo de ejemplo por el dispositivo USB. En este diagrama, los bloques marcados "descriptor de obtener configuración", "descriptor MOD" y "SET\_CONFIGURATION" representan el dispositivo USB que recibe información desde el exterior (el anfitrión USB). A partir de los mensajes de descriptor de obtener configuración, el dispositivo USB mantiene wLength del descriptor de configuración, y continúa el procedimiento de detección. Cuando el dispositivo USB recibe el descriptor MOD, gestiona el descriptor MOD según sea apropiado, y termina el procedimiento de detección. Cuando el dispositivo USB recibe SET\_CONFIGURATION, termina el procedimiento de detección. El flujo de detección a modo de ejemplo ilustrado en la figura 3 corresponde de manera general al ejemplo mostrado en la figura 2.

La figura 4 ilustra un procedimiento para generar una base 16 de datos de conocimiento de comportamiento de OS. En primer lugar, se conecta un dispositivo USB a un anfitrión USB que tiene un OS conocido (etapa S41). El dispositivo USB se presenta como un dispositivo de almacenamiento masivo USB o algún otro dispositivo USB soportado de manera habitual, y realiza una pseudoenumeración similar a la etapa S13 de la figura 1 para recopilar datos de comportamiento a partir del anfitrión USB (etapa S42). Estas etapas se repiten para múltiples anfitriones USB que ejecutan diferentes OS conocidos (etapa S43). Después, se lleva a cabo un análisis para descubrir correlaciones entre factores de comportamiento de OS y los tipos de OS conocidos, y para diseñar un algoritmo para detectar el tipo de OS basándose en factores de comportamiento de OS y sus correlaciones con los tipos de OS conocidos (etapa S44). La etapa S44 puede realizarse por un ser humano o por un ordenador, o por un ser humano con la ayuda de un ordenador. Cuando se realiza por un ordenador, la etapa S44 es esencialmente un procedimiento de aprendizaje de máquina. Tras la etapa S44, las etapas S41 a S43 pueden repetirse para recopilar más datos de comportamiento de OS cuando se desee llevar a cabo iteraciones adicionales de la etapa S44. El resultado del procedimiento de aprendizaje, tal como un sistema de puntuación representado por las tablas II y III anteriores, y/u otros algoritmos de detección de OS, se almacenan en la base 16 de datos de conocimiento de comportamiento de OS (etapa S45).

El procedimiento ilustrado en la figura 4 es un procedimiento de aprendizaje, y puede usarse para acumular una base 16 de datos de conocimiento sin ningún conocimiento previo. Por tanto, la base 16 de datos de conocimiento de comportamiento de OS puede generarse por el fabricante de dispositivos USB usando el procedimiento de aprendizaje. Sin embargo, el comportamiento de OS existentes con respecto a dispositivos USB soportados de manera habitual (por ejemplo dispositivo de almacenamiento masivo USB) tiende a ser relativamente constante; por tanto, una base 16 de datos generada por una parte puede usarse por múltiples fabricantes de dispositivos USB. Cuando se añaden nuevos OS al conjunto de OS, o cuando se producen cambios en la industria de USB tales como cambio en las especificaciones o adición de nuevas clases de dispositivo, etc., puede ser necesario volver a realizar el procedimiento de aprendizaje para actualizar o reconstruir la base 16 de datos de conocimiento de comportamiento de OS. También puede ser deseable equipar cada dispositivo USB con un algoritmo de aprendizaje para permitir que un proveedor de sistema mejore la precisión de detección y cobertura de tipo de OS del algoritmo de detección de anfitrión USB, o para permitir que un usuario final realice solución re problemas en el procedimiento de detección de anfitrión USB.

- Una segunda realización de la presente invención proporciona un método de detección de OS al tiempo que evita la influencia no deseable de la fase de BIOS del ordenador anfitrión. Un dispositivo USB se configura tras conectarse el dispositivo a un ordenador, así como cuando el dispositivo ya está físicamente conectado al anfitrión y el anfitrión entra en la fase de BIOS. Un ordenador anfitrión entra normalmente en la fase de BIOS cuando se arranca el ordenador o se despierta a partir de un estado de hibernación. En la fase de BIOS, el comportamiento de algunos anfitriones puede interferir de manera indeseable con el procedimiento de detección de OS de anfitrión del dispositivo USB.
- Por ejemplo, en algunos OS, si el dispositivo USB se ha configurado en la fase de BIOS, el dispositivo no se reconfigurará posteriormente. Esto interfiere con el procedimiento de detección de OS porque el procedimiento de enumeración de dispositivo USB en la fase de BIOS habitualmente no tiene nada que ver con la fase de OS inminente. Por tanto, si el dispositivo no puede realizar la enumeración de dispositivo USB tras entrar en la fase de OS, el tipo de OS determinado en la fase de BIOS puede ser incorrecto y proporcionar al usuario un dispositivo USB inutilizable.
- Para evitar tal interferencia indeseada de la BIOS, se proporciona un método según una segunda realización de la presente invención, mediante el cual el dispositivo USB usa una ID específica de proveedor para impedir que la BIOS configure el dispositivo.
- Convencionalmente, antes de usar un dispositivo USB, la BIOS o el OS consulta al dispositivo para solicitar descriptores varias veces para identificar el tipo de dispositivo USB, y carga el accionador relacionado para gestionar el dispositivo USB si es necesario. Este procedimiento se ilustra de manera general en la figura 5. Si el dispositivo USB es un dispositivo específico de proveedor que requiere un accionador específico de proveedor, el dispositivo especificará una id de clase de dispositivo "0xff". Por consiguiente, el anfitrión USB cargará un accionador específico de proveedor para gestionar un dispositivo USB de este tipo (véase la figura 5, etapa S54).
- La segunda realización de la presente invención aprovecha esta característica del anfitrión USB. Según la segunda realización, cuando el anfitrión USB consulta al dispositivo USB para solicitar descriptor de dispositivo por primera vez (es decir, antes de un mensaje de SET\_ADDRESS a partir del anfitrión), el dispositivo USB devuelve un descriptor de dispositivo que tiene una id de clase de dispositivo "0xff". Esto puede hacer que la BIOS detenga el procedimiento de enumeración de USB y, como resultado, el dispositivo puede configurarse posteriormente por el OS. Este procedimiento se ilustra en la figura 6.
- Tal como se muestra en la figura 6, cuando la BIOS del anfitrión USB solicita el descriptor de dispositivo por primera vez, el dispositivo USB devuelve un descriptor de dispositivo falso que tiene una id de clase de dispositivo "0xff". Alguna BIOS también solicitará el descriptor de configuración, en cuyo caso el dispositivo USB devolverá un descriptor de configuración falso 0xff. La etapa de solicitar el descriptor de configuración no siempre se produce. En este punto, dado que la id de clase de dispositivo indica un dispositivo específico de proveedor, la BIOS detendrá el procedimiento de enumeración de USB.
- Posteriormente, en la fase de OS (es decir, tras concluirse la fase de BIOS), el anfitrión USB envía un mensaje SET\_ADDRESS al dispositivo USB. Después de eso, cuando el dispositivo USB recibe otra solicitud del descriptor de dispositivo a partir del anfitrión USB, el dispositivo USB devuelve un descriptor para un dispositivo de almacenamiento masivo USB u otro dispositivo USB soportado de manera habitual. El anfitrión USB también solicita el descriptor de configuración, y el dispositivo USB devuelve un descriptor apropiado (la etapa de devolver el descriptor de configuración puede omitirse). En este momento, puede llevarse a cabo el procedimiento de la primera realización, que incluye recopilar factores de comportamiento de OS, determinar el tipo de OS, y realizar la enumeración de dispositivo USB específica para el tipo de OS determinado.
- Obsérvese que tras concluirse la fase de BIOS, la mayoría de los anfitriones USB enviarán otra solicitud de descriptor de dispositivo tras SET\_ADDRESS, de modo que no se necesita desacoplar y volver a acoplar el dispositivo USB para iniciar un procedimiento de enumeración de USB. En algunos OS, el anfitrión USB no vuelve a solicitar el descriptor de dispositivo. En tal situación, el dispositivo USB puede simular un desacoplamiento de USB y después un acoplamiento de USB para iniciar el procedimiento.
- Haciendo ahora referencia a la máquina de estados ilustrada en la figura 3, si el dispositivo USB impide satisfactoriamente que la BIOS complete el procedimiento de enumeración de USB, la máquina de estados no entrará en el estado END\_DETECTING (fin detección). Sin embargo, si el dispositivo USB no logra impedir que la BIOS complete el procedimiento de enumeración de USB, la máquina de estados entrará en el estado END\_DETECTING sin haber determinado correctamente el tipo de OS. En este caso, el dispositivo USB puede iniciar otra enumeración de dispositivo USB simulando un desacoplamiento de USB y después un acoplamiento de USB.
- Para adaptar un dispositivo USB a diferentes aplicaciones, es deseable proporcionar un procedimiento configurable por el cliente para la enumeración de dispositivo. Una tercera realización de la presente invención permite tal procedimiento de enumeración de dispositivo configurable por el cliente.



