

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **3 014 611**

51 Int. Cl.:

**G06F 8/33** (2008.01)

**G06F 8/34** (2008.01)

**G06F 8/38** (2008.01)

**G06F 9/451** (2008.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **08.02.2022 E 22155721 (8)**

97 Fecha y número de publicación de la concesión europea: **29.01.2025 EP 4224310**

54 Título: **Métodos y sistemas para la reidentificación en una interfaz de usuario**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**23.04.2025**

73 Titular/es:

**BLUE PRISM LIMITED (100.00%)  
2 Cinnamon Park Crab Lane Crab Lane  
Fearnhead  
Warrington, Cheshire WA2 0XP, GB**

72 Inventor/es:

**SENSOY, MURAT**

74 Agente/Representante:

**IZQUIERDO BLANCO, María Alicia**

ES 3 014 611 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

## DESCRIPCIÓN

Métodos y sistemas para la reidentificación en una interfaz de usuario

### 5 Campo de la divulgación

La presente divulgación se refiere a métodos y sistemas para la reidentificación en una interfaz de usuario. Más específicamente, la presente divulgación se refiere a sistemas y métodos implementados por ordenador para realizar la minería de descripciones para un concepto, y para realizar la comprobación de reidentificación para un elemento de interfaz de usuario en una interfaz de usuario actual.

### Antecedentes

Los procesos informáticos guiados por el hombre son omnipresentes en muchos campos de la tecnología y la empresa. Las interfaces gráficas de usuario (GUI) modernas han demostrado ser de valor inestimable a la hora de permitir a los operadores humanos usar sistemas informáticos para llevar a cabo tareas, a menudo complejas, de procesamiento de datos y/o control de sistemas. Sin embargo, aunque las GUI permiten a menudo a los operadores humanos acostumbrarse rápidamente a realizar nuevas tareas, suponen una barrera importante para cualquier automatización adicional de las tareas.

La automatización tradicional de los flujos de trabajo se dirige a automatizar las tareas que suelen realizar los operadores usando GUI, de tal manera que un sistema informático pueda llevar a cabo la misma tarea sin necesidad de modificar sustancialmente el software subyacente que se está usando para realizar la tarea. Inicialmente, esto requería exponer las interfaces de programación de aplicaciones (API) del software para que se pudiesen diseñar manualmente scripts para que ejecutasen la funcionalidad requerida del software con el fin de realizar la tarea requerida.

La automatización robótica de procesos (RPA) es una forma emergente de tecnología de automatización de procesos empresariales que utiliza robots de software o trabajadores de inteligencia artificial (AI). En las herramientas convencionales de automatización de flujos de trabajo, un desarrollador de software elabora una lista de acciones para automatizar una tarea usando interfaces internas de programas de aplicación. En cambio, los sistemas RPA desarrollan una lista de acciones observando cómo un usuario realiza una tarea en la interfaz gráfica de usuario de la aplicación y, a continuación, realizan la automatización repitiendo esas tareas directamente en la interfaz gráfica de usuario. A continuación, los robots RPA pueden generar los comandos de entrada adecuados para que la interfaz gráfica de usuario haga que el sistema informático lleve a cabo un proceso dado. Esto permite la automatización de procesos, convirtiendo los procesos atendidos en procesos desatendidos. Las ventajas de este enfoque son múltiples e incluyen una mayor escalabilidad permitiendo que varios robots RPA realicen la misma tarea en múltiples sistemas informáticos, junto con una mayor repetibilidad, ya que se reduce o incluso se elimina la posibilidad de error humano en un proceso dado.

La Patente de Estados Unidos US 11.232.170 divulga la detección de elementos gráficos usando una técnica de objetivo unificado combinada de ejecución en serie y en paralelo retardada. La Solicitud de Patente de Estados Unidos US 2022/0012024 divulga descriptores de objetos de interfaz de usuario (UI), bibliotecas de objetos UI, repositorios de objetos UI y navegadores de objetos UI para la automatización de procesos robóticos. La Solicitud de Patente de Estados Unidos US 2020/0401431 divulga un método implementado por ordenador que puede recibir una solicitud de un motor de automatización de procesos robóticos para identificar un elemento de control de interfaz de usuario objetivo en una página web representada por un marco de datos maestro actual.

### Sumario

Los enfoques existentes para la automatización se basan habitualmente en la extracción de identificadores únicos para los elementos de la interfaz de usuario (UI) y la construcción de la automatización sobre la base de estos identificadores. Ejemplos de tales identificadores para aplicaciones basadas en web son las expresiones del lenguaje de rutas del lenguaje de marcado extensible (XPath) y los selectores de las hojas de estilo en cascada (CSS), que también pueden combinarse con varias propiedades del elemento de UI. El identificador único habitualmente es definido manualmente por el desarrollador del RPA durante el modelado de la aplicación o el desarrollo del flujo de trabajo. Sin embargo, estos identificadores únicos pueden verse fácilmente afectados por los cambios en la UI. Por ejemplo, una expresión XPath puede ser invalidada por la adición de un nuevo ancestro (por ejemplo, un contenedor invisible para el usuario) o un cambio de identificadores de un elemento ancestro existente.

Algunos identificadores de elementos de UI son más resistentes a los cambios en la UI y un experto en el dominio puede usar tales identificadores para identificar los elementos de UI. Al hacerlo, el experto puede usar el método de ensayo y error, además del conocimiento del dominio. Los métodos y sistemas propuestos en la presente se inspiran en esto - se basan en la idea de mantener las representaciones de la UI a lo largo del tiempo y explotar estas versiones posiblemente diferentes de la UI para extraer identificadores más robustos y consistentes de manera

automática usando técnicas de minería de patrones. Además, de acuerdo con algunas realizaciones descritas en la presente, el conocimiento del dominio puede capturarse parcialmente como una jerarquía de conceptos, que puede entenderse como una ontología de los elementos de la UI.

5 Las realizaciones descritas en la presente están dirigidas a métodos y sistemas para llevar a cabo la minería de descripciones de un concepto, y para llevar a cabo la comprobación de reidentificación de un elemento de UI en una UI actual.

10 De acuerdo con un aspecto de la presente divulgación, se proporciona un método implementado por ordenador para realizar la minería de descripciones de un concepto. El método comprende: adquirir una pluralidad de descripciones de referencia, en donde cada una de la pluralidad de descripciones de referencia está asociada con un elemento de interfaz de usuario (UI) correspondiente a un tipo de elemento de UI representado por el concepto; generar, basándose en la pluralidad de descripciones de referencia, por lo menos una de una descripción local para el concepto y una descripción global para el concepto, en donde la descripción local para el concepto comprende un componente de descripción que es específico de una UI respectiva, y en donde la descripción global para el concepto comprende un componente de descripción que no es específico de una UI; y almacenar la por lo menos una de la descripción local y la descripción global generadas de tal manera que se asocie con un identificador de concepto correspondiente al concepto.

20 De acuerdo con otro aspecto de la presente divulgación, se proporciona un método implementado por ordenador para realizar una comprobación de reidentificación de un elemento de interfaz de usuario (UI), en una UI actual. El método comprende: recuperar, para el elemento de UI, por lo menos uno de una primera descripción local almacenada y de una primera descripción global almacenada, sobre la base de un identificador del concepto que representa el elemento de UI; y consultar una representación del UI actual usando la por lo menos una primera descripción local almacenada y una primera descripción global almacenada para determinar si puede usarse la descripción respectiva para reidentificar el elemento de UI.

30 De acuerdo con otro aspecto de la presente divulgación, se proporciona un sistema para realizar minería de descripciones para un concepto. El sistema comprende: una unidad de adquisición configurada para adquirir una pluralidad de descripciones de referencia, en donde cada una de la pluralidad de descripciones de referencia está asociada con un elemento de interfaz de usuario (UI) correspondiente a un tipo de elemento de UI representado por el concepto; una unidad de generación configurada para generar, basándose en la pluralidad de descripciones de referencia, por lo menos una de una descripción local para el concepto y una descripción global para el concepto, en donde la descripción local para el concepto comprende un componente de descripción que es específico de una UI respectiva, y en donde la descripción global para el concepto comprende un componente de descripción que no es específico para una UI; y una unidad de almacenamiento configurada para almacenar por lo menos una de la descripción local y la descripción global generadas de tal manera que se asocia con un identificador de concepto correspondiente al concepto.

40 De acuerdo con otro aspecto de la presente divulgación, se proporciona un sistema para realizar la comprobación de reidentificación de un elemento de interfaz de usuario (UI), en una UI actual. El sistema comprende: una unidad de recuperación configurada para recuperar, para el elemento de UI, por lo menos uno de una primera descripción local almacenada y de una primera descripción global almacenada, sobre la base de un identificador del concepto que representa el elemento de UI; y una unidad de consulta configurada para consultar una representación de la UI actual usando la por lo menos una primera descripción local almacenada y la primera descripción global almacenada para determinar si puede usarse la descripción respectiva para reidentificar el elemento de UI.

Las características opcionales se exponen en las reivindicaciones dependientes adjuntas.

50 Estos y otros aspectos de la presente divulgación se apreciarán y comprenderán mejor cuando se consideren conjuntamente con la siguiente descripción y los dibujos que la acompañan. La siguiente descripción, aunque indica varias realizaciones de la divulgación y numerosos detalles específicos de la misma, se ofrece a título de ilustración y no como limitación. Pueden realizarse muchas sustituciones, modificaciones, adiciones o reordenamientos dentro del alcance de la divulgación, y la divulgación incluye todas esas sustituciones, modificaciones, adiciones o reordenamientos.

