

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 372 956**

51 Int. Cl.:

**H04L 29/06** (2006.01)

**G06F 9/42** (2006.01)

**H04L 29/08** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **05110545 .0**

96 Fecha de presentación: **09.11.2005**

97 Número de publicación de la solicitud: **1667399**

97 Fecha de publicación de la solicitud: **07.06.2006**

54

Título: **TRANSFERENCIA DE FORMA FLEXIBLE DE DATOS DE APLICACIÓN TИPEADOS.**

30

Prioridad:  
**03.12.2004 US 3099**

45

Fecha de publicación de la mención BOPI:  
**30.01.2012**

45

Fecha de la publicación del folleto de la patente:  
**30.01.2012**

73

Titular/es:  
**MICROSOFT CORPORATION  
ONE MICROSOFT WAY  
REDMOND WA 98052, US**

72 Inventor/es:

**Wortendyke, David;  
Pharies, Stefan Harrington;  
Balayoghan, Vaithialingam B.;  
Purdy, Douglas M.;  
Christensen, Erik Bo;  
Wolf, Kenneth David;  
Coulson, Michael James;  
Maruchek, Michael Jon;  
Vernal, Michael Steven;  
Jethanandani, Natasha Harish y  
Sturgell, Ryan Thomas**

74

Agente: **Carpintero López, Mario**

ES 2 372 956 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

Transferencia de forma flexible de datos de aplicación tipeados

**Referencias cruzadas a aplicaciones relacionadas**

No aplicable

5 **Antecedentes de la invención**

1. **El campo de la invención**

La presente invención se refiere a mensajería electrónica y, más particularmente, a la transferencia de forma flexible de datos de aplicación tipeados.

2. **Antecedentes y técnica relevante**

10 Los sistemas de ordenadores y la tecnología relacionada afectan a muchos aspectos de la sociedad. Verdaderamente la capacidad de los sistemas de ordenadores para procesar información ha transformado el modo en el que vivimos y trabajamos. Los sistemas de ordenadores ahora realizan comúnmente una serie de tareas (por ejemplo, el procesamiento de textos, programación, y gestión de bases de datos) que antes del advenimiento de los sistemas de ordenadores se realizaban manualmente. Más recientemente, los sistemas de ordenadores se han  
15 acoplado a unos y otros dispositivos electrónicos para formar ambas redes de ordenadores cableadas e inalámbricas sobre las cuales los sistemas de ordenadores y otros dispositivos electrónicos pueden transferir datos electrónicos. Como resultado, muchas tareas realizadas en un sistema de ordenador (por ejemplo, la comunicación de voz, el acceso al correo electrónico, el control de dispositivos electrónicos domésticos, la búsqueda de páginas Web, y la impresión de documentos) incluyen el intercambio de mensajes electrónicos entre varios sistemas de  
20 ordenadores y/o otros dispositivos electrónicos a través de redes de ordenadores cableadas y/o inalámbricas.

Las redes de hecho se han vuelto tan prolíficas, que un simple sistema de computación habilitado para la red puede comunicar con uno cualquiera de millones de otros sistemas de computación difundidos a través del globo sobre una conglomeración de redes a menudo denominada como la "Internet". Tales sistemas de computación pueden incluir ordenadores personales de sobremesa, ordenadores personales portátiles o tabletas; Asistentes Digitales  
25 Personales (PDA); teléfonos; o cualquier otro ordenador o dispositivo capaz de comunicar sobre una red digital.

Para comunicar sobre una red, un sistema de computación (denominado en el presente documento como un "sistema de computación de envío" construye o de otro modo accede a un mensaje electrónico y transmite el mensaje electrónico sobre una red a otro sistema de computación (denominado en el presente documento como un "sistema de computación receptor"). El mensaje electrónico puede leerse por un usuario humano como cuando el  
30 mensaje electrónico es un correo electrónico o un mensaje instantáneo, o puede leerse, en cambio por una aplicación que corre sobre el sistema de computación receptor. El mensaje electrónico puede estar construido por una aplicación que corre sobre el sistema de computación de envío, con la posible asistencia de un usuario humano.

