

DESTRUCCIÓN DE OBJETO BASADA EN FLUJO DE TRABAJO

ANTECEDENTES

5 En el campo de la informática, muchos escenarios implican un dispositivo de almacenamiento de un conjunto de objetos, que generalmente comprenden un conjunto de recursos de objetos como, por ejemplo, funciones, tipos de datos primitivos, referencias a otros objetos, y contenedores de datos tales como matrices, tablas hash, y árboles. Varios hilos ("threads") que ejecutan dentro de uno o más entornos de ejecución (*por ejemplo*, utilizando diferentes lenguajes de programación, dominios de ejecución y/o plataformas) pueden crear y tener acceso a los objetos, incluyendo almacenar las referencias (*por ejemplo*, direcciones de memoria o punteros inteligentes) a los objetos que se encuentran actualmente en uso por los respectivos hilos. Un objeto puede ser simultánea y/o consecutivamente accedido por varios hilos y/u objetos, y se puede utilizar un contador de referencia para rastrear el número de referencias a cada objeto que están en manos de los hilos y almacenados en otros objetos. En algunos casos, el acceso concurrente a través de un recurso compartido particular puede lograrse mediante un mecanismo de sincronización que asegura el acceso exclusivo al recurso de objeto por un hilo en cualquier momento. El mecanismo de sincronización, tal como una exclusión mutua, un semáforo, o una sección crítica, puede organizar solicitudes pendientes; puede conceder acceso exclusivo sobre el recurso de objeto a un hilo seleccionado, negando al mismo tiempo otros hilos que solicitan acceso al objeto; y puede esperar una terminación del acceso por el hilo seleccionado antes de transferir el acceso exclusivo a un segundo hilo con una solicitud de acceso pendiente.

15 Cuando un hilo finaliza su uso de un objeto, libera su referencia (implícita o explícitamente), y el recuento de referencias se disminuye por uno. Cuando el recuento de referencia para un objeto se reduce a cero, se considera que el objeto ya no estaba en uso y listo para destrucción. El dispositivo puede finalizar los recursos de objeto del objeto, y puede reclamar la memoria y los recursos de cómputo que estaban reservados y/o consumidos por el objeto. Alternativamente, un hilo puede solicitar explícitamente una cesión de un objeto, y el dispositivo puede iniciar el proceso de finalización, que puede incluir alterar cualesquiera referencias existentes al objeto que son mantenidas por otros hilos y/u otros objetos.

BREVE DESCRIPCIÓN DE LA INVENCION

30 Esta Breve Descripción se proporciona para introducir una selección de conceptos en una forma simplificada que se describen a continuación en la Descripción Detallada. Esta Breve Descripción no pretende identificar los factores clave o características esenciales de la materia objeto reclamada, ni pretende ser utilizada para limitar el alcance de la materia objeto reclamada.

35 Cuando un hilo intenta destruir un objeto, pueden surgir algunas circunstancias que impiden que el hilo complete el proceso de destrucción de objeto. Como un primer ejemplo, el objeto puede incluir un recurso de objeto que fue creado por un segundo hilo dentro de un entorno de ejecución diferente, en

donde el proceso para destruir el recurso de objeto de acuerdo con los convenios del entorno de ejecución diferente puede no estar totalmente disponible para el inicio del hilo. Como segundo ejemplo, el objeto puede incluir un recurso de objeto que está exclusivamente reservado para su utilización por un segundo hilo, tal como un bloqueo de sincronización, y el primer hilo no podrá interrumpir el acceso exclusivo al segundo hilo sin corromper el segundo hilo o inducir comportamientos imprevistos del dispositivo. Como tercer ejemplo, el objeto puede incluir un recurso de objeto que actualmente está ocupado (*por ejemplo*, un recurso para manejar más de un recurso que puede tomar un tiempo para la liberación) y el bloqueo del hilo hasta que el recurso de objeto pasa a estar disponible para su destrucción puede imponer e indefinir una demora en el hilo.

Cuando surgen dichas circunstancias, un hilo se puede iniciar la destrucción del objeto, pero puede ser incapaz de completar el proceso de destrucción de objeto. El hilo puede simplemente abortar el proceso, pero esto puede dejar el objeto en un estado inconsciente y/o el objeto parcialmente destruido puede persistir indefinidamente como una pérdida de recursos. El hilo puede intentar destruir el objeto en un momento posterior, pero si algunas tareas no se pueden realizar por el hilo independientemente de cualquier tiempo, la prueba repetida del hilo puede representar un drenaje de recurso improductivo y consistente sobre el dispositivo. Además, en algunos casos, la destrucción del objeto puede requerir la participación de varios hilos (*por ejemplo*, diferentes hilos pueden mantener acceso exclusivo sobre diferentes porciones del objeto), de tal forma que ningún hilo puede iniciar y completar el proceso de destrucción de objeto. En estas circunstancias, la incapacidad del hilo para completar el proceso de destrucción de objeto puede ser difícil de resolver en una manera de prueba y error descoordinada, en ausencia de un proceso de destrucción de objeto más coordinado en donde varios hilos pueden participar.

En vista de estas circunstancias, la presente descripción proporciona técnicas que facilitan la destrucción de objeto de objetos por hilos, particularmente (aunque no exclusivamente) en un entorno de recuento de referencia. De acuerdo con estas técnicas, el proceso de destrucción de un objeto puede ser manejado por un flujo de trabajo de destrucción de objeto. Un hilo puede invocar al flujo de trabajo de destrucción de objeto de con una solicitud para destruir el objeto (*por ejemplo*, en respuesta a que el hilo disminuye el recuento de referencia sobre el objeto a cero), y puede invocar al flujo de trabajo de destrucción de objeto para iniciar la destrucción de objeto. El flujo de trabajo de destrucción de objeto puede realizar las tareas del proceso de destrucción de objeto para este objeto en particular (*por ejemplo*, destruyendo los recursos de objeto embebidos en el objeto; reclamando la memoria y otros recursos computacionales; y liberando cualquier resto de referencias al objeto mantenidas por otros hilos y objetos). El flujo de trabajo de destrucción de objeto también puede verificar que el hilo solicitante está autorizado a realizar cada tarea del proceso de destrucción de objeto. Si la destrucción de objeto se completa, el flujo de trabajo de destrucción de objeto puede notificar al hilo de inicio que la solicitud fue satisfecha completamente. Sin embargo, si el proceso de destrucción de objeto falla (*por ejemplo*, porque el hilo de inicio está restringido a la realización de al menos una de las tareas del proceso de destrucción de objeto), el flujo de trabajo de destrucción de objeto puede crear una nueva referencia al objeto y

agregar la nueva referencia a una lista de destrucción de objeto. Ya que la nueva referencia representa una última referencia al objeto, el flujo de trabajo de destrucción puede incrementar el contador de referencia del objeto de cero a uno, lo que indica que el objeto está actualmente retenido de acuerdo con la semántica de recuento de referencia. El proceso de destrucción de objeto puede entonces notificar al primer hilo de la falla para completar la destrucción de objeto del objeto.

En un momento posterior, el mismo hilo o un hilo diferente que ha utilizado también el objeto, puede descubrir la existencia de la última referencia al objeto en la lista de destrucción de objeto. La última referencia puede ser liberada para reducir el recuento de referencia sobre el objeto de uno a cero, incitando al flujo de trabajo de destrucción de objeto a reanudar el proceso de destrucción de objeto a favor del hilo que solicita. La segunda solicitud (en un momento posterior y/o mediante un hilo diferente) puede permitir que el flujo de trabajo de destrucción de objeto complete el proceso de destrucción de objeto, liberando así plenamente los recursos de objeto del objeto, e incitando al flujo de trabajo de destrucción de objeto a notificar al hilo solicitante de la terminación de la destrucción de objeto. De esta manera, el flujo de trabajo de destrucción de objeto puede manejar varias complejidades del proceso de destrucción del objeto, y puede encapsular un proceso de destrucción de objeto basado en el flujo de trabajo que permite que los hilos se coordinen e interactúen para lograr la completa destrucción del objeto, de acuerdo con las técnicas presentadas en este documento.

Para el cumplimiento de lo que antecede y fines relacionados, la siguiente descripción y dibujos adjuntos establecen ciertos aspectos ilustrativos e implementaciones. Pero éstas son indicativas de algunas de las diversas maneras en las cuales uno o más aspectos pueden ser empleados. Otros aspectos, ventajas y características novedosas de la descripción se harán evidentes a partir de la siguiente descripción detallada considerada junto con los dibujos anexos.

DESCRIPCIÓN DE LOS DIBUJOS

La Figura 1 es una ilustración de un escenario ilustrativo modalizando un conjunto de hilos que comparten el acceso a, e intentan destruir, un conjunto de objetos.

La Figura 2 es una ilustración de un escenario ilustrativo modalizando un conjunto de hilos que destruyen objetos mediante un flujo de trabajo de destrucción de objeto de acuerdo con las técnicas presentadas en este documento.

La Figura 3 es una ilustración de un flujo de trabajo de destrucción de objeto ilustrativo que facilita la destrucción de un objeto a favor de un conjunto de hilos de acuerdo con las técnicas presentadas en este documento.

La Figura 4 es un escenario ilustrativo modalizando un dispositivo ilustrativo utilizando un sistema ilustrativo que destruye un objeto utilizando un flujo de trabajo de destrucción de objeto de acuerdo con las técnicas presentadas en este documento.

La Figura 5 es un método ilustrativo para destruir un objeto a favor de un hilo usando un flujo de trabajo de destrucción de objeto de acuerdo con las técnicas presentadas en este documento.

La Figura 6 es una ilustración de un ejemplo de un dispositivo de almacenamiento legible por computadora que almacena instrucciones que hacen que un dispositivo realice de acuerdo con las técnicas que se presentan en este documento.

La Figura 7 es una ilustración de un escenario ilustrativo modalizando una protección de un objeto que varios hilos intentan destruir a través de un flujo de trabajo de destrucción de objeto de acuerdo con las técnicas que se presentan en este documento.

La Figura 8 es una ilustración de un escenario ilustrativo modalizando un mecanismo de destrucción de una gráfica de objeto por un flujo de trabajo de destrucción de objeto de acuerdo con las técnicas presentadas en este documento.

La Figura 9 es una ilustración de un escenario ilustrativo modalizando el marcado de un objeto como un estado de zombi, durante una destrucción del objeto por un flujo de trabajo de destrucción de objeto, de acuerdo con las técnicas presentadas en este documento.

La Figura 10 es una ilustración de un escenario ilustrativo modalizando una retención de un objeto por un flujo de trabajo de destrucción de objeto de acuerdo con las técnicas presentadas en este documento.

La Figura 11 es una ilustración de un escenario ilustrativo modalizando varios mecanismos mediante los cuales un segundo hilo reanuda el proceso de destrucción de objeto previamente iniciado por un primer hilo de acuerdo con las técnicas presentadas en este documento.

La Figura 12 es una ilustración de un entorno de cómputo ilustrativo en donde una o más de las provisiones establecidas en el presente documento pueden ser implementadas.

DESCRIPCIÓN DETALLADA

Ahora se describirá la materia objeto con referencia a los dibujos, en donde números similares se utilizan para referirse a elementos similares en todo. En la siguiente descripción, para fines de explicación, se establecen numerosos detalles específicos con el fin de proporcionar una comprensión profunda de la materia objeto reclamada. Puede ser evidente, sin embargo, que la materia objeto reclamada pueda ser practicada sin estos detalles específicos. En otros casos, las estructuras y los dispositivos se muestran en forma de diagrama de bloques con el fin de facilitar la descripción de la materia objeto reclamada.

A. Introducción

La Figura 1 es una ilustración de un escenario ilustrativo 100 en donde un conjunto de objetos 102 es accedido por un conjunto de hilos 106 dentro de un entorno de cómputo, en donde los hilos 106 intentan destruir los objetos 102 después de uso.

En este escenario ilustrativo 100, un primer hilo 106 ejecuta dentro de un primer entorno de ejecución 108 (*por ejemplo*, una aplicación C compilado y ejecutado de forma nativa por el dispositivo) y un segundo hilo 106 ejecuta en un segundo, entorno de ejecución diferente 108 (*por ejemplo*, una

aplicación de .NET ejecutada por un motor de ejecución de .NET). Los respectivos hilos 106 pueden crear y acceder a varios objetos 102 que, respectivamente, almacenan un conjunto de recursos de objeto 104, tales como los tipos de datos primitivos; referencias a otros objetos 102; y/o estructuras de datos, tales como funciones, listas, tablas hash, y árboles, que contienen otros datos y/o referencias a otros objetos 102. Los hilos 106 pueden participar en la creación y la actualización de los objetos 102, tales como solicitudes de creación de recurso 110 para crear un recurso de objeto dentro de un objeto 102.

Como se muestra en el escenario ilustrativo 100 de la Figura 1, los hilos 106 también pueden compartir el acceso a objetos respectivos 102. Por ejemplo, dentro de un primer objeto 102, se puede crear un primer recurso de objeto 104 por el primer hilo 106, y puede ser accedido y/o actualizado por el segundo hilo 106. En algunos casos, la compartición puede ocurrir sin restricción, tal como un recurso de objeto 104 que es libre y concurrentemente accesible a cualquier número de hilos 106. Sin embargo, en otras circunstancias, la compartición puede involucrar un mecanismo de sincronización, tal como un bloqueo de recurso de objeto 112 que otorga acceso exclusivo sobre el recurso de objeto 104 a un hilo particular 106. El bloqueo de recurso de objeto 112 puede denegar el acceso al recurso de objeto 104 por cualesquiera otros hilos 106 durante el tiempo que el hilo autorizado exclusivamente 106 retiene el bloqueo de recurso de objeto 112 y puede registrar tales solicitudes de acceso incumplidas por otros hilos 106, opcionalmente bloqueando la ejecución de tales otros hilos 106 hasta que dicho acceso pueda ser concedido. Cuando dicho hilo 106 completa el acceso exclusivo y libera el bloqueo de recurso de objeto 112, un componente del entorno de cómputo (*por ejemplo*, un manejador de bloqueo de sincronización) puede revisar una lista de solicitudes pendientes para tener acceso al recurso de objeto 104, puede elegir una de las solicitudes pendientes (*por ejemplo*, según un orden de prioridad), y puede notificar el hilo que inició la solicitud seleccionada de la concesión del bloqueo de recurso de objeto 112 (opcionalmente desbloqueando el hilo 106 durante la tramitación de la solicitud).

En tales escenarios, los hilos 106 pueden retener los objetos respectivos 102 durante el ciclo de vida útil de los objetos. Al finalizar el ciclo de vida de un objeto particular 102, un hilo 106 puede tratar de destruir el objeto 102 a fin de recuperar los recursos computacionales utilizados por él, tales como la memoria del sistema, el espacio de almacenamiento en una memoria caché o memoria secundaria, y el uso del procesador. El proceso de destrucción de objeto puede activarse explícitamente (*por ejemplo*, un hilo 106 puede invocar una operación Disponer en un objeto 102) o implícitamente (*por ejemplo*, un hilo 106 puede almacenar sólo la referencia a un objeto 102, y la porción de código que utiliza la referencia puede ir fuera de alcance, lo que se traduce en un objeto 102 sin referencias restantes). El proceso de destrucción de objeto invocado por el hilo 106 puede implicar la destrucción de cada uno de los recursos de objeto 104 que comprenden el objeto 102, y luego destruir un contenedor de objeto del objeto 102.

Además, algunas circunstancias pueden permitir que un objeto 102 que fue creado por un primer hilo 106, y que se comparte entre el primer hilo 106 y un segundo hilo 106, sea destruido ya sea por el primer hilo 106 o el segundo hilo 106. Por ejemplo, en el caso del escenario ilustrativo 100 de la Figura 1, el primer hilo 106 puede crear un primer objeto 102 y un primer recurso de objeto 104 contenido en él, y el segundo hilo 106 puede iniciar una solicitud de destrucción de objeto 114 para destruir el objeto 102.

La solicitud puede ser cumplida al destruir el recurso de objeto 104 y luego al destruir el objeto 102 y el completamiento exitoso de la destrucción de objeto puede ser indicado al segundo hilo 106 que invocó la solicitud de destrucción de objeto 114 a través de un valor de retorno que indica un éxito 206. La destrucción del objeto 102 por cualquier hilo 106 puede afectar, por lo tanto, el otro hilo 106 que también
 5 accede al objeto 102 en una manera compartida. En tales circunstancias, el entorno de cómputo podrá ya sea denegar la solicitud del primer hilo 106 para destruir el objeto 102, o puede hacer cumplir la solicitud restableciendo las referencias al objeto 102 mantenidas por el segundo hilo 106 a un valor predeterminado o valor nulo, permitiendo así que el segundo hilo 106 evite una excepción de referencia de memoria no válida. De esta manera, los hilos 106 pueden interoperar para compartir el acceso a los
 10 objetos 102 y los recursos de objeto 104 contenidos en él, y un hilo 106 podrá iniciar y completar un proceso de destrucción de objeto para los objetos respectivos 102 a la finalización del ciclo de vida de un objeto.