Un procedimiento de enumeración de dispositivo configurable por el cliente es útil en situaciones, por ejemplo, en las que un dispositivo USB está diseñado para tener hardware, firmware y/o software que le permiten funcionar como diferentes tipos de dispositivos USB. Por ejemplo, un dispositivo USB puede estar diseñado de modo que puede funcionar como ISD (un tipo de dispositivo de interfaz de red soportado en Windows™ Vista™ y OS de Windows™ posteriores), o un RNDIS (un tipo de dispositivo de interfaz de red USB soportado en Windows™ XP y OS de Windows™ posteriores), o un CDC/ECM (un tipo de dispositivo de interfaz de red USB soportado por MAC™ OS y Linux™), para realizar funciones de interfaz de red. Después, cuando el dispositivo USB se conecta a un anfitrión, el dispositivo USB detecta el tipo de OS del anfitrión, y se presenta al anfitrión USB como uno de los dispositivos de interfaz de red anteriores dependiendo del tipo de OS detectado, de modo que puede soportarse de manera apropiada por el anfitrión para llevar a cabo funciones de interfaz de red.

Un procedimiento de enumeración de dispositivo configurable por el cliente según la tercera realización puede implementarse usando una máquina de estados tal como se ilustra de manera general en la figura 7, y se muestran dos ejemplos en las figuras 8-9. La figura 7 ilustra de manera general la transición de estados en el flujo de enumeración de dispositivo. El estado inicial S71 es el estado en el que está el dispositivo USB tras haberse completado la etapa de detección de OS. Este estado puede denominarse el estado de MSD (dispositivo de almacenamiento masivo) en esta divulgación porque el dispositivo USB se ha presentado como dispositivo de almacenamiento masivo durante el procedimiento de detección de OS. Debe observarse que el estado inicial (el resultado de detección de OS) no es específico para ningún aspecto configurable por el cliente del procedimiento de enumeración.

Después, se realiza una etapa de traducción de estado (etapa S72), usando una tabla 77 de traducción definida por el cliente, para traducir el estado del estado inicial S71 a otro estado definido de la máquina de estados (denominado primer estado S73). La tabla de traducción define una asociación entre el resultado de detección de OS y los estados. Por ejemplo, el procedimiento de detección de OS puede generar tres resultados posibles: 1: Windows Vista y Windows 7; 2: Linux; 3: MAC OS; 4: Windows XP SP2 y posterior; y 5: otros; y la tabla de traducción puede tener la siguiente forma:

Traducción = (1, VISTA7)(2, LINUXMAC)(3, LINUXMAC)(4, XPSP)(5, OTRO)

en la que VISTA7 y LINUXMAC son dos estados en la máquina de estados. Por tanto, tras la etapa de traducción de estado S72, el primer estado S73 puede ser los estados VISTA7 o LINUXMAC.

Después, el dispositivo USB realiza un reinicio de USB (etapa S74). Por ejemplo, esto puede realizarse mediante un desacoplamiento y acoplamiento de USB. La máquina de estados está en un segundo estado S75 tras el reinicio de USB. Después, la máquina de estados pasa por una transición de estados denominada generalmente etapa de "enumeración de dispositivo configurable" S76 en la figura 7, aplicando reglas de transición de estados definidas por el cliente almacenadas en una base 78 de datos de comportamiento de enumeración del cliente.

En este caso se usan dos ejemplos específicos de flujo de procedimiento de enumeración de dispositivo USB con fines de ilustración, tal como se muestra en las figuras 8 y 9. En el procedimiento mostrado en la figura 8, el dispositivo USB se presenta inicialmente como MSD (el estado inicial) y realiza la detección de OS. El procedimiento de enumeración de dispositivo que va a llevarse a cabo depende del resultado de la detección de OS y puede configurarse por el cliente. En este ejemplo particular, el procedimiento de enumeración está configurado de tal manera que: si el OS detectado es Windows Vista o posterior, se lleva a cabo un procedimiento de enumeración de dispositivo para ISD (en el que el dispositivo se presenta al anfitrión como ISD notificando el descriptor de dispositivo para ISD); si el OS detectado es Windows XP SP2 o XP SP3, se lleva a cabo un procedimiento de enumeración de dispositivo para RNDIS + MSD; si el OS detectado es Windows XP o XP SP1, o un OS desconocido, se lleva a cabo un procedimiento de enumeración de dispositivo para RNDIS + MSD; y si el OS detectado es MAC-OS o Linux, se lleva a cabo un procedimiento de enumeración de dispositivo para CDC + MSD. Estos procedimientos se representan como los cuatro estados marcados "ISD", "RNDIS + MSD", "RNDIS + MSD" y "CDC + MSD" en la figura 8. Tal como se muestra en la figura 8, estos cuatro estados no experimentan transición a otros estados.

En esta divulgación, la notación "+" indica un dispositivo USB compuesto que tiene múltiples unidades funcionales.

Los procedimientos de enumeración para múltiples dispositivo USB tipos están previamente almacenados en el dispositivo USB, sin embargo, el cliente puede configurar el dispositivo para especificar qué procedimientos de enumeración se realizan, en qué condiciones, en qué secuencia, etc. En una implementación particular, el cliente configura el procedimiento proporcionando elementos de configuración que definen transición de estados, en la que los elementos de configuración pueden tener el siguiente formato: (estado, tipo de dispositivo, tiempo, siguiente estado).

En este caso, "estado" y "siguiente estado" son nombres de estados de la máquina de estados. "Tipo de dispositivo" es el tipo de dispositivo USB como el que se presentará el dispositivo USB en la enumeración de dispositivo en el estado actual. "Tiempo" es un parámetro que determina la transición al siguiente estado. Tiempo = n significa el estado caducará y experimentará transición al siguiente estado en n segundos, y tiempo = 0 significa que el estado nunca caducará (es decir, es para siempre).

Además del valor de “tiempo”, también se aplican las siguientes reglas de transición de estados. En primer lugar, la cuenta atrás de caducidad se detiene si se recibe un mensaje de establecer configuración de USB para los tipos de dispositivo CDC, RNDIS e ISD, de modo que el estado no experimentará transición al siguiente estado. En segundo lugar, si se recibe un mensaje de eyección de MSD, el estado experimentará inmediatamente transición al siguiente estado.