### Breve descripción de los dibujos

60 Las realizaciones de la presente divulgación se describirán ahora, a modo de ejemplo únicamente, con referencia a los dibujos acompañantes, en los que:

La Figura 1 es un diagrama de flujo de un método implementado por ordenador para realizar la minería de descripciones de un concepto de UI de usuario, de acuerdo con una realización de la presente divulgación; La Figura 2 es un diagrama de flujo de un método para realizar la comprobación de reidentificación de un elemento de UI, en una UI actual, de acuerdo con una realización de la presente divulgación.

La Figura 3 es un diagrama de bloques de un sistema de minería de descripciones de un concepto, de acuerdo con una realización de la presente divulgación;

La Figura 4 es un diagrama de bloques de un sistema para realizar la comprobación de reidentificación de un elemento de UI en una UI actual, de acuerdo con una realización de la presente divulgación; y

La Figura 5 es un diagrama que ilustra un proceso ejemplar para realizar la minería de descripciones de un concepto y realizar la comprobación de reidentificación de un elemento de UI en una UI actual, de acuerdo con una realización de la presente divulgación.

## Descripción detallada

En la siguiente descripción, se exponen numerosos detalles específicos con el fin de proporcionar una comprensión exhaustiva de las realizaciones. Sin embargo, para un experto en la técnica será evidente que no es necesario emplear el detalle específico para poner en práctica las realizaciones. En otros ejemplos, no se han descrito en detalle materiales o métodos bien conocidos para evitar oscurecer la presente divulgación.

La Figura 1 es un diagrama de flujo de un método implementado por ordenador para realizar la minería de descripciones de un concepto de UI (en lo sucesivo denominado simplemente en la presente "concepto"), de acuerdo con una realización de la presente divulgación.

El método 100 comienza en el paso S110 en el que se adquieren una pluralidad de descripciones de referencia. Cada una de la pluralidad de descripciones de referencia se asocia con un elemento de interfaz de usuario (UI) correspondiente a un tipo de elemento de UI representado por el concepto.

En el contexto de la presente realización, una descripción (como una descripción de referencia, una descripción local, y una descripción global) puede ser indicativa de por lo menos una de una ubicación y un atributo de un elemento de UI respectivo o un concepto respectivo. Por ejemplo, en algunas realizaciones, cada una de la pluralidad de descripciones de referencia adquiridas en el paso S110 puede estar en la forma de uno de: un selector de Hojas de Estilo en Cascada (CSS), una expresión de Lenguaje de Ruta de Lenguaje de Marcado Extensible (XPath), y una descripción de imagen y/o audio. Una descripción de imagen puede ser, por ejemplo, una descripción de visión por ordenador (CV).

Por ejemplo, el concepto en cuestión puede representar un elemento de UI del tipo "botón OK". En este ejemplo, cada una de la pluralidad de descripciones de referencia puede ser una descripción para un "botón OK" diferente, por ejemplo de diferentes UI y/o diferentes versiones de una UI. Por ejemplo, la pluralidad de descripciones de referencia pueden ser expresiones XPath de botones OK de diferentes UI y/o diferentes versiones de una UI específica.

En algunas realizaciones, la pluralidad de descripciones de referencia puede comprender por lo menos una descripción de referencia histórica local y una descripción de referencia no local. En estas realizaciones, una descripción de referencia histórica local es específica de una UI actual (por ejemplo, una UI que está siendo usada por un usuario), y una descripción de referencia no local no es específica de la UI actual. Una descripción de referencia histórica local corresponde a un elemento de la UI que forma parte de la UI actual. Una descripción de referencia no local corresponde a un elemento de la UI que no forma parte de la UI actual, por ejemplo, un elemento de la UI que forma parte de una UI similar a la UI actual.

A continuación, en el paso S120, se genera por lo menos una descripción local para el concepto y una descripción global para el concepto basándose en la pluralidad de descripciones de referencia adquiridas en el paso S110. La descripción local para el concepto comprende un componente de descripción que es específico de una UI respectiva, por ejemplo, una UI actual. La descripción global para el concepto comprende un componente de descripción que no es específico de una UI, y por lo tanto la descripción global puede reutilizarse en diferentes UI (por ejemplo, diferentes aplicaciones o páginas web). En algunos casos, en el paso S120 pueden generarse múltiples descripciones locales/globales que corresponden a diferentes formas paralelas de identificar un concepto/elemento de UI respectivo. Por ejemplo, las descripciones generadas pueden incluir una expresión XPath, un selector CSS, una descripción CV, etc. En estos ejemplos, incluso si una forma/tipo de descripción se vuelve inválida, todavía puede usarse otra forma/tipo con el propósito identificar un elemento de UI. Además, en algunas realizaciones, la primera descripción (es decir, la descripción que se volvió inválida) puede actualizarse usando datos del elemento de UI identificado usando la segunda (es decir, la descripción que se usa con el propósito de identificar el elemento de UI).

En algunas realizaciones, generar una descripción local en S120 puede comprender actualizar una descripción local existente para el concepto. Alternativamente, o además, en algunas realizaciones la generación de una descripción global en el paso S120 puede comprender actualizar una descripción global existente para el concepto. Estas descripciones existentes pueden almacenarse en una base de datos (por ejemplo, el almacenamiento "Descripciones de Conceptos de UI" mostrado en la Figura 5).

La generación de las descripciones locales y/o globales puede implicar la realización de minería de patrones

con respecto a por lo menos la pluralidad de descripciones de referencia. En más detalle, en algunas realizaciones, la generación de una descripción local para el concepto en S120 puede comprender la determinación de un componente de descripción común a partir de una pluralidad de descripciones de referencia históricas locales. Alternativamente, o además, generar una descripción global para el concepto en S120 puede comprender determinar un componente de descripción común a partir de una pluralidad de descripciones de referencia no locales. En estas realizaciones, la determinación del componente de descripción común puede realizarse usando un modelo de aprendizaje automático entrenado. Por ejemplo, la determinación del componente de descripción común puede realizarse usando una serie de técnicas de aprendizaje automático como redes neuronales basadas en visión por ordenador, árboles de decisión usando información estructurada de un árbol DOM (por ejemplo, etiquetas, contenido de texto), o métodos de espacio de versiones basados en características de los elementos de la interfaz de usuario.

Posteriormente, en el paso S130, la por lo menos una de la descripción local y la descripción global generadas, generadas en el paso S120 se almacena de tal manera que esté asociada a un identificador de concepto correspondiente al concepto. El identificador de concepto identifica unívocamente el concepto en una colección de conceptos. Por ejemplo, el identificador de concepto puede indicar un orden del tipo de elemento de UI representado por el concepto en una ontología de una pluralidad de diferentes tipos de elementos de UI. La colección de conceptos puede ser una jerarquía de conceptos en la que los conceptos se presentan sobre la base de los tipos de elementos de UI representados por el concepto, por ejemplo, "botón OK" y "botón de inicio de sesión" serían subconceptos incluidos en el concepto más general de "botón de envío" en la jerarquía. La jerarquía de conceptos puede representar el conocimiento del dominio de un experto en el campo de la automatización, más concretamente en la automatización de procesos robóticos.

En algunas realizaciones, el método 100 puede comprender además la adquisición de una entrada de usuario aceptando o rechazando la por lo menos una de la descripción local generada y la descripción global generada. En estas realizaciones, el almacenamiento de la descripción local generada y/o de la descripción global generada en el paso S130 solo puede realizarse tras la adquisición de la entrada del usuario que acepta la descripción local generada o la descripción global generada respectiva.

Como se ha mencionado anteriormente con referencia a S120, en algunas realizaciones la generación de una descripción local para el concepto en S120 puede comprender determinar un componente de descripción común a partir de una pluralidad de descripciones de referencia históricas locales. En estas realizaciones, almacenar la descripción local generada en S130 puede comprender almacenar el componente de descripción común determinado como por lo menos parte de la descripción local.

Además, como se ha mencionado anteriormente con referencia a S120, en algunas realizaciones generar una descripción global para el concepto en S120 puede comprender determinar un componente de descripción común a partir de una pluralidad de descripciones de referencia no locales. En estas realizaciones, almacenar la descripción global generada en S130 puede comprender almacenar el componente de descripción común determinado como por lo menos parte de la descripción global.

Aunque no se muestra en el diagrama de flujo de la Figura 1, en algunas realizaciones el método 100 puede comprender además adquirir, para cada una de la pluralidad de descripciones de referencia adquiridas en el paso S110, una representación de UI correspondiente. En estas realizaciones, la generación de una descripción local para el concepto o una descripción global para el concepto en el paso S120 puede basarse además en una interrelación entre una descripción de referencia y la representación de UI correspondiente.

En algunas realizaciones una representación de UI puede comprender por lo menos una de una representación estructurada de la UI respectiva y una representación no estructurada de la UI respectiva. Por ejemplo, una representación estructurada puede comprender un árbol de modelo de objeto de documento (DOM). Como otro ejemplo, la representación no estructurada de una UI puede comprender una captura de pantalla de la UI respectiva.

Además, aunque no se muestra en el diagrama de flujo de la Figura 1, en algunas realizaciones el método 100 puede comprender además, antes de adquirir una representación de una UI correspondiente para una descripción de referencia respectiva, generar una representación de UI actualizada eliminando contenido de información de una representación de UI original. Por tanto, en estas realizaciones, la representación adquirida puede ser la representación actualizada (en lugar de la representación original). Como ejemplo, la representación de UI original puede comprender un árbol DOM, y el contenido de información (que se elimina) puede comprender nodos de texto y/o nodos de imagen en el árbol DOM. La representación actualizada puede considerarse como una versión de "privacidad garantizada" de la representación de la UI, ya que se ha eliminado de esta versión el contenido de la información que podría causar problemas de privacidad.

