

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 626 705**

51 Int. Cl.:

G06F 9/06 (2006.01)

G06F 9/44 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **25.03.2011 PCT/US2011/030068**

87 Fecha y número de publicación internacional: **20.10.2011 WO11129989**

96 Fecha de presentación y número de la solicitud europea: **25.03.2011 E 11769274 (9)**

97 Fecha y número de publicación de la concesión europea: **15.03.2017 EP 2558930**

54 Título: **Composición de presentación independiente de plataforma**

30 Prioridad:

15.04.2010 US 760565

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

25.07.2017

73 Titular/es:

**MICROSOFT TECHNOLOGY LICENSING, LLC
(100.0%)
One Microsoft Way
Redmond, Washington 98052, US**

72 Inventor/es:

**BYKOV, EVGUENI N.;
FINDIK, FERIT;
BENSON, RYAN S. y
OTRYSHKO, VOLODYMYR V.**

74 Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 626 705 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

Composición de presentación independiente de plataforma

Antecedentes

- 5 La calidad de una experiencia de usuario (UX) está basada en cómo de bien está alineada la UX con las expectativas del usuario. Teniendo que tratar con muchos tipos de datos, muchos orígenes de datos y muchas plataformas de UX, los diseñadores tienen que realizar una elección a partir de enfoques poco atractivos que incluyen escribir código de presentación para una persona específica que consume datos específicos desde orígenes de datos para una plataforma de UX específica, o proporcionar una UX dirigida de manera amplia que no cumple las necesidades de cualquier persona única.
- 10 Por ejemplo, los sistemas de composición de UX existentes tales como HTML (lenguaje de marcaje hipertexto), XAML (lenguaje de marcaje de aplicación extensible) y XSLT (transformaciones de lenguaje de hoja de estilos extensible) están diseñados de manera que el código de marcaje puede desarrollarse para una plataforma específica. Si el desarrollador desea que el código funcione en varias plataformas se ha de crear una lógica personalizada en el código para manejar las diferencias de la plataforma. Además, los sistemas de composición de UX existentes requieren que se defina de manera explícita una presentación para cada elemento de interfaz de datos. La funcionalidad que permite la generación dinámica de elementos de UX basándose en las estructuras de datos subyacentes que representan los elementos está limitada a no existente, especialmente si las estructuras de datos son complejas y/o no heredables.
- 15 Como resultado de estas limitaciones, el mercado masivo (por ejemplo, correo electrónico) puede haber sido servido, pero las comunidades más pequeñas de usuarios (por ejemplo, el administrador de intercambio o el propietario de servicio de CRM (gestión de relación con el cliente) están desatendidos.
- 20 El documento US 2007/130205 A1 desvela que los metadatos permiten al desarrollador especificar un conjunto de eventos 215 para cada control que se incluyen en la interfaz de usuario. Los metadatos permiten que el desarrollador de formularios de UI especifique los controles a añadirse a la UI; definir eventos personalizados en estos controles añadidos (o añadir eventos a los controles existentes); y definir los manejadores de eventos mediante código en un ensamblaje detrás del código para estos nuevos eventos personalizados (o modificar manejadores personalizados existentes anulando el comportamiento por defecto). Se expone un Modelo de Objeto (OM) que permite al desarrollador leer/modificar propiedades de los controles en la forma de manera que los manejadores de eventos se definen en el código detrás del ensamblaje. El nombre y localización del ensamblaje detrás del código se captura como parte de los metadatos. El motor de representación recibe los metadatos que definen la UI a través de un intérprete y representa el formulario de la UI.
- 25 El documento US 2007/180432 A1 desvela sistemas y procedimientos para personalización de aplicaciones de internet y sus interfaces de usuario asociadas en un entorno basado en explorador. Las interfaces de usuario típicamente se entregan a un usuario como una página web o parte de la misma. La personalización se aplica tanto a aspectos estáticos como funcionales de los elementos de interfaz de usuario. La realización incluye adicionalmente un sistema de personalización que incluye un motor de personalización y una interfaz de perfil de usuario, la interfaz de personalización configurada para modificar datos de personalización que caracterizan el modelo de interacción de manera que se modifica el modelo de interacción en más de una de la pluralidad de interfaces de usuario de aplicación, y un repositorio de datos que incluye un registro de datos para almacenar los datos de personalización, siendo accesible el registro de datos usando los metadatos.
- 30 El documento de Arron Ferguson: "Creating a declarative XML UI language (Build a UI and the accompanying framework in the Java language)" Developer Works, 1 de septiembre de 2009, desvela procedimientos para crear un conjunto de etiquetas de UI de XML declarativas con una estructura de Java adjunta que analiza, valida, construye y finalmente une los componentes de GUI declarados a lógica empresarial en tiempo de ejecución.
- 35 Gordon Erlebacher y col: "Designing grid tag libraries and grid beans", 13 de noviembre de 2006, desvela la implementación de una biblioteca de bibliotecas de etiquetas Grid y beans de Grid para desarrollo de portales web de Grid.
- 40 Es el objeto de la presente invención proporcionar un procedimiento y sistema mejorados para presentación independiente de plataforma.
- 45 Este objeto se resuelve mediante la materia objeto de las reivindicaciones independientes.
- Las realizaciones se definen mediante las reivindicaciones dependientes.
- 50 Lo siguiente presenta un resumen simplificado para proporcionar un entendimiento básico de algunas realizaciones novedosas descritas en el presente documento. Este resumen no es una vista general extensiva, y no se pretende para identificar elementos clave/críticos o para delimitar el alcance de los mismos. Su único fin es presentar algunos conceptos en una forma simplificada como una introducción a la descripción más detallada que se presenta más
- 55

adelante.

La arquitectura desvelada incluye un motor de composición de presentación accionado por configuración independiente de plataforma. El motor de composición permite la generación dinámica de experiencia de usuario (UX) multiplataforma basándose en un contrato de datos. Por composición, el usuario puede seleccionar las partes, interacciones y restricciones entre la interacción y las partes, así como la colocación con respecto entre sí.

La UX está compuesta dinámicamente de componentes que están dirigidos a clases de datos particulares. En tiempo de ejecución, las implementaciones de componente dependientes de plataforma se seleccionan automáticamente mediante el motor basándose en la plataforma de ejecución del anfitrión de la composición.

La arquitectura desvelada permite que un usuario cree o personalice una UX sin escribir código componiendo desde múltiples miniaplicaciones de presentación que pueden acceder a muchos orígenes de datos que funcionan en muchas plataformas. Las composiciones están dirigidas tanto a una clase de datos como tipo de presentación y pueden estar predefinidas o generarse.

Para la consecución de los fines anteriores y relacionados, se describen en el presente documento ciertos aspectos ilustrativos en relación con la siguiente descripción y los dibujos anexos. Estos aspectos son indicativos de las diversas maneras en las que los principios desvelados en el presente documento pueden ponerse en práctica y todos los aspectos y equivalentes de los mismos se pretende que estén dentro del alcance de la materia objeto reivindicada. Otras ventajas y características novedosas serán evidentes a partir de la siguiente descripción detallada cuando se considera en conjunto con los dibujos.

Breve descripción de los dibujos

La Figura 1 ilustra un sistema de visualización de acuerdo con la arquitectura desvelada.
 La Figura 2 ilustra un sistema de visualización alternativo de acuerdo con la arquitectura desvelada.
 La Figura 3 ilustra una composición ejemplar compuesta por el motor de composición.
 La Figura 4 ilustra un componente primario que incluye el contexto de datos y un componente base visual de un sistema de composición.
 La Figura 5 ilustra una definición de componente.
 La Figura 6 ilustra un registro de componente para encontrar o seleccionar componentes.
 La Figura 7 ilustra un diagrama declarativo que representa el uso de variables en el motor de composición.
 La Figura 8 ilustra un procedimiento de visualización de acuerdo con la arquitectura desvelada.
 La Figura 9 ilustra aspectos adicionales del procedimiento de la Figura 8.
 La Figura 10 ilustra un procedimiento de visualización alternativo.
 La Figura 11 ilustra aspectos adicionales del procedimiento de la Figura 10.
 La Figura 12 ilustra un procedimiento de obtención de componentes en el motor de composición.
 La Figura 13 ilustra un procedimiento más detallado de obtención de componentes en el motor de composición.
 La Figura 14 ilustra un diagrama de bloques de un sistema informático que ejecuta la composición de acuerdo con la arquitectura desvelada.

Descripción detallada

La arquitectura desvelada es un motor de composición de presentación. El motor de composición es una estructura de composición genérica que está expuesta como un conjunto de servicios que permiten al usuario “pegar” juntos (componer) diferentes componentes, y la composición (composición de salida del motor) del componente o componentes. Mediante la composición, el usuario puede seleccionar las partes, interacciones y restricciones entre la interacción y las partes, así como la colocación de las partes con respecto entre sí. El motor es una estructura de presentación neutral para tanto los componentes de UI (interfaz de usuario) como distintos de UI.

Un componente es el bloque de construcción reutilizable más pequeño de la declaración de UI para el motor de composición, y que es identificable por el nombre, y opcionalmente, dirigido a un tipo de datos. Un componente puede ser un componente de base (componente de unidad) o un componente de contenedor (componente compuesto). El contexto de datos es una instancia de datos objetivo para un componente. En otras palabras, el contexto de datos es un conjunto de par nombre/valor que representa datos asociados con un componente. Las entradas de contexto de datos soportan notificaciones de cambio, y las composiciones pueden iniciar cambios y/o escuchar los cambios iniciados por otras composiciones. El motor de composición ensambla componentes para un anfitrión particular como una experiencia de usuario que es independiente de la plataforma. El anfitrión de virtualización es el entorno de ejecución de la composición para una plataforma particular (tiempo de ejecución).

Las composiciones están dirigidas tanto a una clase de datos como tipo de presentación y pueden estar predefinidas o generarse. Aunque puede haber múltiples componentes compuestos, la cadena de componentes finaliza en el componente de base concreto, (por ejemplo, un control de cuadro de texto (TextBox), componente de consulta de base de datos, etc.).

El motor de composición permite la generación dinámica de UX (experiencia de usuario) multiplataforma basándose en un contrato de datos. La UX está compuesta dinámicamente de componentes que están dirigidos a clases de

datos particulares. En tiempo de ejecución, las implementaciones de componentes dependientes de plataforma se seleccionan automáticamente por el motor basándose en la plataforma de ejecución del anfitrión de la composición.

Se hace ahora referencia a los dibujos, donde los números de referencia similares se usan para hacer referencia a elementos similares a lo largo del presente documento. En la siguiente descripción, para fines de explicación, se exponen numerosos detalles específicos para proporcionar un entendimiento minucioso de la misma. Puede ser evidente, sin embargo, que las realizaciones novedosas pueden ponerse en práctica sin estos detalles específicos. En otros casos, estructuras y dispositivos bien conocidos se presentan en forma de diagramas de bloques para facilitar una descripción de los mismos. La intención es cubrir todas las modificaciones, equivalentes y alternativas que caen dentro del alcance de la materia objeto reivindicada.

5 La Figura 1 ilustra un sistema 100 de visualización de acuerdo con la arquitectura desvelada. El sistema 100 incluye un almacén 102 de definiciones de almacén que incluyen definiciones 104 de componente y definiciones 106 de datos para componentes y datos asociados con una experiencia de usuario. Las definiciones 104 de componente pueden incluir definiciones para componente de base, componentes de contenedor y composiciones de componentes de base y de contenedor. De esta manera, una composición existente de los componentes está fácilmente disponible para la selección dinámica y composición en el componente 110 de salida.