Sin embargo, en tales circunstancias, un hilo particular 106 que inicia una solicitud de destrucción de objeto 114 puede ser incapaz de completar el proceso de destrucción de objeto, por una variedad de
 15 razones.

Como un primer ejemplo, un objeto 102 puede incluir recursos de objeto 104 que fueron creados, respectivamente, dentro de los diferentes entornos de ejecución 108, cada uno de los cuales puede utilizar una convención particular para crear, así como para destruir recursos de objeto 104 de objetos respectivos. En tales circunstancias, un hilo 106 que ejecuta dentro de un entorno de ejecución particular
 20 108 puede desconocer y/o es incapaz invocar el proceso de destrucción de recurso de objeto que se utiliza por un entorno de ejecución diferente 108 que creó el recurso de objeto 104. Por ejemplo, en el caso del escenario ilustrativo 100 de la Figura 1, el segundo objeto 102 incluye un tercer recurso de objeto 104 que fue creado por una solicitud de creación de recurso 110 del primer hilo 106 ejecutando dentro del primer entorno de ejecución 108, y también un cuarto recurso de objeto 104 que fue creado
 25 por una solicitud de creación de recurso 110 del segundo hilo 106 que ejecuta dentro del segundo entorno de ejecución 108. El primer hilo 106 (*por ejemplo*, un proceso nativo C) puede ser incapaz de iniciar la destrucción del cuarto recurso de objeto 104 (que fue creado por el motor de ejecución de .NET); similarmente, el segundo hilo 106 (*por ejemplo*, un proceso manejado en C#) puede ser incapaz de iniciar la destrucción del tercer recurso de objeto 104 (que fue creado por el C proceso nativo). En
 30 consecuencia, ningún hilo 106 puede ser capaz de iniciar y completar la destrucción del segundo objeto 102, y cualquier solicitud de destrucción de objeto 114 iniciada por cualquier hilo 106 puede resultar en una falla 118.

Como un segundo ejemplo, un recurso de objeto 104 de un objeto 102 puede ser utilizado exclusivamente por un segundo hilo 106 que mantiene un bloqueo de recurso de objeto 112, y un primer
 35 hilo 106 que inicia una solicitud para destruir el objeto 102 puede no estar permitido para destruir el recurso de objeto 104 y, por lo tanto, el objeto 102, hasta el segundo hilo 106 completa su acceso exclusivo y abandona el bloqueo de recurso de objeto 112. Además, como se muestra en el escenario ilustrativo 100 de la Figura 1, el tercer objeto 102 contiene un quinto recurso objeto 104 para el cual el

segundo hilo 106 mantiene un primer bloqueo de recurso de objeto 112, y también un sexto recurso de objeto 104 para el cual el primer hilo 106 mantiene un segundo bloqueo de recurso de objeto 112. En este caso, ningún hilo 106 es capaz, por sí mismo, tanto de iniciar como de completar una solicitud de destrucción de objeto 114, y cualquier solicitud de este tipo puede resultar en una falla 118.

5 En aún otras circunstancias puede surgir que obstruir la realización de una destrucción de objeto de un objeto 102 por un hilo particular 106 como un tercer ejemplo, un recurso de objeto 104 puede tener criterios de pre-requisito particulares que serán cumplidos antes de que el recurso de objeto 104 sea destruido, tal como destruyendo primero un segundo recurso de objeto 104 del mismo objeto 102 o un objeto diferente 102; simultáneamente destruyendo un segundo recurso de objeto 104 del objeto 102; y/o
10 sólo permitiendo que un hilo particular 106 (*por ejemplo*, un hilo 106 que sirve como un manejador del recurso de objeto 104) invoque la destrucción del recurso de objeto 104. Como un cuarto ejemplo, el objeto 102 y/o recurso de objeto 104 pueden estar involucrados en una gráfica de objeto cíclico, que comprende un conjunto de objetos 104 que hagan referencia a uno y otro en una manera cíclica, y en donde cada objeto 102 puede ser destruido sólo después de que los otros objetos 102 sean destruidos.
15 En ausencia de un reconocimiento de esta circunstancia, la gráfica de objeto cíclico puede llevar a un callejón sin salida que se opone a la destrucción de cualquiera de los objetos 102. Como un quinto ejemplo, un recurso de objeto 104 puede estar simplemente ocupado; *por ejemplo*, un recurso de objeto 104 puede comprender un puntero inteligente a través de un dispositivo externo que está realizando un proceso ininterrumpido, en donde la destrucción del recurso de objeto 104 antes de la finalización del
20 proceso ininterrumpido puede dañar el dispositivo externo. Si bien puede ser viable para el hilo 106 iniciar la solicitud de destrucción de objeto 114 para tal objeto 102 para bloquear hasta que el recurso de objeto 104 quede disponible para la destrucción, este bloqueo puede provocar que el hilo 106 deje de responder, posiblemente durante un tiempo prolongado y posiblemente de duración indeterminada.

En tales circunstancias, la incapacidad de un hilo 106 para completar una solicitud de destrucción
25 de objeto 114 sobre un objeto 102 puede ser imprevista, de tal forma que un error o excepción no controlada provoca que el hilo 106 deje de responder, accidente, o involucrarse en comportamientos imprevistos. En algunos casos, cuando un hilo 106 encuentra un obstáculo mientras se procesa una solicitud de destrucción de objeto 114, el hilo 106 puede abandonar la tarea de destrucción. No obstante, si el entorno de cómputo carece de cualquier mecanismo para reanudar y potencialmente completar la
30 destrucción de objeto del objeto 102, el objeto 102 puede persistir indefinidamente como una pérdida de recurso que consume recursos de cómputo tales como memoria y ciclos de procesador sin efecto productivo. Además, cuando un hilo 106 destruye parcialmente un objeto 102 y luego aborta la destrucción debido a un obstáculo, el objeto 102 puede quedar en un estado indeterminado, inconsistente, en donde algunos recursos consumidos por el objeto 102 pueden no satisfacer un modelo
35 implícito para el uso del recurso. Por ejemplo, el entorno de cómputo puede rastrear el uso de la red de acuerdo con los hilos 106 que hacen referencia a los objetos 102 en donde los punteros inteligentes de la red se almacenan y se utilizan. Sin embargo, el entorno de cómputo no puede anticipar la circunstancia de que un objeto 102 utiliza la red, pero no se hace referencia por cualquier hilo 106, así como un

resultado inesperado puede crear comportamientos imprevistos en otros hilos 106 del entorno de cómputo, tal como un consumo inidentificable y potencialmente interminable de la capacidad de la red. En otros casos, un hilo 106 puede estar preparado para manejar una incapacidad para completar la destrucción de un objeto 102, pero puede ser incapaz de determinar por qué la destrucción de objeto del

5 objeto 102 ha fallado, ni ningún mecanismo adecuado para continuar el proceso de destrucción de objeto. Por ejemplo, el hilo 106 puede reconocer que algún otro hilo 106 es responsable de la destrucción de un recurso de objeto 104 en particular de un objeto 102, pero puede ser incapaz de identificar y/o notificar cualquier otro hilo 106 de la solicitud de destrucción de objeto 114. Como otra de las posibles desventajas

10 potenciales, en algunas circunstancias, los criterios de requisito previo de destruir los recursos de objeto 104 de un objeto 102 pueden ser complejos y sutiles, de tal forma que la configuración de cualquier hilo particular 106 para manejar las complejidades del proceso de destrucción de objeto para cualquier objeto particular 102 puede delegar una cantidad indeseable de complejidad para un desarrollador del hilo 106. Además, los criterios que se refieren a la destrucción de un recurso de objeto 104 pueden evolucionar con el transcurso del tiempo (*por ejemplo*, un nuevo criterio puede definirse según mecanismos

15 ambientales recientemente implementados del entorno de cómputo), imponiendo una carga de desarrollo significativo y potencialmente prohibitivo sobre el desarrollador de un hilo particular 106. Estas y otras complicaciones pueden surgir en el contexto de una solicitud de destrucción de objeto 114 de un hilo 106 sobre los recursos de objeto 104 de un objeto 102.

20 B. Técnicas Presentadas

En vista de las complejidades de la destrucción de objeto de objetos 102, incluyendo aquellas que se ilustran en el escenario ilustrativo 100 de la Figura 1, la presente descripción proporciona técnicas para facilitar los hilos 106 en la destrucción de objetos 102 y en escenarios de direcciones particulares en donde un primer hilo 106 inicia una solicitud de destrucción de objeto 114 pero es incapaz de completar la

25 destrucción de objeto del objeto 102. Estas técnicas involucran la representación del proceso de destrucción de objeto como un flujo de trabajo de destrucción de objeto, que encapsula y realiza las tareas de destrucción de objeto de un objeto particular 102 a favor de varios hilos 102.

La Figura 2 es una ilustración de un escenario ilustrativo 200 que presenta un flujo de trabajo de destrucción de objeto 202 que destruye un objeto 102 de acuerdo con las técnicas que se presentan en este documento. En este escenario ilustrativo 200, en un primer tiempo 216, un primer hilo 106 inicia una

30 solicitud de destrucción de objeto 114 para destruir un objeto 102 que comprende dos recursos de objeto 104. En este ejemplo particular, la destrucción de objeto se inicia cuando el primer hilo 106 libera una referencia al objeto 102 que pasa a ser la última referencia 206 al objeto 102. La liberación de la referencia 206 produce un recuento de referencia 204 de las referencias 206 al objeto 102 para disminuir de uno a cero, lo que indica que ya ningún hilo 106 en el entorno de cómputo ha retenido una referencia

35 206, el objeto 102 ha alcanzado el final del ciclo de vida del objeto y está listo para ser destruido. En consecuencia, el primer hilo 106 inicia la destrucción del objeto 102 en respuesta a la reducción del recuento de referencia 204 a cero. Sin embargo, más que intentar realizar la destrucción de objeto como

tal, el hilo 106 presenta la solicitud de destrucción de objeto 114 a un flujo de trabajo de destrucción de objeto 202, lo que inicia la destrucción del objeto 102 a favor del hilo 106. El flujo de trabajo de destrucción de objeto 202 comienza iniciando una destrucción de recurso de objeto 208 para el primer recurso de objeto 104, que produce un éxito 116.

5 Como se ilustra en el escenario ilustrativo 200 de la Figura 2, en un segundo tiempo 218, el flujo de trabajo de destrucción de objeto 202 puede continuar la destrucción del objeto 102 a favor del primer hilo 106 iniciando una destrucción de recurso de objeto 208 del segundo recurso de objeto 104, pero se puede encontrar una falla 118. Por ejemplo, el segundo recurso de objeto 104 puede ser exclusivamente accedido mediante un hilo diferente 106 que mantiene un bloqueo de recurso de objeto 112; puede haber
10 sido creado por un hilo diferente 106 dentro de un entorno de ejecución diferente 108 que utiliza procesos de destrucción de recurso de objeto diferentes y potencialmente desconocidos; y/o simplemente puede estar ocupado, tal como esperando procesos ininterrumpidos por otros recursos o dispositivos. En respuesta a encontrar la falla 118, el flujo de trabajo de destrucción de objeto 202 puede iniciar una creación de referencia 210 de una nueva referencia 206 al objeto 102, y agregar la nueva referencia 206
15 a una lista de destrucción de objeto 212. La lista de destrucción de objeto 212 comprende de un almacenamiento de últimas referencias 206 a objetos 102 que están en el proceso de ser destruidos (y, más concretamente, objetos 102 para los cuales un proceso de destrucción de objeto fue iniciado previamente por el flujo de trabajo de destrucción de objeto 202 a favor de al menos un hilo 106, pero en donde un obstáculo impide que el flujo de trabajo de destrucción de objeto 202 complete la solicitud de
20 destrucción de objeto 114). La nueva referencia 204, que ahora comprende la última referencia 206 al objeto 102, es insertada por el flujo de trabajo de destrucción de objeto 202 en la lista de destrucción de objeto 212, y el flujo de trabajo de destrucción de objeto incrementa el recuento de referencia 204 del objeto 102 de cero a uno. El flujo de trabajo de destrucción de objeto 202 también puede reportar la falla 118 al primer hilo 106, que podrá reanudar la ejecución como si el objeto 102 hubiera sido
25 completamente destruido.

 Como se ilustra en el escenario ilustrativo 200 de la Figura 2, en un tercer tiempo 220, un segundo hilo 106 puede descubrir y/o ser notificado de la última referencia 206 al objeto 102 en la lista de destrucción de objeto 212. El segundo hilo 106 pudo haber accedido previamente al objeto 102 (o incluso pudo haber creado el objeto 102) y, por lo tanto, puede iniciar una liberación de referencia 214 de la
30 referencia 206. La liberación de referencia 214 provoca que el recuento de referencia 202 se disminuya de uno a cero, lo que reinicia la solicitud de destrucción de objeto 114 por el flujo de trabajo de destrucción de objeto 202, pero en este tercer tiempo 220, a favor del segundo hilo 106 en lugar del primer hilo 106. El flujo de trabajo de destrucción de objeto 202 puede iniciar de nuevo una destrucción de recurso de objeto 208 del segundo recurso de objeto 104 del objeto 102, y este intento puede producir
35 un éxito 116 (*por ejemplo*, ya que el flujo de trabajo de destrucción de objeto 202 puede destruir el recurso de objeto 104 a favor del segundo hilo 106, pero no el primer hilo 106). Habiendo completado la destrucción de objeto del objeto 102, el flujo de trabajo de destrucción de objeto 202 puede informar un éxito 106 al segundo hilo 106. De esta manera, el flujo de trabajo de destrucción de objeto 202 logra una

destrucción de objeto del objeto 102 a través de un proceso iterativo involucrando la cooperación de ambos hilos 106 en conformidad con las técnicas presentadas en este documento.

C. Efectos Técnicos

5 Los varios usos de las técnicas presentadas en este documento pueden resultar en una variedad de efectos técnicos.

Un primer efecto técnico que se puede obtener de acuerdo con las técnicas aquí presentadas implica la destrucción exitosa de un objeto 102 de acuerdo con un flujo de trabajo de destrucción de objeto 202. El flujo de trabajo de destrucción de objeto 202 proporciona un proceso iterativo de
 10 destrucción de objeto que puede implicar la cooperación de múltiples hilos 106, y/o un hilo 106 en dos o más veces sucesivas, como un proceso incremental para destruir los recursos de objeto 104 del objeto 102. Tal proceso iterativo puede permitir una travesía de una amplia gama de posibles obstáculos potenciales a la iniciación y terminación de una destrucción de objeto por un solo hilo 106 en un solo paso, y puede reducir la incidencia de excepciones, bloqueo de hilo, eliminación de objeto interrumpida
 15 que da lugar a un objeto parcialmente eliminado 102 en un estado inconsistente, fugas de recurso causadas por objetos 102 eliminados incompletamente, comportamiento inesperado de los hilos 106 y/o el entorno de cómputo debido a objetos 102 inconsistentes y posiblemente fugados, la interrupción de los procesos ininterrumpidos, y escenarios de atascamiento en donde ningún hilo 106 individual es capaz de destruir el objeto 102 sin la cooperación de otros hilos 106.

20 Un segundo efecto técnico que se puede obtener de acuerdo con las técnicas presentadas en este documento implica la utilización, por parte del flujo de trabajo de destrucción de objeto 202, de recursos preexistentes del entorno de cómputo en el proceso de destrucción de objeto, incluyendo técnicas de recuento de referencias que ya están disponibles dentro del entorno de cómputo, y la semántica automatizada de iniciar la destrucción de objeto después de que el recuento de referencia 204
 25 de un objeto 102 se ha reducido a cero. Además, el uso de la lista de destrucción de objeto 212, como un almacenamiento de referencias 206 a objetos 102 que están listos para ser destruidos (y para el cual un intento anterior por el flujo de trabajo de destrucción de objeto 202 para destruir el objeto 102 a favor de uno o más hilos 106 resultó en una falla 118), sirve como un mecanismo de coordinación entre los hilos 106; *por ejemplo*, la presencia de una referencia 206 en la lista de destrucción de objeto 212 sirve como
 30 una solicitud tácita de los hilos 106 del entorno de cómputo para intentar destruir el objeto 102, y los intentos por cada hilo 106 puede llegar a producir un éxito 116 que completa o, al menos, avanza, la destrucción del objeto 102. En consecuencia, los hilos que ya están configuradas para utilizar las referencias 206 pueden ser capaces de participar en el proceso de destrucción de objeto incremental mediado por el flujo de trabajo de destrucción de objeto 202, sin necesidad de reconfiguración o cualquier
 35 toma de conciencia por el desarrollador de la semántica y de la mecánica de esas técnicas.

Un tercer efecto técnico que es factible de acuerdo con las técnicas presentadas en este documento involucra una reducción de la complejidad del desarrollo en favor de los hilos 106 en la manipulación de la destrucción de objeto. La delegación de los detalles y la mecánica de destrucción de

objeto de los hilos 106 al flujo de trabajo de destrucción de objeto 202 permite a los hilos 106 hacer caso omiso de los progresos y las complejidades del proceso de destrucción de objeto; *por ejemplo*, los hilos 106 no necesitan ser configurados para manejar las fallas 118, los diversos motivos para las mismas y/o la comunicación de las solicitudes a otros hilos 106 para hacer un nuevo intento de destruir el objeto 102.