En algunas realizaciones, el método 100 puede comprender además, antes de generar por lo menos una descripción local y una descripción global para el concepto en el paso S120, determinar si se cumple una condición de activación predeterminada. En estas realizaciones, la generación de por lo menos una de una descripción local y de una descripción global en el paso S120 sólo puede realizarse si se determina que se cumple la condición de

activación predeterminada. La condición de activación predeterminada puede estar asociada con por lo menos uno de: un intervalo de tiempo predeterminado, el almacenamiento de una nueva instancia de concepto para el concepto, la instancia de concepto comprendiendo una ubicación (por ejemplo, coordenadas espaciales, o un identificador como un selector CSS, o un atributo de objeto Win32) de un elemento de UI respectivo y una representación de una UI respectiva, y una entrada de usuario de instrucción. Una nueva instancia de concepto puede referirse a una instancia de concepto asociada con un elemento de UI que no puede ser identificado por una descripción existente almacenada con respecto al identificador de concepto correspondiente.

Como ejemplo de una condición de activación predeterminada, la generación de las descripciones locales y/o globales puede activarse manualmente mediante la recepción de una entrada de usuario que solicite que se realice la minería de descripciones. Como otro ejemplo, la generación de las descripciones locales y/o globales puede ser activada por una instrucción predefinida para realizar la minería de descripciones para un intervalo de tiempo/frecuencia específicos, por ejemplo, cada lunes.

Aunque no se muestra en la Figura 1, en algunas realizaciones el método 100 puede comprender además: adquirir una entrada de usuario, la entrada de usuario correspondiente a la asignación del identificador de concepto a un primer elemento de UI en una UI respectiva, y almacenar una instancia de concepto para el identificador de concepto. La instancia de concepto puede comprender una representación de la UI respectiva, una descripción del primer elemento de UI, y el identificador de concepto. Por ejemplo, una instancia de concepto para el concepto "botón OK" puede comprender una representación (por ejemplo, un árbol DOM) de la UI que contiene el botón OK, una expresión XPath del botón OK, y el identificador de concepto para "botón OK" (que puede ser el propio "botón OK" o un identificador arbitrario). Por tanto, en el contexto de la presente divulgación cada elemento de la UI puede ser considerado como una instancia de un concepto. Como se ha mencionado anteriormente, el almacenamiento de una (nueva) instancia de concepto puede ser una condición de activación predeterminada para iniciar la generación de por lo menos una descripción local y una descripción global.

La Figura 2 es un diagrama de flujo de un método para realizar la comprobación de reidentificación de un elemento de UI, en una UI actual, de acuerdo con una realización de la presente divulgación. En algunas realizaciones, el elemento de UI puede ser parte de una acción de un flujo de trabajo de automatización. Además, en algunas realizaciones, realizar la comprobación de reidentificación del elemento de UI como se describe a continuación puede formar parte de la ejecución de un flujo de trabajo de automatización.

El método 200 comienza en el paso S210 en el que por lo menos una de una primera descripción local almacenada y una primera descripción global almacenada se recupera para el elemento de UI. Esta recuperación se basa en un identificador de concepto que representa el elemento de UI. Por ejemplo, si el elemento de UI es un "botón OK", puede estar asociado con un identificador de concepto que identifica "botón OK" como el concepto asociado. La primera descripción local y/o la primera descripción global pueden haber sido generadas para el concepto basándose en el método descrito anteriormente con referencia a la Figura 1.

Como se ha mencionado anteriormente, en algunas realizaciones el elemento de UI puede formar parte de una acción de un flujo de trabajo de automatización. En estas realizaciones, antes de recuperar la por lo menos una de una primera descripción local almacenada y una primera descripción global almacenada en el paso S210, el método puede comprender además adquirir el flujo de trabajo de automatización, el flujo de trabajo de automatización incluyendo por lo menos la acción que implica el elemento de UI. En una implementación práctica, el flujo de trabajo de automatización puede incluir una pluralidad de acciones, cada una de las cuales implica una serie de elementos de UI. La adquisición del flujo de trabajo de automatización puede comprender la recepción de una entrada de usuario correspondiente a la acción en el flujo de trabajo de automatización. Por ejemplo, la entrada del usuario puede corresponder a una operación para crear la acción correspondiente en el flujo de trabajo de automatización.

Volviendo a la Figura 2, en el paso S220, se consulta una representación de la UI actual usando la por lo menos una primera descripción local almacenada y una primera descripción global almacenada, para determinar si puede usarse la descripción respectiva para reidentificar el elemento de UI. Por ejemplo, la primera descripción local puede ser una expresión XPath y en el paso S220 puede consultarse un árbol DOM que representa la UI actual para determinar si hay una coincidencia para esta expresión XPath en el árbol DOM. Si se determina que hay una coincidencia, entonces la reidentificación del elemento UI tiene éxito. Por otro lado, si se determina que no hay una coincidencia, entonces la reidentificación del elemento de UI no tiene éxito. En el contexto de la presente divulgación, se entiende que los términos "reidentificación" y "reidentificar" se refieren al proceso de identificar un elemento de UI a partir de una UI (por ejemplo, mediante la búsqueda de una coincidencia en la representación de la UI), o más específicamente a partir de una representación de la UI (por ejemplo, un árbol DOM). Por tanto, no es un prerrequisito que el elemento de UI haya sido "identificado" previamente para que pueda ser "reidentificado" en el paso S220. Además, los términos "reidentificación" y "reidentificar" también pueden referirse a un escenario en el que el elemento de UI no puede identificarse usando una descripción local/global y, por lo tanto, se usa una descripción local/global diferente para realizar una "reidentificación" (por ejemplo, mediante la búsqueda de una coincidencia en la representación de la UI). Esto se explicará con más detalle a continuación con respecto a los escenarios en los que no puede usarse la primera descripción local/global para reidentificar el elemento de UI y se recupera una segunda

descripción local/global para comprobar la reidentificación.

En algunas realizaciones en las que una primera descripción local almacenada y una primera descripción global almacenada están ambas disponibles, el método puede comprender recuperar, en el paso S210, por lo menos la primera descripción local basada en el identificador de concepto, y consultar, en el paso S220, la presentación de la UI actual usando la primera descripción local. De esta manera, el método puede priorizar el uso de la descripción local sobre una descripción global en la comprobación de reidentificación, que puede proporcionar resultados más fiables ya que una descripción local es específica de la UI actual.

En algunas realizaciones, el método puede comprender además, almacenar la representación de la IU actual y la primera descripción local y/o la primera descripción global de tal manera que se asocien con el identificador de concepto, si se determina en el paso S220 que la primera descripción local y/o la primera descripción global pueden usarse para reidentificar el elemento de UI.

Alternativamente, si se determina en el paso S220 que la primera descripción local y/o la primera descripción global no pueden usarse para reidentificar el elemento de UI, el método puede comprender además descartar la primera descripción local y/o la primera descripción global que no pueden usarse para la identificación. Alternativa o adicionalmente, si se determina en el paso S220 que la primera descripción local y/o la primera descripción global no pueden usarse para reidentificar el elemento de UI, el método puede comprender además recuperar por lo menos una de: una segunda descripción local almacenada y una segunda descripción global almacenada basadas en el identificador de concepto, la segunda descripción local siendo diferente de la primera descripción local y la segunda descripción global siendo diferente de la segunda descripción global, y consultar la representación de la UI actual usando la por lo menos una de la segunda descripción local y la segunda descripción global para determinar si la descripción respectiva puede usarse para reidentificar el elemento de UI.

Como otra alternativa más, en algunas realizaciones si se determina que la primera descripción local y/o la primera descripción global no pueden usarse para reidentificar el elemento de UI, el método puede comprender además solicitar a la entrada del usuario que indique al elemento de UI en la UI actual y/o realizar la minería de descripciones para el concepto. La minería de descripciones puede realizarse de acuerdo con el método descrito anteriormente con referencia a la Figura 1.

Se apreciará que, en algunas realizaciones de la presente divulgación, los métodos ilustrados en las Figuras 1 y 2 pueden combinarse en ciertas realizaciones para permitir que se lleven a cabo tanto la minería de descripciones como la comprobación de la reidentificación. En la Figura 5 se ilustra un proceso ejemplar en el que se llevan a cabo tanto la minería de descripciones como la comprobación de la reidentificación, que se explicará con más detalle a continuación haciendo referencia al dibujo.

La Figura 3 es un diagrama de bloques de un sistema para realizar la minería de descripciones para un concepto, de acuerdo con una realización de la presente divulgación.

Como se muestra en la Figura 3, se proporciona un sistema 300 que comprende una unidad de adquisición 310, una unidad de generación 320 y una unidad de almacenamiento 330.

La unidad de adquisición 310 está configurada para adquirir una pluralidad de descripciones de referencia. Cada una de la pluralidad de descripciones de referencia está asociada con un elemento de UI correspondiente a un tipo de elemento de UI representado por el concepto.

Como se ha explicado anteriormente con referencia a la Figura 1, una descripción (como una descripción de referencia, una descripción local y una descripción global) puede ser indicativa de por lo menos una de una ubicación y un atributo de un elemento de UI respectivo o un concepto respectivo. Por ejemplo, en algunas realizaciones, cada una de la pluralidad de descripciones de referencia adquiridas por la unidad de adquisición 310 puede estar en la forma de uno de: un selector de Hojas de Estilo en Cascada (CSS), una expresión de Lenguaje de Ruta de Lenguaje de Marcado Extensible (XPath), y una descripción de imagen y/o audio. Una descripción de imagen puede ser, por ejemplo, una descripción de visión por ordenador (CV).