Un motor 108 de composición que compone automáticamente y de manera declarativa una instancia de un componente 110 de salida basándose en una definición de almacén. El componente de salida es específico a una experiencia de usuario de un anfitrión de visualización de diferentes anfitriones 112.

20 El componente 110 de salida incluye un componente de base, un componente de contenedor o una combinación de componentes de base y de contenedor. El componente 110 de salida está compuesto basándose en el tipo de datos objetivo de la experiencia de usuario. El sistema 100 puede comprender adicionalmente un registro de componente mediante el cual se busca un componente basándose en el tipo de dato objetivo. El componente de salida une las propiedades de componente asociadas a elementos de contexto de datos para enlazar componentes secundarios. El motor 108 de composición incluye variables globales que posibilitan el intercambio de datos entre componentes de salida en contextos de datos no relacionados.

25 La Figura 2 ilustra un sistema 200 de visualización alternativo de acuerdo con la arquitectura desvelada. El sistema 200 incluye las entidades del sistema 100 de la Figura 1, así como el contexto 202 de datos, una anulación 204 de personalización (privada) y la implementación 206 de componente. El componente 110 de salida puede estar compuesto basándose en el contexto 202 de datos en lugar de en una definición de componente existente. Es decir, basándose en los datos, puede crearse un componente personalizado y emitirse puramente basándose en los datos 30 202 de contexto (instancia de datos en la UX objetivo). El motor 108 de composición emplea la anulación 204 de personalización que está compuesta por una definición de componente seleccionado para anular una variable global con una variable privada.

35 La Figura 3 ilustra una composición 300 ejemplar compuesta mediante el motor de composición. En este punto, la composición 300 se describe en términos de componentes de base (por ejemplo, un componente 302 de base de panel apilado (StackPanel)) y un componente 304 de contenedor. En este punto, el componente 302 de base incluye dos componentes de base de cuadro de texto: un primer componente de base de cuadro de texto que muestra el texto "ABC" y un segundo componente de base de cuadro de texto que muestra el texto "DEF". El componente 302 de base también incluye un componente de base de botón.

40 El componente de base es una implementación concreta dirigida a plataformas específicas, el nodo hoja del procedimiento de la composición, y puede ser visual o no visual. A continuación se encuentra un ejemplo de una definición de componente de base (en términos de tipo de componente).

```

45 <ComponentType ID="EventQuery">
    <Parameter>
    <Parameter Name="Scope" Type="String" />
        <Parameter Name="Output" Type="IEnumerable" />
    </Parameters>
</ComponentType>
50 <ComponentImplementation TypeId="EventQuery">
    <SupportedPlatforms>
        <Platform>WPF</Platform>
    </SupportedPlatforms>
    <Unit>
55 <MefFactory>
<ContractName>Company.EnterpriseManagement.EventQuery</ContractName>
    </MefFactory>
    <Properties
        <Property Name="QueryVerb" Direction="In">Events</Property>
60 <Property Name="Scope" Direction="In">$Parameter/Scope</Property>

```

```

        <Property Name="Output"
Direction="Out">$Parameter/Output$</Property>
        </Properties>
5      </Unit>
      </ComponentImplementation>

```

La fabricación de MEF (estructura de extensibilidad gestionada) solicita un tiempo de ejecución MEF para extraer tipos correspondientes a partir de ensamblajes de UI registrados. Obsérvese que MEF es solo una manera de implementación de ejemplo; pueden empelarse también otras implementaciones de fabricación.

- 10 El componente de contenedor (también, componente compuesto) es el contenedor para los componentes de base (también, componentes de unidad), no tiene una implementación personalizada, y es independiente de plataforma. A continuación hay un ejemplo de una definición de componente compuesto.

```

      <ComponentImplementation TypeId="SampleComponent">
        <SupportedPlatforms>
          <Platform>All</Platform>
        </SupportedPlatforms>
        <Composite>
          <Variables>
            <Variable Id="abc" Type="String" />
          </Variables>
          <Component TypeId="SampleContainerComponent">
            <Parameter Id="Child1">
              <Component TypeId="SampleComponent1"'>'
                <Parameter Id="Bla">$Variable/abc$</Parameters>
            </Component>
          </Parameter>
            <Parameter Id="Child2">
              <Component TypeId="SampleComponent2">
                <Parameter Id="Bla">$Variable/abc$</Parameter>
            </Component>
          </Parameter>
        </Component>
      </Composite>
    </ComponentImplementation>

```

- 35 En la Figura 3, el componente 304 de contenedor combina los componentes de base, como se ilustra en el siguiente código, donde la propiedad A (PropertyA) es "ABC" y la propiedad B (PropertyB) es "DEF".

```

<Component TypeId="StackPanel">
<Parameter Name="Child">
40  <Component TypeId="Edit">
      <Target>$Target/propertyA$</Target>
      </Component>
      <Component TypeId="Edit">
        <Target>$Target/propertyB$</Target>
      </Component>
45  <Component TypeId="Button">
      </Component>
    </Parameter>

```

- 50 Cada componente (base o contenedor) se respalda con una instancia de contexto de datos (Data Context). El contexto de datos es una colección de par clave(cadena)-valor(objeto) que soporta la notificación de propiedad cambiada y la notificación de ajuste de error. El componente puede unir sus propiedades a los elementos de contexto de datos para enlazar componentes secundarios juntos.

- 55 La Figura 4 ilustra un componente 400 primario que incluye el contexto 402 de datos (como elementos de contexto de datos) y un componente 404 de base visual (por ejemplo, el componente 302 de base de panel apilado de la Figura 3) de un sistema de composición. En este punto, el contexto 402 de datos incluye tres propiedades: la propiedad A, la propiedad B y la propiedad C, con valores correspondientes "ABC", "DEF" y "XYZ". El componente 404 de base visual incluye las uniones a la propiedad A y la propiedad B mediante uniones de datos en 2 sentidos correspondientes, mientras que los componentes 406 de datos múltiples de un modelo 408 de vista se unen a la propiedad B y la propiedad C. En otras palabras, el componente 400 primario está compuesto que incluye las uniones de propiedades a componentes secundarios (los componentes de base del cuadro de texto) y las uniones de las propiedades de los componentes 406 de datos.
- 60

La Figura 5 ilustra una definición 500 de componente. La definición 500 de componente comprende dos partes: una declaración 502 de tipo y una definición 504 de implementación. Cada componente tiene un nombre, y atributos de tipo objetivo opcionales que pueden usarse para buscar un componente relevante. Una declaración de tipo de ejemplo (en términos de componente) es la siguiente:

```
5 <ComponentType ID="SampleComponent" Target="String" Accessibility="Internal">
```

Un componente también puede incluir un subnodo de parámetros que define la forma de los datos (por ejemplo, cadena) que se espera que se pasen, como se ilustra a continuación (en términos de componente):

```

10 <ComponentType ID="AnotherComposition">
    <Parameters>
        <Parameter Name="Parameter1" Type="String" />
        <Parameter Name="SelectedText" Type="String" BindingDirection="Both" />
    </Parameters>

```

La Figura 6 ilustra un registro 600 de componente para unir o seleccionar componentes. El registro 600 mantiene una lista de todos los componentes definidos y composiciones de componente. Por ejemplo, el componente de base (por ejemplo, panel apilado/botón/editar de la Figura 3) puede averiguarse (buscarse) basándose en ID de tipo (TypeId) y tipo de objetivo (TargetType) (el tipo de datos objetivo). La salida es entonces el componente de base que corresponde al tipo de objetivo o Id de tipo y tipo de objetivo.

Para establecer una propiedad en un componente, se usa un nodo de "parámetro", como se muestra en el siguiente código de muestra:

```
20 <Component Id="EventView">
    <Parameter Id="Scope">Microsoft.SystemCenter.SqlDB </Parameter>

```

En este caso, la propiedad "Scope" del componente "EventView" se establece a texto "Company.SystemCenter.SqlDB". En ocasiones, sin embargo, los parámetros no son estáticos, sino que se unen a otros elementos. En general, se hace referencia en el formulario \$<protocol>/<protocol-specific string>.

25 Por ejemplo, se unen dos componentes a una variable "abc":

```

30 <ComponentImplementation TypeId="SampleComponent">
    <SupportedPlatforms>
        <Platform>All</Platform>
    </SupportedPlatforms>
    <Composite>
        <Variables>
            <Variable Id="abc" Type="String" />
        </Variables>
        <Component TypeId="SampleComponent1">
            <Parameter Id="A">${Variable/abc}</Parameter>
        </Component>
        <Component TypeId="SampleComponent2">
            <Parameter Id="A">${Variable/abc}</Parameter>
        </Component>
    </Composite>
40 </ComponentImplementation>

```

A continuación hay una lista de ejemplo de protocolos de referencia en un nodo de Parámetro:

```

45 $Parameter/<propertyName>$ Parámetro pasado a componente
    $Variable/<propertyName>$ Variable declarada en componente
    $Target/<propertyName>$ Propiedad de una instancia objetivo pasada a componente
    $Target$ Instancia objetivo

```

Las variables globales pueden usarse para posibilitar el intercambio de datos entre composiciones (componentes de base y/o componentes de contenedor) en contextos de datos no relacionados. En primer lugar, se declara una variable global:

```
50 <GlobalVariable ID="GlobalSelectedItem" Type="String"/>
```

La variable puede referenciarse en el componente de base y componentes de contenedor usando

```
$GlobalVariable/<variable name>$.
```

Cualquier componente dado (por ejemplo, base, contenedor) puede anular la variable con una implementación

privada de modo que el componente secundario ve una copia privada de la variable, ilustrada en el siguiente código de ejemplo:

```

5  <Variables>
    <GlobalVariableOverride GlobalVariableId="GlobalSelectedItem" />
  </Variables>

```

La Figura 7 ilustra un diagrama 700 declarativo que representa el uso de variables en el motor de composición. En 702, se declara una variable global "A". En 704, se pasa un nombre de parámetro "Blah" a un componente. En 706, se crea una copia local de la variable global. En 708 y 710, se usan copias locales de la variable en lugar de la variable global.

10 Se incluye en el presente documento un conjunto de diagramas de flujo representativos de metodologías ejemplares para realizar aspectos novedosos de la arquitectura desvelada. Aunque, para fines de simplicidad de explicación, la una o más metodologías mostradas en el presente documento, por ejemplo, en forma de un gráfico de flujo o diagrama de flujo, se muestran y describen como una serie de actos, se ha de entender y apreciar que las metodologías no están limitadas al orden de los actos, ya que algunos actos pueden tener lugar, de acuerdo con las mismas, en un orden diferente y/o de manera concurrente con otros actos a partir de los mostrados y descritos en el presente documento. Por ejemplo, los expertos en la materia entenderán y apreciarán que una metodología podría representarse como alternativa como una serie de estados o eventos interrelacionados, tales como en un diagrama de estado. Además, no todos los actos ilustrados en una metodología pueden requerirse para una implementación novedosa.

20 La Figura 8 ilustra un procedimiento de visualización de acuerdo con la arquitectura desvelada. En 800, se recibe una solicitud para que se emplee un componente en un entorno de ejecución. En 802, se busca una definición de componente asociada con el componente. En 804, se selecciona una o más definiciones de datos para una definición de componente encontrada. En 806, la una o más definiciones de datos están compuestas automáticamente con la definición de componente para emitir el componente en el entorno de ejecución en tiempo de ejecución del entorno.