- 5 El flujo de trabajo de destrucción de objeto 202 incluso puede automatizar el proceso de reiniciar la destrucción del objeto 102 por el mismo hilo 106 en un momento posterior, en vez del hilo 106 teniendo la responsabilidad de reiniciar el intento posteriormente. Cada hilo 106 puede configurarse sencillamente para iniciar la solicitud de destrucción de objeto 114 y, a continuación, continuar el procesamiento antes de que la destrucción de objeto se haya completado y aún independientemente del resultado de la
- 10 solicitud de destrucción del objeto inicial 114. El flujo de trabajo de destrucción de objeto 202, por lo tanto, facilita la eficiencia de los hilos 106 reduciendo el bloqueo y/o complejidad del hilo en participar en el proceso de destrucción de objeto, y permitiendo a los hilos 106 reanudar el procesamiento productivo mientras la destrucción de objeto es manejada en otros lugares.

- Un cuarto efecto técnico que se puede obtener de acuerdo con las técnicas presentadas en este
- 15 documento involucra una sofisticación del proceso de destrucción de objeto. Al encapsular los detalles de la destrucción de objeto, el flujo de trabajo de destrucción de objeto 202 puede configurarse para manejar una amplia variedad de objetos 102 y recursos de objeto 104 que serán destruidos; una amplia variedad de circunstancias que pueden complicar la destrucción de éste; y una amplia variedad de técnicas para facilitar la manipulación de determinados tipos y circunstancias de los objetos 102 y recursos de objeto
- 20 104.

- La Figura 3 es una ilustración de un escenario ilustrativo 300 modalizando un ejemplo de flujo de trabajo de destrucción de objeto 202. En este escenario ilustrativo 300, el flujo de trabajo de destrucción de objeto 202 está formulado para la destrucción de objeto, en favor de dos hilos 106, de un objeto en particular 102 modalizando un sólido conjunto de recursos de objeto 104. El ejemplo del flujo de trabajo
- 25 de destrucción de objeto 202 está representado como un conjunto de tareas 302 que se realizarán en el objeto 102, tales como marcar el objeto 102 como pendiente de destrucción, así como de destruir recursos particulares 102. Además, las tareas respectivas 302 están asociadas con uno o más criterios de tarea 304, tales como uno o más pre-requisitos para ser satisfechos antes de iniciar la tarea 302 (*por ejemplo*, un pre-requisito de haber completado primero la destrucción de otros recursos de objeto 104),
- 30 y/o circunstancias en donde la tarea 302 puede ser realizada (*por ejemplo*, una identificación de los tipos de hilos 106 a favor de la cual el flujo de trabajo de destrucción de objeto 202 puede realizar la tarea 302, y/o la identificación de un segundo recurso de objeto 104 que va a ser destruido concurrentemente con un primer recurso de objeto 104). El flujo de trabajo de destrucción de objeto 202 también puede comprender un conjunto de indicadores de estado 306 de las tareas respectivas 302 que indican si la
- 35 tarea 302 se ha completado. Algunas tareas 302 del flujo de trabajo de destrucción de objeto 202 pueden realizarse concurrentemente con otras tareas 302 y/o en cualquier orden secuencial (*por ejemplo*, la destrucción del primer recurso de objeto 104 y el segundo recurso de objeto 104 puede realizarse de forma simultánea, y/o en cualquier orden secuencial), mientras que otras tareas 302 pueden especificar

criterios 304 que imponen un orden secuencial y/o simultaneidad particular.

Cuando un hilo 106 en particular disminuye el recuento de referencia 204 del objeto 102 de uno a cero, el hilo 106 puede iniciar una solicitud de destrucción de objeto 114. El flujo de trabajo de destrucción de objeto 202 puede responder al examinar la lista de tareas 302, así como los respectivos criterios de tarea 304 e indicadores de estado 306, para identificar una o más tareas 302 que son elegibles para ser realizadas a favor del hilo 106. El flujo de trabajo de destrucción de objeto 202 puede entonces realizar las tareas identificadas 302 a favor del hilo 106 ya sea hasta que las tareas 302 se completen (de manera que la destrucción de objeto es un éxito 116), o hasta que no haya más tareas 302 que sean elegibles para ser realizadas a favor de este hilo 106, punto en el cual el flujo de trabajo de destrucción de objeto 202 inserta una nueva referencia 206 al objeto 102 (como la última referencia 206) en la lista de destrucción de objeto 212, como un mecanismo para reanudar la destrucción de objeto del objeto 102 en un momento posterior y, posiblemente, a favor de un hilo diferente 106. De esta manera, el flujo de trabajo de destrucción de objeto 202 puede permitir y estar de acuerdo con un sólido conjunto de tareas 302 y criterios de tarea 304 en la destrucción de los recursos de objeto 104 y el objeto 102. Muchos de esos efectos técnicos que se pueden obtener mediante la destrucción de objeto 102 utilizando un flujo de trabajo de destrucción de objeto 202 de acuerdo con las técnicas presentadas en este documento.

D. Modalidades Ilustrativas

La Figura 4 presenta una ilustración de un escenario ilustrativo 400 modalizando una modalidad ilustrativa de las técnicas presentadas en este documento, ilustrada como un dispositivo ilustrativo 402 que destruye los objetos 102 de acuerdo con las técnicas presentadas en este documento. En este escenario ilustrativo 400, el dispositivo 402 comprende un procesador 404, el cual ejecuta un conjunto de hilos 106, *por ejemplo*, a favor de varias aplicaciones y/o el entorno de cómputo del dispositivo 302. El dispositivo 402 además comprende una memoria 406 que almacena instrucciones que, cuando se ejecutan por el procesador 404, implementan un sistema ilustrativo 408 que destruye objetos 102 a favor de los hilos 106 de acuerdo con las técnicas que se presentan en este documento.

El sistema ilustrativo 408 comprende una lista de destrucción de objeto 212, que almacena referencias 206 a objetos 102 que serán destruidos (y, más específicamente, a objetos 102 para los cuales la destrucción de objeto está pendiente, después de al menos un intento anterior para destruir el objeto 102 a favor de al menos un hilo 106 que ha tenido como resultado una falla 118). El sistema ilustrativo 408 comprende además el flujo de trabajo de destrucción de objeto 202, que es ejecutado por el procesador 304 a favor de un primer hilo 106. El flujo de trabajo de destrucción de objeto 202 responde para liberar una última referencia 206 al objeto 102 (*por ejemplo*, reduciendo un recuento de referencia 204 de uno a cero) al intentar destruir el objeto 102 (*por ejemplo*, mediante la realización de diversas tareas 302, tales como la destrucción de recursos de objeto individuales 104 del objeto 102). En respuesta a la falla 118 para completar la destrucción de objeto del objeto 102, el flujo de trabajo de destrucción de objeto 202 añade una nueva referencia 206 al objeto 102 a la lista de destrucción de

objeto 212, en donde la nueva referencia 206 representa la última referencia 206 al objeto 102. La creación de la nueva referencia 206 también puede provocar un incremento del recuento de referencia 204 de las referencias 206 al objeto 102 de cero a uno. Además, el flujo de trabajo de destrucción de objeto 202 responde a la identificación en la lista de destrucción de objeto 212 de una última referencia 206 al objeto 102, a favor del primer hilo 106, añadido a favor de un segundo hilo 106, liberando la última referencia 206 al objeto 102. Dicha liberación, a su vez, invoca el flujo de trabajo de destrucción de objeto 202 con una solicitud de destrucción objeto 114 del objeto 102, a favor del primer hilo 106, lo cual puede resultar en un éxito 116 (o al menos progresión) de la destrucción de objeto del objeto 102. De esta manera, el dispositivo ilustrativo 402 y el sistema ilustrativo 406 de la Figura 4 facilitan los hilos 106 en la destrucción de objetos 102 utilizando un flujo de trabajo de destrucción de objeto 202 de acuerdo con las técnicas presentadas en este documento.

La Figura 5 es una ilustración de un método ilustrativo 500 para destruir objetos 102 a favor de hilos 106 mediante el uso de un flujo de trabajo de destrucción de objeto 202, de acuerdo con las técnicas presentadas en este documento. El método ilustrativo 500 involucra un dispositivo que comprende un procesador 404, y puede implementarse, *por ejemplo*, como un conjunto de instrucciones almacenadas en un componente de memoria de un dispositivo, tal como un circuito de memoria, un plato de un disco duro, un dispositivo de almacenamiento de estado sólido o un disco magnético u óptico y organizadas de tal manera que, cuando se ejecutan en un procesador del dispositivo, hacen que el dispositivo opere de acuerdo con las técnicas que se presentan en este documento.

El método ilustrativo 500 comienza en 502 e involucra la ejecución 504 del flujo de trabajo de destrucción de objeto 202 en el procesador 404 del dispositivo. Específicamente, la ejecución del flujo de trabajo de destrucción de objeto 202 genera 506 una lista de destrucción de objeto 212, que almacena las referencias 206 a objetos 102 a ser destruidos. La ejecución del flujo de trabajo de destrucción de objeto 202, a favor 508 de un primer hilo 106 y sensible a la liberación de una última referencia 206 a un objeto 102 accedido por el primer hilo 106, además intenta 510 destruir el objeto 102; y responde a una falla 118 para completar la destrucción del objeto 102, además añade 512 una nueva referencia 206 al objeto 102 a la lista de destrucción de objeto 212 como la última referencia 206 al objeto 102, en donde la inclusión de la última referencia 206 al objeto 102 en la lista de destrucción de objeto 212 incita a otros hilos 106 (opcionalmente incluyendo el primer hilo 106) a intentar destruir el objeto 102. Habiendo logrado la destrucción de objeto del objeto 102 a favor de los hilos 102, el método ilustrativo 500 implementa el flujo de trabajo de destrucción de objeto 202 y la lista de destrucción de objeto 212 de acuerdo con las técnicas presentadas aquí, y así termina en 514.

Otra modalidad involucra un medio legible por computadora que comprende instrucciones ejecutables por procesador configuradas para aplicar las técnicas que se presentan en este documento. Tales medios legibles por computadora pueden incluir varios tipos de medios de comunicación, tal como una señal que puede propagarse a través de varios fenómenos físicos (*por ejemplo*, una señal electromagnética, una señal de onda de sonido o una señal óptica) y en varios escenarios con cable (*por ejemplo*, a través de una conexión Ethernet o cable de fibra óptica) y/o escenarios inalámbricos (*por*

ejemplo, una red de área local inalámbrica (WLAN) tal como WiFi, una red de área personal (PAN), tal como Bluetooth, o una red celular o por radio) y que codifica un conjunto de instrucciones legibles por computadora que, cuando se ejecutan en un procesador de un dispositivo, hacen que el dispositivo implemente las técnicas que se presentan en este documento. Tales medios legibles por computadora también pueden incluir (como una clase de tecnologías que excluye las comunicaciones multimedia)- dispositivos de memoria legible por computadora, tales como un semiconductor de memoria (*por ejemplo*, un semiconductor que utiliza tecnologías de memoria estática de acceso aleatorio (SRAM), memoria de acceso aleatorio dinámico (DRAM), y/o memoria de acceso aleatorio dinámico síncrono (SDRAM)), un plato de un disco duro, un dispositivo de memoria flash o un disco magnético u óptico (tal como un CD-R, DVD-R o disco flexible), que codifica un conjunto de instrucciones legibles por computadora que, cuando se ejecutan por un procesador de un dispositivo, hacen que el dispositivo para implemente las técnicas que se presentan en este documento.

Un medio legible por computadora ilustrativo que puede ser elaborado en estas formas se ilustra en la Figura 6, en donde la implementación 600 comprende un dispositivo de memoria legible por computadora 602 (por ejemplo, un CD-R, DVD-R o un plato de una unidad de disco duro), en el que se codifican datos legibles por computadora 604. Estos datos legibles por computadora 604 a su vez comprenden un conjunto de instrucciones de computadora 606 que, cuando se ejecutan en un procesador 404 de un dispositivo 610, hacen que el dispositivo 610 opere de acuerdo a los principios establecidos en este documento. En una primera modalidad, las instrucciones ejecutables por procesador 606 pueden activar una configuración de un dispositivo para destruir objetos 102 utilizando un flujo de trabajo de destrucción de objeto 202 de acuerdo con las técnicas presentadas aquí, tal como el dispositivo ilustrativo 402 en el escenario ilustrativo 400 de la Figura 4. En dicha primera modalidad, las instrucciones ejecutables por procesador 606 pueden proporcionar un sistema que, cuando se ejecuta en un dispositivo hace que el dispositivo destruya los objetos 102 utilizando un flujo de trabajo de destrucción de objeto 202 de acuerdo con las técnicas presentadas aquí, tal como el sistema ilustrativo 408 en el escenario ilustrativo 400 de la Figura 4. En una tercera modalidad, las instrucciones ejecutables por procesador 606 del dispositivo de almacenamiento legible por computadora 602 pueden proporcionar un método 608 que hace que un dispositivo destruya los objetos 102 utilizando un flujo de trabajo de destrucción de objeto 202 de acuerdo con las técnicas que aquí se presentan, tal como el método ilustrativo 500 de la Figura 5. Muchos de estos medios legibles por computadora pueden ser concebidos por aquellos expertos en la materia que están configurados para funcionar de acuerdo con las técnicas presentadas en este documento.

E. Variaciones

Las técnicas aquí descritas pueden ser elaboradas con variaciones en muchos aspectos, y algunas variaciones pueden presentar ventajas adicionales y/o reducir las desventajas con respecto a otras variaciones de estas y otras técnicas. Además, algunas variaciones pueden ser aplicadas en combinación, y algunas combinaciones pueden modalizar ventajas adicionales y/o desventajas reducidas

a través de una cooperación sinérgica. Las variaciones pueden ser incorporados en diversas modalidades (*por ejemplo*, el dispositivo ilustrativo 400 y/o el sistema ilustrativo 408 de la Figura 4; el método ilustrativo 500 de la Figura 5; y/o el dispositivo de almacenamiento legible por computadora ilustrativo 602 de la Figura 6) para conferir ventajas individuales y/o sinérgicas con base en tales modalidades.

E1. Escenarios

Un primer aspecto que puede variar entre las modalidades de las técnicas presentadas en el presente documento se refiere a la gama de escenarios en los que estas técnicas pueden ser utilizadas.

Como un primera variante de este primer aspecto, las técnicas presentadas aquí pueden ser utilizadas con una variedad de dispositivos, tales como estaciones de trabajo; computadoras portátiles; tabletas; teléfonos móviles; consolas de juegos y dispositivos de juegos portátiles; reproductores multimedia portátiles o no portátiles; dispositivos de visualización multimedia tales como televisores, aparatos electrodomésticos; y dispositivos de automatización; y componentes de computación integrado con un dispositivo portátil integrando gafas o un reloj.

Como una segunda variante de este primer aspecto, las técnicas presentadas aquí pueden ser utilizadas con una variedad de hilos 106, tales como aplicaciones accionadas por GUI, accionadas por consola y/o IU-menos; servicios y demonios; controladores de dispositivo; y procesos del sistema operativo reales o virtualizados, tales como procesos de núcleo y procesos del sistema de archivos.

Varios hilos y modelos de programación de hilos pueden ser compatibles con las técnicas presentadas en este documento. Los hilos 106 pueden ejecutar dentro de diferentes tipos de entornos de ejecución 108, tales como entornos de ejecución nativa 108 provistos en un lenguaje de máquina y ejecutados directamente por el procesador 404; aplicaciones manejadas que son ejecutadas por un motor en tiempo de ejecución dentro de un determinado entorno de tiempo de ejecución; aplicaciones virtualizadas que se ejecutan dentro de una máquina virtual, tal como un navegador web y los scripts o manuscritos ejecutados por un intérprete. Además, muchos tipos de objetos 102 y recursos de objeto 104 pueden ser manejados y, en particular, destruidos por las técnicas presentadas aquí, incluyendo colecciones ligeras de objetos no escritos y/o ligeramente escritos (*por ejemplo*, almacenamientos de clave/valor, tales como objetos JSON) y objetos fuertemente organizados y escritos (*por ejemplo*, una jerarquía profunda de objetos basados en clases con herencia y escritura polimórfica). El flujo de trabajo de destrucción de objeto 202 también puede invocarse para destruir cualquier objeto 102 dentro del entorno de cómputo del dispositivo 402; sólo para objetos 102 para los cuales la destrucción de objeto presenta una complejidad, tales como objetos 102 que son compartidos por múltiples hilos 106 y/o entornos de ejecución 108; y/o sólo para objetos 102 para los cuales un hilo 106 ha intentado iniciar y completar el proceso de destrucción de objeto y ha encontrado una falla 118.