Estos elementos de configuración los proporciona el cliente y se almacenan en el dispositivo USB. Por ejemplo, para implementar el flujo mostrado en la figura 8, el cliente puede proporcionar elementos de configuración de la siguiente manera:

```
FLUJO 1 = (VISTA7, ISD, 0, VISTA7)(XPSP, RNDIS+MSD, 0, XPSP)(LINUXMAC, CDC+MSD, 0,
LINUXMAC)(OTRO, RNDIS+MSD, 0, OTRO)
```

De manera similar, en el ejemplo mostrado en la figura 9, el dispositivo USB se presenta inicialmente como MSD (el estado inicial) y realiza la detección de OS. El procedimiento de enumeración está configurado de tal manera que: si el OS detectado es Windows Vista y posterior, se lleva a cabo un procedimiento de enumeración de dispositivo para ISD; si el OS detectado es Windows XP SP2 y XP SP3, o Windows XP y XP SP1, o un OS desconocido, se lleva a cabo un procedimiento de enumeración de dispositivo para MSD; y si el OS detectado es MAC-OS o Linux, se lleva a cabo un procedimiento de enumeración de dispositivo para CDC. Estos procedimientos se representan como los tres estados marcados “ISD”, “MSD” y “CDC” en la figura 9. En este flujo, el estado MSD experimenta transición al estado “RNDIS” cuando se produce la caducidad o si se recibe un MSD EJECT (expulsar MSD). Dicho de otro modo, mientras se realiza la enumeración para un MSD, se recibe un MSD EJECT o se produce una caducidad, el dispositivo USB comienza a realizar un procedimiento de enumeración para RNDIS.

Para implementar el flujo mostrado en la figura 9, el cliente puede proporcionar elementos de configuración de la siguiente manera:

```
FLUJO 2 = (VISTA7, ISD, 0, VISTA7)(LINUXMAC, CDC, 0, LINUXMAC)(OTRO, MSD, 16, OTRO 1)(OTRO, RNDIS,
0, OTRO 1)
```

Debe observarse en este caso que en las representaciones de las figuras 8 y 9, los estados se marcan por el tipo de dispositivo USB tal como “ISD”, etc., mientras que en el formato de elemento de configuración descrito anteriormente, los estados se identifican por el tipo de OS tal como “VISTA7”, etc. Estas etiquetas diferentes se refieren a las mismas cosas. Debe entenderse que el uso de etiquetas diferentes se debe simplemente a conveniencia en la representación gráfica y textual, respectivamente.

Debe observarse que la implementación del procedimiento de enumeración de dispositivo para cada tipo de OS y cada tipo de dispositivo puede lograrse por los expertos en la técnica sin experimentación excesiva, haciendo referencia a las normas relevantes y otra documentación disponible.

Además del enfoque de transición de estados descrito anteriormente, pueden usarse otros métodos adecuados para implementar procedimientos de enumeración configurables.

La figura 10 es un diagrama de flujo de alto nivel que ilustra un procedimiento de enumeración de dispositivo USB realizado por un dispositivo USB que implementa las realizaciones primera, segunda y tercera de la presente invención. Este procedimiento se realiza cuando comienza el procedimiento de enumeración de dispositivo USB, es decir cuando se conecta el dispositivo USB a un anfitrión USB o cuando se arranca el anfitrión o se reanuda a partir de un estado de hibernación o suspensión con el dispositivo USB ya conectado. En primer lugar, si el dispositivo USB determina que el anfitrión USB está en la fase de BIOS, el dispositivo USB detiene que se complete la enumeración de dispositivo en la fase de BIOS (etapa S101). Por ejemplo, si el dispositivo USB recibe una solicitud de descriptor de dispositivo antes de SET\_ADDRESS, el dispositivo USB devuelve un descriptor de dispositivo falso que contiene una id de clase de dispositivo “0xff” (dispositivo específico de proveedor). Este procedimiento corresponde a la segunda realización mostrada en la figura 6. Después, cuando el anfitrión está fuera de la fase de BIOS, el dispositivo USB realiza un procedimiento para detectar el tipo de OS del anfitrión (etapa S102). En este procedimiento se usa la base de datos de conocimiento de comportamiento de OS. Este procedimiento se describe en más detalle en la primera realización mostrada en las figuras 1-3. Después, tras determinarse el tipo de OS, el dispositivo USB realiza un procedimiento de enumeración de dispositivo usando un procedimiento seleccionado basándose en el tipo de OS detectado del anfitrión, y el procedimiento de enumeración de dispositivo puede ser configurable por el cliente para permitir que el dispositivo USB se presente como una de múltiples clases de dispositivos USB dependiendo del tipo de OS detectado (etapa S103). En este procedimiento se usa la base de datos de comportamiento de enumeración de cliente. Este procedimiento se describe en más detalle en la tercera realización mostrada en las figuras 7-9.

Puede implementarse un procedimiento adicional dentro del contexto anterior, mediante el cual el dispositivo USB detecta si un accionador de dispositivo apropiado está instalado en el anfitrión, y si no es así, hace que el accionador instale el accionador. Este método de detección de accionador usa los descriptors de OS de Microsoft (MOD) como medios para pasar información. Tal como se muestra en la figura 11, después de enchufar por primera vez el dispositivo USB en el anfitrión, el anfitrión emite una consulta al dispositivo sobre si el dispositivo soporta MOD. La

consulta es un mensaje de obtener descriptor de cadena con índice 0xEE. En respuesta, el dispositivo devuelve el descriptor de cadena de OS con un código de proveedor particular, tal como 0x66. El dispositivo puede elegir qué valor particular se usa como código de proveedor en esta respuesta. Después, el anfitrión emite una solicitud de MOD con el código de proveedor que recibió a partir del dispositivo, es decir 0x66 en este ejemplo. Cuando el dispositivo recibe esta solicitud de MOD con un código de proveedor 0x66, es decir el mismo código de proveedor que el código anteriormente enviado por el dispositivo al anfitrión, el dispositivo determina que el anfitrión no tiene el accionador apropiado instalado.

Las etapas anteriores son una parte del procedimiento de detección de tipo de OS de anfitrión (primera realización). Dicho de otro modo, el anfitrión emite las solicitudes como si fueran una parte de la enumeración de USB normal, pero el dispositivo usa información en estas solicitudes para detectar si el anfitrión tiene el accionador apropiado instalado. Estas etapas son una parte del procedimiento de detección de OS descrito como primera realización de la presente invención. En este punto, el dispositivo ha determinado el tipo de OS del anfitrión.

Cuando el dispositivo determina que no se ha instalado el accionador apropiado en el anfitrión, el dispositivo pasa a un estado de "accionador no instalado", que puede ser uno de los estados en la máquina de estados descrita en la tercera realización. En este estado, el dispositivo simula un desacoplamiento y acoplamiento de USB. Después de eso, en el siguiente procedimiento de enumeración de USB, el dispositivo se presenta como un CD-ROM u otro MSD, que tiene un programa de ejecución automática almacenado en el mismo. El programa de ejecución automática realiza, o hace que otro programa de configuración (también almacenado en el CD-ROM simulado) realice, las siguientes funciones: instalar el programa accionador apropiado, también almacenado en el CD-ROM simulado; modificar el código de proveedor almacenado en anfitrión para dar otro código de proveedor, 0x99 en este ejemplo, diferente del código de proveedor recibido anteriormente a partir del dispositivo, 0x66 en este ejemplo; y hacer que el anfitrión emita un comando de MSD EJECT, más específicamente, un comando de iniciar-detener unidad de SCSI con LoEj igual a expulsar medios.

Tras MSD EJECT, el anfitrión emite otra solicitud de MOD, pero esta vez con el nuevo código de proveedor 0x99 que ha modificado el programa de configuración o ejecución automática. A partir del valor de código de proveedor, el dispositivo determina que el anfitrión tiene ahora instalado el programa accionador apropiado, y simula un desacoplamiento y acoplamiento para iniciar un procedimiento de enumeración de dispositivo USB real. En este procedimiento de enumeración real, el dispositivo está en un estado de "accionador instalado", y se presenta como el dispositivo USB objetivo, tal como CD-ROM + NIC en el ejemplo ilustrado.