La Figura 9 ilustra aspectos adicionales del procedimiento de la Figura 8. En 900, se busca la definición de componente basándose en un tipo de dato del componente solicitado cuando no se encuentra la definición de componente. En 902, se crea un componente personalizado basándose en la ausencia de la solicitud del componente. En 904, se aplica una variable global al componente para posibilitar el intercambio de datos entre contextos de datos no relacionados. En 906, se anula una variable global con una variable privada para imponer la variable privada en componentes secundarios del componente. En 908, se crea un componente de contenedor cuando no se encuentra el componente solicitado. En 910, se carga el componente de contenedor con propiedades de tipo de datos asociadas de componentes de base. En 912, el componente de contenedor se emite como el componente.

35 La Figura 10 ilustra un procedimiento de visualización alternativo. En 1000, se recibe una solicitud para un componente basándose en un entorno de ejecución de componente. En 1002, se busca una definición de componente asociada con el componente. En 1004, se selecciona una o más definiciones de datos para la definición de componente si se encuentra la definición de componente. En 1006, se crea un componente personalizado basándose en un tipo de datos relacionado con el componente solicitado cuando no se encuentra la definición de componente. En 1008, se aplica una variable global al componente o se personaliza para posibilitar el intercambio de datos entre contextos de datos no relacionados. En 1010, la una o más definiciones de datos se componen automáticamente con la definición de componente para emitir el componente en el entorno de ejecución en tiempo de ejecución del entorno.

45 La Figura 11 ilustra aspectos adicionales del procedimiento de la Figura 10. En 1100, la variable global se anula con una variable privada para imponer la variable privada en componentes secundarios del componente. En 1102, se crea un componente de contenedor cuando no se encuentra el componente solicitado. En 1104, el componente de contenedor se carga con propiedades de tipo de datos asociadas a componentes de base. En 1106, el componente de contenedor se emite como el componente. En 1108, se definen datos a pasarse al componente mediante un nodo de parámetro. En 1110, se compone un componente primario que incluye uniones de propiedades a componentes secundarios y uniones de las propiedades a componentes de datos.

La Figura 12 ilustra un procedimiento de obtención de componentes en el motor de composición. En 1200, se obtiene un componente (por ejemplo, mediante una solicitud "obtener componente"). En 1202, se obtiene el tipo de dato objetivo. En 1204, se realiza una comprobación para determinar si existe un componente para el tipo de dato objetivo. Si no, el flujo va a 1206 para crear un contenedor de componente. El componente puede tener una o más propiedades de datos asociadas. En 1208, se obtienen tipos de propiedades para el componente. En 1210, se realiza una solicitud "obtener componente" para los tipos de propiedades. En 1212, el componente se añade al flujo de contenedor de vuelta a 1208 para continuar hasta que se complete. Después de que se han aplicado todas las propiedades y tipos al contenedor, el flujo va a 1214 para devolver los resultados. En 1204, si la comprobación determina que existe un componente para el tipo de dato objetivo, el flujo va a 1216 para seleccionar el componente,

y a continuación devuelve los resultados, en 1214.

La Figura 13 ilustra un procedimiento más detallado de obtención de componentes en el motor de composición. En 1300, se reciben los parámetros de función tal como tipo de objetivo y tipo de datos. En 1302, se realiza una comprobación para un componente definido para el tipo y nombre de objetivo. Si es así, el flujo va a 1304 para verificar la interfaz. Como alternativa, si no existe componente para el tipo y nombres de objetivo, el flujo va a 1306 de manera que para cada propiedad que usa la búsqueda de sistema de tipo, hay un componente definido para el tipo de datos. En caso afirmativo, el flujo va a 1304 para verificar la interfaz. El acceso al sistema 1308 de tipo se proporciona para realizar la comprobación y verificar la interfaz. Una vez que se verifica la interfaz, el flujo va a 1310 para comprobar el tipo de componente. Si hay un componente de unidad, el flujo va a 1312 para crear una instancia del componente de unidad para la plataforma correcta y pasar todos los parámetros declarados al componente de unidad. En 1314, el cargador puede cargar un conjunto para el sistema de composición de UX (por ejemplo, XAML, MEF, etc.).

Si el tipo de componente, en 1310, es un componente compuesto, el flujo va a 1316 para andar a los nodos secundarios en una configuración (por ejemplo, escrita en XML) que establece los valores en los componentes de la composición. Esto incluye recibir valores de parámetro desde el nodo o nodos 1318 de parámetro, los parámetros desde el nodo o nodo 1320 de componente, y un conjunto de parámetros desde nodos secundarios, según se proporciona desde datos creados a partir de los parámetros en 1322. El ID de tipo se envía desde el nodo 1320 de componente para crear los datos 1322. En 1324, el componente referenciado por el objetivo como datos y nombre, al igual que el nombre para la búsqueda, basándose en información de nombre y objetivo recibida desde el nodo 1320 de componente.

Uno o más componentes pueden residir en un procedimiento y/o hilo de ejecución, y un componente puede localizarse en un ordenador y/o distribuirse entre dos o más ordenadores. La palabra “ejemplar” puede usarse en el presente documento para significar que sirve como un ejemplo, instancia o ilustración. Cualquier aspecto o diseño descrito en el presente documento como “ejemplar” no se ha de interpretar necesariamente como preferido o ventajoso sobre otros aspectos o diseños.

Haciendo referencia ahora a la Figura 14, se ilustra un diagrama de bloques de un sistema 1400 informático que ejecuta la composición de acuerdo con la arquitectura desvelada. Para proporcionar un contexto adicional para diversos aspectos de la misma, la Figura 14 y la siguiente descripción se pretende que proporcionen una descripción general breve del sistema 1400 informático adecuado en el que pueden implementarse los diversos aspectos. Aunque la descripción anterior es en el contexto general de instrucciones ejecutables por ordenador que pueden ejecutarse en uno o más componentes, los expertos en la materia reconocerán que también puede implementarse una realización novedosa en combinación con otros módulos de programa y/o como una combinación de hardware y software.

El sistema 1400 informático para implementar diversos aspectos incluye el ordenador 1402 que tiene la unidad o unidades 1404 de procesamiento, un almacén legible por ordenador tal como una memoria 1406 de sistema, y un bus 1408 de sistema. La unidad o unidades 1404 de procesamiento pueden ser cualquiera de diversos procesadores comercialmente disponibles tales como único procesador, multi-procesador, unidades de único núcleo y unidades multi-núcleo. Además, los expertos en la materia apreciarán que los procedimientos novedosos pueden ponerse en práctica con otras configuraciones de sistema informático, que incluyen miniordenadores, ordenadores centrales, así como ordenadores personales (por ejemplo, de sobremesa, portátil, etc.), dispositivos informáticos de mano, electrónica basada microprocesador o de consumidor programable, y similares, cada uno de los cuales puede estar acoplado de manera operativa a uno o más dispositivos asociados.

La memoria 1406 de sistema puede incluir el almacén legible por ordenador (medio de almacén físico) tal como la memoria 1410 volátil (VOL) (por ejemplo, memoria de acceso aleatorio (RAM)) y memoria no volátil (NON-VOL) 1412 (por ejemplo, ROM, EPROM, EEPROM, etc.). Un sistema básico de entrada/salida (BIOS) puede almacenarse en la memoria 1412 no volátil, e incluye las rutinas básicas que facilitan la comunicación de datos y señales entre componentes en el ordenador 1402, tales como durante el arranque. La memoria 1410 volátil puede incluir también una RAM de alta velocidad tal como RAM estática para almacenar en caché datos.

El bus 1408 de sistema proporciona una interfaz para componentes de sistema que incluye, pero sin limitación, la memoria 1406 de sistema a la unidad o unidades 1404 de procesamiento. El bus 1408 de sistema puede ser cualquiera de varios tipos de estructura de bus que pueden interconectar adicionalmente a un bus de memoria (con o sin un controlador de memoria), y un bus periférico (por ejemplo, PCI, PCIe, AGP, LPC, etc.), usando cualquiera de una diversidad de arquitecturas de bus comercialmente disponibles.

El ordenador 1402 incluye adicionalmente el subsistema o subsistemas 1414 de almacén legible por máquina y la interfaz o interfaces 1416 de almacén para interconectar el subsistema o subsistemas 1414 de almacén al bus 1408 de sistema y a otros componentes informáticos deseados. El subsistema o subsistemas 1414 de almacén (medio de almacén físico) pueden incluir uno o más de una unidad de disco duro (HDD), una unidad de disco flexible magnético (FDD), y/o unidad de almacén de disco óptico (por ejemplo, una unidad de CD-ROM, unidad de DVD), por ejemplo. La interfaz o interfaces 1416 de almacén pueden incluir tecnologías de interfaz tales como EIDE, ATA,

SATA e IEEE 1394, por ejemplo.

Uno o más programas y datos pueden almacenarse en el subsistema 1406 de memoria, un subsistema 1418 de memoria legible y extraíble por máquina (por ejemplo, tecnología de factor de forma de unidad flash), y/o el subsistema o subsistemas 1414 de almacén (por ejemplo, óptico, magnético, estado sólido), que incluye un sistema 1420 operativo, uno o más programas 1422 de aplicación, otros módulos 1424 de programa y datos 1426 de programa.

El uno o más programas 1422 de aplicación, otros módulos 1424 de programa, y datos 1426 de programa pueden incluir las entidades y componentes del sistema 100 de la Figura 1, las entidades y componentes del sistema 200 de la Figura 2, la composición 300 de la Figura 3, el componente 400 primario de la Figura 4, la definición 500 de componente de la Figura 5, el registro 600 de la Figura 6, el diagrama 700 de la Figura 7, y los procedimientos representados mediante los diagramas de flujo de las Figuras 8-13, por ejemplo.

En general, los programas incluyen rutinas, procedimientos, estructuras de datos, otros componentes de software, etc., que realizan tareas particulares o implementan tipos de datos abstractos particulares. Todo o porciones del sistema 1420 operativo, aplicaciones 1422, módulos 1424 y/o datos 1426 pueden también almacenarse en memoria caché tal como la memoria 1410 volátil, por ejemplo. Se ha de apreciar que la arquitectura desvelada puede implementarse con diversos sistemas operativos comercialmente disponibles o combinaciones de sistemas operativos (por ejemplo, como máquinas virtuales).

El subsistema o subsistemas 1414 de almacén y los subsistemas (1406 y 1418) de memoria sirven como medio legible por ordenador para almacén de datos volátil y no volátil, estructuras de datos, instrucciones ejecutables por ordenador y así sucesivamente. Tales instrucciones, cuando se ejecutan mediante un ordenador u otra máquina, pueden provocar que el ordenador u otra máquina realice uno o más actos de un procedimiento. Las instrucciones para realizar los actos pueden almacenarse en un medio, o podrían almacenarse a través de múltiples medios, de modo que las instrucciones aparecen colectivamente en el uno o más medios de almacén legibles por ordenador, independientemente de si todas las instrucciones están en el mismo medio.

Medio legible por ordenador puede ser cualquier medio disponible que pueda accederse mediante el ordenador 1402 e incluye medio interno y/o externo volátil y no volátil que es extraíble o no extraíble. Para el ordenador 1402, el medio adapta el almacén de datos en cualquier formato digital adecuado. Debería apreciarse por los expertos en la materia que pueden emplearse otros tipos de medio legible por ordenador tales como unidades zip, cinta magnética, tarjetas de memoria flash, unidades de flash, cartuchos y similares, para almacenar instrucciones ejecutables por ordenador para realizar los procedimientos novedosos de la arquitectura desvelada.