En algunas realizaciones, la pluralidad de descripciones de referencia puede comprender por lo menos una descripción de referencia histórica local y una descripción de referencia no local. En estas realizaciones, una descripción de referencia histórica local es específica de una UI actual, y una descripción de referencia no local no es específica de la UI actual. Por ejemplo, una descripción de referencia histórica local puede corresponder a un elemento de UI que forma parte de la interfaz de usuario actual. Por ejemplo, una descripción de referencia no local puede corresponder a un elemento de la UI que no es parte de la UI actual, pero que es similar a la UI actual.

La unidad de generación 320 está configurada para generar por lo menos una de una descripción local para el concepto y una descripción global para el concepto sobre la base de la pluralidad de descripciones de referencia adquiridas por la unidad de adquisición 310. La descripción local para el concepto comprende un componente de

descripción que es específico de la interfaz de usuario (UI) respectiva. La descripción local para el concepto comprende un componente de descripción que es específico a una interfaz de usuario (UI) respectiva. La descripción global para el concepto comprende un componente de descripción que no es específico de una UI, y por lo tanto la descripción global puede ser reutilizada en diferentes UI (por ejemplo, diferentes aplicaciones o páginas web). La generación de las descripciones locales y/o globales puede implicar la realización de minería de patrones con respecto a por lo menos la pluralidad de descripciones de referencia. En algunos casos, múltiples las descripciones locales/globales que corresponden a diferentes maneras paralelas para identificar un concepto/elemento de UI respectivo pueden ser generadas por la unidad de generación 320. Por tanto, en algunas realizaciones, si una forma/tipo de descripción se vuelve inválida, en su lugar puede usarse otra forma/tipo de descripción para identificar un elemento de UI. Además, la descripción que se ha vuelto inválida puede ser actualizada usando datos del elemento de UI identificado usando la otra descripción.

En algunas realizaciones, la unidad de generación 320 puede estar configurada para generar una descripción local actualizando una descripción local existente para el concepto. Alternativamente, o además, en algunas realizaciones la unidad de generación 320 puede estar configurada para generar una descripción global actualizando una descripción global existente para el concepto. Estas descripciones existentes pueden almacenarse en un almacén de Descripciones de Conceptos de UI (por ejemplo, el almacén de "Descripciones de Conceptos de UI" mostrado en la Figura 5).

La unidad de generación 320 puede estar configurada para generar una descripción local para el concepto determinando un componente de descripción común a partir de una pluralidad de descripciones de referencia históricas locales. Alternativamente, o además, la unidad de generación 320 puede estar configurada para generar una descripción global para el concepto determinando un componente de descripción común a partir de una pluralidad de descripciones de referencia no locales. Además, en estas realizaciones, la unidad de generación 320 puede estar configurada para determinar el componente de descripción común usando un modelo de aprendizaje automático entrenado. Por ejemplo, la determinación del componente de descripción común puede realizarse usando una serie de técnicas de aprendizaje automático como redes neuronales basadas en visión por ordenador, árboles de decisión usando información estructurada de un árbol DOM (por ejemplo, etiquetas, contenido de texto), o métodos de espacio de versiones basados en características de los elementos de la interfaz de usuario.

La unidad de almacenamiento 330 está configurada para almacenar por lo menos una de la descripción local y la descripción global generadas, de tal manera que se asocie con un identificador de concepto correspondiente al concepto. En algunas realizaciones, el identificador de concepto puede identificar unívocamente el concepto en una colección de conceptos. Por ejemplo, el identificador de concepto puede indicar un orden del tipo de elemento de UI representado por el concepto en una ontología de una pluralidad de diferentes tipos de elementos de UI. La colección de conceptos puede ser una jerarquía de conceptos en la que los conceptos se presentan sobre la base de los tipos de elementos de UI representados por el concepto, por ejemplo, "botón OK" y "botón de inicio de sesión" serían subconceptos incluidos en el concepto más general de "botón de envío" en la jerarquía. La jerarquía de conceptos puede representar el conocimiento del dominio de un experto en el campo de la automatización, más concretamente en la automatización de procesos robóticos.

En algunas realizaciones, la unidad de adquisición 310 puede estar configurada además para adquirir una entrada de usuario aceptando o rechazando la por lo menos una de las descripción local generada y la descripción global generada. En estas realizaciones, la unidad de almacenamiento 330 está configurada para almacenar únicamente la descripción local generada y/o la descripción global generada después de que la unidad de adquisición 310 adquiera la entrada del usuario aceptando la descripción local generada o descripción global generada respectiva.

Como se ha mencionado anteriormente, en algunas realizaciones la unidad de generación 320 puede estar configurada para generar una descripción local para el concepto determinando un componente de descripción común a partir de una pluralidad de descripciones de referencia históricas locales. En estas realizaciones, la unidad de almacenamiento 330 puede estar configurada para almacenar la descripción local generada almacenando el componente de descripción común determinado como por lo menos parte de la descripción local.

Además, como se ha mencionado anteriormente, en algunas realizaciones la unidad de generación 320 puede estar configurada para generar una descripción global para el concepto determinando un componente de descripción común a partir de una pluralidad de descripciones de referencia no locales. En estas realizaciones, la unidad de almacenamiento 330 puede estar configurada para almacenar la descripción global generada almacenando el componente de descripción común determinado como por lo menos parte de la descripción global.

En algunas realizaciones, la unidad de adquisición 310 puede estar configurada además para adquirir, para cada una de la pluralidad de descripciones de referencia, una representación de UI correspondiente. En estas realizaciones, la unidad de generación 320 puede estar configurada para generar una descripción local para el concepto o una descripción global para el concepto basándose además en una interrelación entre una descripción de referencia y la representación de UI correspondiente.



Una representación de UI en algunas realizaciones puede comprender por lo menos una de una representación estructurada de la UI respectiva y una representación no estructurada de la UI respectiva. Por ejemplo, una representación estructurada puede comprender un árbol de modelo de objeto de documento (DOM). Como otro ejemplo, la representación no estructurada de una UI puede comprender una captura de pantalla de la UI respectiva.

En algunas realizaciones, la unidad de generación 320 puede estar configurada para generar una representación de UI actualizada eliminando contenido de información de una representación de UI original. Esta operación puede realizarse antes de que la unidad de adquisición 310 adquiera una representación de una UI correspondiente para una descripción de referencia respectiva. En estas realizaciones, la representación adquirida puede ser la representación actualizada. Por ejemplo, la representación de UI original puede comprender un árbol DOM, y el contenido de información (que se elimina) puede comprender nodos de texto y/o nodos de imagen en el árbol DOM. La representación actualizada puede ser considerada como una versión de "privacidad garantizada" de la representación de la UI, ya que el contenido de la información que podría provocar problemas de privacidad ha sido eliminado de esta versión.

Aunque no se muestra en la Figura 3, en algunas realizaciones el sistema 300 puede comprender además una unidad de determinación configurada para determinar si se cumple una condición de activación predeterminada, antes de que la unidad de generación 320 genere por lo menos una descripción local y una descripción global para el concepto. En estas realizaciones, la unidad de generación 320 puede configurarse de tal manera que la generación de por lo menos una descripción local y una descripción global sólo se realice si la unidad de determinación determina que se ha cumplido la condición de activación predeterminada. La condición de activación predeterminada puede estar asociada con por lo menos uno de: un intervalo de tiempo predeterminado, el almacenamiento de una nueva instancia de concepto para el concepto, la instancia de concepto que comprende una ubicación (por ejemplo, coordenadas espaciales) de un elemento de UI respectivo y una representación de una interfaz de usuario respectiva, y una entrada de usuario de instrucción. Una nueva instancia de concepto puede referirse a una instancia de concepto asociada con un elemento de UI que no puede ser identificado por una descripción existente almacenada con respecto al identificador de concepto correspondiente.

Como ejemplo de una condición de activación predeterminada, la generación de las descripciones locales y/o globales por la unidad de generación 3120 puede activarse manualmente mediante por medio de la recepción de una entrada de usuario que solicite que se realice la minería de descripciones. Como otro ejemplo, la generación de las descripciones locales y/o globales puede ser activada por una instrucción predefinida para realizar la minería de descripciones para un intervalo de tiempo/frecuencia específicos, por ejemplo, cada lunes.

En algunas realizaciones, la unidad de adquisición 310 puede estar configurada además para adquirir una entrada de usuario, la entrada de usuario correspondiente a la asignación del identificador de concepto a un primer elemento de UI en una UI respectiva. Además, en estas realizaciones, la unidad de almacenamiento 330 puede estar configurada además para almacenar una instancia de concepto para el identificador de concepto, la instancia de concepto comprendiendo: una representación de la UI respectiva, una descripción del primer elemento de la UI, y el identificador de concepto. Por ejemplo, una instancia de concepto para el concepto "botón OK" puede comprender una representación (por ejemplo, un árbol DOM) de la UI que contiene el botón OK, una expresión XPath del botón OK, y el identificador de concepto para "botón OK" (que puede ser el propio "botón OK" o un identificador arbitrario). Por tanto, en el contexto de la presente divulgación cada elemento de UI puede ser considerado como una instancia de un concepto. Como se ha mencionado anteriormente, el almacenamiento de una (nueva) instancia de concepto puede ser una condición de activación predeterminada para iniciar la generación de por lo menos una descripción local y una descripción global.

Aunque no se ilustra en la Figura 3, en algunas realizaciones los componentes que incluyen la unidad de adquisición 310, la unidad de generación 320, y la unidad de almacenamiento 330 pueden implementarse como una combinación de un circuito de procesamiento y una memoria en el sistema 300.