En el método anterior, el valor del código de proveedor modificado por el programa de configuración o ejecución automática puede usarse como manera para informar al dispositivo USB sobre como qué dispositivo USB objetivo debe presentarse en el procedimiento de enumeración de USB real. Por tanto, diferentes programas accionadores pueden modificar el código de proveedor para dar diferentes valores, de modo que la solicitud de MOD posterior se emitirá con estos diferentes códigos de proveedor. Cuando el dispositivo USB recibe la solicitud de MOD con un de tales códigos de proveedor, el dispositivo se presentará como dispositivo objetivo particular basándose en el valor de código de proveedor. También puede usarse el código de proveedor en el descriptor de cadena de OS para pasar otra información entre el dispositivo USB y el anfitrión.

Los procedimientos de enumeración de dispositivo USB según las diversas realizaciones se implementan en el dispositivo USB mediante hardware, firmware o software. En una implementación a modo de ejemplo, los procedimientos se realizan mediante un controlador de dispositivo USB (o una sección de control que incluye múltiples módulos de control) del dispositivo USB que ejecuta programas (firmware o software) almacenados en una memoria de programa del dispositivo USB. En particular, los programas que implementan múltiples procedimientos de enumeración de dispositivo USB se almacenan en el dispositivo USB, y puede realizarse uno seleccionado de los múltiples procedimientos de enumeración de dispositivo basándose en el tipo de OS detectado del anfitrión. El cliente puede configurar el flujo de enumeración de dispositivo USB (tal como se describió en la tercera realización) almacenando programas nuevos o modificados en la memoria de programa. Los datos relacionados (base de datos de conocimiento de comportamiento de OS, base de datos de comportamiento de enumeración de cliente, etc.) también se almacenan en una memoria del dispositivo USB.

**REIVINDICACIONES**

1. Método de enumeración de dispositivo USB implementado en un dispositivo USB que comprende un controlador (22) de dispositivo USB, que comprende:
- 5 (a) determinar un tipo de sistema operativo (OS) de un anfitrión USB conectado al dispositivo USB, en el que la etapa (a) incluye:
- (a1) presentar el dispositivo USB al anfitrión USB como dispositivo USB soportado de manera habitual;
- (a2) realizar una pseudoenumeración y recopilar factores de comportamiento de OS de anfitrión a partir de mensajes recibidos a partir del anfitrión USB, en el que los factores de comportamiento de OS de anfitrión incluyen uno o más factores seleccionados de un grupo que consiste en: mensajes relacionados con descriptor de OS de Microsoft, wLength de un primer mensaje de descriptor de obtener configuración, wLength de un segundo mensaje de descriptor de obtener configuración, mensajes relacionados con descriptor de asociación de interfaz, e identidad de una segunda solicitud tras SET\_ADDRESS; y
- 10 (a3) determinar el tipo de OS del anfitrión USB basándose en los factores de comportamiento de OS de anfitrión recopilados;
- 15 (b1) hacer que el controlador (22) de dispositivo USB se desacople del anfitrión USB enviando un comando de apagado al controlador (22) de dispositivo USB; y
- (b2) hacer que el controlador (22) de dispositivo USB se acople de nuevo al anfitrión USB, presentándose como dispositivo USB objetivo; y
- 20 (b3) realizar una enumeración de dispositivo USB con el anfitrión USB usando un procedimiento de enumeración de dispositivo USB seleccionado de una pluralidad de procedimientos de enumeración de dispositivo USB almacenados en el dispositivo USB basándose en el tipo de OS determinado del anfitrión USB.
2. Método según la reivindicación 1, en el que el dispositivo USB soportado de manera habitual es un dispositivo de almacenamiento masivo USB o un dispositivo de interfaz humana USB.
3. Método según la reivindicación 1, en el que el dispositivo USB tiene una base de datos de conocimiento de comportamiento de OS previamente almacenada que incluye información referente a factores de comportamiento conocidos de una pluralidad de tipos de OS conocidos, y en el que la etapa (a3) se realiza usando la base de datos de conocimiento de comportamiento de OS previamente almacenada.
- 25 4. Método según la reivindicación 3, en el que la pluralidad de tipos de OS conocidos incluyen Windows Vista, Win7, Windows XP, Linux y MAC OS; o comprendiendo el método además:
- 30 (c) antes de la etapa (a), generar la base de datos de conocimiento de comportamiento de OS, que comprende:
- (c1) presentar el dispositivo USB a un anfitrión USB conectado al dispositivo USB como dispositivo USB soportado de manera habitual, teniendo el anfitrión USB un tipo de OS conocido;
- (c2) recopilar factores de comportamiento de OS de anfitrión a partir de mensajes recibidos a partir del anfitrión USB;
- 35 (c3) repetir las etapas (c1) y (c2) para una pluralidad de anfitriones USB que tienen cada uno un tipo de OS conocido; y
- (c4) analizar los factores de comportamiento de OS de anfitrión recopilados en la etapa (c2) para descubrir relaciones entre los factores de comportamiento de OS y los tipos de OS.
5. Método según la reivindicación 1, en el que la etapa (a) incluye:
- 40 (a1) presentar el dispositivo USB al anfitrión USB como dispositivo USB soportado de manera habitual;
- (a2) si se recibe una solicitud de descriptor de dispositivo a partir del anfitrión USB antes de recibir cualquier mensaje que establece una dirección de dispositivo, devolver un descriptor de dispositivo al anfitrión USB que indica un dispositivo específico de proveedor;
- 45 (a3) si se recibe una solicitud de descriptor de dispositivo a partir del anfitrión USB tras recibir un mensaje que establece una dirección de dispositivo, devolver un descriptor de dispositivo que indica un dispositivo USB soportado de manera habitual;
- (a4) tras la etapa (a3), recopilar factores de comportamiento de OS de anfitrión a partir de mensajes recibidos a partir del anfitrión USB; y

(a5) determinar el tipo de OS del anfitrión USB basándose en los factores de comportamiento de OS de anfitrión recopilados.

5 6. Método según la reivindicación 1, que comprende además: antes de la etapa (a), almacenar elementos de configuración en el dispositivo USB que definen un flujo de procedimiento de enumeración, en el que la etapa (b) se realiza según el flujo de procedimiento definido por los elementos de configuración; o

10 en el que la etapa (a) determina además si un programa accionador apropiado para un dispositivo USB objetivo está instalado en el anfitrión USB, y en el que la etapa (b) incluye, si el accionador no está instalado, presentar el dispositivo USB como dispositivo de almacenamiento masivo (MSD) que almacena el accionador y un programa de ejecución automática que hace que el accionador se instale en el anfitrión, en el que preferiblemente la etapa (a) incluye determinar si el programa accionador está instalado en el anfitrión USB basándose en un valor de un código de proveedor en una solicitud de MOD (descriptor de OS de Microsoft) a partir del anfitrión USB.

7. Dispositivo USB que comprende:

un circuito de interfaz de USB;

15 un circuito de control que controla el circuito de interfaz de USB, en el que el circuito de control incluye un controlador (22) de dispositivo USB; y

un código de programa de almacenamiento en memoria ejecutado por el circuito de control, estando el código de programa configurado para hacer que el circuito de control ejecute un procedimiento que incluye:

(a) determinar un tipo de sistema operativo (OS) de un anfitrión USB conectado al dispositivo USB, en el que la etapa (a) incluye:

20 (a1) presentar el dispositivo USB al anfitrión USB como dispositivo USB soportado de manera habitual;

25 (a2) recopilar factores de comportamiento de OS de anfitrión a partir de mensajes recibidos a partir del anfitrión USB, en el que los factores de comportamiento de OS de anfitrión incluyen uno o más factores seleccionados de un grupo que consiste en: mensajes relacionados con descriptor de OS de Microsoft, wLength de un primer mensaje de descriptor de obtener configuración, wLength de un segundo mensaje de descriptor de obtener configuración, mensajes relacionados con descriptor de asociación de interfaz, e identidad de una segunda solicitud tras SET\_ADDRESS; y

(a3) determinar el tipo de OS del anfitrión USB basándose en los factores de comportamiento de OS de anfitrión recopilados;

30 (b1) hacer que el controlador (22) de dispositivo USB se desacople del anfitrión USB enviando un comando de apagado al controlador (22) de dispositivo USB; y

(b2) hacer que el controlador (22) de dispositivo USB se acople de nuevo al anfitrión USB, presentándose como dispositivo USB objetivo; y

35 (b3) realizar una enumeración de dispositivo USB con el anfitrión USB usando un procedimiento de enumeración de dispositivo USB seleccionado de una pluralidad de procedimientos de enumeración de dispositivo USB almacenados en el dispositivo USB basándose en el tipo de OS determinado del anfitrión USB.