Un usuario puede interactuar con el ordenador 1402, programas y datos usando dispositivos 1428 de entrada de usuario externos tales como un teclado y a ratón. Otros dispositivos 1428 de entrada de usuario externos pueden incluir un micrófono, un control remoto de IR (infra rojos), una palanca de mandos, un control de juegos, sistemas de reconocimiento de cámara, un lápiz óptico, pantalla táctil, sistemas de gestos (por ejemplo, movimiento del ojo, movimiento de la cabeza, etc.), y/o similares. El usuario puede interactuar con el ordenador 1402, programas y datos usando los dispositivos 1430 de entrada de usuario integrados tales como un panel táctil, micrófono, teclado, etc., donde el ordenador 1402 es un ordenador portátil, por ejemplo. Estos y otros dispositivos de entrada están conectados a la unidad o unidades 1404 de procesamiento a través de interfaz o interfaces 1432 de dispositivo de entrada/salida (E/S) mediante el bus 1408 de sistema, pero pueden conectarse mediante otras interfaces tales como un puerto paralelo, puerto de serie IEEE 1394, un puerto de juegos, un puerto USB, una interfaz de IR, etc. La interfaz o interfaces 1432 de dispositivo de E/S también facilitan el uso de periféricos 1434 tales como impresoras, dispositivos de audio, dispositivos de cámara y así sucesivamente, tal como una tarjeta de sonido y/o capacidad de procesamiento de audio integrado.

Una o más interfaz o interfaces 1436 gráficas (también comúnmente denominadas como una unidad de procesamiento de gráficos (GPU)) proporcionan señales de gráficos y de vídeo entre el ordenador 1402 y la pantalla o pantallas 1438 externas (por ejemplo, LCD, plasma) y/o pantallas integradas 1440 (por ejemplo, para el ordenador portátil). La interfaz o interfaces 1436 de gráficos pueden fabricarse también como parte de la placa del sistema informático.

El ordenador 1402 puede operar en un entorno en red (por ejemplo, basado en IP) usando conexiones lógicas mediante un subsistema 1442 de comunicaciones alámbricas/inalámbricas a una o más redes y/u otros ordenadores. Los otros ordenadores pueden incluir estaciones de trabajo, servidores, encaminadores, ordenadores personales, dispositivos de entretenimiento basados en microprocesador, dispositivos de pares y otros nodos de red comunes, y típicamente incluyen muchos o todos los elementos descritos con relación al ordenador 1402. Las conexiones lógicas pueden incluir conectividad alámbrica/inalámbrica a una red de área local (LAN), una red de área extensa (WAN), punto caliente y así sucesivamente. Los entornos de interconexión en red LAN y WAN son comunes en oficinas y compañías y facilitan las redes informáticas por toda la empresa, tales como intranets, todas las cuales pueden conectarse a una red de comunicaciones global tal como internet.

Cuando se usa en un entorno de interconexión en red el ordenador 1402 se conecta a la red mediante un

- subsistema 1442 de comunicación alámbrico/inalámbrico (por ejemplo, un adaptador de interfaz de red, subsistema transceptor integrado, etc.) para comunicar con redes alámbricas/inalámbricas, impresoras alámbricas/inalámbricas, dispositivos 1444 de entrada alámbricos/inalámbricos, y así sucesivamente. El ordenador 1402 puede incluir un módem u otro medio para establecer comunicaciones a través de la red. En un entorno en red, los programas y datos con relación al ordenador 1402 pueden almacenarse en el dispositivo de memoria/almacén remoto, ya que están asociados con un sistema distribuido. Se apreciará que las conexiones de red mostradas son ejemplares y que pueden usarse otros medios de establecimiento de un enlace de comunicaciones entre los ordenadores.
- El ordenador 1402 es operable para comunicar con dispositivos o entidades alámbricas/inalámbricas usando las tecnologías de radio tales como la familia de normas del IEEE 802.xx, tal como dispositivos inalámbricos dispuestos de manera operativa en comunicación inalámbrica (por ejemplo, técnicas de modulación a través del aire del IEEE 802.11) con, por ejemplo, una impresora, escáner, ordenador de sobremesa y/o portátil, asistente digital personal (PDA), satélite de comunicaciones, cualquier pieza de equipo o localización asociada con una etiqueta detectable inalámbricamente (por ejemplo, un quiosco, puesto de periódicos, baño), y teléfono. Esto incluye al menos Wi-Fi (o Fidelidad Inalámbrica) para puntos calientes, WiMax, y tecnologías inalámbricas Bluetooth™. Por lo tanto, las comunicaciones pueden ser una estructura predefinida como con una red convencional o simplemente una comunicación ad hoc entre al menos dos dispositivos. Las redes Wi-Fi usan tecnologías de radio denominadas IEEE 802.11x (a, b, g, etc.) para proporcionar conectividad inalámbrica segura, fiable, rápida. Una red Wi-Fi puede usarse para conectar los ordenadores entre sí, a internet y a redes alámbricas (que usan medios y funciones relacionados con IEEE 802.3).
- Los aspectos ilustrados y descritos pueden ponerse en práctica en entornos informáticos distribuidos donde ciertas tareas se realizan mediante dispositivos de procesamiento remotos que están enlazados a través de una red de comunicaciones. En un entorno informático distribuido, los módulos de programa pueden localizarse en almacén local y/o remoto y/o sistema de memoria.
- Lo que se ha descrito anteriormente incluye ejemplos de la arquitectura desvelada. Por supuesto, no es posible describir cada combinación concebible de componentes y/o metodologías, aunque un experto en la materia puede reconocer que son posibles combinaciones y permutaciones adicionales. Por consiguiente, la arquitectura novedosa se pretende que abarque todas tales alteraciones, modificaciones y variaciones que caen dentro del espíritu y alcance de las reivindicaciones adjuntas. Adicionalmente, hasta el punto que el término “incluye” se usa en cualquiera de la descripción detallada o las reivindicaciones, tal término se pretende que sea inclusivo de una manera similar las expresiones “que comprende” ya que “que comprende” se interpreta cuando se emplea como una palabra transicional en una reivindicación.

REIVINDICACIONES

1. Un sistema de visualización implementado por ordenador que tiene un medio legible por ordenador que almacena instrucciones ejecutables ejecutadas mediante un procesador, que comprende:
 - 5 un almacén (102) de definiciones de almacén que incluyen definiciones (104) de componente y definiciones (106) de datos para componentes y datos asociados con una experiencia de usuario; y
 - un motor (108) de composición que compone automáticamente y de manera declarativa una instancia (110) de un componente de salida basándose en una definición de almacén que incluye una definición de componente y al menos una definición de datos para emitir el componente de salida en un entorno de ejecución dado en tiempo de ejecución del entorno, siendo el componente de salida específico de una experiencia de usuario de un anfitrión (112) de visualización particular,
 - 10 en el que el componente de salida comprende un componente (304) de contenedor independiente de plataforma, en el que el componente de contenedor es el contenedor para componentes (302) de base, en el que un componente de base es una implementación concreta dirigida a plataformas específicas y seleccionada automáticamente mediante el motor de composición en tiempo de ejecución basándose en la plataforma de ejecución,
 - 15 y en el que el motor de composición emplea una anulación (706) de personalización que está compuesta por una definición de componente seleccionada para anular una variable global con una variable privada.
2. El sistema de la reivindicación 1, en el que el componente de salida incluye un componente de base, un componente de contenedor, o una combinación de componentes de base y de contenedor.
- 20 3. El sistema de la reivindicación 1 o 2, en el que el componente de salida está compuesto basándose en un tipo de datos objetivo.
4. El sistema de la reivindicación 3, que comprende adicionalmente un registro (600) de componentes mediante el cual se busca un componente basándose en el tipo de datos objetivo.
5. El sistema de una de las reivindicaciones 1 a 4, en el que el componente de salida está compuesto basándose en un contexto (402) de datos.
- 25 6. El sistema de una de las reivindicaciones 1 a 5, en el que el componente (400) de salida une propiedades de componente asociadas a elementos de contexto (402) de datos para enlazar componentes (404, 406) secundarios.
7. El sistema de una de las reivindicaciones 1 a 6, en el que el motor (700) de composición incluye variables globales que posibilitan el intercambio de datos entre componentes de salida en contextos de datos no relacionados.
- 30 8. Un procedimiento de visualización implementado por ordenador ejecutable mediante un procesador y memoria, que comprende:
 - recibir (800; 1000) una solicitud para que se emplee un componente en un entorno de ejecución;
 - 35 buscar (802; 1002) un almacén (102) de definiciones de almacén para una definición de componente asociada con el componente;
 - seleccionar (804; 1004), mediante un motor (108) de composición una o más definiciones de datos para una definición de componente encontrada; y
 - componer (806; 1010) automáticamente, mediante el motor de composición, la una o más definiciones de datos con la definición de componente para emitir el componente en el entorno de ejecución en tiempo de ejecución del entorno,
 - 40 en el que el componente es específico de una experiencia de usuario de un anfitrión de visualización particular, en el que el componente comprende un componente de contenedor independiente de plataforma, siendo el componente de contenedor el contenedor para componentes de base,
 - en el que un componente de base es una implementación concreta dirigida a plataformas específicas y seleccionada automáticamente mediante el motor de composición en tiempo de ejecución basándose en la plataforma de ejecución,
 - 45 y en el que el procedimiento comprende adicionalmente anular (906; 1100) una variable global con una variable privada para imponer la variable privada en componentes secundarios del componente.
9. El procedimiento de la reivindicación 8, que comprende adicionalmente buscar (900) para la definición de componente basándose en un tipo de datos del componente solicitado cuando no se encuentra la definición de componente.
- 50 10. El procedimiento de la reivindicación 8, que comprende adicionalmente crear (902) un componente personalizado basándose en la ausencia del componente solicitado.
11. El procedimiento de una de las reivindicaciones 8 a 10, que comprende adicionalmente aplicar (904; 1008) una variable global al componente para posibilitar el intercambio de datos entre contextos de datos no relacionados.

12. El procedimiento de una de las reivindicaciones 8 a 11, que comprende adicionalmente:

crear (908; 1102) un componente de contenedor cuando no se encuentra el componente solicitado;
cargar (910; 1104) el componente de contenedor con propiedades de tipo de datos asociadas a componentes de base; y

5 emitir (912; 1106) el componente de contenedor como el componente.

13. El procedimiento de una de las reivindicaciones 8 a 12, que comprende adicionalmente componer (1110) un componente primario que incluye uniones de propiedades a componentes secundarios y uniones de las propiedades a los componentes de datos.