Como una tercera variante de este primer aspecto, muchos tipos de flujos de trabajo de destrucción de objeto 202 pueden ser utilizados. Como un primer ejemplo, se utilizan a menudo patrones de flujo de trabajo para muchos tipos de tareas en varios entornos de cómputo y muchos de estos flujos

de trabajo se aplican a procesos de recurso intensivo y computacionalmente caros, tales como los pasos intrincados involucrados en la finalización de una transacción de base de datos. El flujo de trabajo de destrucción de objeto 202 aquí presentado se utiliza como un patrón para organizar la destrucción de objetos 102, y puede ser adecuado para una amplia variedad de objetos 102, incluyendo objetos comparativamente ligeros que son típicamente destruibles en respuesta a una sola solicitud de destrucción de objeto 114 (pero que pueden encontrar complicaciones en situaciones inusuales, tales como el bloqueo exclusivo mediante un hilo diferente 106) y objetos complejos 102 que almacenan un conjunto sólido de recursos de objeto 104 con criterios de tareas detalladas 304, como se ilustra en el escenario ilustrativo 300 de la Figura 3. Un flujo de trabajo de destrucción de objeto 202 generalizado también puede escalar para facilitar la destrucción de una amplia gama de esos objetos 102, *por ejemplo*, todo un conjunto de objetos 102 utilizados por una aplicación robusta, en donde tales objetos 102 incluyen tanto los objetos ligeros 102 que normalmente son fáciles de destruir como objetos complejos 102 que presentan amplios criterios de tarea 304 en la destrucción de los mismos. Las técnicas presentadas aquí pueden ser adecuadas para usarse con una amplia gama de tipos de objeto y complejidad.

Como un segundo ejemplo, el flujo de trabajo de destrucción de objeto 202 puede comprender un componente del sistema operativo de un dispositivo 402; un componente de uno o más tiempos de ejecución proporcionados por una plataforma o elemento arquitectónico de tal dispositivo 402; y/o una biblioteca en tiempo de ejecución instalada en el dispositivo 402, que puede ser accesible a todos los hilos 106 para la destrucción de cualquiera de los objetos 102, y/o pueden ser proporcionados para un conjunto específico de hilos 106 y/u objetos 102 (*por ejemplo*, un componente de navegador web que facilita la destrucción de objeto de los objetos 102 utilizados por aplicaciones web). El flujo de trabajo de destrucción de objeto 202 también puede facilitar un conjunto distribuido de hilos 106 que ejecutan en una colección de dispositivos 402, como una colección de hilos 106 distribuidos en un conjunto de servidores que comprenden una granja de servidores y/o puede ser invocada para manejar la destrucción de objeto de los objetos 102 distribuidos en una gran variedad de dispositivos, tal como una colección compartida de objetos almacenados y/o utilizados por una gran variedad de servidores de objeto. Como tercer ejemplo, el flujo de trabajo de destrucción de objeto 202 puede estar embebido en uno o más objetos 102 que son manejados por el flujo de trabajo de destrucción de objeto 202; por el contrario, el flujo de trabajo de destrucción de objeto 202 puede encapsular los objetos 102 manejados por ello; y/o la flujo de trabajo de destrucción de objeto 202 y los objetos 102 pueden existir independientemente y/o dentro de un contexto compartido, tal como un entorno de tiempo de ejecución de uno o más hilos 106. Como cuarto ejemplo, la relación y/o la interconexión de los hilos 106 y el flujo de trabajo de destrucción de objeto 202 puede ser definida de varias maneras. Por ejemplo, el flujo de trabajo de destrucción de objeto 202 puede comprender un administrador que maneja la destrucción del objeto a favor de los hilos 106; como un servicio que los hilos 106 pueden invocar para ayudar con la destrucción de objeto de objetos 102; y/o como un manuscrito o script, equipo de recurso o una biblioteca de funciones que los hilos 106 podrán consultar y/o utilizar para realizar tareas individuales 302 de la destrucción de objeto de

uno o más objetos 102. Los hilos 106 también pueden delegar diversas funciones al flujo de trabajo de destrucción de objeto 202; *por ejemplo*, un hilo 106 y/o el entorno de ejecución 108 puede exponer una expresión lambda, función o devolución de llamada que el flujo de trabajo de destrucción de objeto 202 puede invocar para solicitar la participación del hilo 106 en las tareas 302 implicadas en la destrucción de objeto de un objeto 102.

Como una cuarta variante de este primer aspecto, muchos tipos de las listas de destrucción de objeto 212 pueden ser utilizados. Como un primer ejemplo, la lista de destrucción de objeto 212 puede ser implementada como muchos tipos de estructuras de datos, incluyendo una matriz; una lista vinculada; una cola; un conjunto sin ordenar; una tabla, como una tabla hash; y una estructura de árbol. Como un segundo ejemplo, la lista de destrucción de objeto 212 puede ser implementada como una lista de destrucción de objeto global 212 que almacena últimas referencias 206 a cualquier objeto 102 del entorno de cómputo que se va a destruir. Alternativamente, se pueden proporcionar una o más listas de destrucción de objeto 212, en donde cada lista de destrucción de objeto 212 almacena últimas referencias 206 a objetos 102 de un tipo particular, tal como objetos 102 de un determinado tipo de objeto, y/u objetos 102 utilizados dentro de un contexto seleccionado; objetos 102 utilizados por hilos 106 seleccionados, tales como los hilos 106 de un determinado lenguaje de programación y/o entorno de ejecución 108; y/o para objetos 102 que almacenan un conjunto determinado de recursos de objeto 104, tales como recursos de objeto 104 de un determinado tipo de recurso de objeto 104. Como tercer ejemplo, la lista de destrucción de objeto 212 puede almacenar varios tipos de últimas referencias 206, tal como un puntero de memoria en una ubicación en donde el objeto identificado 102 está almacenado o un identificador único del objeto 102, tal como un identificador globalmente único (GUID) y/o un identificador uniforme de recurso (URI) del objeto 102. Como alternativa o adicionalmente, la lista de destrucción de objeto 212 puede almacenar referencias 206 a recursos de objeto particular 104 que se van a destruir en el curso de la destrucción de uno o más objetos 102 que incluyen y/o hacen referencia a tales recursos de objeto 104. Muchos de estas variaciones de escenarios de cómputo y variaciones arquitectónicas pueden ser adecuadas para el uso de un flujo de trabajo de destrucción de objeto 202 para facilitar la destrucción de objetos 102 de acuerdo con las técnicas que se presentan en este documento.

E2. Flujo de Trabajo de Destrucción de Objeto

Un segundo aspecto que puede variar entre las modalidades de las técnicas presentadas en el presente documento se refiere al flujo de trabajo de destrucción de objeto 202 que se utiliza por los hilos 106 para lograr una destrucción de objeto de varios objetos 102 del entorno de cómputo.

Como un primera variante de este segundo aspecto, el flujo de trabajo de destrucción de objeto 202 puede generarse *ad hoc* a fin de manejar la destrucción de objeto de un objeto particular 102, y/o para manejar la destrucción de objetos 102 por un hilo en particular 106. Alternativamente, el flujo de trabajo de destrucción de objeto 202 puede ser generado para manejar la destrucción de cualquier objeto 102 de dentro del entorno de cómputo a favor de cualquier hilo 106, o para manejar un determinado subconjunto de objetos 102 que son utilizados por un determinado subconjunto de hilos 106. Como un

ejemplo, el entorno de cómputo puede caracterizar un contexto que involucra una colección de hilos 106 que interactúan a través de un determinado conjunto de objetos 102, tales como los hilos 106 y los objetos 102 implicados en un determinado servicio, plataforma, biblioteca y/o aplicación. Además, el flujo de trabajo de destrucción de objeto 202 puede contener varios tipos de información acerca del objeto 102 y/o los hilos 106 implicados en la destrucción de objeto del objeto(s) 102. Como un ejemplo, el flujo de trabajo de destrucción de objeto puede identificar, para un determinado objeto 102, un conjunto de hilos 106 que ha accedido y/o que puede tener acceso a un objeto particular 102. El almacenamiento de dicha información puede permitir que el flujo de trabajo de destrucción de objeto 202 emita una notificación relativa a la destrucción de un objeto particular 102 específicamente a los hilos 106 que han accedido y/o pueden tener acceso al objeto 102, *por ejemplo*, con el fin de informarles del progreso de la destrucción de objeto del objeto 102 y/o para solicitar la participación de estos hilos 106 en la destrucción de objeto. Además, el flujo de trabajo de destrucción de objeto 202 puede incluir varios tipos de procesos de destrucción de información que pueden ser aplicables a diversos objetos 102, tales como las tareas 302 implicadas en la destrucción del objeto 102, los criterios de tarea 304 aplicables a los mismos, y/o el estado 306 de las tareas respectivas 302 en la destrucción de objeto.

Como una segunda variante de este segundo aspecto, la destrucción de objeto de un objeto 102 por el flujo de trabajo de destrucción de objeto 202 puede ser iniciada de varias formas. Como un primer ejemplo, un hilo 106 puede iniciar una solicitud de destrucción de objeto explícita 114, tal como invocar una instrucción de disposición o interconexión de disposición para desechar el objeto 102. Como segundo ejemplo, un hilo 106 puede iniciar una solicitud de destrucción de objeto implícita 114, tal como liberar una referencia 206 al objeto 102 que pasa para constituir la última referencia 206 al objeto y que solicita el decremento de un recuento de referencia 204 de uno a cero, o que cierra un ámbito del hilo 106 que contiene la última referencia 206 al objeto 102. Como tercer ejemplo, un hilo 106 puede auto-terminarse y/o puede ser terminado por otro hilo 106, y la terminación del hilo 106 puede incluir liberar una última referencia 206 al objeto 102 que fue mantenido por el hilo 106.

Como una tercera variante de este segundo aspecto, el flujo de trabajo de destrucción de objeto 202, mientras realiza una destrucción de objeto de un objeto 102 a favor de un primer hilo 106 que inició la solicitud de destrucción de objeto 114, puede encontrar muchos tipos de obstáculos que impiden la realización de la destrucción de objeto a favor del primer hilo 106. Como un primer ejemplo, el flujo de trabajo de destrucción de objeto 202 puede descubrir que una porción seleccionada del objeto 102 está asociada con un proceso de destrucción de objeto de un segundo hilo 106 (*por ejemplo*, una convención de destrucción de objeto utilizada por un entorno de ejecución 108 del segundo hilo 106, tal como una convención del lenguaje de programación y/o plataforma del segundo hilo 106), y puede encontrar una falla 118 al determinar que la porción seleccionada del objeto 102 está asociada con el proceso de destrucción de objeto que será invocado dentro del segundo hilo 106. Como un segundo ejemplo, el flujo de trabajo de destrucción de objeto 202 puede descubrir que un recurso de objeto 104 del objeto 102 se mantiene exclusivamente por un hilo 106 diferente a través de un bloqueo de recurso de objeto 112 que ni el primer hilo 106 ni el flujo de trabajo de destrucción de objeto 202 es capaz de interrumpir. Como un

tercer ejemplo, el flujo de trabajo de destrucción de objeto 202 puede determinar que un recurso de objeto 104 del objeto 102 está en un estado de ocupado (*por ejemplo*, un puntero inteligente de dispositivo que está realizando un proceso ininterrumpido usando un dispositivo externo, en donde la destrucción del recurso de objeto 104 durante el proceso ininterrumpido puede resultar en la corrupción de datos o daño en el dispositivo externo). Como un cuarto ejemplo, el flujo de trabajo de destrucción de objeto 202 simplemente puede descubrir que el primer hilo 106 que inicia la solicitud de discusión de objeto 114 no tiene permiso para destruir el objeto 102. Muchos de estos obstáculos pueden surgir durante la destrucción de objeto por el flujo de trabajo de destrucción de objeto 202.

La Figura 7 es una ilustración de un escenario ilustrativo 700 modalizando una cuarta variante de este segundo aspecto, en donde el flujo de trabajo de destrucción de objeto 202 permite que un objeto 102 sea marcado como un objeto protegido que puede ser destruido sólo por un hilo seleccionado 106, y en donde el flujo de trabajo de destrucción de objeto 202 niega solicitudes de destrucción de objeto 114 iniciadas por cualquier otro hilo 106. En este escenario ilustrativo 700, un objeto particular 102 está marcado 702 como un objeto protegido que sólo puede ser destruido a petición de un segundo hilo 106. En un primer momento 704, un primer hilo 106 inicia una solicitud de destrucción de objeto 114, *por ejemplo*, intentando reducir el recuento de referencia 204 de uno a cero. Sin embargo, el flujo de trabajo de destrucción de objeto 202 determina que el objeto 102 se marca para evitar la destrucción de objeto por el primer hilo 106 y, por lo tanto, rechaza la solicitud de destrucción de objeto 114 del primer hilo 106 (incluyendo la negación de permitir que el primer hilo 106 disminuya el recuento de referencia 204 de uno a cero, y/o realizar una liberación de referencia 214 de la última referencia 206 en la lista de destrucción de objeto 212), y devuelve una falla 118 de la solicitud de destrucción de objeto 114 al primer hilo 106. En un segundo momento 706, el flujo de trabajo de destrucción de objeto 202 recibe una solicitud de destrucción de objeto 202 desde el segundo hilo 106, y comprueba que la marcación 702 del objeto 102 permite que el segundo hilo 106 inicie la destrucción de objeto. El flujo de trabajo de destrucción de objeto 202, por lo tanto, realiza una liberación de referencia 214 de la última referencia 206 en la lista de destrucción de objeto 212, disminuye el recuento de referencia 204 de uno a cero, y reporta un éxito 116 de, al menos, iniciar la solicitud de destrucción de objeto 114 del objeto 102.

La Figura 8 presenta una ilustración de un escenario ilustrativo 800 modalizando una quinta variación de este segundo aspecto, implicando una gráfica objeto cíclico 802 de objetos 102 que respectivamente contienen referencias 206 a otros objetos 102 de la gráfica de objeto 802, formando un ciclo de referencia. Cuando el flujo de trabajo de destrucción de objeto 202 recibe, desde un hilo 106, una solicitud de destrucción de objeto 114 para destruir un objeto 102 de la gráfica de objeto cíclico 802, el flujo de trabajo de destrucción de objeto 202 puede encontrar una dificultad, en que ninguno de los objetos 102 es capaz de ser totalmente destruido sin destruir primero uno o más de otros objetos 102 de la gráfica de objeto cíclico 802 que almacena una referencia 206 al objeto 102 identificado en la solicitud de destrucción de objeto 114. El conjunto completo de objetos 102 que comprenden la gráfica de objeto cíclico 802 puede, por lo tanto, persistir más allá del ciclo de vida de una parte o la totalidad de los objetos 102 debido a las interrelaciones de los objetos 102. A fin de abordar esta cuestión, el flujo de

trabajo de destrucción de objeto 202 puede proporcionar un mecanismo de sincronización 804 sobre el conjunto completo de objetos 102 de la gráfica de objeto cíclico 802 (*por ejemplo*, un bloqueo exclusivo sobre todos los objetos 102). Al recibir una solicitud de destrucción de objeto 114 para destruir uno, varios o todos los objetos de la gráfica de objeto cíclico 802, el flujo de trabajo de destrucción de objeto 202 puede en primer lugar, asegurar el acceso exclusivo a los 102 objetos de la gráfica de objeto cíclico 802 mediante el mecanismo de sincronización 804. El flujo de trabajo de destrucción de objeto 202 puede entonces iniciar la destrucción de objeto de los objetos 102 de acuerdo con una secuencia de destrucción 806 (*por ejemplo*, destruyendo los objetos 102 en un orden específico). En particular, la destrucción de objeto de cada objeto 102 puede realizarse independientemente de las referencias 206 al objeto 102 mantenidas por otros objetos 102 de la gráfica de objeto cíclico 802, ya que el mecanismo de sincronización 804 mantenido por el flujo de trabajo de destrucción de objeto 202 excluye que cualesquiera otros hilos 106 accedan a los objetos 102 y potencialmente encuentren una referencia no válida 206 a un objeto anteriormente destruido 102. El flujo de trabajo de destrucción de objeto 202 puede retener el mecanismo de sincronización 804 hasta que la secuencia de destrucción 806 sobre la gráfica de objeto cíclico 802 quede completa y todos esos objetos 102 se hayan destruido, permitiendo la destrucción de la gráfica de objeto cíclico 802.