8. Dispositivo USB según la reivindicación 7, en el que el dispositivo USB soportado de manera habitual es un dispositivo de almacenamiento masivo USB o un dispositivo de interfaz humana USB.

40 9. Dispositivo USB según la reivindicación 7, en el que la memoria almacena una base de datos de conocimiento de comportamiento de OS previamente almacenada que incluye información referente a factores de comportamiento conocidos de una pluralidad de tipos de OS conocidos, y en el que la etapa (a3) se realiza usando la base de datos de conocimiento de comportamiento de OS previamente almacenada.

10. Dispositivo USB según la reivindicación 9, en el que la pluralidad de tipos de OS conocidos incluyen Windows Vista, Win7, Windows XP, Linux y MAC OS; o en el que el procedimiento comprende además:

45 (c) antes de la etapa (a), generar la base de datos de conocimiento de comportamiento de OS, que comprende:

(c1) presentar el dispositivo USB a un anfitrión USB conectado al dispositivo USB como dispositivo USB soportado de manera habitual, teniendo el anfitrión USB un tipo de OS conocido;

(c2) recopilar factores de comportamiento de OS de anfitrión a partir de mensajes recibidos a partir del anfitrión USB;

(c3) repetir las etapas (c1) y (c2) para una pluralidad de anfitriones USB que tienen cada uno un tipo de OS conocido; y

(c4) analizar los factores de comportamiento de OS de anfitrión recopilados en la etapa (c2) para descubrir relaciones entre los factores de comportamiento de OS y los tipos de OS.

5 11. Dispositivo USB según la reivindicación 7, en el que la etapa (a) incluye:

(a1) presentar el dispositivo USB al anfitrión USB como dispositivo USB soportado de manera habitual;

(a2) si se recibe una solicitud de descriptor de dispositivo a partir del anfitrión USB antes de recibir cualquier mensaje que establece una dirección de dispositivo, devolver un descriptor de dispositivo al anfitrión USB que indica un dispositivo específico de proveedor;

10 (a3) si se recibe una solicitud de descriptor de dispositivo a partir del anfitrión USB tras recibir un mensaje que establece una dirección de dispositivo, devolver un descriptor de dispositivo que indica un dispositivo USB soportado de manera habitual;

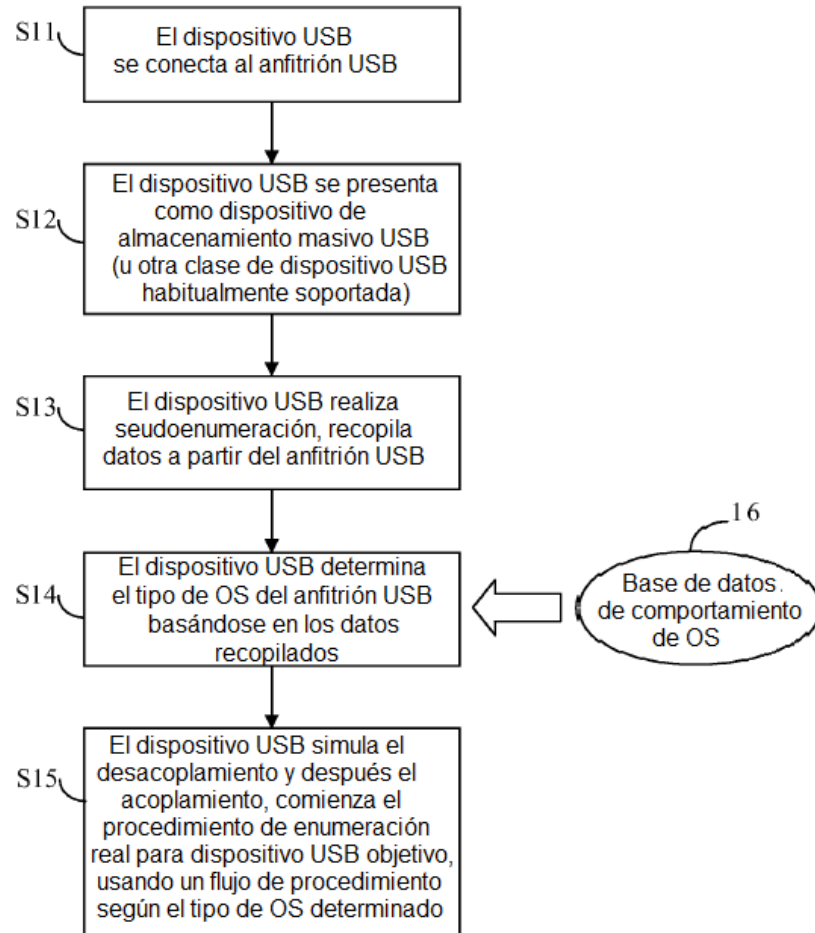
(a4) tras la etapa (a3), recopilar factores de comportamiento de OS de anfitrión a partir de mensajes recibidos a partir del anfitrión USB; y

15 (a5) determinar el tipo de OS del anfitrión USB basándose en los factores de comportamiento de OS de anfitrión recopilados.

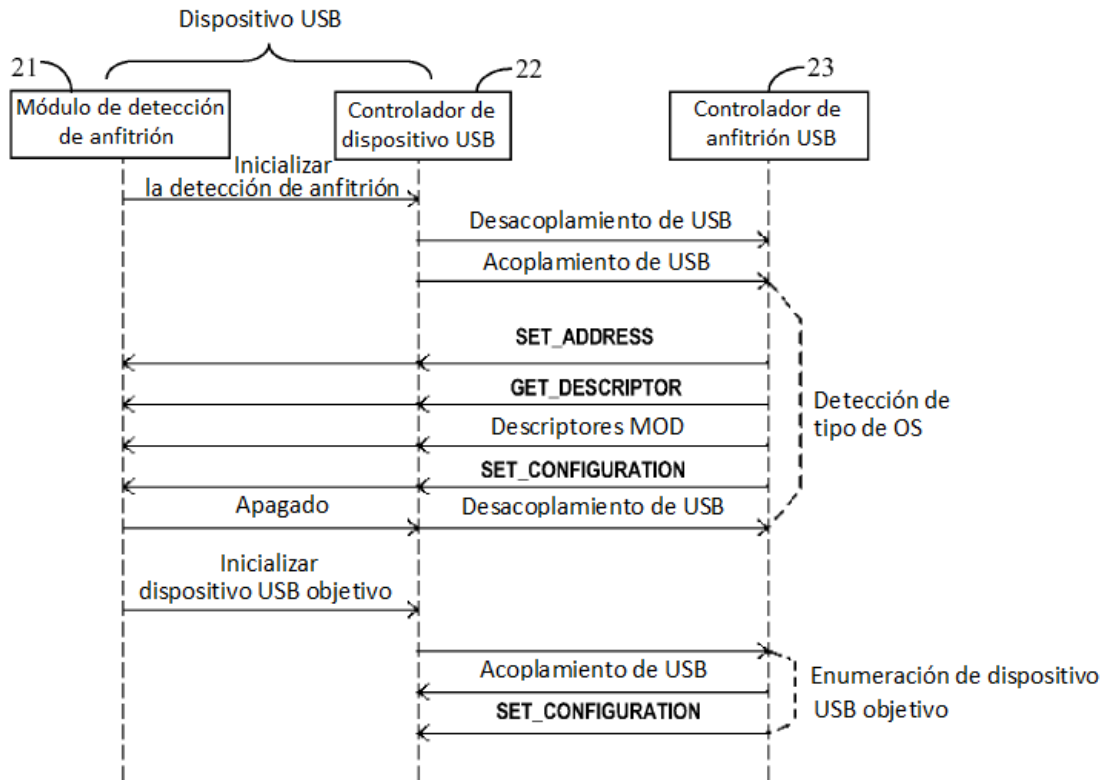
12. Dispositivo USB según la reivindicación 7, en el que el procedimiento comprende además: antes de la etapa (a), almacenar elementos de configuración en el dispositivo USB que definen un flujo de procedimiento de enumeración, en el que la etapa (b) se realiza según el flujo de procedimiento definido por los elementos de configuración; o