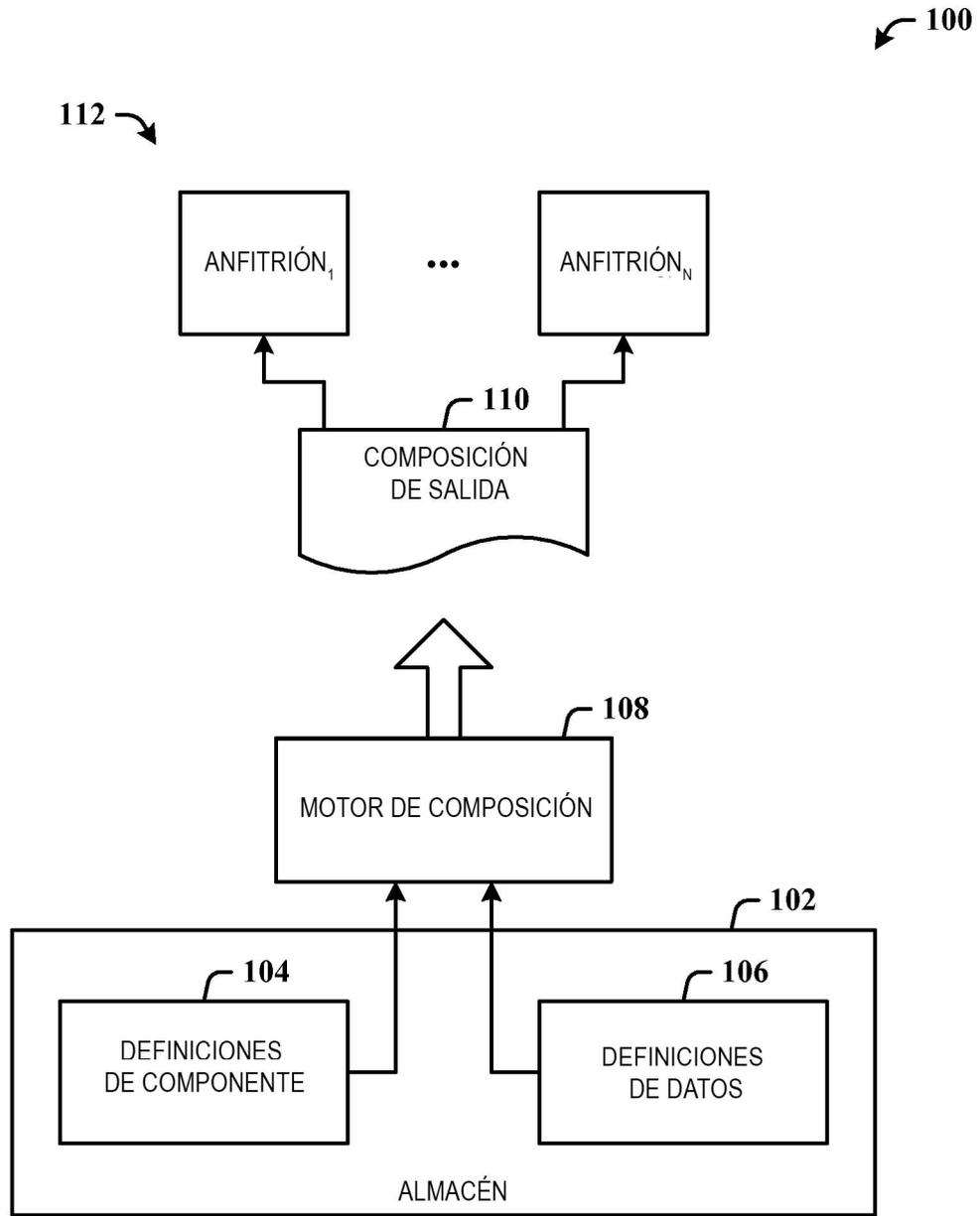


FIG. 1

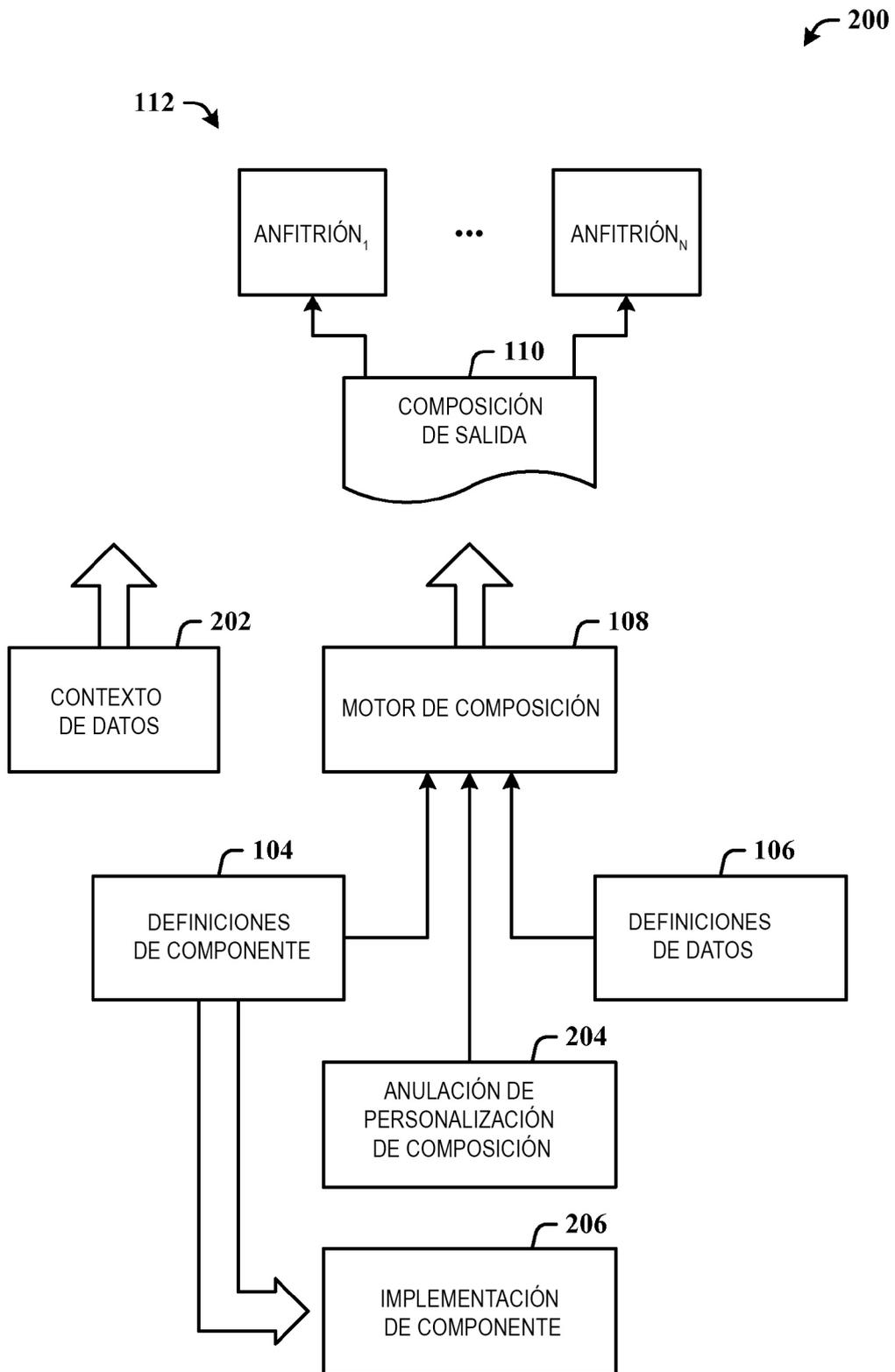


FIG. 2

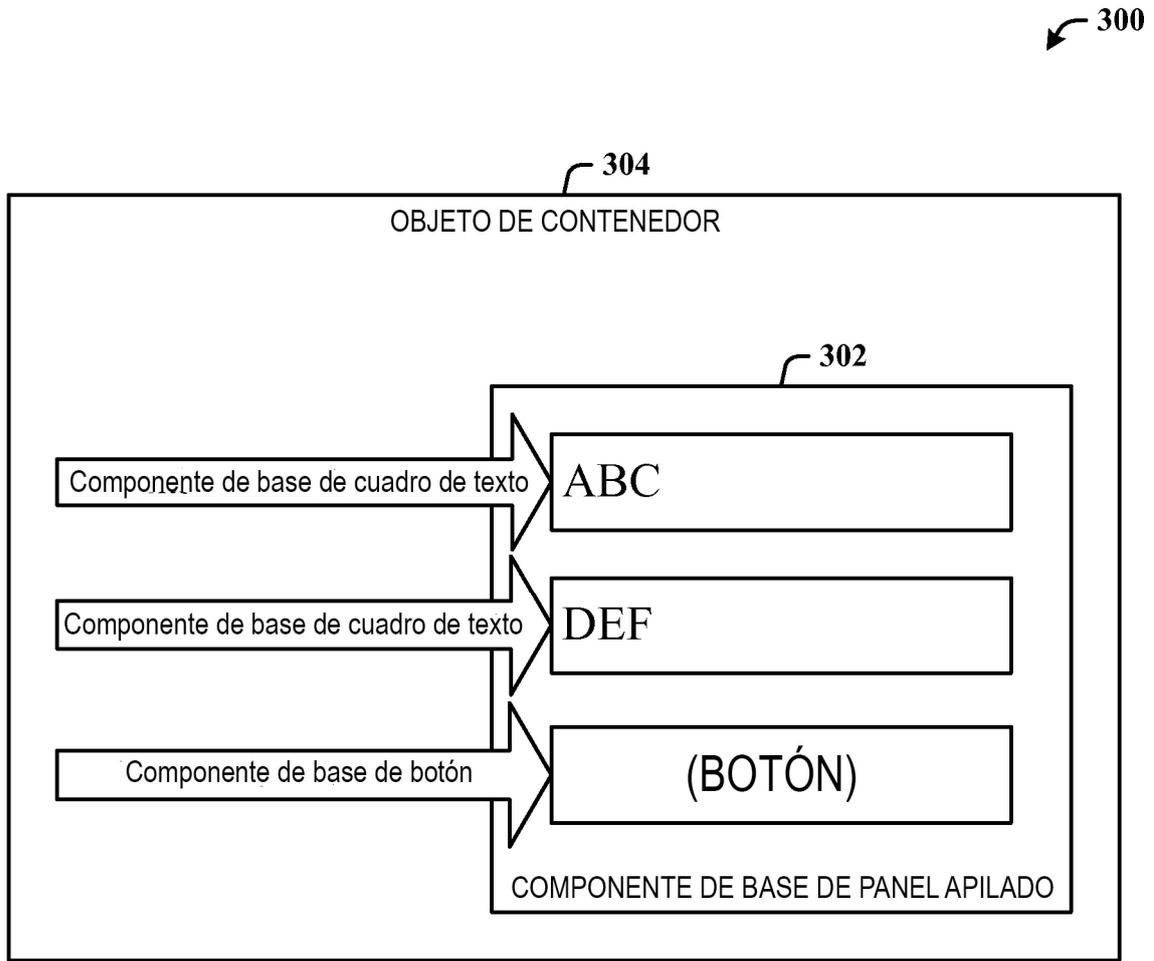


FIG. 3