La Figura 4 es un diagrama de bloques de un sistema para realizar la comprobación de reidentificación de un elemento de UI en una UI actual, de acuerdo con una realización de la presente divulgación. En algunas realizaciones, el elemento de UI puede formar parte de una acción de un flujo de trabajo de automatización.

Tal como se muestra en la Figura 4, se proporciona un sistema 400 que comprende una unidad de recuperación 410 y una unidad de consulta 420. La unidad de recuperación 410 está configurada para recuperar, para el elemento de UI, por lo menos una de una primera descripción local almacenada y una primera descripción global almacenada, basándose en un identificador de concepto que representa el elemento de UI. Esta recuperación se basa en un identificador de concepto que representa el elemento de UI. Por ejemplo, si el elemento de UI es un "botón OK", puede estar asociado con un identificador de concepto que identifica "botón OK" como el concepto asociado. La primera descripción local y/o la primera descripción global pueden haber sido generadas para el concepto basándose en el método descrito anteriormente con referencia a la Figura 1 o pueden haber sido generadas por el sistema 300 de la manera descrita con referencia a la Figura 3.

Como se ha mencionado anteriormente, en algunas realizaciones el elemento de UI puede ser parte de una acción de un flujo de trabajo de automatización. En estas realizaciones, el sistema 400 puede comprender además una unidad de adquisición (no ilustrada en la Figura 4) configurada para adquirir el flujo de trabajo de automatización antes de que la unidad de recuperación 410 recupere la por lo menos una de una primera descripción local almacenada y una primera descripción global almacenada, el flujo de trabajo de automatización incluyendo por lo menos la acción que implica el elemento de UI. En una implementación práctica, el flujo de trabajo de automatización puede incluir una pluralidad de acciones, cada una de las cuales implica una serie de elementos de UI. La unidad de adquisición puede estar configurada para adquirir el flujo de trabajo de automatización mediante la recepción de una entrada de usuario correspondiente a la acción en el flujo de trabajo de automatización. Por ejemplo, la entrada del usuario puede corresponder a una operación para crear la acción correspondiente en el flujo de trabajo de automatización.

La unidad de consulta 420 está configurada para consultar una representación de la UI actual usando la por lo menos una primera descripción local almacenada y la primera descripción global almacenada, para determinar si la descripción respectiva puede usarse para reidentificar el elemento de UI. Por ejemplo, la descripción local puede ser una expresión XPath y la unidad de consulta 420 puede estar configurada para consultar un árbol DOM que represente la UI actual para determinar si existe una coincidencia para esta expresión XPath en el árbol DOM. Si se determina que hay una coincidencia, entonces la reidentificación del elemento de UI ha tenido éxito. Por otro lado, si se determina que no hay una coincidencia, entonces la reidentificación del elemento de UI no ha tenido éxito. En algunas realizaciones, la realización de la comprobación de reidentificación del elemento de UI por parte de la unidad de consulta 420 puede considerarse parte de la ejecución de un flujo de trabajo de automatización.

Como se ha explicado anteriormente con referencia a la Figura 2, los términos "reidentificación" y "reidentificar" pueden referirse al proceso de identificar un elemento de UI a partir de una UI, o más específicamente a partir de una presentación de UI, sin que sea necesario que el elemento de UI se haya "identificado" anteriormente. Además, los términos "reidentificación" y "reidentificar" también pueden referirse a un escenario en el que el elemento de UI no puede identificarse usando una descripción local/global y, por lo tanto, se usa una descripción local/global diferente para realizar la "reidentificación".

En algunas realizaciones en las que una primera descripción local almacenada y una primera descripción global almacenada están ambas disponibles, la unidad de recuperación 410 puede estar configurada para recuperar por lo menos la primera descripción local basándose en el identificador de concepto. Además, la unidad de consulta 420 puede estar configurada para consultar la presentación de la UI actual usando la primera descripción local. De esta manera, el sistema 400 puede priorizar el uso de una descripción local sobre una descripción global en la comprobación de reidentificación, que puede proporcionar resultados más fiables, ya que una descripción local es específica de la UI actual.

En algunas realizaciones, el sistema 400 puede comprender además una unidad de almacenamiento configurada para almacenar una representación de la UI actual y la primera descripción local y/o la primera descripción global de tal manera que estén asociadas con el identificador de concepto, si la unidad de consulta 420 determina que pueden usarse la primera descripción local y/o la primera descripción global para reidentificar el elemento de UI.

Alternativamente, el sistema 400 puede comprender además una unidad de descarte configurada para descartar la primera descripción local y/o la primera descripción global si la unidad de consulta 420 determina que la primera descripción local y/o la primera descripción global no pueden usarse para reidentificar el elemento de UI. Además, alternativa o adicionalmente, si la unidad de consulta 420 determina que la primera descripción local y/o la primera descripción global no pueden usarse para reidentificar el elemento de UI, la unidad de recuperación 410 puede configurarse además para recuperar por lo menos una de: una segunda descripción local almacenada y una segunda descripción global almacenada sobre la base del identificador de concepto, la segunda descripción local siendo diferente de la primera descripción local y la segunda descripción global siendo diferente de la segunda descripción global. Además, la unidad de consulta 420 puede estar configurada además para consultar la representación de la UI actual usando la por lo menos una de la segunda descripción local y la segunda descripción global para determinar si puede usarse la descripción respectiva para reidentificar el elemento de la UI.

Como otra alternativa más, en algunas realizaciones el sistema 400 puede comprender además una unidad de solicitud configurada para solicitar la entrada del usuario para indicar el elemento de UI en la UI actual y/o realizar la minería de descripciones para el concepto, si la unidad de consulta 420 determina que la primera descripción local y/o la primera descripción global no pueden usarse para reidentificar el elemento de UI. La minería de descripciones puede realizarse de acuerdo con el método descrito anteriormente con referencia a la Figura 1 y/o realizarse por el sistema 300 como se ha descrito anteriormente con referencia a la Figura 3.

Aunque no se ilustra en la Figura 4, en algunas realizaciones, los componentes que incluyen la unidad de recuperación 410 y la unidad de consulta 420 pueden implementarse como una combinación de un circuito de procesamiento y una memoria en el sistema 400.

Se apreciará que en algunas realizaciones de la presente divulgación, las funciones de los sistemas 300 y

400 ilustrados en las Figuras 3 y 4 pueden combinarse en ciertas realizaciones como un único sistema para permitir que dicho sistema lleve a cabo tanto la minería de descripciones como la comprobación de la reidentificación.

La Figura 5 es un diagrama que ilustra un proceso ejemplar para realizar la minería de descripciones de un concepto y realizar la comprobación de reidentificación de un elemento de UI en una UI actual, de acuerdo con una realización de la presente divulgación.

Como se muestra en la Figura 5, el proceso se ilustra mediante una serie de pasos, numerados de (i) a (xi) en el diagrama para facilitar la referencia. Estos pasos se explican con más detalle a continuación con referencia a un usuario, que puede ser un desarrollador de automatización que usa un editor RPA para construir flujos de trabajo. En el contexto de la presente realización, el usuario está construyendo un flujo de trabajo con respecto a una UI de aplicación abierta usando elementos de UI específicos en la UI de aplicación.

(i) El usuario resalta el elemento de la UI de interés. En el ejemplo mostrado en la Figura 5, se trata del botón OK de la UI de la aplicación.

(ii) El editor RPA o un sistema (por ejemplo, el sistema 300 de la Figura 3) captura el elemento resaltado. En otras palabras, el elemento resaltado está ubicado dentro de la UI de la aplicación actual.

(iii) Como se ha explicado anteriormente con referencia a la Figura 1, por ejemplo, los elementos de la UI pueden considerarse instancias o ejemplos de conceptos. Por ejemplo, el botón OK en la UI actual, como se muestra en la Figura 5, puede considerarse una instancia del concepto "botón OK" o una instancia del concepto "botón de inicio de sesión", que a su vez pueden considerarse ambos un subconcepto del concepto "botón de envío". El usuario puede crear estos conceptos y colocarlos en una posición adecuada dentro de una jerarquía de conceptos, antes de resaltar las instancias respectivas en las UI. Como alternativa, el usuario puede usar directamente las jerarquías de conceptos existentes.

(iv) Una representación de la UI completa y la ubicación del elemento de UI resaltado dentro de la UI se almacenan junto con el identificador del concepto (que también puede denominarse "ID del concepto") como una instancia en el almacenamiento "Representación de la UI y objetos resaltados". La representación de la UI puede ser no estructurada (por ejemplo, una captura de pantalla de la UI), o puede ser estructurada (por ejemplo, un árbol DOM), o ambas. La ubicación del elemento de interés dentro de la representación de la UI puede ser en forma de coordenadas espaciales o de un identificador que la identifica de manera única, como un selector CSS o una expresión XPath. Si hay problemas de privacidad, puede usarse una representación estructurada de la interfaz de usuario, como DOM, pero eliminando todo el contenido informativo (por ejemplo, los nodos de texto e imágenes). La siguiente tabla proporciona ejemplos de la información almacenada para instancias bajo el concepto de "botón OK". Para cada aplicación o sitio web, las instancias pueden almacenarse por separado para cada concepto.