20 en el que la etapa (a) determina además si un programa accionador apropiado para un dispositivo USB objetivo está instalado en el anfitrión USB, y en el que la etapa (b) incluye, si el accionador no está instalado, presentar el dispositivo USB como dispositivo de almacenamiento masivo (MSD) que almacena el accionador y un programa de ejecución automática que hace que el accionador se instale en el anfitrión, en el que preferiblemente la etapa (a) incluye determinar si el programa accionador está instalado en el anfitrión USB basándose en un valor de un código de proveedor en una solicitud de MOD (descriptor de OS de Microsoft) a partir del anfitrión USB.

25

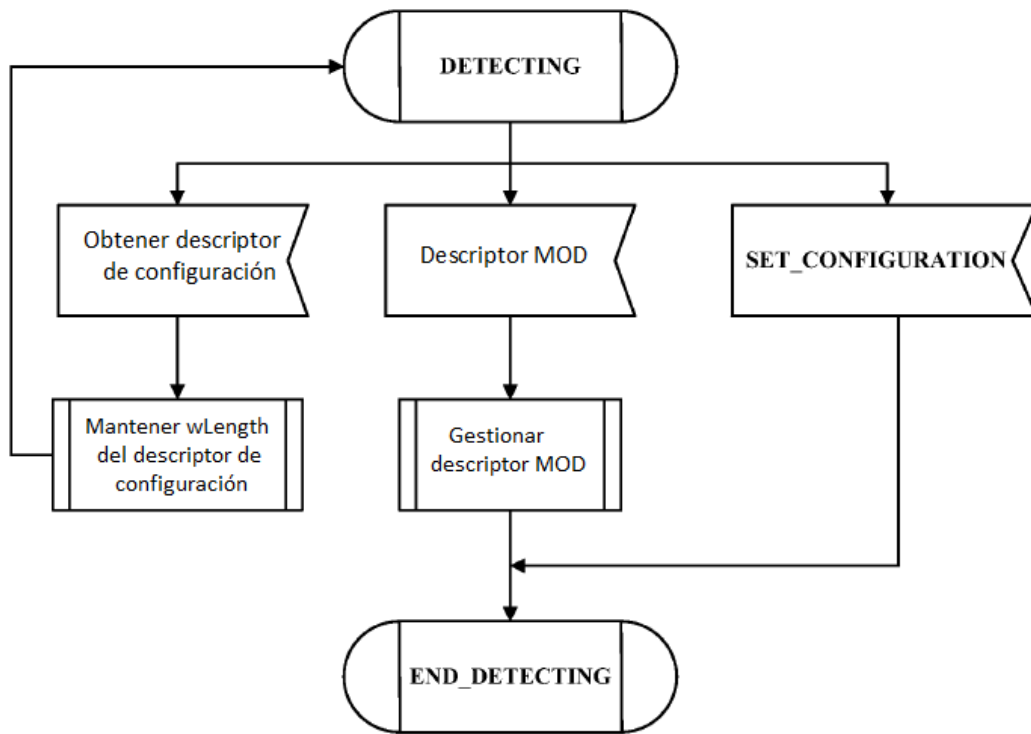


**Fig. 1**

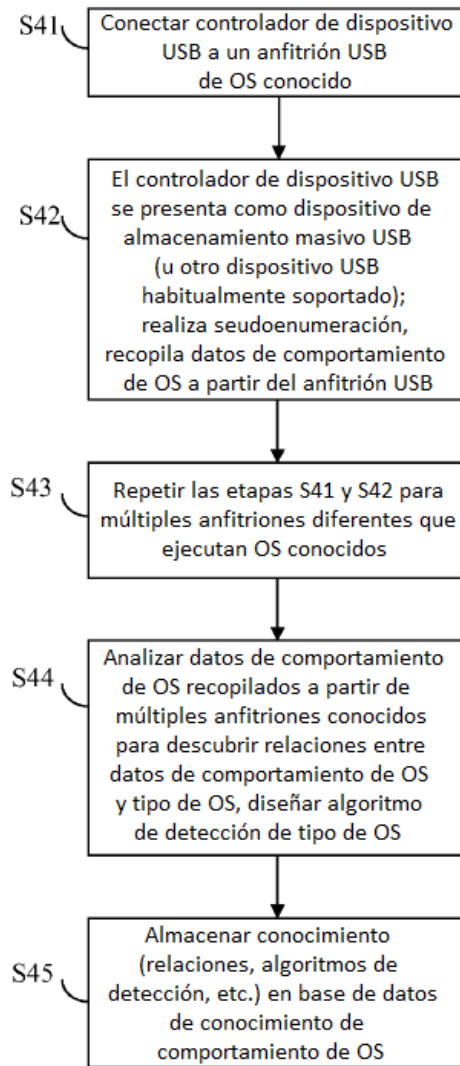