La Figura 9 presenta una ilustración de un escenario ilustrativo 900 modalizando una sexta variación de este segundo aspecto, en donde el flujo de trabajo de destrucción de objeto 202 marca objetos 102 para los que se ha iniciado la destrucción de objeto, pero todavía no se ha terminado, como marcados con un estado de zombi 902. En este escenario ilustrativo 900, en un primer momento 908, un primer hilo 106 inicia una solicitud de destrucción de objeto 114 para destruir un objeto particular 102, y el flujo de trabajo de destrucción de objeto 202 inicia la destrucción de objeto 102. Al iniciar la destrucción de objeto, el flujo de trabajo de destrucción de objeto 202 puede aplicar al objeto 102 una marca de un estado de zombi 902, indicando a otros hilos 106 que la destrucción de objeto del objeto 102 está en progreso. En respuesta a encontrar una falla 118 para completar la destrucción de objeto del objeto 102, el flujo de trabajo de destrucción de objeto 202 realiza una creación de referencia 210, añade una nueva referencia 206 al objeto 102 a una lista de destrucción de objeto 212 como la última referencia 206 al objeto 102, e incrementa un contador de referencia 204 de cero a uno. El flujo de trabajo de destrucción de objeto 202 entonces espera para que otro hilo 106 reanude el proceso de destrucción de objeto. Sin embargo, en un segundo momento 910 antes de que dicha destrucción de objeto se complete, un segundo hilo 106 puede iniciar una solicitud de acceso 904 sobre el objeto 102, tal como una solicitud para almacenar datos en el objeto 102. La marcación del objeto 102 como un estado de zombi 902 puede resultar en la generación de una advertencia 906 para el segundo hilo 106 indicando el estado de zombi 902 del objeto 102, y la advertencia de que la lectura de datos de y/o escritos en el objeto 102 puede no ser exacta, actual y/o válida y/o pueden perderse durante la destrucción de objeto. De esta manera, el flujo de trabajo de destrucción de objeto 202 asegura al objeto 102 del acceso por parte de otros hilos 106 durante la tramitación de la destrucción de objeto, y la reducción de excepciones, comportamiento inconsistente, y/o la pérdida de datos que pueda resultar de que los hilos 106 inadvertidamente accedan

al objeto 102 durante la destrucción de objeto. Muchas de estas técnicas pueden ser utilizadas por el flujo de trabajo de destrucción de objeto 202 para llevar a cabo la destrucción de objeto de los objetos 102 de acuerdo con las técnicas que se presentan en este documento.

5 **E3. Lista de Destrucción de Objeto**

Un tercer aspecto que puede variar entre las modalidades de las técnicas presentadas en este documento implica el uso de la lista de destrucción de objeto 212 por el flujo de trabajo de destrucción de objeto 202 para almacenar referencias 206 a los objetos 102 a ser destruidos (y, más concretamente, para las cuales por lo menos un hilo 106 ha iniciado una solicitud de destrucción de objeto 114 que ha
10 tenido como resultado una falla 118).

La Figura 10 es una ilustración de un escenario ilustrativo 1000 que modaliza un primera variante de este segundo aspecto, en donde el flujo de trabajo de destrucción de objeto 202 permite que un objeto 102 sea marcado con un estado de retención 1002, de manera que el objeto 102 será retenido más allá de la liberación de referencia 214 de la última referencia en cualquier hilo 106. Por ejemplo, el objeto 102
15 puede comprender una instancia de una biblioteca en tiempo de ejecución o servicio que es para permanecer disponible a los hilos 106 del entorno de cómputo, incluso aunque ningún hilo 106 está actualmente utilizando el objeto 102. En este escenario ilustrativo 1000, un hilo 106 que almacena una última referencia 206 al objeto 102 inicia una solicitud de destrucción de objeto 114 para liberar la última referencia 206, reduce el recuento de referencia 204 de uno a cero y destruye el objeto 102. Sin
20 embargo, el flujo de trabajo de destrucción de objeto 202 identifica el objeto 102 como habiendo sido marcado como un objeto retenido 1002, para el cual una última referencia 206 va a ser mantenida en lugar de liberada. El flujo de trabajo de destrucción de objeto 202 pueden, por lo tanto, reportar una falla 118 para reducir el recuento de referencia 204, liberar la última referencia 206, y/o destruir el objeto 102. En su lugar, el flujo de trabajo de destrucción de objeto 202 puede almacenar la última referencia 206 en la
25 lista de destrucción de objeto 212, aún puede rechazar las solicitudes por distintos hilos 106 para liberar la última referencia 206 y/o reducir el recuento de referencia 204 de uno a cero. El flujo de destrucción de objeto 202, por lo tanto, utiliza la lista de destrucción de objeto 212 como una memoria caché de objeto retenido que almacena en memoria caché una referencia 206 al objeto retenido 102, garantizando así que el objeto 102 no va a ser destruido mientras detiene el marcado como un objeto retenido 1002.

Además, aunque la memoria caché de objeto retenido almacena la referencia 206 guardada en memoria caché en el objeto retenido 1002, otras referencias 206 al objeto retenido 1002 dentro de los respectivos hilos se marcan como una referencia débil al objeto retenido 1002, y no son rastreados por el recuento de referencia 204. En respuesta a una solicitud para eliminar la marcación del objeto 102 como un objeto retenido 1002, el flujo de trabajo de destrucción de objeto 202 puede cambiar cualquier referencia 206 en
35 los hilos 106 para referencias fuertes 206 (y puede incrementar el contador de referencia 204 en consecuencia), de manera que la liberación de todas esas referencias 204 invoca la destrucción de objeto del objeto 102.

Como una segunda variante de este tercer aspecto, los hilos 106 y el flujo de trabajo de

destrucción de objeto 202 pueden interactuar de diversas maneras para comunicarse acerca de la lista de destrucción de objeto 212, incluyendo la adición de una referencia 206 a un objeto 102 a la lista de destrucción de objeto 212 para la cual distintos hilos 106 pueden iniciar una solicitud de destrucción de objeto 114.

5 La Figura 11 presenta una ilustración de un escenario ilustrativo 1100 modalizando dos de dichas técnicas en virtud de las cuales los hilos 106 y el flujo de trabajo de destrucción de objeto 202 utilizan la lista de destrucción de objeto 212 para coordinar la destrucción de objeto de un objeto 102. Como un primer ejemplo 1106, un primer hilo 106 puede iniciar una solicitud de destrucción de objeto 114, durante la cual el flujo de trabajo de destrucción de objeto 202 puede encontrar una falla 118, provocando que el
10 flujo de trabajo de destrucción de objeto 202 agregue una nueva referencia 206 a la lista de destrucción de objeto 212 como una última referencia al objeto 102. Además, en respuesta a la adición de la nueva referencia 206 al objeto 102 para la lista de destrucción de objeto 212, el flujo de trabajo de destrucción de objeto 202 además puede notificar a un segundo hilo 106 que también tiene acceso al objeto 102 acerca de la adición de la nueva referencia 206 al objeto 102 a la lista de destrucción de objeto 212. La
15 notificación 1102 puede incitar al segundo hilo 106 a liberar la última referencia 206, incitando así a que el flujo de trabajo de destrucción de objeto 202 reinicie el proceso de destrucción de objeto a favor del segundo hilo 106.

En esta variación, se pueden muchas técnicas mediante las cuales el flujo de trabajo de destrucción de objeto 202 identifica al segundo hilo 106 como un receptor de la notificación 1102. Como
20 un primer ejemplo, la notificación 1102 puede también ser iniciada por el flujo de trabajo de destrucción de objeto 202, y/o por la lista de destrucción de objeto 212 al recibir la última referencia 206 al objeto 102. Como segundo ejemplo, el segundo hilo 106 además puede comprender un hilo superior 106 que es superior al primer hilo 106, y que comparte el acceso al objeto 106 (*por ejemplo*, un hilo de un tiempo de ejecución de administración que maneja al primer hilo 106), y el flujo de trabajo de destrucción de objeto
25 202 puede notificar al hilo superior 106 de la adición de la nueva referencia 206 al objeto 102 a la lista de destrucción de objeto 212. Como un tercer ejemplo, el flujo de trabajo de destrucción de objeto 202 puede buscar entre una colección de hilo de 106 hilos que son ejecutados por el procesador para identificar el segundo hilo como uno de varios hilos que tiene acceso a y/o está accediendo actualmente o ha accedido previamente, al objeto 102. Como un cuarto ejemplo, la notificación 1102 puede ser enviada
30 al segundo hilo 106 en una variedad de maneras. Por ejemplo, el segundo hilo 106 puede también comprender una cola de mensaje de los mensajes que se van a enviar al segundo hilo 106 y el flujo de trabajo de destrucción de objeto 202 puede insertar la notificación 1102 en la cola de mensaje del segundo hilo 106, en donde el mensaje notifica al segundo hilo 106 de la adición de la nueva referencia 206 al objeto 102 a la lista de destrucción de objeto 212. Como un quinto ejemplo, si el segundo hilo 106
35 al cual la notificación 1102 va a ser enviada está en un estado suspendido, el flujo de trabajo de destrucción de objeto 202 puede activar el segundo hilo 106 para procesar la notificación 1102 y continuar la destrucción de objeto del objeto 102.

La Figura 11 también ilustra un segundo ejemplo 1108 de una interacción entre los hilos 106 y la

lista de destrucción de objeto 212, en donde los respectivos hilos 106 periódicamente revisan la lista de destrucción de objeto 212 para identificar la presencia de referencias 206 a objetos 102 que están actualmente y/o han sido accedidas por el hilo 106. En este segundo ejemplo 1108, un primer hilo 106 puede iniciar una solicitud de destrucción de objeto 114, durante la cual el flujo de trabajo de destrucción de objeto 202 puede producir una falla 118, provocando que el flujo de trabajo de destrucción de objeto 202 agregue una nueva referencia 206 a la lista de destrucción de objeto 212 como una última referencia al objeto 102. Un segundo hilo 106 puede sondear 1104 la lista de destrucción de objeto 212, y al descubrir la última referencia 206 al objeto 102, puede determinar si el objeto 102 está actualmente y/o ha sido accedido por el segundo hilo 102. En respuesta a determinar que el segundo hilo 106 tiene una relación con el objeto 102, el segundo hilo 106 puede remover la última referencia 206 al objeto 102 de la lista de destrucción de objeto 212 y realizar una liberación de referencia 214 de la última referencia 206, iniciando así una reanudación del flujo de trabajo de destrucción de objeto 202 a favor del segundo hilo 106. Muchas de estas técnicas pueden ser utilizadas por los hilos 106 y el flujo de trabajo de destrucción de objeto 202 para comunicarse entre ellos acerca del procesamiento de destrucción de objeto de los objetos 102 de acuerdo con las técnicas que se presentan en este documento.

F. Entorno de Cómputo

La Figura 12 y la siguiente discusión proporcionan una breve descripción general de un entorno de cómputo adecuado para implementar modalidades de una o más de las disposiciones establecidas en el presente documento. El entorno operativo de la Figura 12 es sólo un ejemplo de un entorno operativo adecuado y no pretende sugerir ninguna limitación en cuanto al alcance de uso o funcionalidad del entorno operativo. Los dispositivos de cómputo ilustrativos incluyen, pero no están limitados a, computadoras personales, computadoras de servidor, dispositivos de mano o dispositivos portátiles, dispositivos móviles (tales como teléfonos móviles, asistentes digitales personales (PDA), reproductores multimedia, etc.), sistemas de multiprocesador, electrónica de consumidor, mini computadoras, computadoras centrales, entornos de computación distribuida que incluyen cualquiera de los sistemas o dispositivos anteriores y similares.

Aunque no sea necesario, las modalidades se describen en el contexto general de "instrucciones legibles por computadora" ejecutadas por uno o más dispositivos de cómputo. Las instrucciones legibles por computadora pueden ser distribuidas a través de medios legibles por computadora (véase más adelante). Las instrucciones legibles por computadora pueden ser implementadas como módulos de programa, tales como funciones, objetos, interfaces de programación de aplicación (API), estructuras de datos y similares, que realizan determinadas tareas o implementan determinados tipos de datos abstractos. Normalmente, la funcionalidad de las instrucciones legibles por computadora puede combinarse o distribuirse como se desee en diversos entornos.

La Figura 12 muestra un ejemplo de un sistema 1200 compuesto por un dispositivo de cómputo 1202 configurado para implementar una o más modalidades aquí proporcionadas. En una configuración, el dispositivo de cómputo 1202 incluye al menos una unidad de procesamiento 1206 y memoria 1208.

Dependiendo de la configuración exacta y el tipo de dispositivo de cómputo, la memoria 1208 puede ser volátil (tal como la RAM, por ejemplo), no volátil (tal como la ROM, memoria flash, etc., por ejemplo) o alguna combinación de las dos. Esta configuración se muestra en la Figura 12 por la línea discontinua 1204.

En otras modalidades, el dispositivo 1202 puede incluir características adicionales y/o funcionalidad. Por ejemplo, el dispositivo 1202 también puede incluir almacenamiento adicional (por ejemplo, removible y/o no removible) incluyendo, pero no limitado a, almacenamiento magnético, almacenamiento óptico, y similares. Tal almacenamiento adicional se ilustra en la Figura 12 por el almacenamiento 1210. En una modalidad, las instrucciones legibles por computadora para implementar uno o más modalidades aquí proporcionadas pueden estar en el almacenamiento 1210. El almacenamiento 1210 también puede almacenar otras instrucciones legibles por computadora para implementar un sistema operativo, un programa de aplicación, y similares. Las instrucciones legibles por computadora pueden ser cargadas en la memoria 1208 para la ejecución por la unidad de procesamiento 1206, por ejemplo.

El término "medios legibles por computadora" utilizado en el presente texto incluye dispositivos de memoria legible por computadora que excluyen otras formas de medios legibles por computadora comprendiendo medios de comunicación, tales como señales. Tales dispositivos de memoria legible por computadora pueden ser volátiles o no volátiles, removibles y/o no removibles y pueden involucrar varios tipos de dispositivos físicos que almacenan instrucciones legibles por computadora u otros datos. La memoria 1208 y el almacenamiento 1210 son ejemplos de medios de almacenamiento por computadora. Los dispositivos de almacenamiento por computadora incluyen, pero no están limitados a, RAM, ROM, EEPROM, memoria flash u otra tecnología de memoria, CD-ROM, discos versátiles digitales (DVD) u otro tipo de almacenamiento óptico, casetes magnéticos, cintas magnéticas y almacenamiento en discos magnéticos u otros dispositivos de almacenamiento magnético.

El dispositivo 1202 también puede incluir la conexión(es) de comunicación 1216 que permite al dispositivo 1202 comunicarse con otros dispositivos. La conexión(es) de comunicación 1216 puede incluir, pero no se limita a, un módem, una tarjeta de interfaz de red (NIC), una interfaz de red integrada, un transmisor/receptor de frecuencia de radio, un puerto de infrarrojos, una conexión USB u otras interfaces para la conexión del dispositivos de cómputo 1202 a otros dispositivos de cómputo. La conexión(es) de comunicación 1216 puede incluir una conexión con cable o una conexión inalámbrica. La conexión(es) de comunicación 1216 puede transmitir y/o recibir medios de comunicación.

El término "medios legibles por computadora" puede incluir medios de comunicación. Los medios de comunicación suelen modalizar instrucciones legibles por computadora u otros datos en una "señal de datos modulada" tal como una onda portadora u otro mecanismo de transporte e incluye cualquier medio de suministro de información. El término "señal de datos modulada" puede incluir una señal que tiene una o más de sus características fijadas o cambiadas de tal manera que codifica la información en la señal.

El dispositivo 1202 puede incluir dispositivos(s) de entrada 1214, tales como un teclado, ratón, lápiz óptico, dispositivo de entrada de voz, dispositivo de entrada táctil, cámaras infrarrojas, dispositivos

de entrada de vídeo y/o cualquier otro dispositivo de entrada. En el dispositivo 1202 también se puede incluir un dispositivo de salida(s) 1212 tal como una o más pantallas, altavoces, impresoras y/o cualquier otro dispositivo de salida. El dispositivo(s) de entrada 1214 y el dispositivo(s) de salida 1212 pueden ser conectados al dispositivo 1202 a través de una conexión por cable, conexión inalámbrica, o cualquier combinación de las mismas. En una modalidad, un dispositivo de entrada o un dispositivo de salida desde otro dispositivo de cómputo pueden utilizarse como dispositivo(s) de entrada 1214 o dispositivo(s) de salida 1212 para el dispositivo de cómputo 1202.

Los componentes del dispositivo de cómputo 1202 pueden ser conectados a través de varias interconexiones, tales como un conductor común. Tales interconexiones pueden incluir una Interconexión de Componente Periférico (PCI), tales como PCI Express, un conductor común en serie universal (USB), Firewire (IEEE 1394), una estructura de conductor común óptico y similares. En otra modalidad, los componentes del dispositivo de cómputo 1202 pueden estar interconectados a través de una red. Por ejemplo, la memoria 1208 puede estar compuesta de múltiples unidades de memoria física ubicadas en diferentes lugares físicos interconectados mediante una red.

Aquellos expertos en la materia se darán cuenta de que los dispositivos de almacenamiento utilizados para almacenar instrucciones legibles por computadora pueden ser distribuidos a través de una red. Por ejemplo, un dispositivo de cómputo 1220 accesible a través de la red 1218 puede almacenar instrucciones legibles por computadora para implementar uno o más modalidades aquí proporcionadas. El dispositivo de cómputo 1202 puede tener acceso al dispositivo de cómputo 1220 y descargar una parte o todas las instrucciones legibles por computadora para su ejecución. Alternativamente, el dispositivo de cómputo 1202 pueden descargar piezas las instrucciones legibles por computadora, según sea necesario, o algunas instrucciones pueden ser ejecutadas en el dispositivo de cómputo 1202 y algunas en el dispositivo de cómputo 1220.

G. Uso de Términos

Aunque la materia objeto se ha descrito en el lenguaje específico de características estructurales y/o actos metodológicos, se entiende que la materia objeto definida en las reivindicaciones anexas no se limita necesariamente a las características específicas o los actos descritos anteriormente. Más bien, las características específicas y los actos descritos anteriormente son revelados como formas ilustrativas para implementar las reivindicaciones.