De este modo en algunos entornos, los mensajes se intercambian entre aplicaciones distribuidas con una pequeña interacción directa con el usuario, si la hay. Un protocolo usado para el intercambio de mensajes entre aplicaciones distribuidas (por ejemplo, los servicios Web) es el Protocolo de Acceso de Objetos Simples ("SOAP"). El SOAP es un protocolo ligero previsto para el intercambio de información en un entorno descentralizado, distribuido. El SOAP usa las tecnologías del Lenguaje de Marcado extensible ("XML") para definir una estructura de mensajería extensible que proporciona construcciones de mensajes que pueden intercambiarse sobre una diversidad de protocolos subyacentes, tales como, por ejemplo el Protocolo de Transferencia de Correo Simple ("SMTP"), el Protocolo de Transferencia de Hipertextos ("HTTP"), el Protocolo de Datagramas de Usuario ("UDP"), el Protocolo de Control de Transmisión ("TCP") y las canalizaciones con nombre. Los mensajes del SOAP pueden definirse en un asunto abierto como un conjunto de información del SOAP (por ejemplo, los elementos de XML) que tienen un conjunto de cabeceras de mensajes y un cuerpo del mensaje.  
40

Muchas aplicaciones están diseñadas usando un procedimiento y/o modelos de programación orientados a objetos. Estos modelos de programación frecuentemente usan formatos de tipos de datos (por ejemplo, números enteros, cadenas de caracteres, formatos lógicos o formatos de datos definidos por el usuario) para intercambiar objetos entre módulos. Por ejemplo, un procedimiento puede recibir uno o más objetos tipeados (por ejemplo, parámetros de diversos formatos de tipos de datos diferentes) como entrada. En respuesta, el procedimiento puede generar y sacar valores (potencialmente diferentes) para los objetos tipeados o puede devolver el objeto tipeado de un tipo de datos diferente (por ejemplo, el tipo de datos del procedimiento). El uso de los formatos de datos tipeados es ventajoso ya  
50 que permite al desarrollador diseñar aplicaciones y módulos que usan tipos de datos que son más apropiados y/o más eficaces en base a la aplicación y/o funcionalidad del módulo.

De este modo en entornos de aplicación distribuidos, puede ser que los objetos tipeados se tengan que intercambiar entre diferentes porciones de una aplicación distribuida. Por ejemplo, para facilitar una transición bancaria, un cliente de banco distribuido puede intercambiar una estructura de datos tipeados de la transacción bancaria con un servidor  
55

bancario distribuido. Sin embargo, puede ser difícil diseñar aplicaciones distribuidas que utilicen las ventajas de los formatos de datos tipeados junto con la flexibilidad de los mensajes del SOAP.

El SOAP es un protocolo abierto y de este modo virtualmente no establece requisitos sobre la aplicación de mapeo y/o los datos de usuario dentro de un mensaje del SOAP o viceversa. De este modo, hay mecanismos limitados, si hay alguno, para el mapeo de los objetos tipeados (por ejemplo, los tipos del Lenguaje Común del Tiempo de ejecución ("CLR")) para los mensajes del SOAP correspondientes y para el mapeo de los mensajes del SOAP a los objetos tipeados correspondientes. Una incapacidad para el mapeo significativo entre objetos tipeados y mensajes del SOAP puede hacer más difícil, o incluso imposible, para las aplicaciones distribuidas el intercambio de datos de usuario y/o de aplicación.

Además, los mecanismos típicos para la codificación de mensajes del SOAP dentro de octetos puros limitan el tamaño de los mensajes del SOAP y limitan los tipos de codificadores que pueden usarse. Por lo tanto los sistemas, procedimientos, y productos de programas de ordenador para la transferencia de forma flexible de datos de aplicaciones tipeados serían ventajosos.

La puesta en serie de objetos en la estructura de red .NET usando el Formateador del SOAP se describe en el documento XP-002372766.