Tabla 1 - Ejemplos de instancias almacenadas para el concepto "Botón OK"

Instancias de concepto Botón OK		
www.sap.co.uk/login	Representación UI	Ubicación de elemento
	<xml> ....</xml>	/body/div/span
	<xml> ....</xml>	/body/div/div/span
	<xml> ....</xml>	/body/div/div/span/span
app inicio sesión SAP [windows]	Representación UI	Ubicación de elemento
	<xml> ... .</xml>	[//.../]
	<xml> ... .</xml>	[//.../]
	<xml> ... .</xml>	[//.../]

(v) En este ejemplo, el almacenamiento de una nueva instancia de concepto en la base de datos "Representaciones de UI y objetos resaltados" puede activar la minería de descripciones por parte de un Minero de descripciones de conceptos de UI (que en la presente puede denominarse simplemente "minero de descripciones").

(vi) El minero de descripciones crea y/o actualiza descripciones locales y globales para los conceptos relevantes. Por ejemplo, si el botón OK resaltado en el paso (i) pertenece al sitio web de SAP, el minero de descripciones puede crear y/o actualizar la descripción o descripciones para el concepto "botón OK" para este sitio web específico, así como la descripción o descripciones globales que pueden usarse como descripción o descripciones por defecto para el concepto "botón OK" en otros sitios web y aplicaciones. Las descripciones globales pueden generarse usando toda la información almacenada en diferentes sitios web y aplicaciones para este concepto

específico, mientras que las descripciones locales pueden generarse por sitio web o por aplicación. En algunas realizaciones, el minero de descripciones puede estar compuesto de una serie de varios mineros especializados. Por ejemplo, un minero de XPath puede crear y/o actualizar expresiones de XPath para los elementos respectivos de UI, mientras que un minero de CSS puede crear y/o actualizar selectores de CSS. De manera similar, un minero de Visión Informática (CV) puede usar capturas de pantalla de una UI y crear descripciones CV para el concepto tanto local como globalmente. Esto significa que el minero de descripciones puede producir diferentes tipos de descripciones para el mismo concepto y almacenarlas en un almacén de "Descripciones de Conceptos de UI". Las descripciones "minadas" pueden ser examinadas por usuarios humanos, de manera que puedan ser aceptadas o rechazadas tras su revisión.

En muchos casos, incluso si el elemento de interés puede localizarse automáticamente usando las descripciones existentes, seguiría siendo útil crear descripciones (locales/globales) más eficientes y robustas con el tiempo. Por ejemplo, una expresión de XPath puede ser suficiente para localizar una instancia de un botón de envío en un sitio web, pero el minero de descripciones puede producir un selector de CSS más robusto usando una serie de representaciones de UI almacenadas del mismo sitio web. Por lo tanto, incluso si la expresión de XPath almacenada deja de ser válida en el futuro, el selector de CSS extraído puede usarse para autorreparar el flujo de trabajo y reidentificar el elemento de UI previsto.

- (i) Cada descripción minada se empareja con un identificador de concepto pertinente (es decir, uno correspondiente al concepto que representa el tipo del elemento de UI respectivo), de tal manera que puedan consultarse y accederse a ellos a través de los identificadores de concepto correspondientes.
- (ii) El usuario crea el flujo de trabajo usando identificadores de concepto para referirse a los elementos de interés correspondientes. Por lo tanto, los identificadores de los elementos de la UI y las descripciones del flujo de trabajo están desacoplados. Esto permite la posterior modificación de las descripciones de los elementos de la UI (es decir, los conceptos) sin realizar ningún cambio en los flujos de trabajo existentes.
- (iii) Un trabajador digital puede ejecutar el flujo de trabajo total o parcialmente durante el tiempo de diseño del flujo de trabajo para comprobar si el flujo de trabajo o partes del flujo de trabajo están funcionando correctamente. Al hacerlo, se consultan conceptos para recuperar descripciones de los elementos de la UI correspondientes.
- (iv) El trabajador digital usa las descripciones recuperadas (es decir, expresiones de XPath, selectores de CSS, etc.) para reidentificar los elementos de interés. Si el elemento se reidentifica correctamente, la representación de la UI y el elemento reidentificado automáticamente se almacenan para su uso futuro, por ejemplo, para descubrir mejores descripciones. Si el elemento no puede reidentificarse, se puede pedir al usuario que lo resalte de nuevo.
- (v) Durante la ejecución normal de un flujo de trabajo, si la descripción o descripciones existentes (recuperadas basándose en identificadores de concepto) no pueden usarse para reidentificar un elemento de UI, puede crearse una solicitud para que los operadores humanos resalten el elemento no identificado en una pantalla - los pasos (ix) y (x) pueden repetirse posteriormente y la ejecución del flujo de trabajo puede reanudarse si es posible.

Se apreciará que esta descripción es a modo de ejemplo solamente; pueden realizarse alteraciones y modificaciones a la realización descrita sin apartarse del alcance de la divulgación como se define en las reivindicaciones.

Aunque la presente tecnología se ha descrito en detalle con propósitos de ilustración, basándose en lo que actualmente se consideran las implementaciones más prácticas y preferidas, debe entenderse que tal detalle es únicamente para ese propósito y que la tecnología no se limita a las implementaciones divulgadas, sino que, por el contrario, se pretende que abarque modificaciones y disposiciones equivalentes que están dentro del alcance de las reivindicaciones adjuntas.

La referencia a lo largo de la memoria descriptiva a "una realización" o "un ejemplo" significa que una función, estructura o característica particular descrita en relación con la realización o el ejemplo está incluido en por lo menos una realización de la presente divulgación. Por lo tanto, la aparición de las frases "en una realización" o "en un ejemplo" en varios lugares a lo largo de esta memoria descriptiva no se refieren necesariamente a la misma realización o ejemplo. Además, las funciones, estructuras o características particulares pueden combinarse en cualquier combinación y/o subcombinación adecuada en una o más realizaciones o ejemplos. Además, se aprecia que las figuras proporcionadas con el presente documento son para fines de explicación a las personas expertas en la técnica y que los dibujos no están necesariamente dibujados a escala.

Las realizaciones de acuerdo con la presente divulgación pueden materializarse como un aparato, método o producto de programa informático. Por ejemplo, en algunas realizaciones puede proporcionarse un producto de programa informático que comprende instrucciones que, cuando el programa es ejecutado por un procesador, hacen que el procesador lleve a cabo el método descrito con respecto a la Figura 1 y/o la Figura 2. Como otro ejemplo, en algunas realizaciones se puede proporcionar un medio de almacenamiento legible por ordenador que comprende instrucciones que, cuando son ejecutadas por un ordenador, hacen que el ordenador lleve a cabo los pasos del método descrito en la Figura 1 y/o la Figura 2. Por consiguiente, las presentes realizaciones pueden adoptar la forma de una realización completamente de hardware, una realización completamente de software (incluyendo firmware, software residente, microcódigo, etc.), o una realización que combine aspectos de software y hardware que pueden

denominarse de manera general en la presente como "módulo" o "sistema". Además, las realizaciones de la presente divulgación pueden adoptar la forma de un producto de programa informático incorporado en cualquier medio tangible de expresión que tenga un código de programa utilizable por ordenador incorporado en el medio.

Aunque se describen en relación con un entorno de sistema informático ejemplar, las realizaciones de la presente divulgación son operativas con otros numerosos entornos o configuraciones de sistemas informáticos de propósito general o especial. Ejemplos de sistemas, entornos y/o configuraciones informáticos bien conocidos que pueden ser adecuados para su uso con aspectos de la divulgación incluyen, pero no se limitan a, dispositivos informáticos móviles, ordenadores personales (por ejemplo, ordenadores de sobremesa), ordenadores de servidor, dispositivos de mano o portátiles, sistemas multiprocesador, consolas de juegos, sistemas basados en microprocesador, decodificadores, electrónica de consumo programable, teléfonos móviles, PC de red, miniordenadores, ordenadores centrales, entornos informáticos distribuidos que incluyen cualquiera de los sistemas o dispositivos anteriores, y similares.

Puede usarse cualquier combinación de uno o más medios utilizables en ordenar o legibles por ordenador. Por ejemplo, un medio legible por ordenador puede incluir uno o más de un disquete de ordenador portátil, un disco duro, un dispositivo de memoria de acceso aleatorio (RAM), un dispositivo de memoria de sólo lectura (ROM), un dispositivo de memoria de sólo lectura programable borrable (EPROM o memoria Flash), un disco compacto portátil de memoria de sólo lectura (CDROM), un dispositivo de almacenamiento óptico y un dispositivo de almacenamiento magnético. El código de programa informático para llevar a cabo las operaciones de las realizaciones de la presente divulgación puede estar escrito en cualquier combinación de uno o más lenguajes de programación.

Los diagramas de flujo y los diagramas de bloques ilustran la arquitectura, la funcionalidad y el funcionamiento de posibles implementaciones de sistemas, métodos y productos de programas informáticos de acuerdo con varias realizaciones de la presente divulgación. A este respecto, cada bloque en los diagramas de flujo o en los diagramas de bloques puede representar un módulo, segmento o porción de código, que comprende una o más instrucciones ejecutables para implementar la función o funciones lógicas especificadas. También se observará que cada bloque de los diagramas de bloques y/o ilustraciones de diagramas de flujo, y combinaciones de bloques en los diagramas de bloques y/o ilustraciones de diagramas de flujo, puede implementarse mediante sistemas basados en hardware de propósito especial que realizan las funciones o actos especificados, o combinaciones de hardware de propósito especial e instrucciones de programa informático. Estas instrucciones de programa informático también pueden almacenarse en un medio legible por ordenador que pueda dirigir un ordenador u otro aparato de procesamiento de datos programable para que funcione de una manera particular, de tal manera que las instrucciones almacenadas en el medio legible por ordenador produzcan un artículo de fabricación que incluya medios de instrucción que implementen la función/acto especificado en los diagramas de flujo y/o diagramas de bloques.