Como se utiliza en esta solicitud, los términos "componente", "módulo", "sistema", "interfaz" y similares se refieren a una entidad relacionada con computadoras, ya sea *hardware*, una combinación de *hardware* y *software*, *software* o *software* en ejecución. Por ejemplo, un componente puede ser, pero no se limita a ser, un proceso que se ejecuta en un procesador, un procesador, un objeto, un archivo ejecutable, un hilo de ejecución, un programa, y/o una computadora. A modo de ilustración, tanto una aplicación que se ejecuta en un controlador como el controlador pueden ser un componente. Uno o más componentes pueden residir dentro de un proceso y/o hilo de ejecución y un componente puede ser localizado en una computadora y/o distribuido entre dos o más computadoras.

Además, la materia objeto reclamada puede ser implementada como un método, aparato, o artículo de fabricación utilizando técnicas estándares de programación y/o de ingeniería para producir *software*, *firmware*, *hardware*, o cualquier combinación de los mismos para controlar una computadora para implementar la materia objeto descrita. El término "artículo de fabricación" tal como se utiliza en el presente documento pretende abarcar un programa de computadora accesible desde cualquier dispositivo legible por computadora, portador o medios. Por supuesto, aquellos expertos en la materia reconocerán que muchas modificaciones pueden hacerse a esta configuración sin apartarse del alcance o del espíritu de la materia objeto reclamada.

Aquí se proporcionan varias operaciones de modalidades. En una modalidad, una o más de las operaciones descritas puede constituir instrucciones legibles por computadora almacenadas en uno o más medios legibles por computadora, que si se ejecutan en un dispositivo de cómputo, harán que el dispositivo de cómputo realice las operaciones descritas. El orden en el que todas o algunas de las operaciones son descritas no debe ser construido como implicando que estas operaciones son necesariamente dependientes del orden. Un ordenamiento alternativo será apreciado por un experto en la materia teniendo el beneficio de esta descripción. Además, se entenderá que no todas las operaciones están necesariamente presentes en cada modalidad aquí proporcionada.

Cualquier aspecto o diseño descrito en este documento como un "ejemplo" no necesariamente tiene que ser interpretado como ventajoso sobre otros aspectos o diseños. Más bien, el uso de la palabra "ejemplo" está destinado a presentar un posible aspecto y/o implementación que puede pertenecer a las técnicas presentadas en este documento. Dichos ejemplos no son necesarios para dichas técnicas, o pretenden ser limitantes. Varias modalidades de dichas técnicas pueden incluir un ejemplo, solo o en combinación con otras características, y/o pueden variar y/u omitir el ejemplo ilustrado.

Como se utiliza en esta solicitud, el término "o" se entiende una "o" amplia en vez de una "o" exclusiva. Es decir, a menos que se especifique lo contrario, o claro del contexto, "X emplea A o B" se entiende cualquiera de las permutaciones inclusivas naturales. Es decir, si X emplea A; X emplea B; o X emplea tanto A como B, luego "X emplea A o B" está satisfecho bajo cualquiera de los casos anteriores. Además, los artículos "un" y "una" tal como se utiliza en esta solicitud y reivindicaciones anexas generalmente pueden interpretarse como "uno o más", a menos que se especifique lo contrario o sea claro del contexto que se dirige a una forma singular.

Además, aunque la descripción ha sido mostrada y descrita con respecto a una o más implementaciones, alteraciones y modificaciones equivalentes se les ocurrirán a otros expertos en la materia basadas en una lectura y comprensión de esta especificación y los dibujos adjuntos. La descripción incluye todas esas modificaciones y alteraciones y sólo está limitada por el alcance de las siguientes reivindicaciones. En particular, respecto a las diversas funciones realizadas por los componentes descritos anteriormente (por ejemplo, elementos, recursos, etc.), los términos utilizados para describir estos componentes están diseñados para corresponder, a menos que se indique lo contrario, a cualquier componente que realiza la función especificada del componente descrito (*por ejemplo*, que es funcionalmente equivalente), aunque no estructuralmente equivalente a la estructura

descrita que realiza la función en las implementaciones de ejemplo aquí ilustradas de la descripción. Además, mientras que una característica particular de la descripción puede haber sido descrita con respecto a sólo una de las varias implementaciones, tal característica puede combinarse con una o más de las otras características de las otras implementaciones como pueden ser deseado y ventajoso para cualquier aplicación dada o particular. Además, en la medida en que los términos "incluye", "que tiene", "tiene", "con", o sus variantes se utilicen en la descripción o las reivindicaciones, tales términos están pensados para ser inclusivos en una manera similar al término "que comprende".

10

15

20

25

30

35

REIVINDICACIONES

1. Un dispositivo que destruye objetos dentro de un entorno de recuento de referencia de un dispositivo, el dispositivo comprendiendo:

5 un procesador, y
una memoria que almacena:
una lista de destrucción de objeto de referencias a objetos a ser destruidos; y
un flujo de trabajo de destrucción de objeto ejecutado por el procesador a favor de un primer hilo que:

10 en respuesta a la liberación de una última referencia a un objeto accedido por el primer hilo:

intentar destruir el objeto; y

en respuesta a la falla de completar la destrucción del objeto, agregar una nueva referencia al objeto en la lista de destrucción de objeto, tal como la última referencia al objeto;

15 y
en respuesta a la identificación, para el objeto, de la última referencia al objeto agregado a la lista de destrucción de objeto por un segundo hilo, liberar la última referencia al objeto.

2. El dispositivo de acuerdo con la reivindicación 1, en donde:

una porción seleccionada del objeto está asociada con un proceso de destrucción de objeto del
20 segundo hilo; y

la falla para completar la destrucción del objeto además comprende: determinar que la porción seleccionada se asocia con el proceso de destrucción de objeto que será invocado en el segundo hilo.

3. El dispositivo de acuerdo con la reivindicación 1, en donde:

una porción seleccionada del objeto está reservada exclusivamente por un hilo distinto del primer
25 hilo; y

la falla para completar la destrucción del objeto además comprende: determinar que la porción seleccionada del objeto está reservada exclusivamente por un hilo distinto del primer hilo.

4. El dispositivo de acuerdo con la reivindicación 1, en donde el flujo de trabajo de destrucción de objeto además comprende:

30 un conjunto de tareas que deben realizarse para destruir el objeto; y
para tareas respectivas:

un criterio de tarea de la tarea; y

un estado de la tarea.

5. El dispositivo de acuerdo con la reivindicación 1, en donde:

35 el flujo de trabajo de destrucción de objeto, en respuesta a la iniciación de la destrucción de objeto de un objeto, marca al objeto con un estado de zombi; y

el objeto, en respuesta a un segundo hilo que accede al objeto mientras se marca como un estado de zombi, devuelve al segundo hilo una advertencia indicando un estado de zombi del objeto.

6. El dispositivo de acuerdo con la reivindicación 1, en donde:
el dispositivo además almacena un conjunto de objeto de objetos comprendiendo una gráfica de objeto cíclico; y

el flujo de trabajo de destrucción de objeto, en respuesta a intentar destruir el objeto:

5 identifica al objeto como un miembro del conjunto de objeto de objetos comprendiendo la gráfica de objeto cíclico;

 invoca un mecanismo de sincronización para obtener acceso exclusivo al conjunto de objeto de la gráfica de objeto cíclico;

10 mientras retiene el acceso exclusivo del mecanismo de sincronización, destruye los objetos del conjunto de objeto de acuerdo con una secuencia de destrucción; y

 en respuesta a destruir todos los objetos de la gráfica de objeto cíclico, liberar el acceso exclusivo del mecanismo de sincronización.

7. El dispositivo de acuerdo con la reivindicación 6, en donde la recepción de la operación de disposición comprende:

15 determinar que el objeto está marcado como un objeto protegido que se puede destruir sólo por un hilo seleccionado distinto del primer hilo; y

 en respuesta a la determinación, negar la destrucción del objeto.

8. Un método para destruir objetos dentro de un entorno de recuento de referencia de un dispositivo que comprende un procesador, el método comprende:

20 ejecutar, en el procesador, un flujo de trabajo de destrucción de objeto que:

 genera un lista de destrucción de objeto de referencias a objetos a ser destruidos; y

 a favor de un primer hilo y en respuesta a liberar una última referencia a un objeto accedido por el primer hilo:

 intentar destruir el objeto; y

25 en respuesta a la falla de completar la destrucción del objeto, agregar una nueva referencia al objeto en la lista de destrucción de objeto como la última referencia al objeto, en donde la inclusión de la última referencia al objeto en la lista de destrucción de objeto incita a otros hilos a acceder al objeto para intentar destruir el objeto.

30 9. El método de acuerdo con la reivindicación 8, en donde la ejecución de las instrucciones además hacen que el dispositivo, en respuesta a la adición de la nueva referencia al objeto en la lista de destrucción de objeto, notifique a un segundo hilo que también tiene acceso al objeto de la adición de la nueva referencia al objeto en la lista de destrucción de objeto como la última referencia al objeto.

35 10. El método de acuerdo con la reivindicación 8, en donde la lista objeto liberable, en respuesta a la recepción de la última referencia al objeto a la lista de objeto liberable, notifica al segundo hilo de la adición de la nueva referencia al objeto a la lista de objeto liberable como la última referencia al objeto.

11. El método de acuerdo con la reivindicación 10, en donde la notificación al segundo hilo además comprende: en respuesta a la identificación de un estado suspendido del segundo hilo, activar el

segundo hilo para procesar la última referencia al objeto en la lista de objeto liberable.

12. Un sistema para la destrucción de objetos dentro de un entorno de recuento de referencia de un dispositivo, el método comprende:

una lista de destrucción de objeto de referencias a objetos a ser destruidos; y

5 un flujo de trabajo de destrucción de objeto que, en respuesta a que un primer hilo libera una última referencia a un objeto accedido por el hilo:

intenta destruir el objeto; y

10 en respuesta a la falla para completar la destrucción del objeto, agregar una nueva referencia al objeto en la lista de destrucción de objeto como la última referencia al objeto, en donde la inclusión de la última referencia al objeto en la lista de destrucción de objeto incita a otros hilos a acceder al objeto para intentar destruir el objeto.

13. El sistema de acuerdo con la reivindicación 12, en donde:

entre los objetos dentro del entorno de recuento de referencia del dispositivo, un conjunto de objeto de sesión de contexto de objetos que se utilizan dentro de una sesión de contexto; y

15 la lista de referencia liberable está asociada con la sesión de contexto para almacenar las últimas referencias a los objetos del conjunto de objeto de sesión de contexto de la sesión contexto.

14. El sistema de acuerdo con la reivindicación 12, en donde:

el objeto es un objeto retenido que va a ser retenido después de que el primer hilo libera la última referencia al objeto; y

20 el sistema además comprende: un objeto retenido en memoria caché que almacena una referencia guardada en memoria caché al objeto fuera del primer hilo y el segundo hilo.

15. El sistema de acuerdo con la reivindicación 14, en donde:

25 mientras el objeto retenido en memoria caché almacena la referencia guardada en memoria caché al objeto retenido, otras referencias del objeto retenido dentro de sus respectivos hilos están marcados como una referencia débil al objeto retenido; y

el flujo de trabajo de destrucción de objeto, en respuesta a remover la referencia guardada en memoria caché al objeto retenido, cambia las referencias débiles al objeto retenido en los respectivos hilos a una referencia fuerte.

RESUMEN

Los objetos en un entorno de cómputo por lo general son utilizados por hilos a través de un ciclo de vida del objeto y se destruyen al final del ciclo de vida del objeto para reclamar los recursos de cómputo utilizados por el objeto. En algunos casos, un hilo que inicia una destrucción de objeto de un objeto puede ser incapaz de completar la destrucción (por ejemplo, un segundo hilo puede mantener un bloqueo de sincronización sobre un recurso de objeto del objeto). Un flujo de trabajo de destrucción de objeto inicia la destrucción de un objeto a favor de un hilo. Si el flujo de trabajo de destrucción de objeto encuentra una falla para completar la destrucción del objeto, una nueva referencia al objeto se inserta en una lista de destrucción de objeto, como la última referencia al objeto. Un segundo hilo descubre y libera la última referencia en la lista de destrucción de objeto, incitando al flujo de trabajo de destrucción de objeto a reasumir el intento para destruir el objeto a favor del segundo hilo.