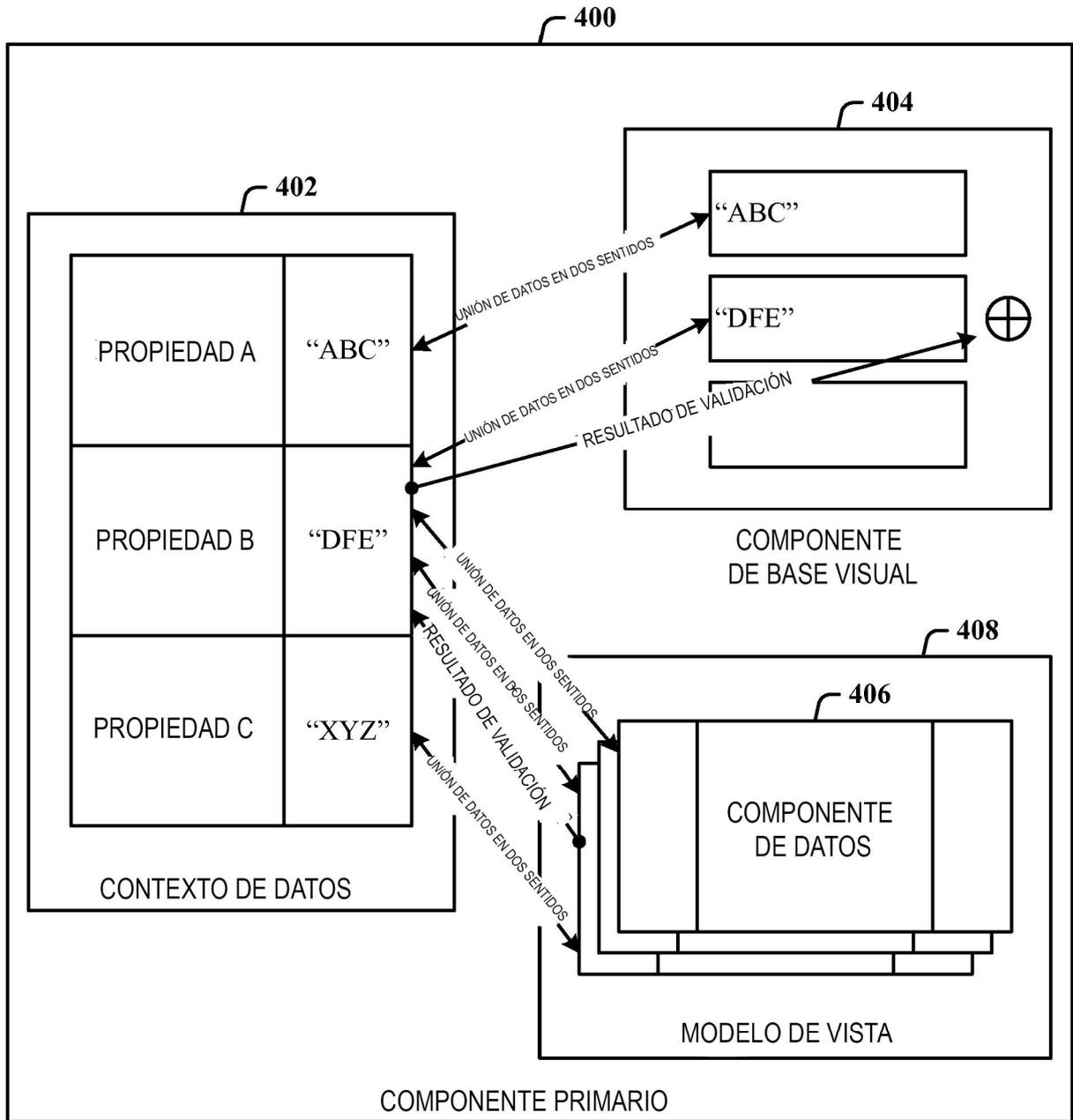


FIG. 4

500

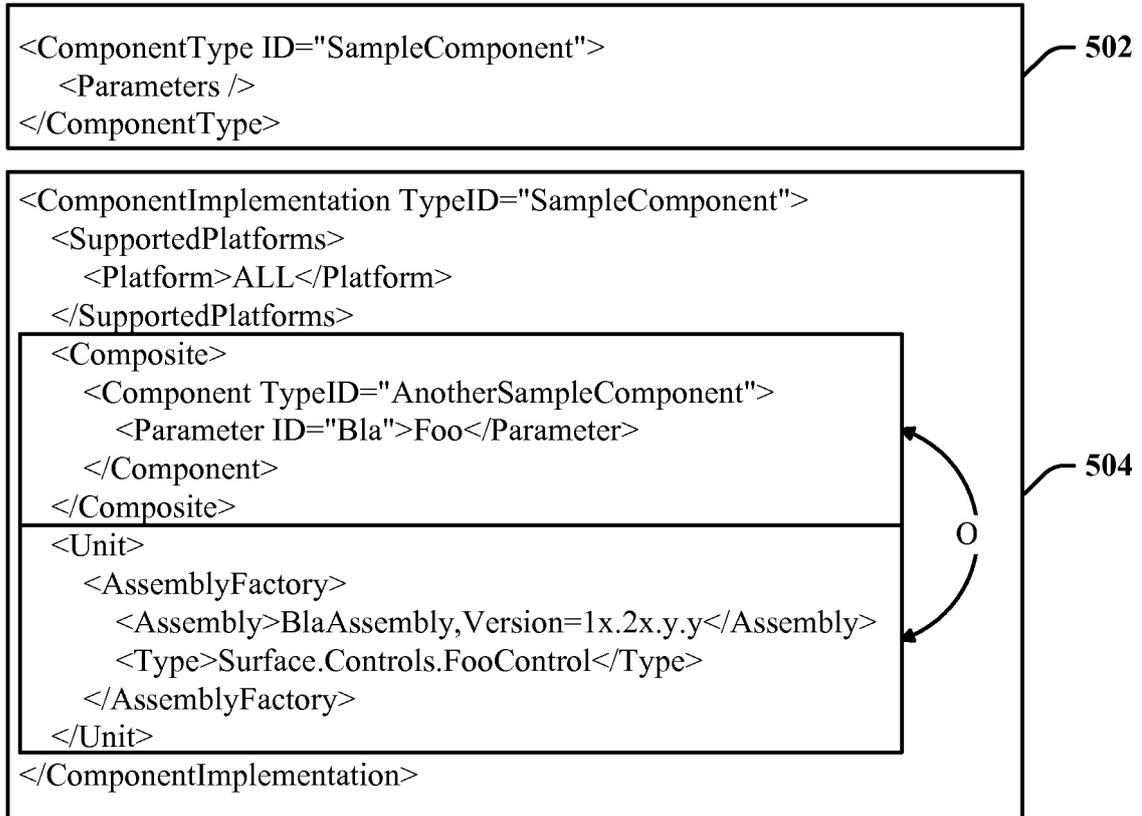


FIG. 5



FIG. 6

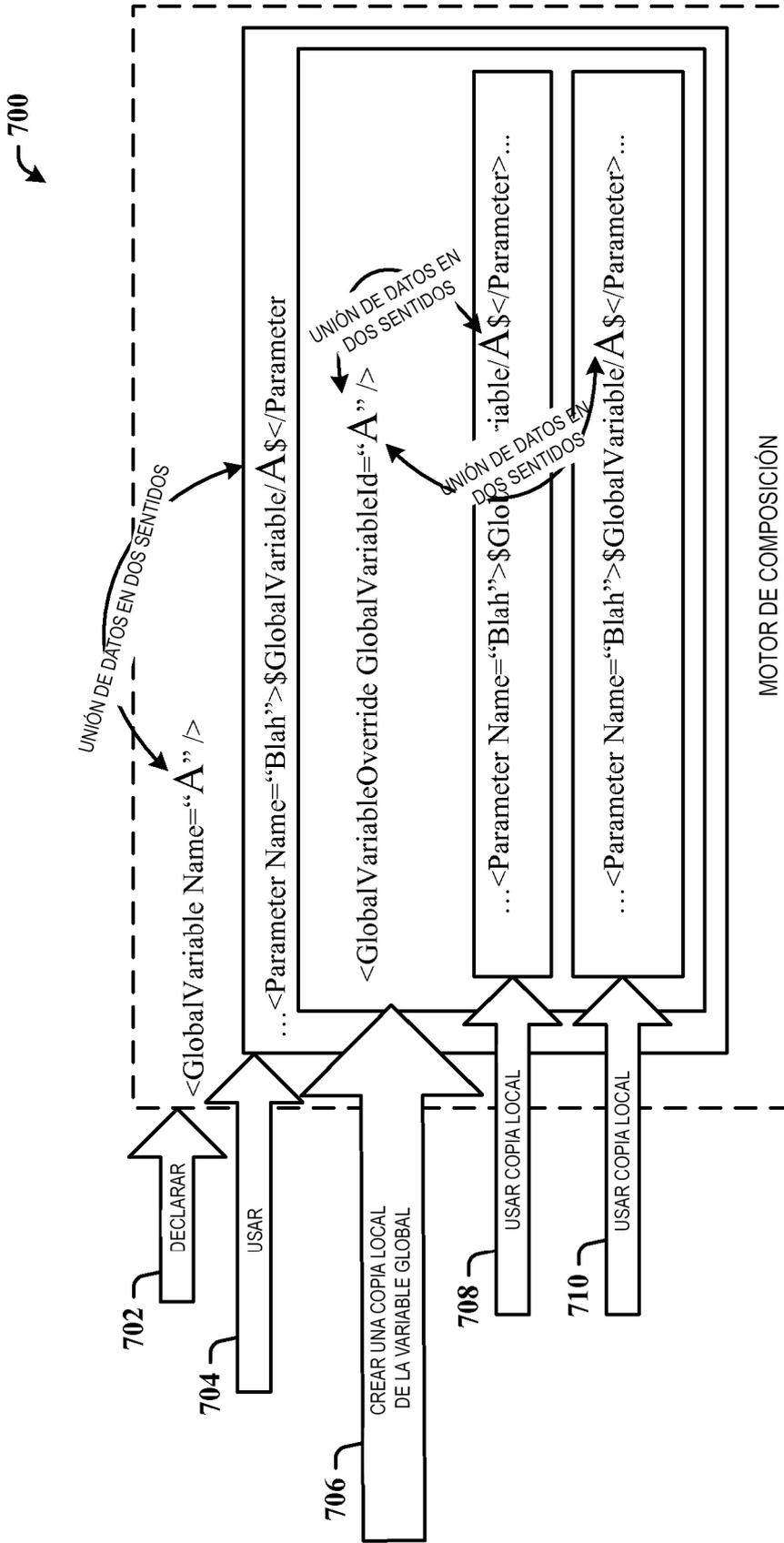


FIG. 7