# REIVINDICACIONES

1. Un método implementado por ordenador (100) para realizar minería de descripciones para un concepto, el método comprendiendo:

adquirir (S110) una pluralidad de descripciones de referencia, donde cada una de la pluralidad de descripciones de referencia está asociada con un elemento de interfaz de usuario, UI, correspondiente a un tipo de elemento de UI representado por el concepto;  
generar (S120), basándose en la pluralidad de descripciones de referencia, por lo menos una de una descripción local para el concepto y una descripción global para el concepto, en donde la descripción local para el concepto comprende un componente de descripción que es específico para una UI respectiva, y en donde la descripción global para el concepto comprende un componente de descripción que no es específico para una UI; y  
almacenar (S130) la por lo menos una de la descripción local y la descripción global generadas, de tal manera que se asocie a un identificador de concepto correspondiente al concepto.

2. El método (100) de acuerdo con la reivindicación 1, en donde el identificador de concepto identifica unívocamente el concepto en una colección de conceptos, la colección de conceptos siendo una ontología de una pluralidad de diferentes tipos de elementos de UI.

3. El método (100) de acuerdo con cualquiera de las reivindicaciones anteriores, en donde cada una de la pluralidad de descripciones de referencia está en forma de uno de: un Selector de Hojas de Estilos de Cascada, CSS, un lenguaje de rutas del lenguaje de marcado extensible, XPath, expresión, y una descripción de imagen y/o audio.

4. El método (100) de acuerdo con cualquiera de las reivindicaciones anteriores, que comprende además:  
adquirir, para cada una de la pluralidad de descripciones de referencia, una representación de UI correspondiente, en donde la generación de una descripción local para el concepto o una descripción global para el concepto se basa además en una interrelación entre una descripción de referencia y la representación correspondiente de la UI.

5. El método (100) de acuerdo con la reivindicación 4, que comprende además, antes de adquirir una representación de una UI correspondiente para una descripción de referencia respectiva:

generar una representación de UI actualizada eliminando contenido informativo de una representación de UI original,  
en donde la representación adquirida es la representación actualizada.

6. El método (100) de acuerdo con cualquiera de las reivindicaciones anteriores, en donde la generación de una descripción local para el concepto comprende determinar un componente de descripción común a partir de una pluralidad de descripciones de referencia locales históricas, y/o en donde la generación de una descripción global para el concepto comprende determinar un componente de descripción común a partir de una pluralidad de descripciones de referencia no locales.

7. El método (100) de acuerdo con cualquiera de las reivindicaciones anteriores, que comprende además, antes de generar por lo menos una de una descripción local y una descripción global para el concepto:

determinar si se cumple una condición de activación predeterminada,  
en donde la generación de por lo menos una de una descripción local y una descripción global sólo se realiza si se determina que se cumple la condición de activación predeterminada.

8. El método (100) de acuerdo con la reivindicación 7, que comprende además:

adquirir una entrada de usuario, en donde la entrada de usuario corresponde a la asignación del identificador de concepto a un primer elemento de UI en una UI respectiva,  
almacenar una instancia de concepto para el identificador de concepto, la instancia de concepto comprendiendo:  
una representación de la UI respectiva, una descripción del primer elemento de UI y el identificador del concepto;  
en donde el almacenamiento de la instancia del concepto es una condición de activación predeterminada.

9. El método (100) de acuerdo con cualquiera de las reivindicaciones anteriores, que comprende además:

adquirir una entrada del usuario aceptando o rechazando la por lo menos una de la descripción local generada y la descripción global generada,  
en donde el almacenamiento de la descripción local generada y/o de la descripción global generada se realiza únicamente tras obtener la entrada de usuario que acepta la descripción local generada o descripción global generada respectivas.

10. Un método implementado por ordenador (200) para realizar una comprobación de reidentificación de un elemento

de interfaz de usuario, en una UI actual, el método comprendiendo:

recuperar (S210), para el elemento de UI, por lo menos una de una primera descripción local almacenada y una descripción global almacenada sobre la base de un identificador de concepto que representa el elemento de UI, en donde cada una de la por lo menos una primera descripción local almacenada y la primera descripción global almacenada se genera basándose en una pluralidad de descripciones de referencia, cada una de la pluralidad de descripciones de referencia está asociada a un elemento de UI correspondiente a un tipo de elemento de UI correspondiente al identificador de concepto, en donde la primera descripción local almacenada comprende un componente de descripción que es específico de una UI respectiva, y en donde la primera descripción global almacenada comprende un componente de descripción que no es específico de la UI; y consultar (S220) una representación de la IU actual usando la por lo menos una primera descripción local almacenada y la primera descripción global almacenada para determinar si la descripción respectiva puede usarse para reidentificar el elemento de la IU.

11. El método (200) de acuerdo con la reivindicación 10, que comprende además, si se determina que la primera descripción local y/o la primera descripción global pueden usarse para reidentificar el elemento de UI, almacenar la representación de la UI actual y la primera descripción local y/o la primera descripción global de tal manera que se asocien con el identificador de concepto.

12. El método (200) de acuerdo con la reivindicación 10, que comprende además, si se determina que la primera descripción local y/o la primera descripción global no pueden usarse para reidentificar el elemento de UI:

recuperar por lo menos una de: una segunda descripción local almacenada y una segunda descripción global almacenada sobre la base del identificador de concepto, en donde la segunda descripción local es diferente de la primera descripción local y en donde la segunda descripción global es diferente de la segunda descripción global; y consultar la representación de la UI actual usando la por lo menos una de la segunda descripción local y la segunda descripción global para determinar si puede usarse la descripción respectiva para reidentificar el elemento de UI.

13. El método (200) de acuerdo con la reivindicación 10, que comprende además, si se determina que la primera descripción local y/o la primera descripción global no pueden usarse para reidentificar el elemento de UI, realizar por lo menos uno de:

solicitar la entrada del usuario para indicar el elemento de UI en la UI actual; y realizar la minería de descripciones para el concepto de acuerdo con cualquiera de las reivindicaciones 1 a 9.

14. Un sistema (300) para realizar minería de descripciones para un concepto, el sistema comprendiendo:

una unidad de adquisición (310) configurada para adquirir una pluralidad de descripciones de referencia, en donde cada una de la pluralidad de descripciones de referencia está asociada con un elemento de interfaz de usuario, UI, correspondiente a un tipo de elemento de UI representado por el concepto; una unidad de generación (320) configurada para generar, basándose en la pluralidad de descripciones de referencia, por lo menos una de una descripción local para el concepto y una descripción global para el concepto, en donde la descripción local para el concepto comprende un componente de descripción que es específico para una UI respectiva, y en donde la descripción global para el concepto comprende un componente de descripción que no es específico para una UI; y una unidad de almacenamiento (330) configurada para almacenar la por lo menos una la descripción local y la descripción global generadas, de tal manera que se asocie a un identificador de concepto correspondiente al concepto.

15. Un sistema (400) para realizar la comprobación de reidentificación de un elemento de interfaz de usuario, UI, en una UI actual, el sistema comprendiendo:

una unidad de recuperación (410) configurada para recuperar, para el elemento de UI, por lo menos una de una primera descripción local almacenada y una primera descripción global almacenada, basándose en un identificador de concepto que representa al elemento de UI, en donde cada una de la por lo menos una de la primera descripción local almacenada y la primera descripción global almacenada se genera basándose en una pluralidad de descripciones de referencia, cada una de la pluralidad de descripciones de referencia estando asociada a un elemento de UI correspondiente a un tipo de elemento de UI correspondiente al identificador de concepto, en donde la primera descripción local almacenada comprende un componente de descripción que es específico de una UI respectiva, y en donde la primera descripción global almacenada comprende un componente de descripción que no es específico para una UI; y una unidad de consulta (420) configurada para consultar una representación de la UI actual usando la por lo menos una primera descripción local almacenada y primera descripción global almacenada para determinar si puede usarse la descripción respectiva para reidentificar el elemento de UI.

16. Un medio de almacenamiento legible por ordenador que comprende instrucciones que, cuando son ejecutadas por un ordenador, hacen que el ordenador lleve a cabo los pasos del método de cualquiera de las reivindicaciones 1 a 13.