15

20

25

30

35

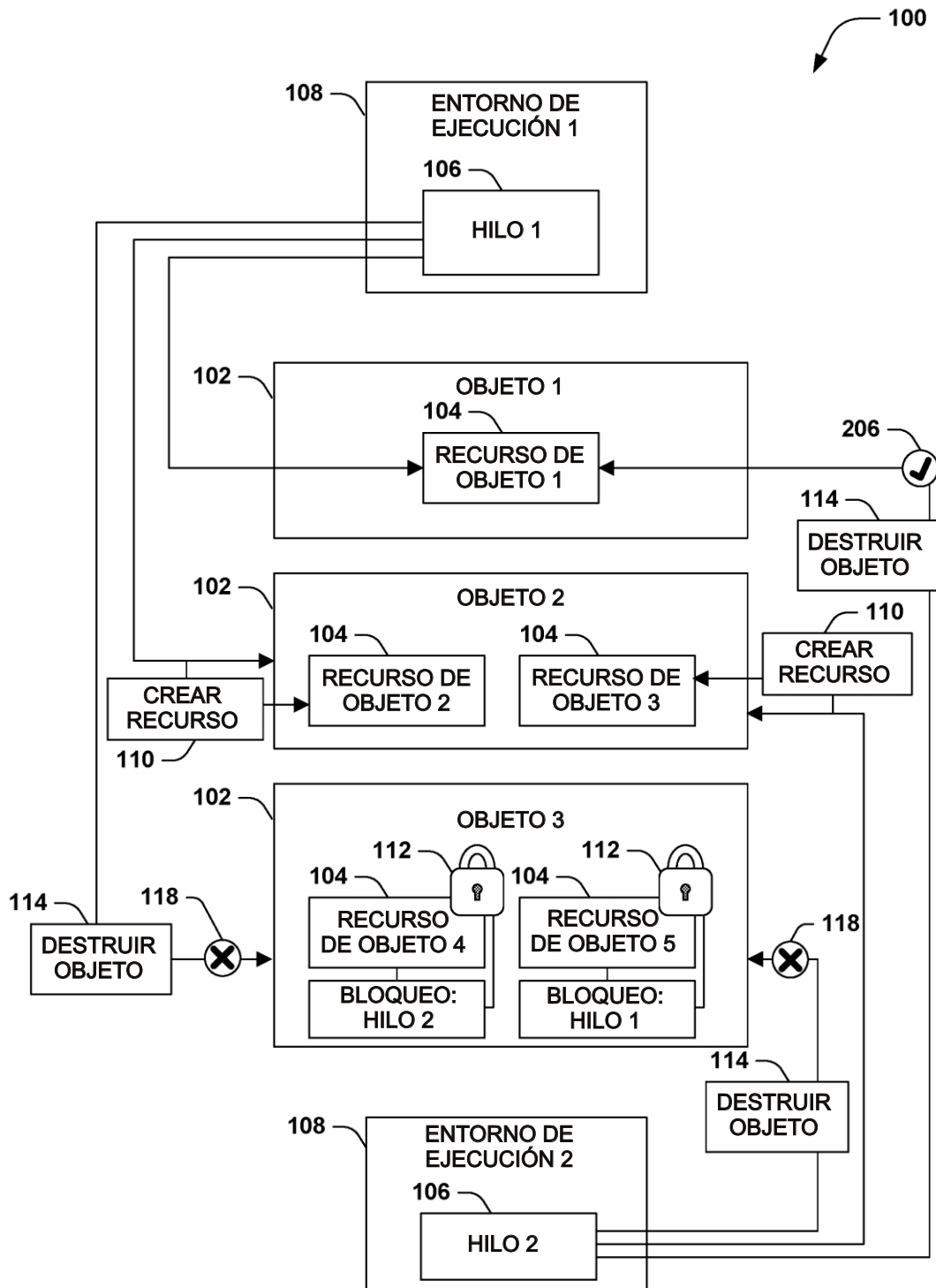


FIG. 1

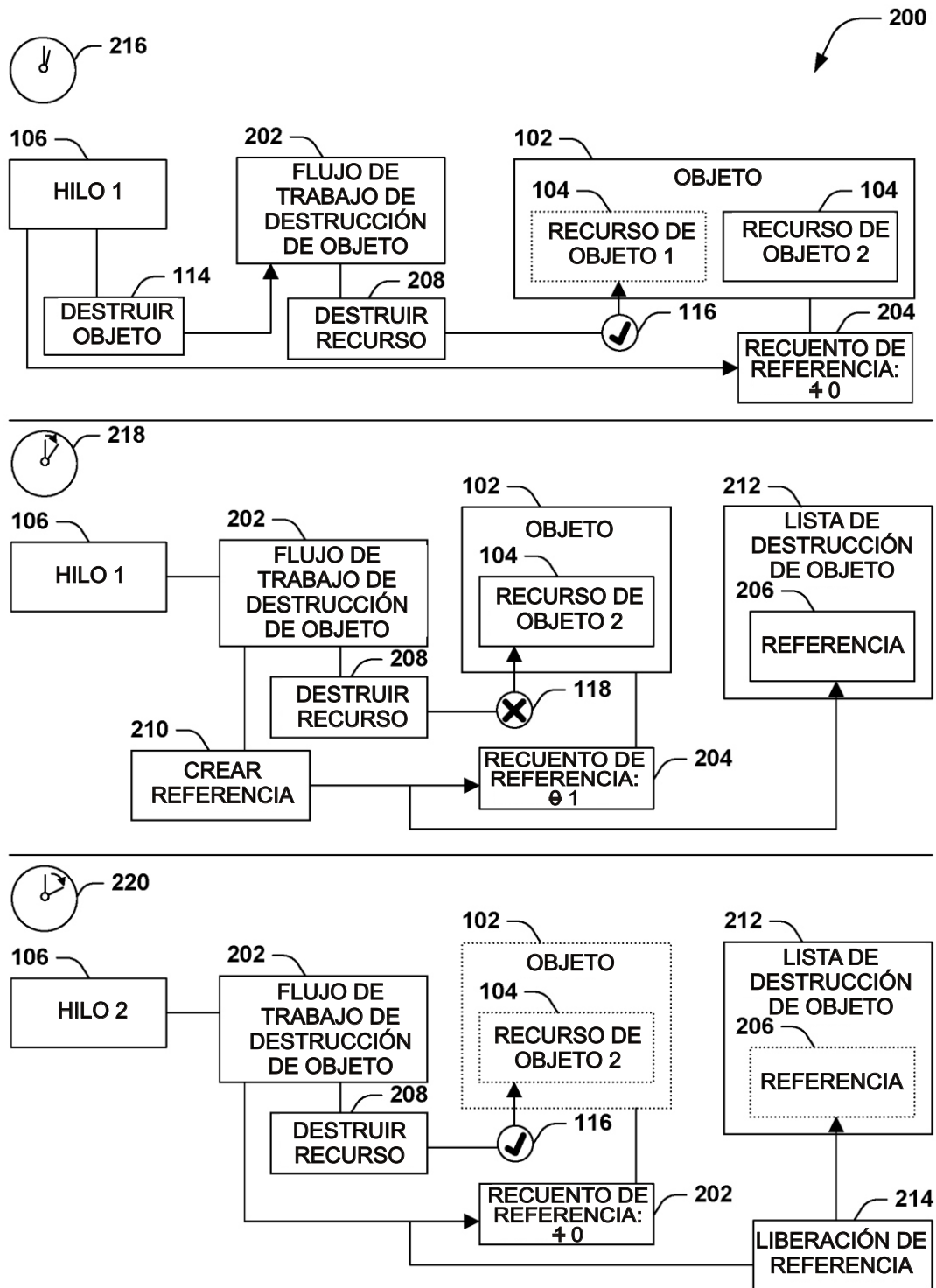


FIG. 2

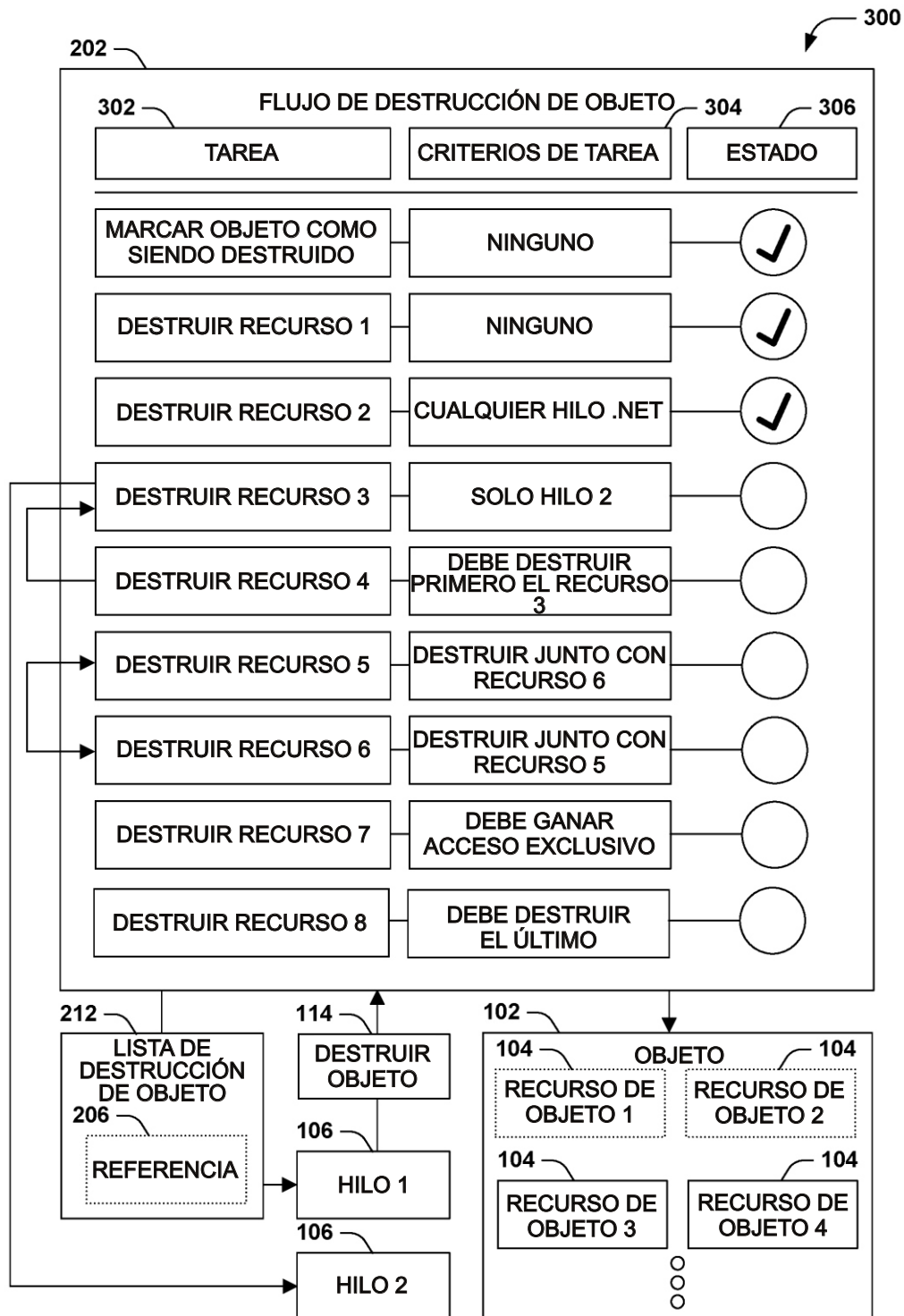


FIG. 3

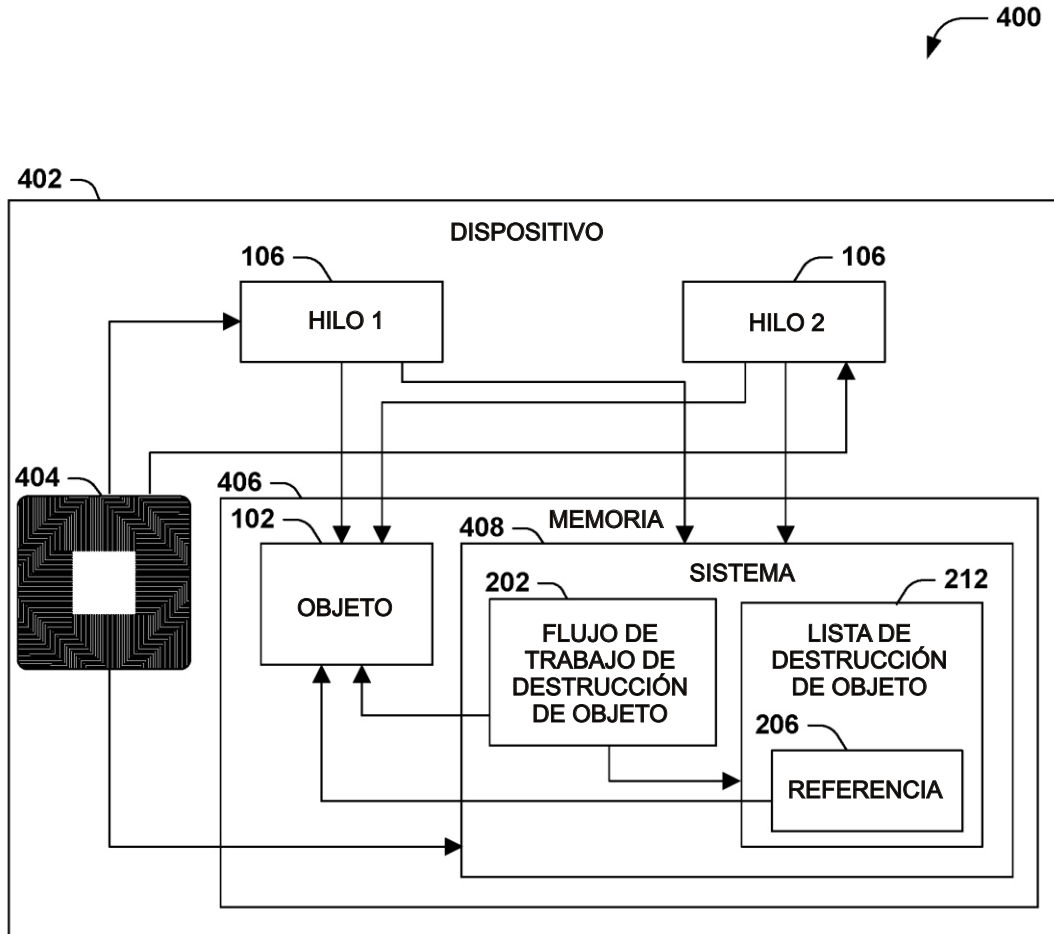


FIG. 4

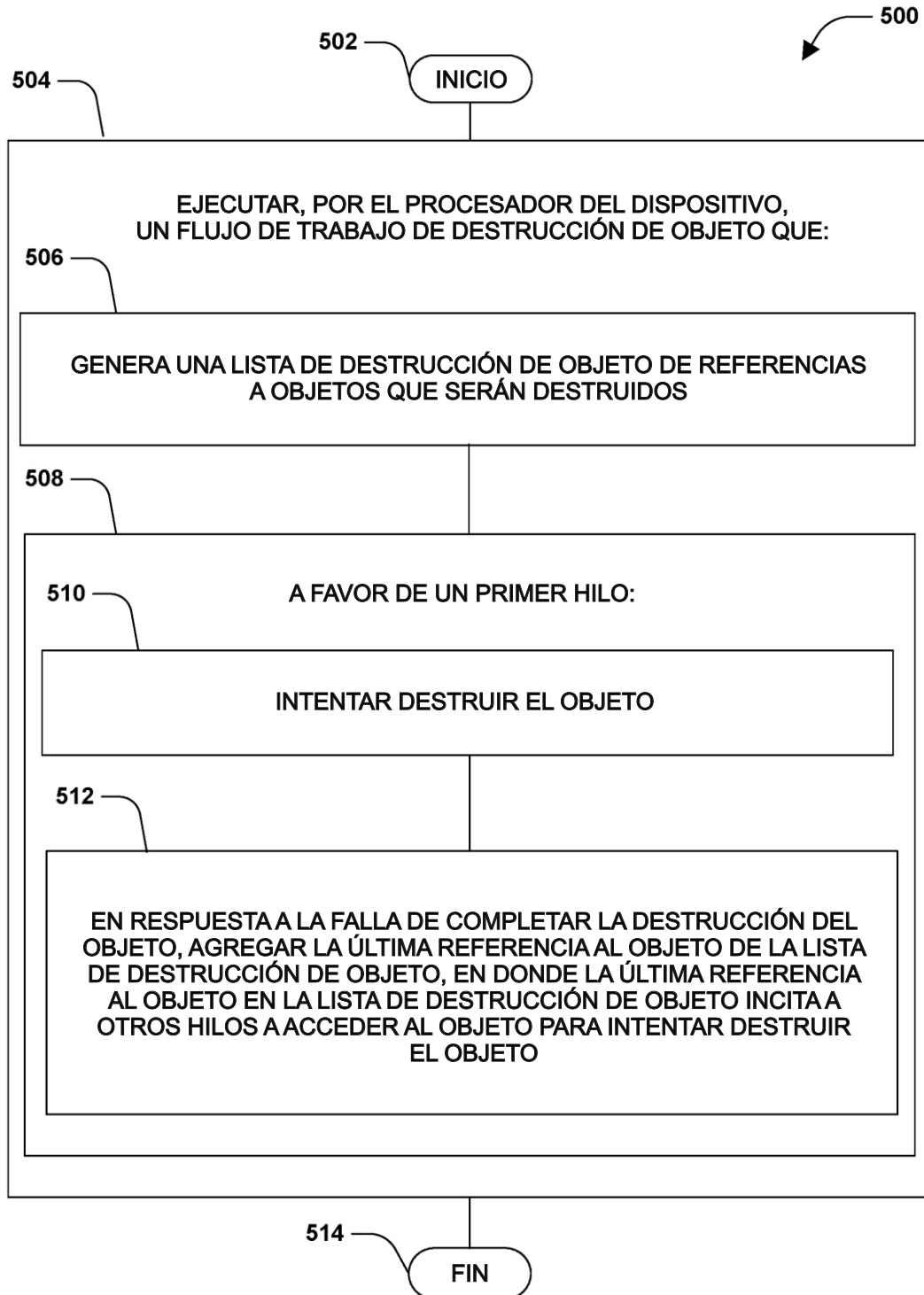


FIG. 5

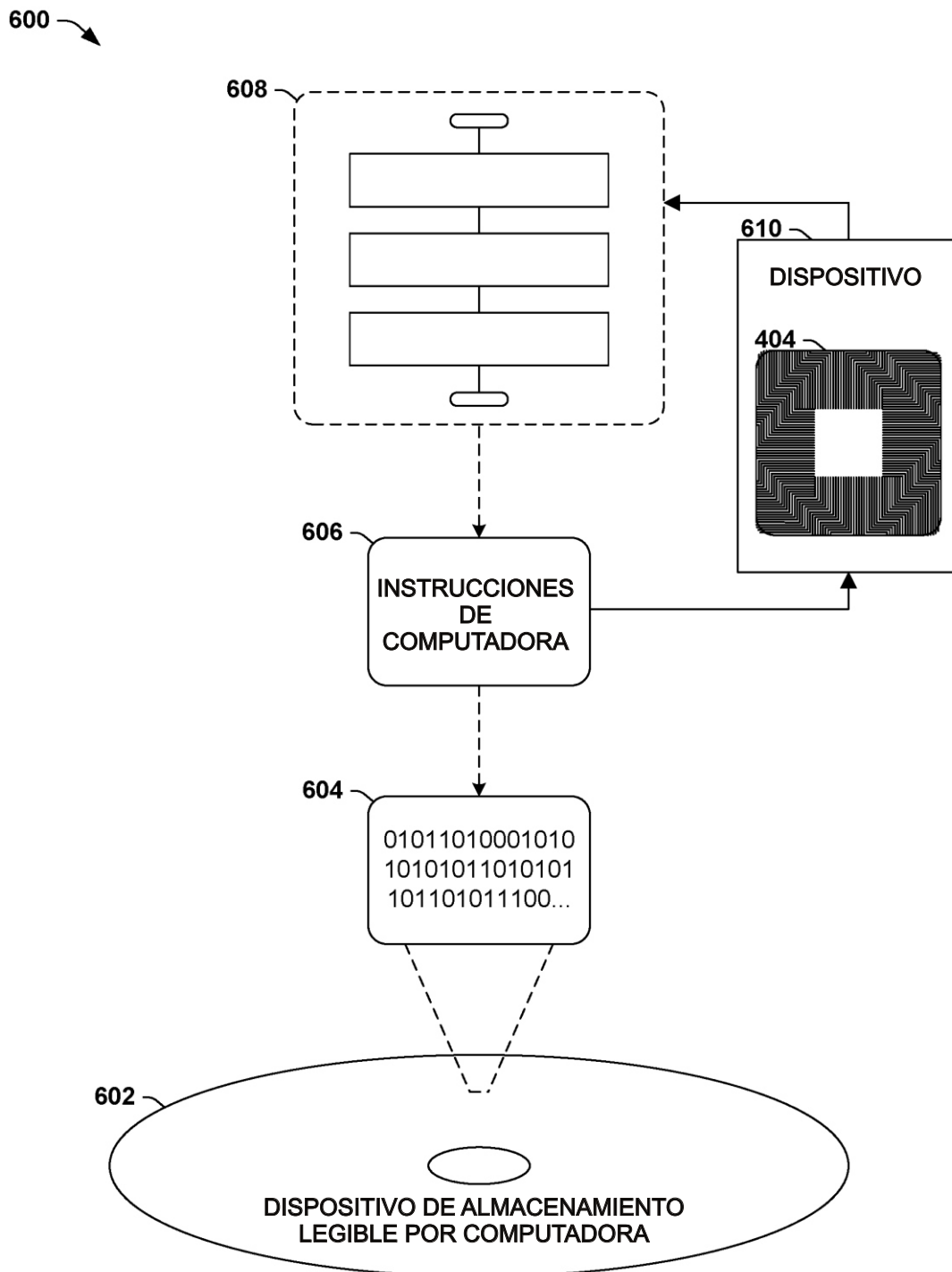