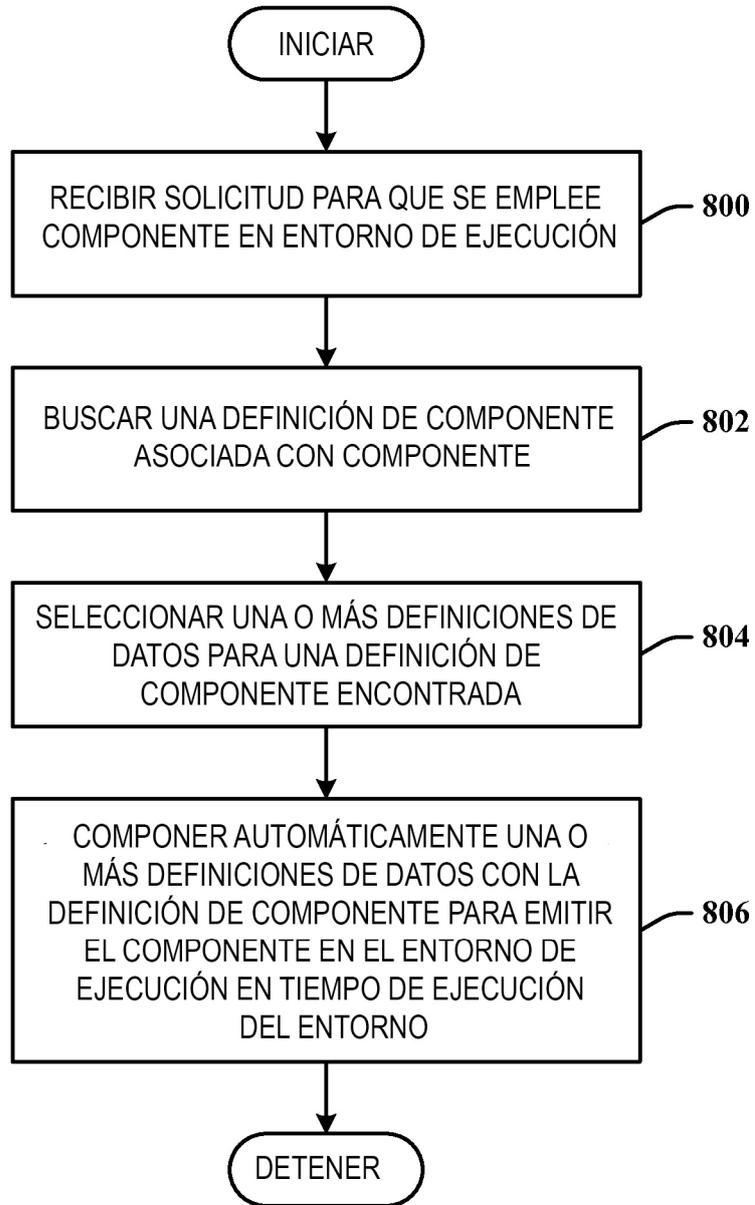


FIG. 8

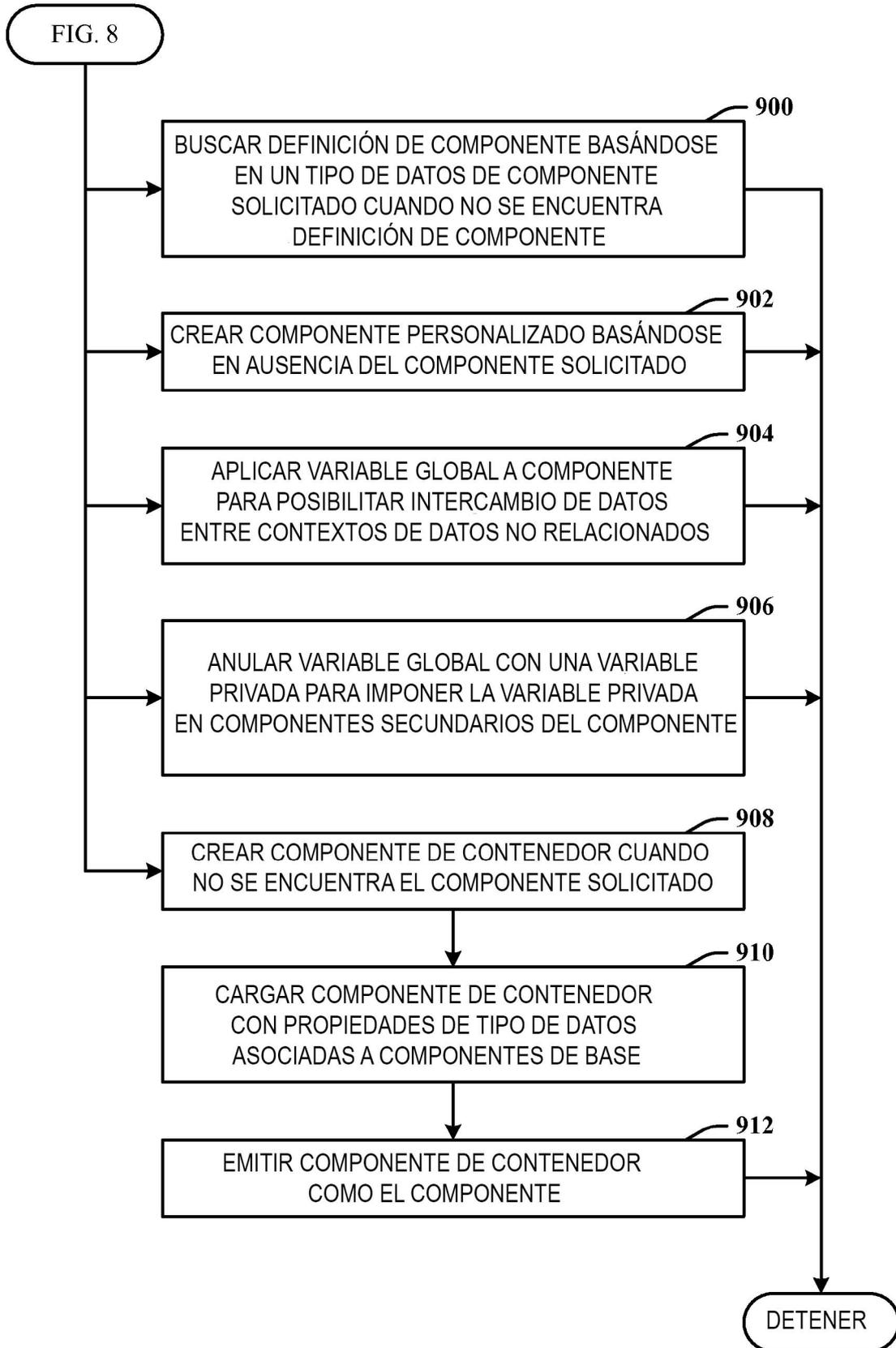


FIG. 9

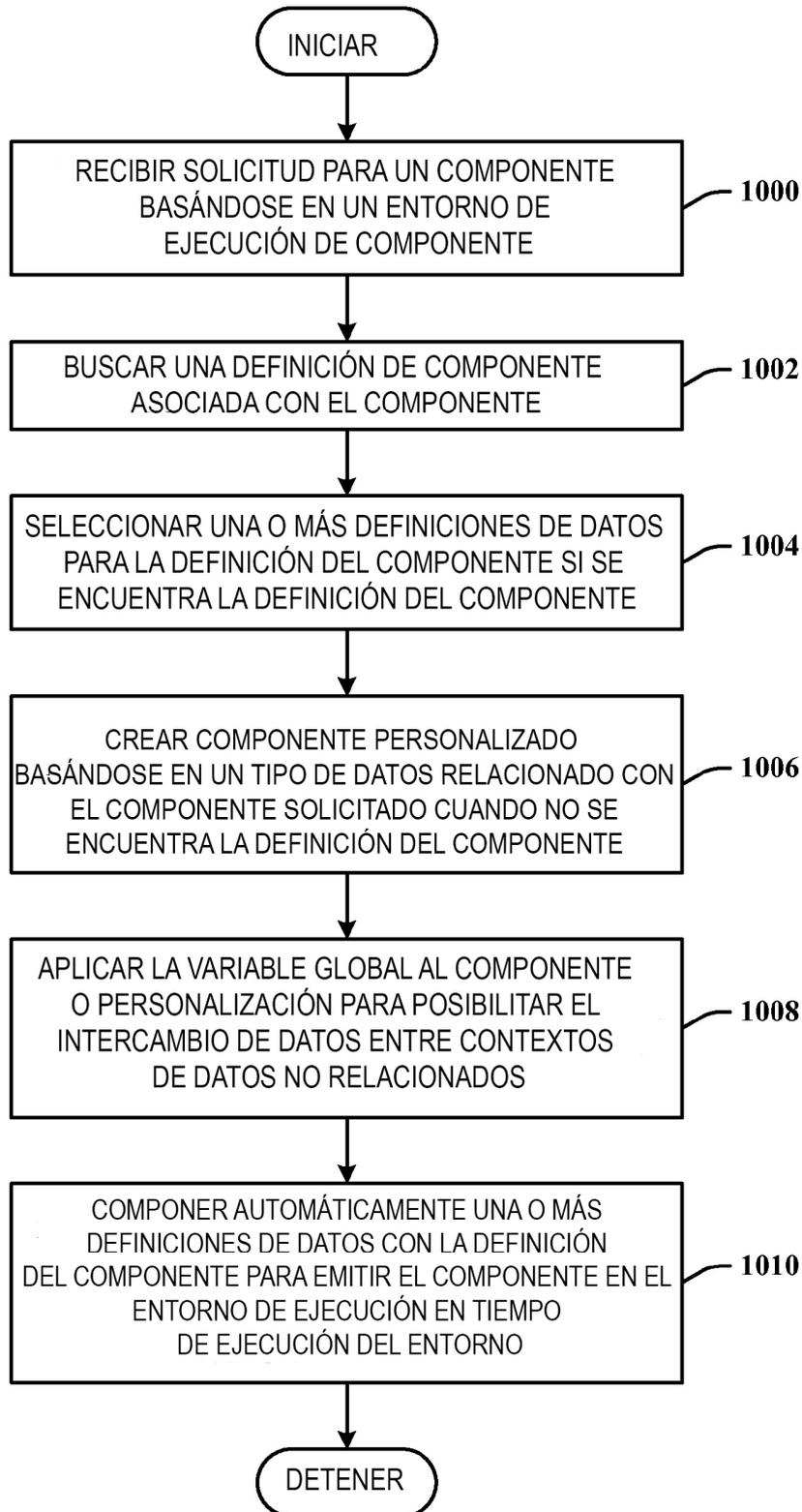


FIG. 10

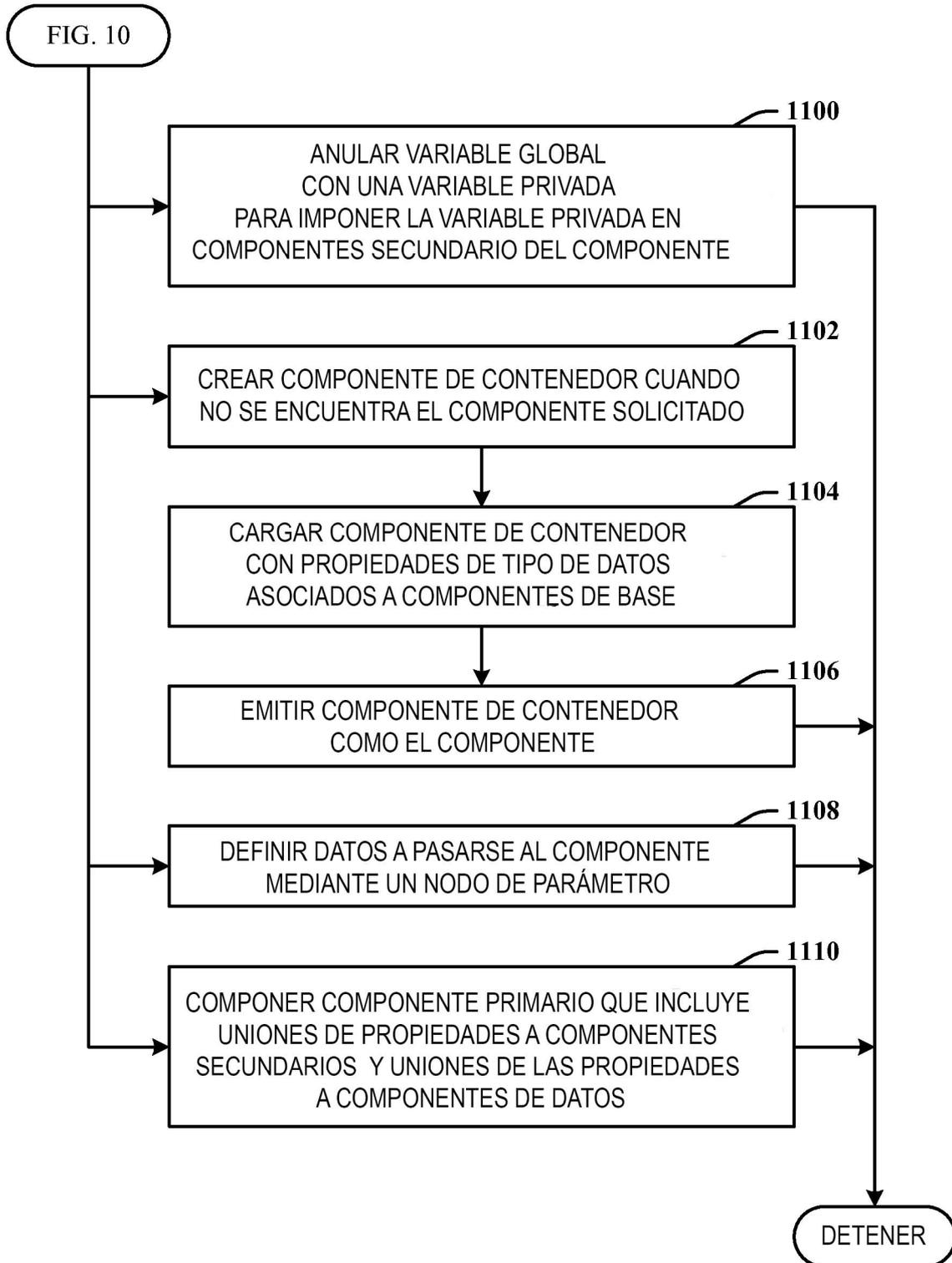