**Fig. 2**

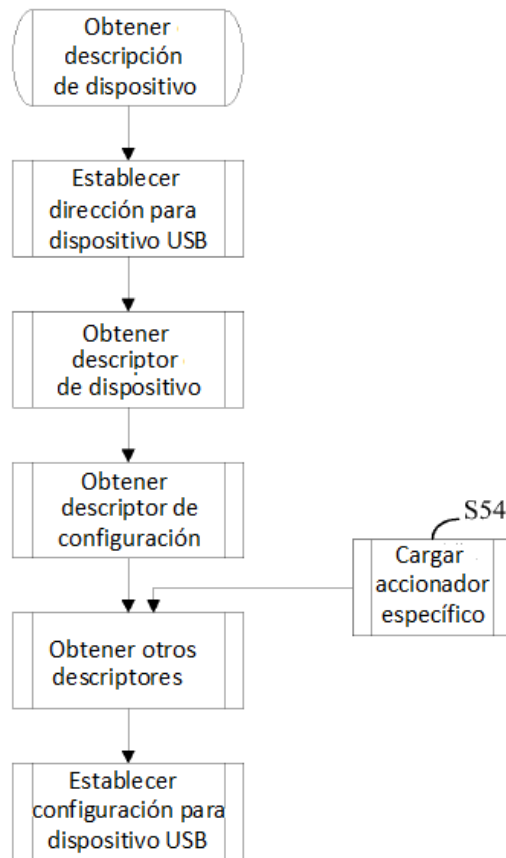




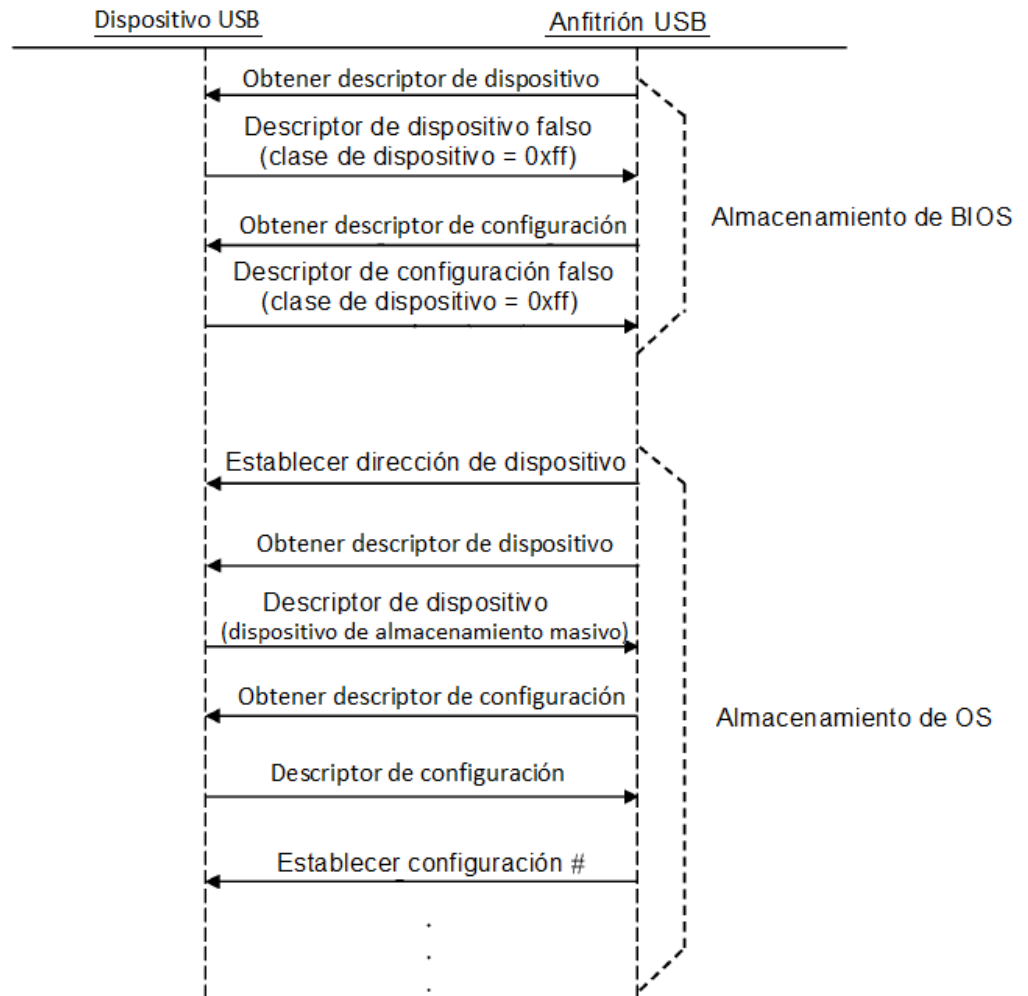
**Fig.3**



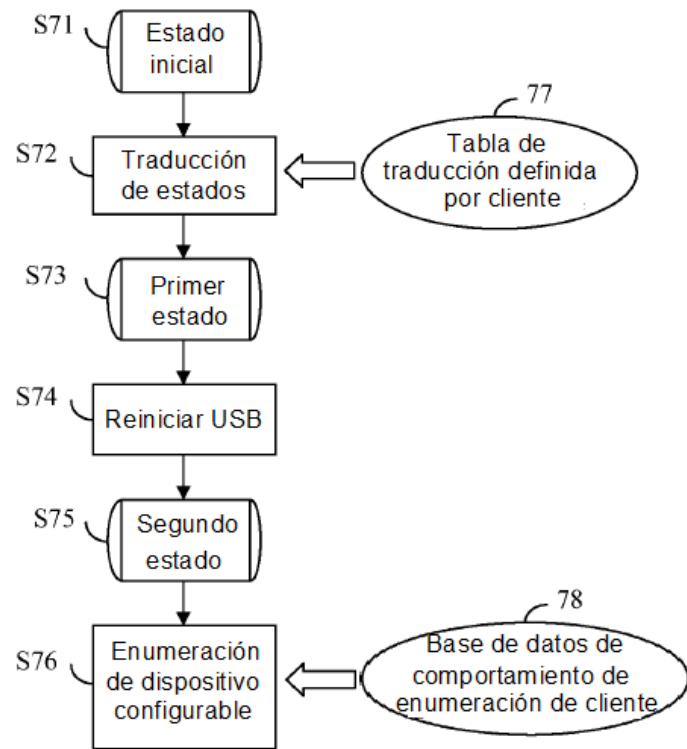
**Fig. 4**



**Fig. 5**



**Fig. 6**



**Fig. 7**

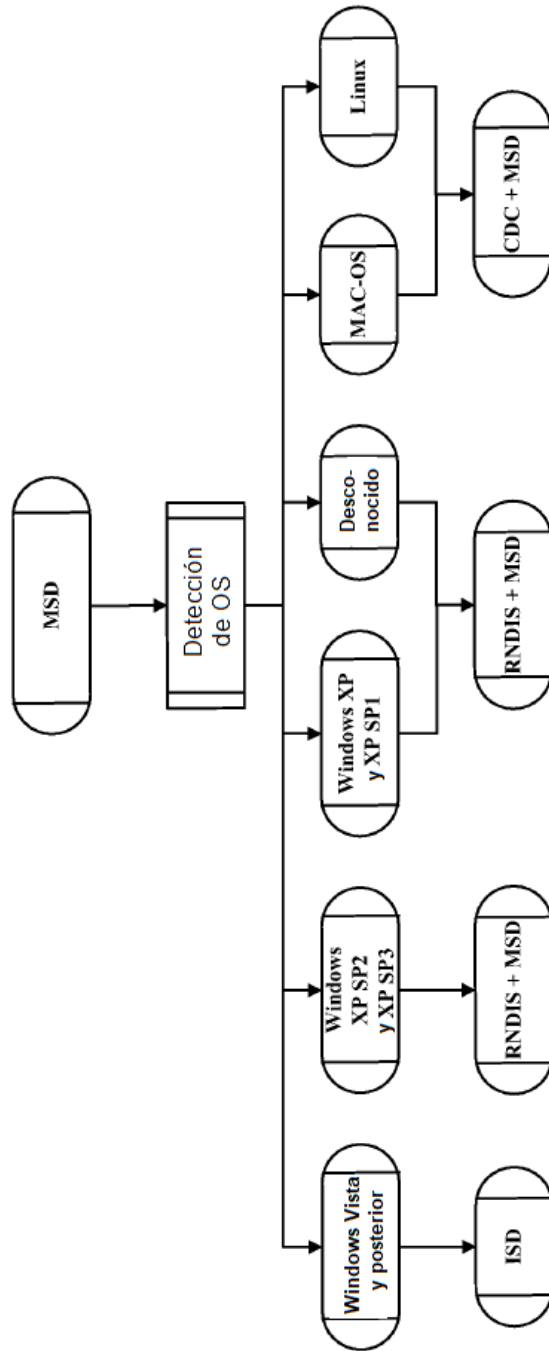
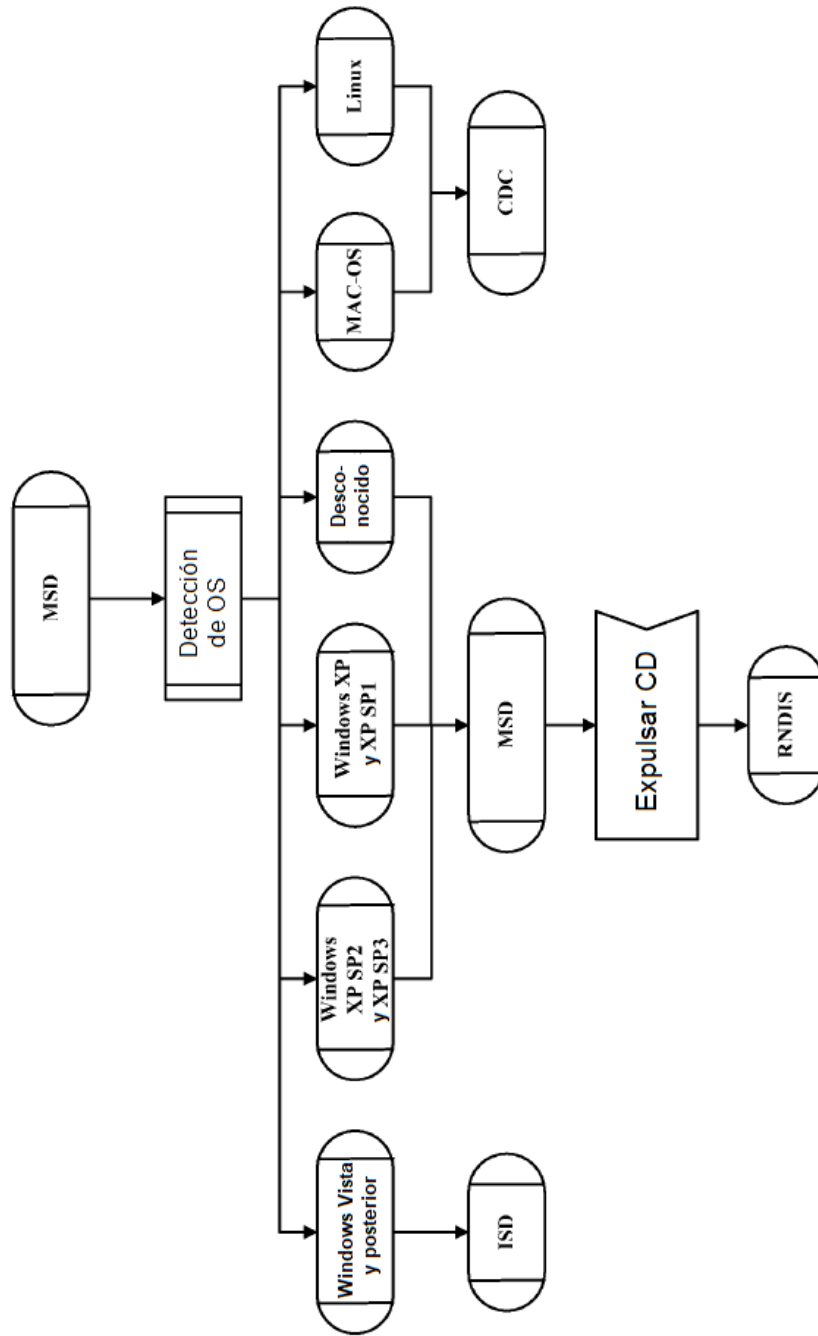
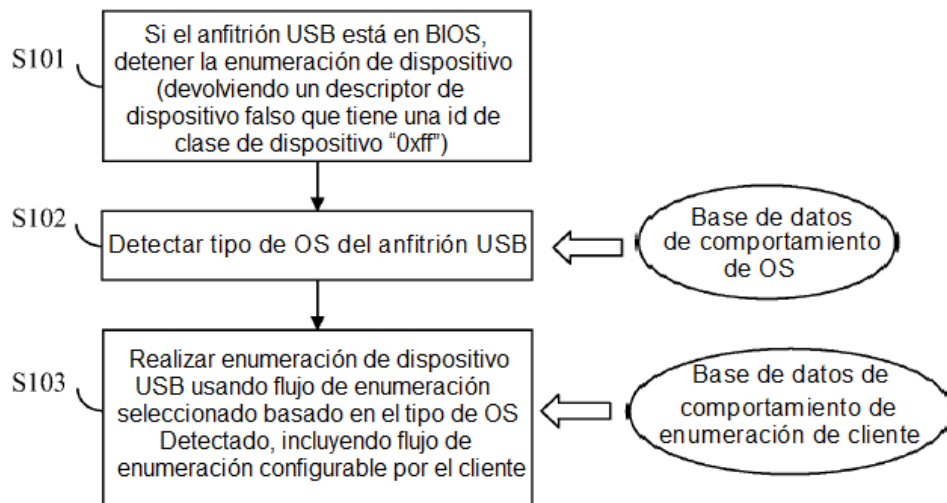


Fig.8



**Fig.9**



**Fig. 10**



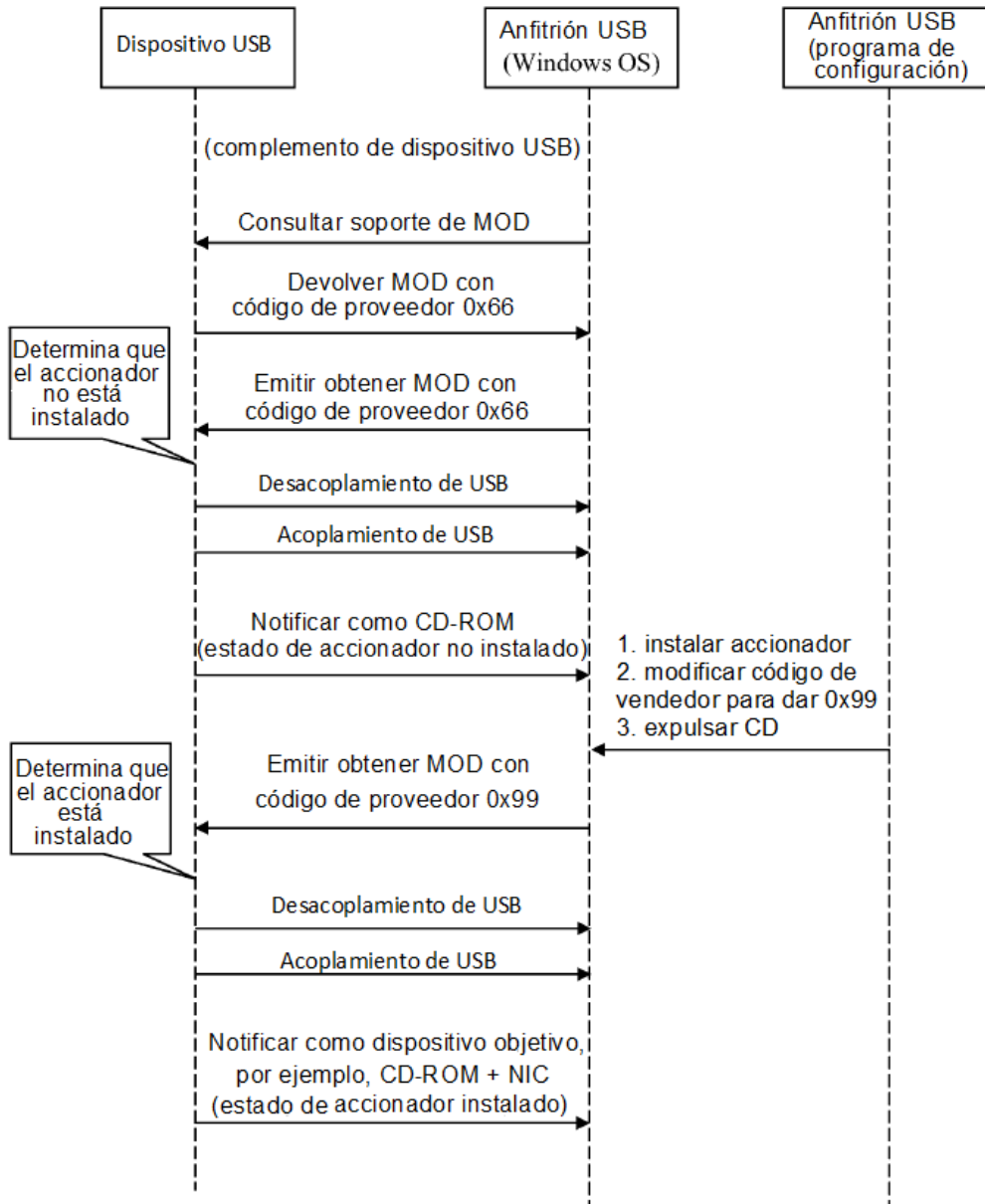


Fig. 11