5

10

15

20

25

30

35

40

45

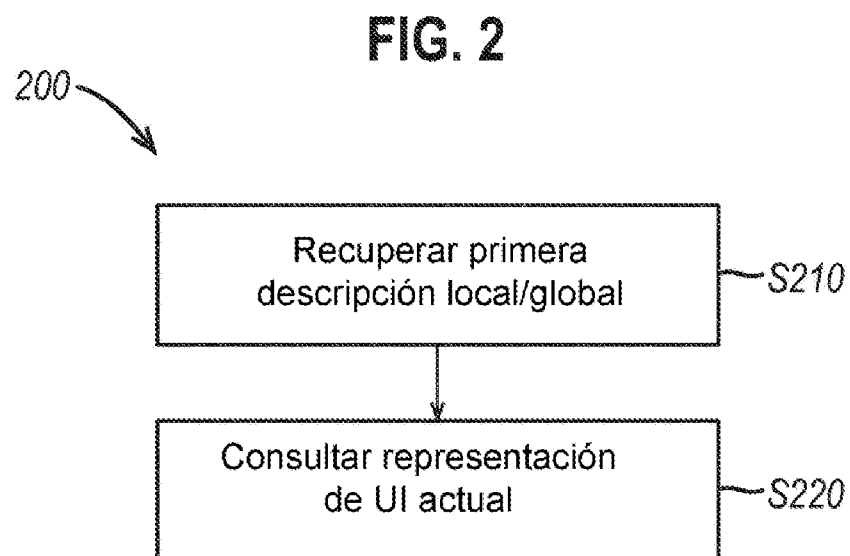
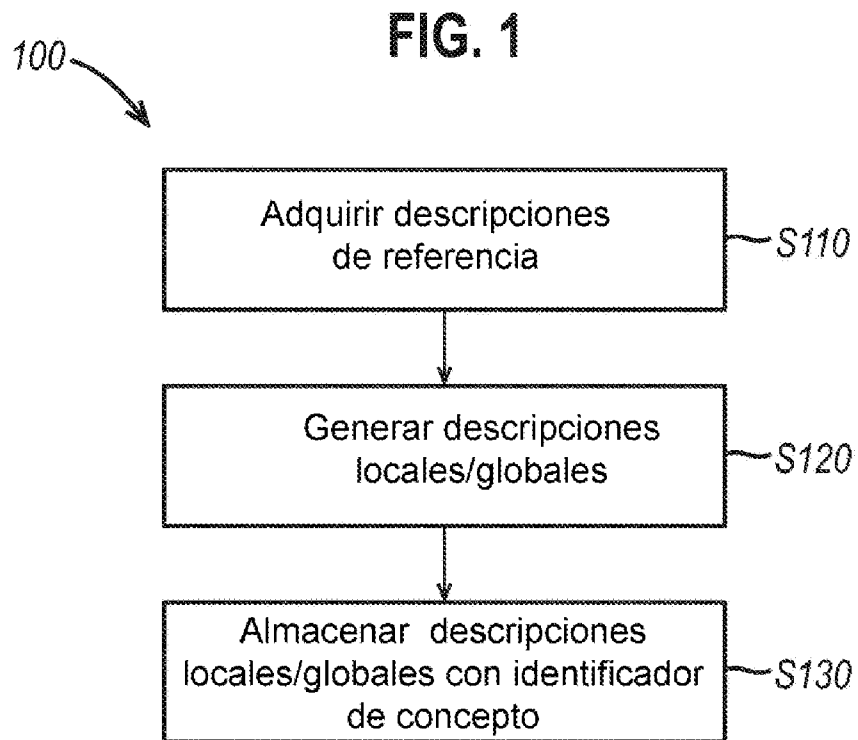
50

55

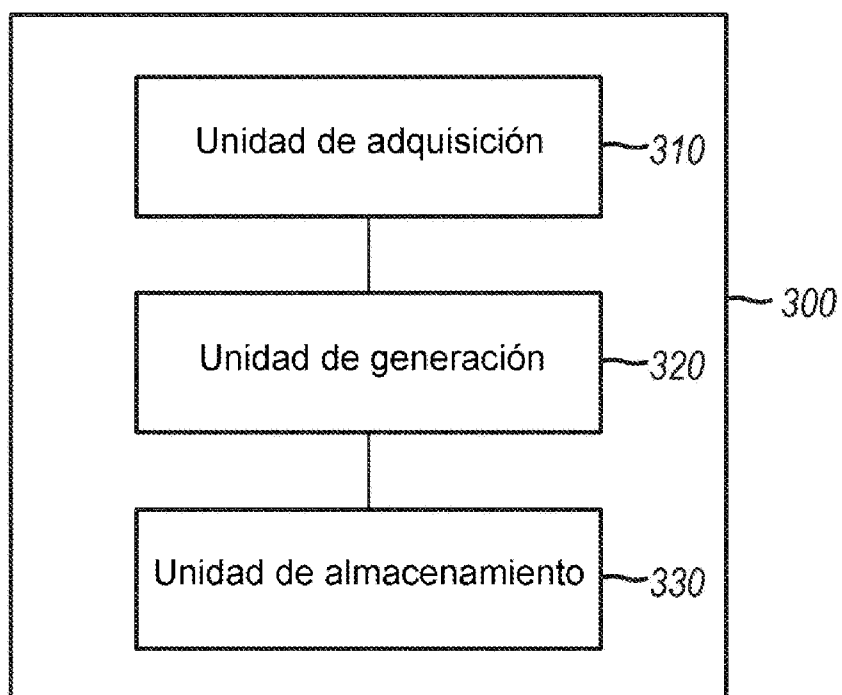
60

65

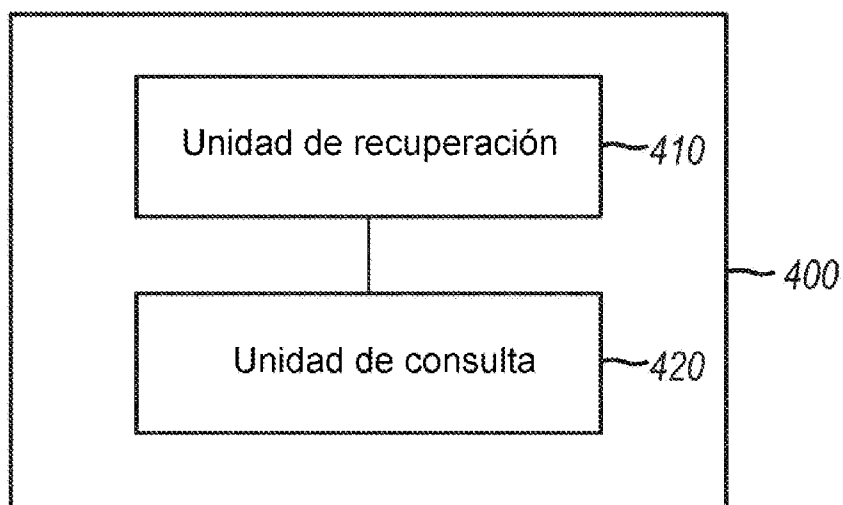




**FIG. 3**



**FIG. 4**



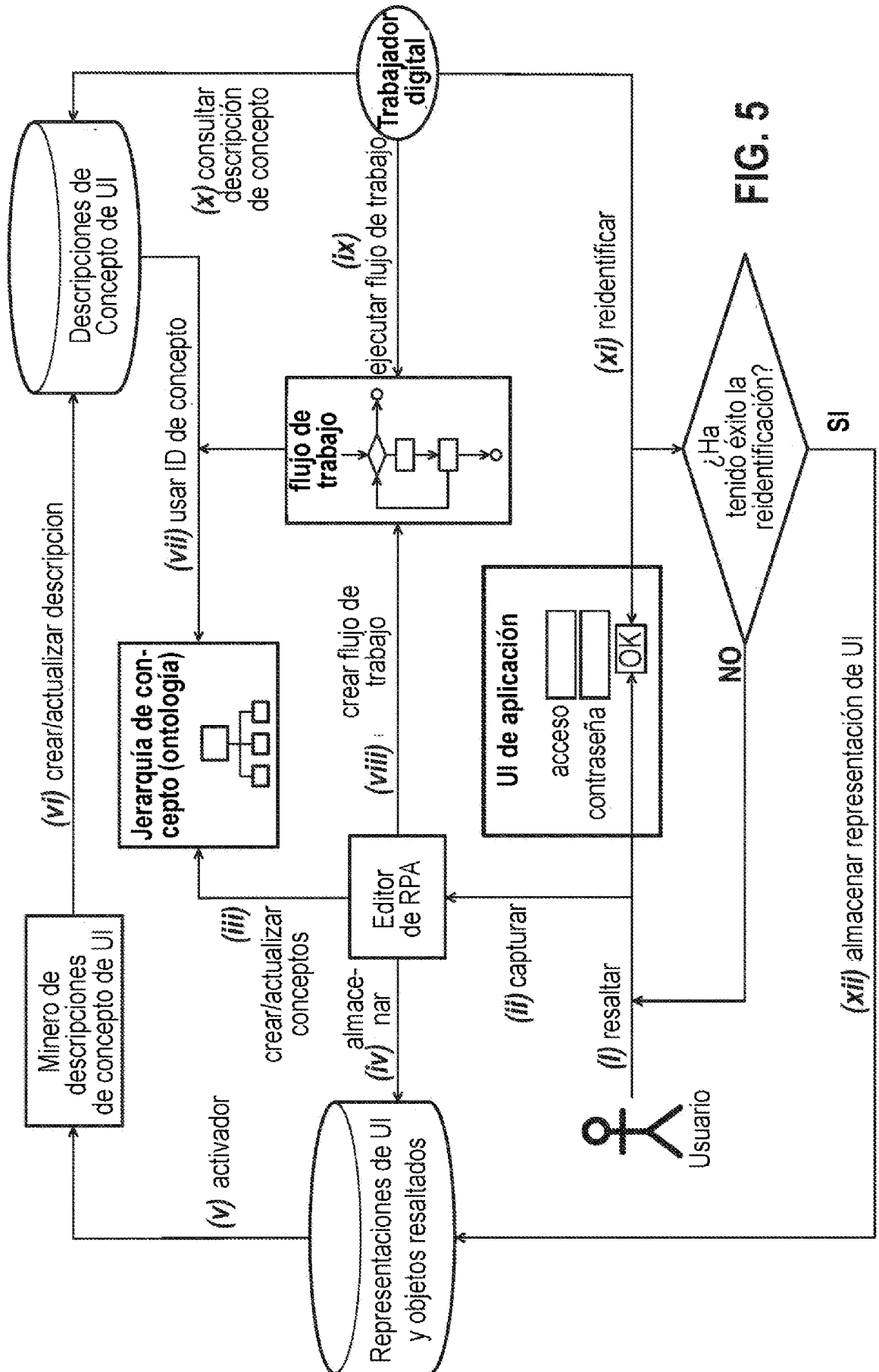


FIG. 5