FIG. 11

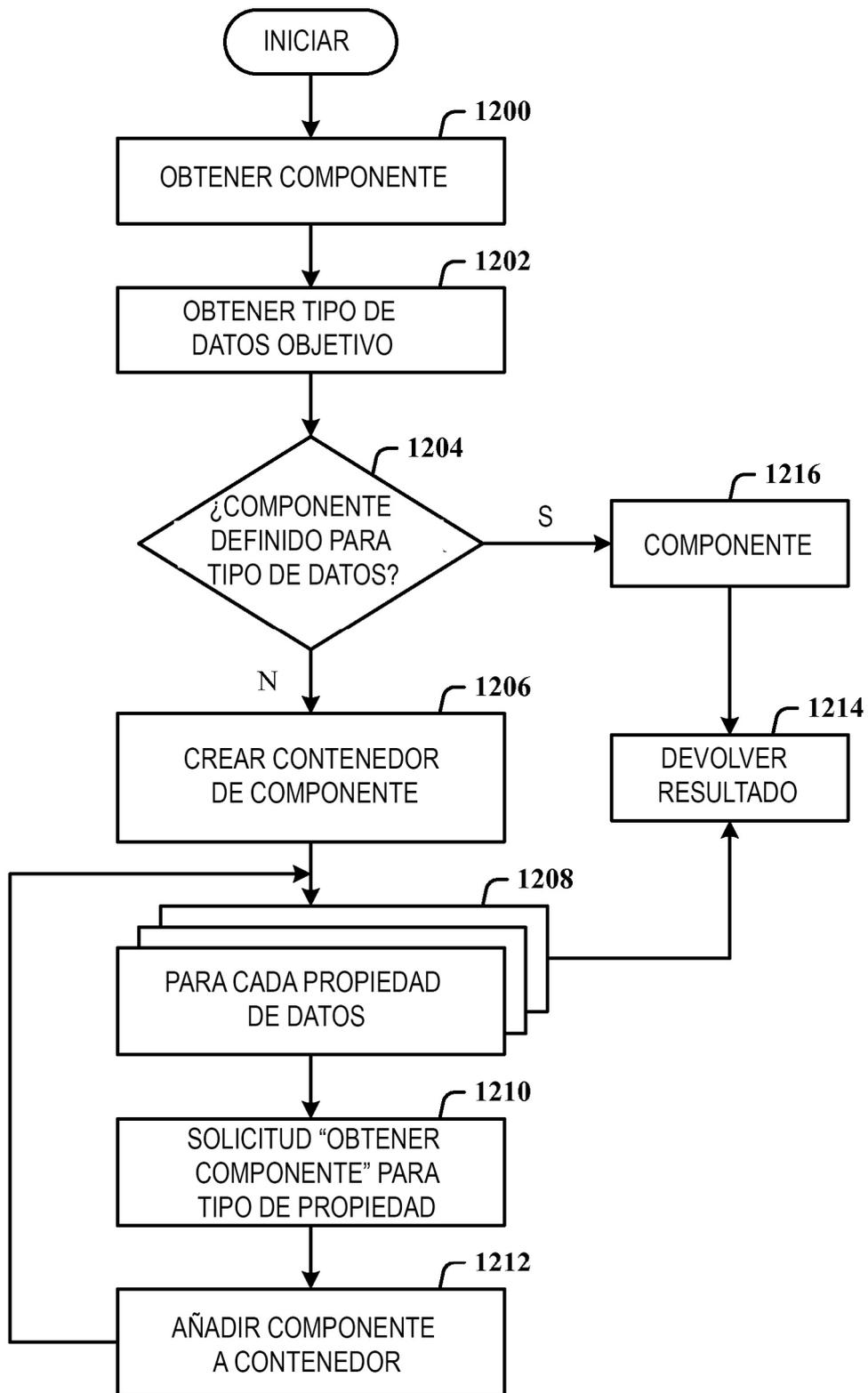


FIG. 12

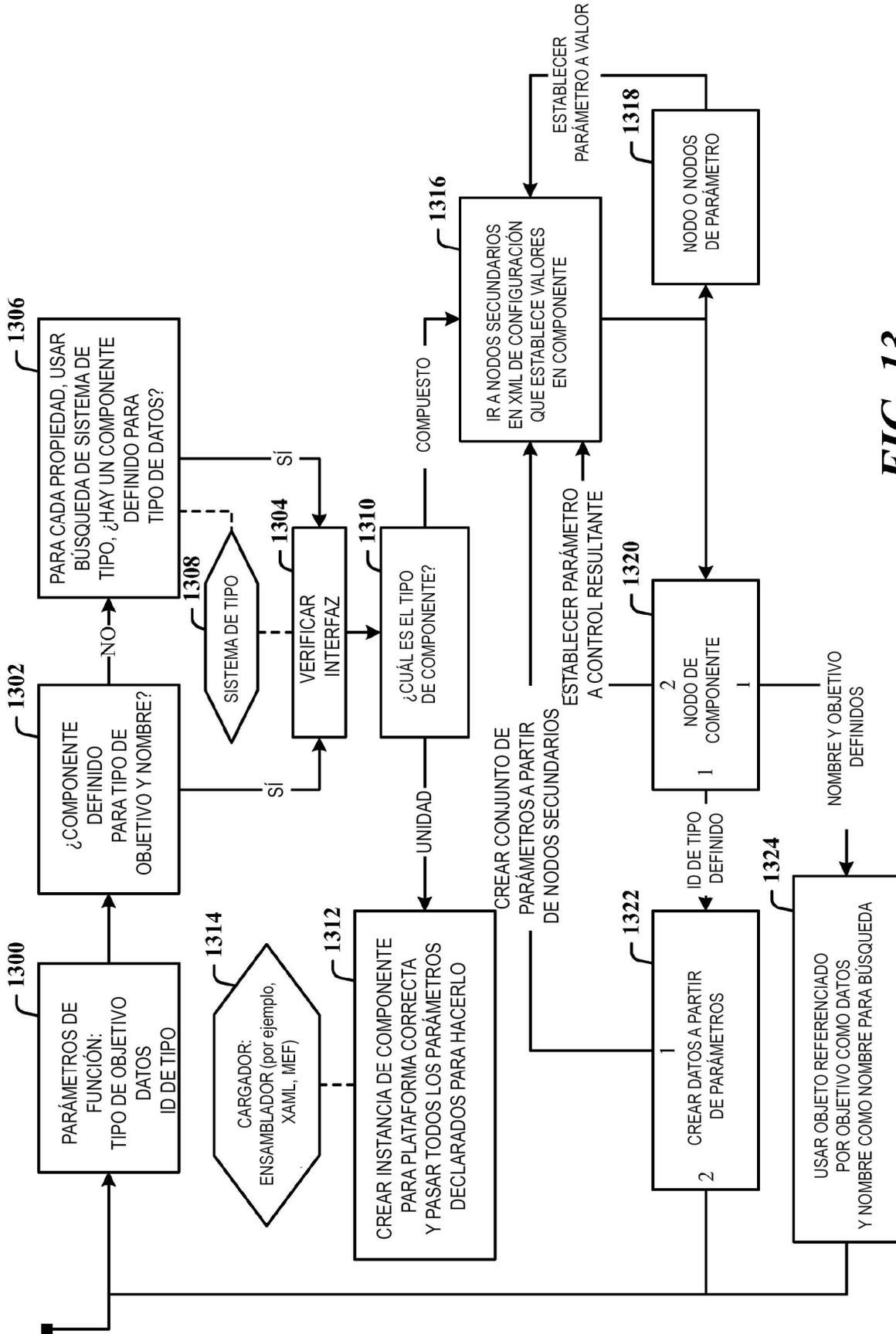


FIG. 13

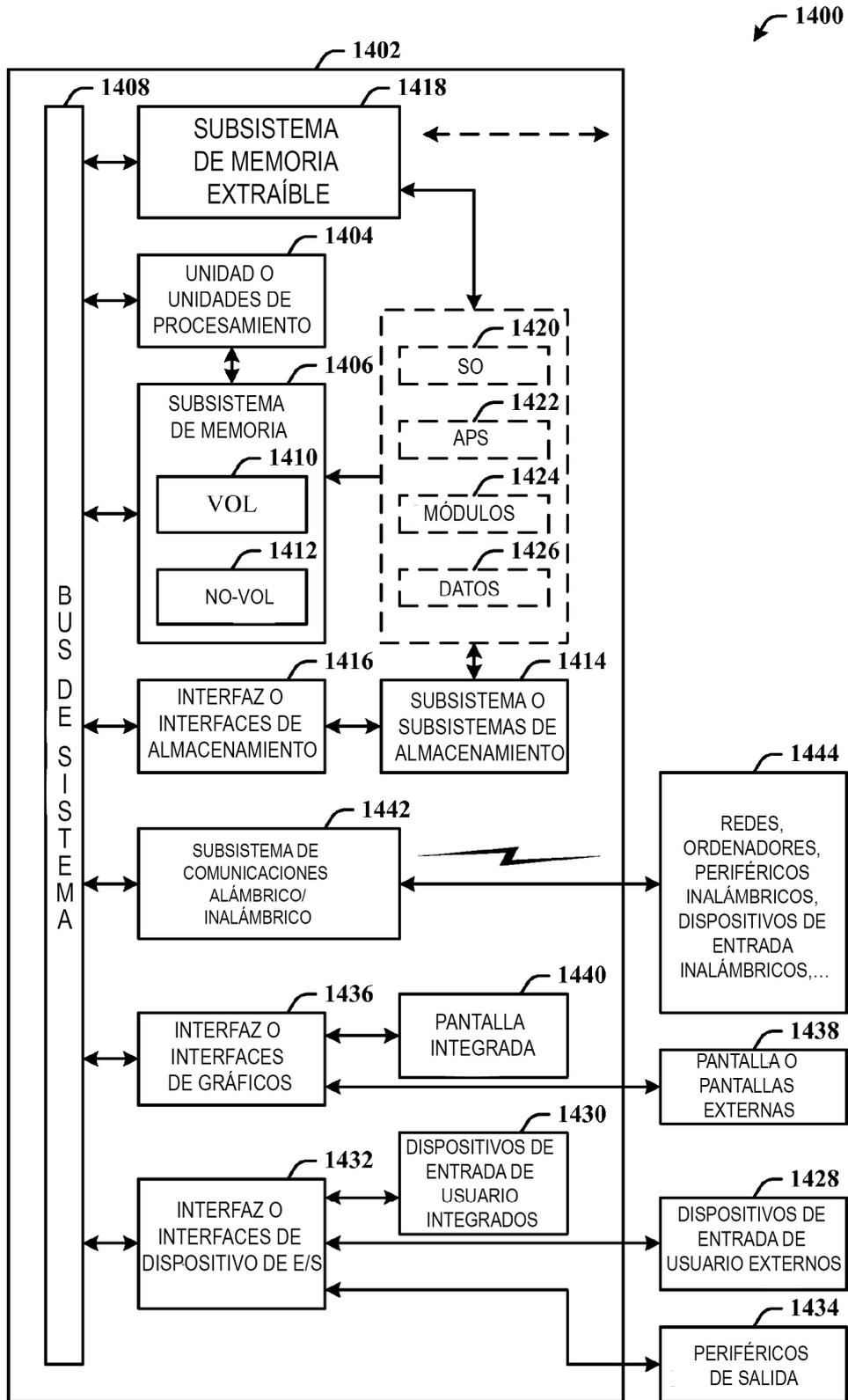


FIG. 14