FIG. 6

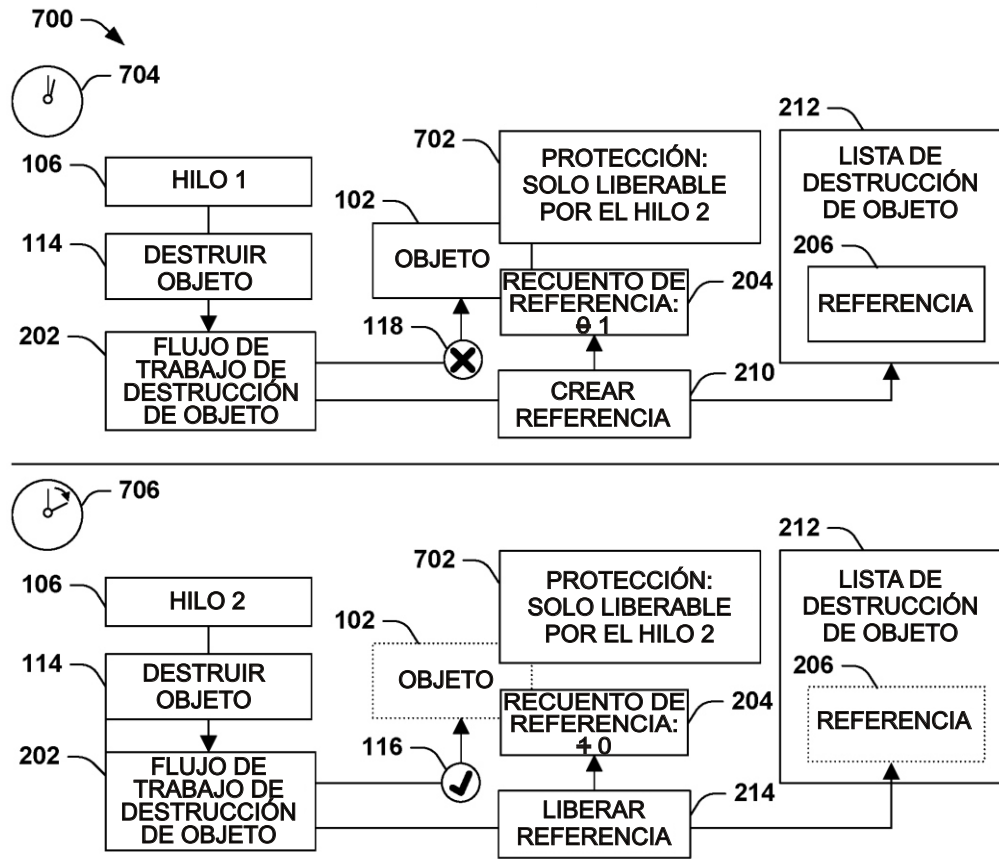


FIG. 7

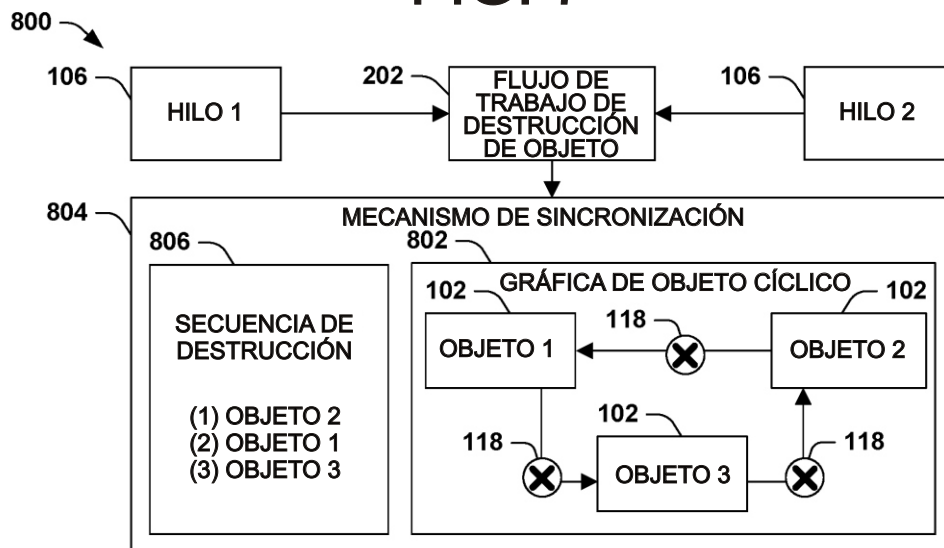


FIG. 8

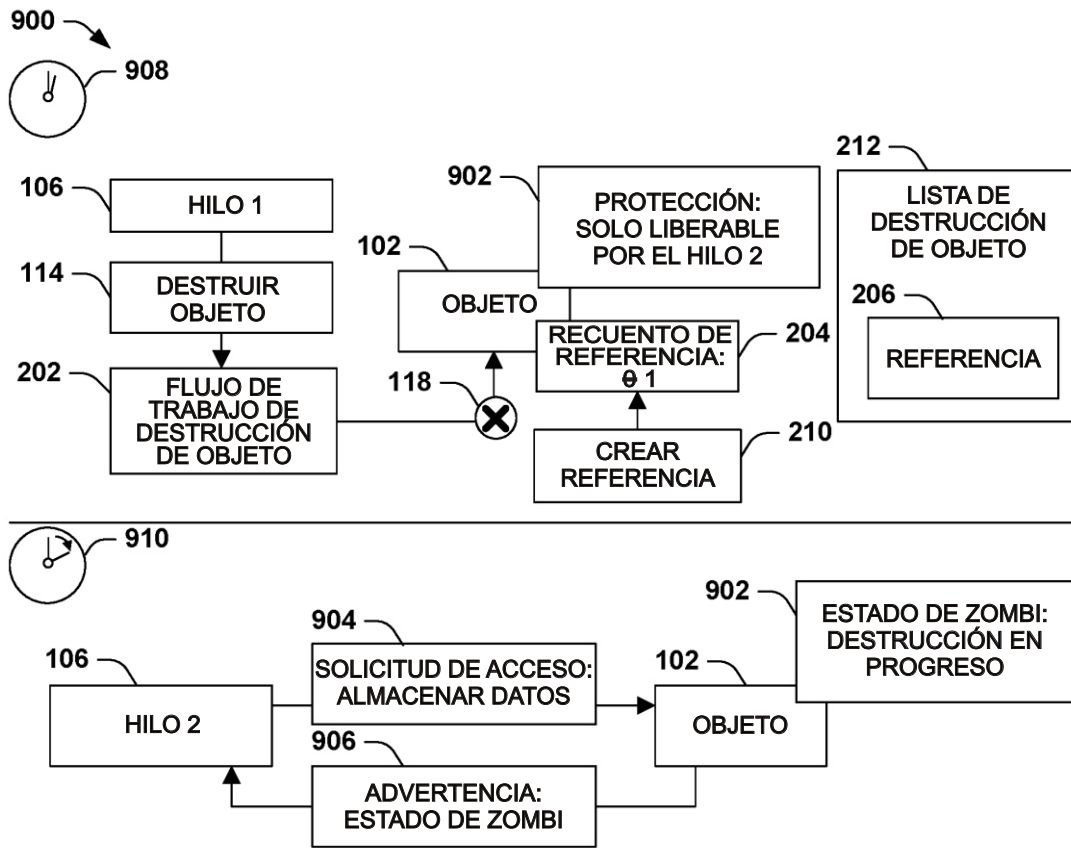


FIG. 9

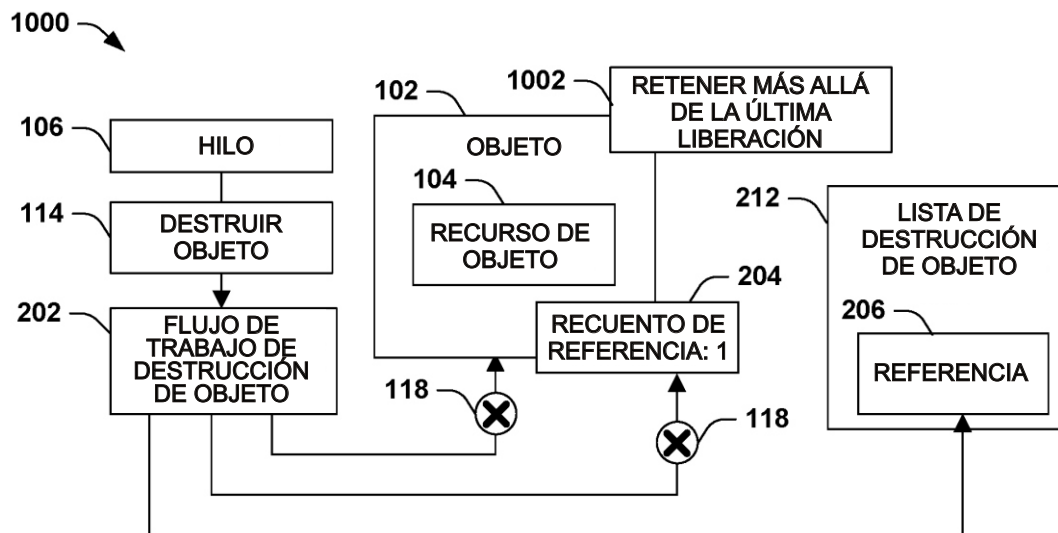


FIG. 10

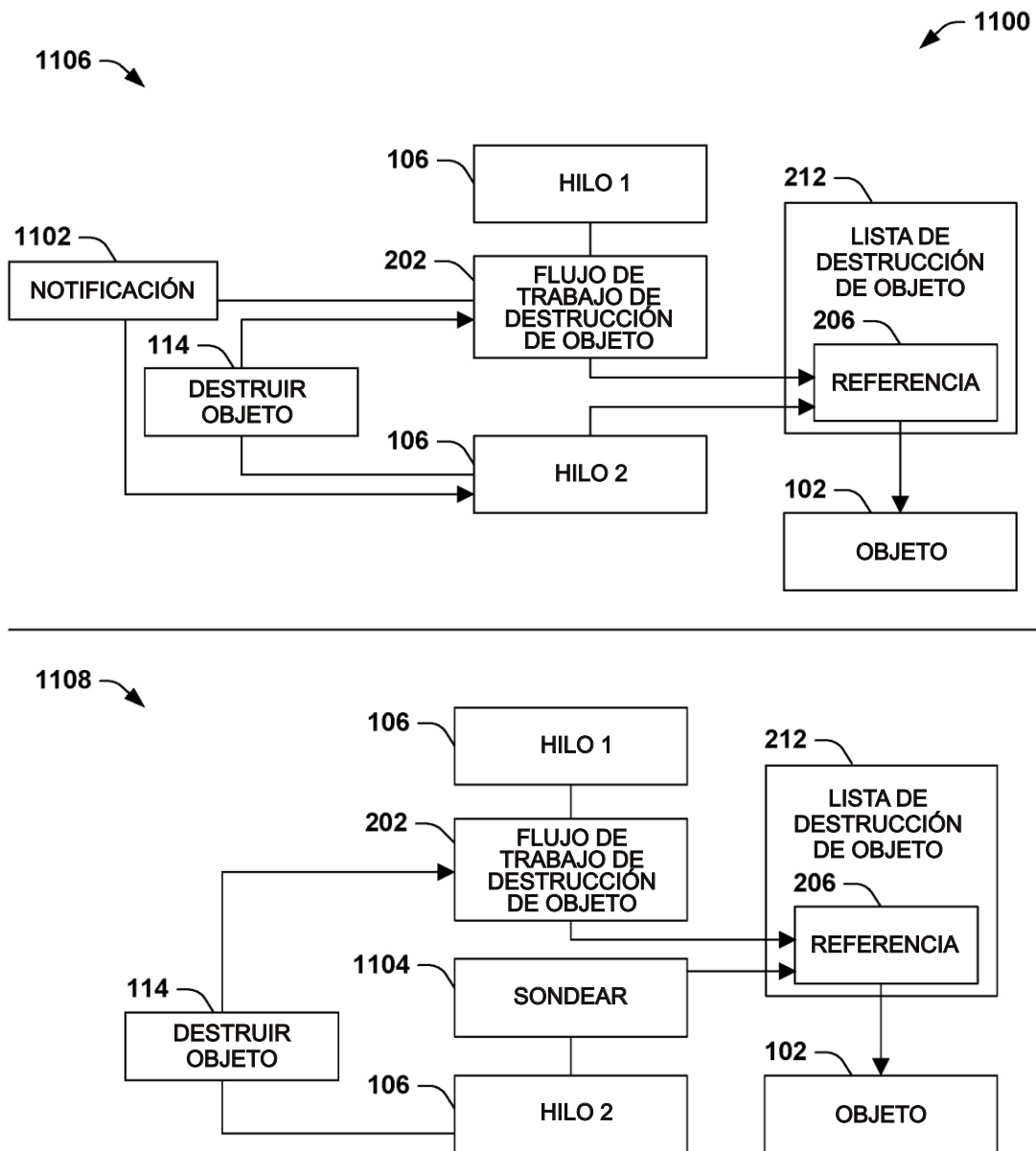


FIG. 11

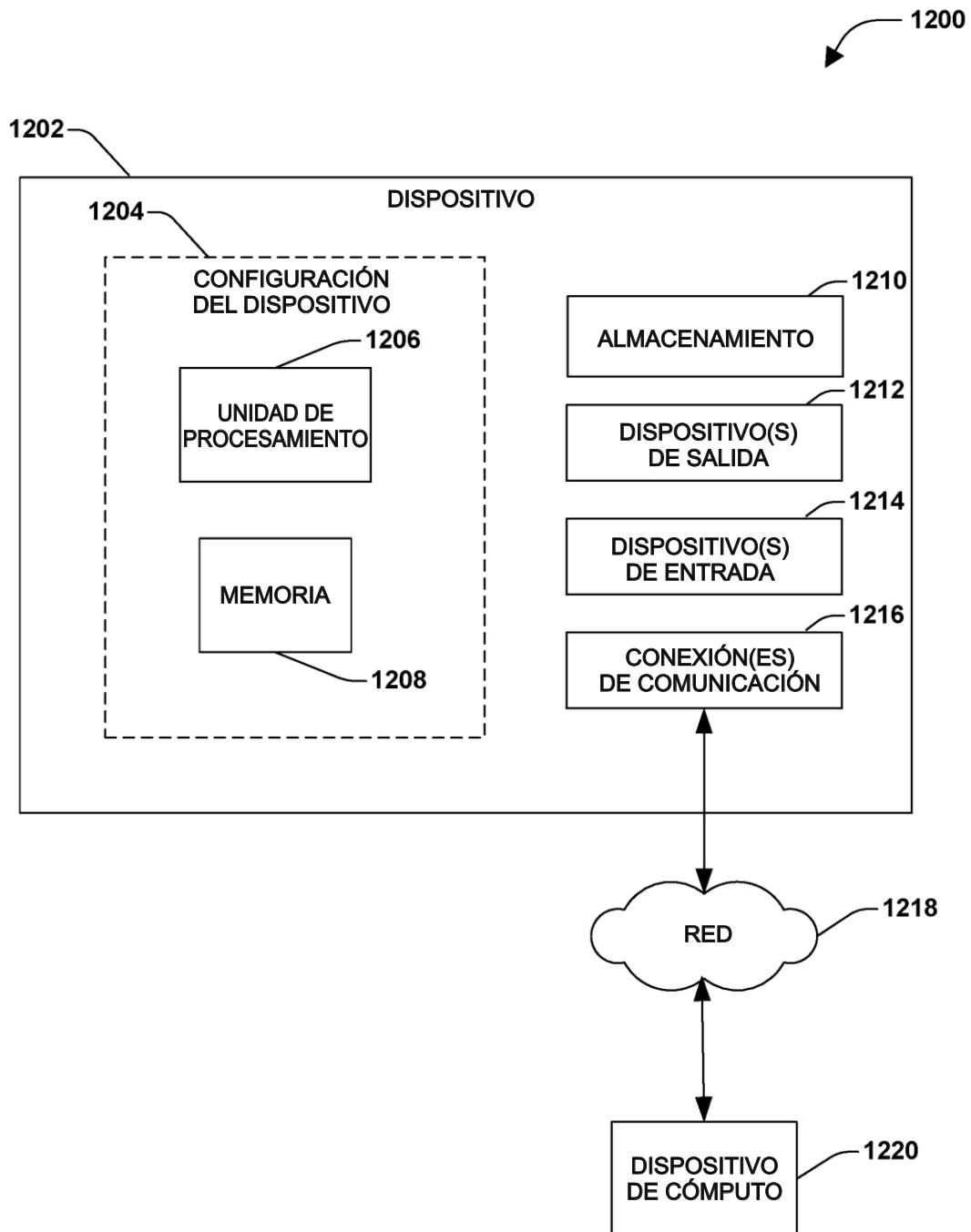


FIG. 12