### **Breve resumen de la invención**

Los problemas anteriores con el estado anterior de la técnica se superan por los principios de la presente invención, que se dirigen hacia sistemas, procedimientos y productos de programas de ordenador para la transferencia de forma flexible de datos tipeados de aplicación. En algunas realizaciones, un sistema de ordenador configura un mensaje del SOAP para transmitir datos tipeados. El código de aplicación (por ejemplo, un cliente proxy) llama a un objeto tipeado (por ejemplo, un procedimiento) con parámetros del objeto tipeado (por ejemplo, parámetros para una invocación de procedimiento). El sistema de ordenador accede a los parámetros del objeto tipeado correspondiente al objeto tipeado. El objeto tipeado está anotado con uno o más de uno atributos de declaración del mensaje de un modelo de declaración de mensajes. El uno o más atributos de declaración del mensaje definen un mapeo entre los parámetros del objeto tipeado y los elementos del SOAP correspondientes.

El sistema de ordenador mapea los parámetros del objeto tipeado accedido en elementos del SOAP de acuerdo con el uno o más atributos de declaración del mensaje. El sistema de ordenador se refiere al uno o más atributos de declaración del mensaje para identificar una localización (por ejemplo, una cabecera del mensaje o el cuerpo del mensaje) dentro de una envoltura del SOAP. El sistema de ordenador inserta el elemento del SOAP en la localización identificada dentro de la envoltura del SOAP. La envoltura del SOAP puede situarse a continuación en un canal, codificarse dentro de octetos puros, y transferir los octetos puros correspondientes a otro sistema de ordenador.

En otras realizaciones, un sistema de ordenador recupera los datos tipeados a partir del mensaje de un SOAP. El sistema de ordenador puede recibir octetos puros desde otro sistema de ordenador, decodificar los octetos puros dentro de una envoltura del SOAP correspondiente, y colocar la envoltura del SOAP en un canal. El sistema de ordenador accede al elemento del SOAP incluido en la envoltura del SOAP.

El sistema de ordenador identifica el objeto tipeado que corresponde al elemento del SOAP. El objeto tipeado se anota con uno o más atributos de declaración del mensaje del modelo de declaración de mensajes. El uno o más atributos de declaración del mensaje definen un mapeo entre los elementos del SOAP y los parámetros del objeto tipeado correspondiente. El sistema de ordenador mapea el elemento del SOAP accedido en parámetros del objeto tipeado de acuerdo con el uno o más atributos de declaración del mensaje. Los parámetros del objeto tipeado se presentan al código de la aplicación (por ejemplo, un servicio) que incluye el objeto tipeado identificado.

Estos y otros objetos y características de la presente invención se harán más completamente evidentes a partir de la siguiente descripción y las reivindicaciones adjuntas, o pueden aprenderse por la práctica de la invención como se mostrará en adelante en el presente documento.

### **Breve descripción de los dibujos**

Para clarificar adicionalmente las anteriores y otras ventajas y características de la presente invención, se presentará una descripción más particular de la invención por referencia a realizaciones específicas de la misma que se ilustran en los dibujos adjuntos. Se aprecia que estos dibujos representan sólo realizaciones típicas de la invención y por lo tanto, no se deben considerar como limitativas de su alcance. La invención se describirá y explicará con una especificidad adicional y detalles mediante el uso de los dibujos adjuntos en los que:

La Figura 1 ilustra un ejemplo de arquitectura de ordenador que facilita el uso del Protocolo de Acceso a Objetos Simples para objetos tipeados intercambiados.

La Figura 2 ilustra un diagrama de flujo de ejemplo de un procedimiento de envío de modo flexible de datos de aplicación tipeados.

La Figura 3 ilustra un diagrama de flujo de ejemplo de un procedimiento de recepción de modo flexible de datos de aplicación tipeados.

La Figura 4 ilustra un entorno de funcionamiento adecuado para los principios de la presente invención.

**Descripción detallada de las realizaciones preferidas**

5 Los problemas anteriores con el estado anterior de la técnica se superan por los principios de la presente invención, que se dirigen hacia procedimientos, sistemas, y productos de programas de ordenador para la transferencia de modo flexible de datos tipeados de aplicación. En algunas realizaciones, un sistema de ordenador configura un mensaje del SOAP para transmitir datos tipeados. El código de aplicación (por ejemplo, un cliente proxy) llama a un objeto tipeado (por ejemplo, un procedimiento) con parámetros del objeto tipeado (por ejemplo, parámetros para una invocación del procedimiento). El sistema de ordenador accede a los parámetros del objeto tipeado correspondiente al objeto tipeado. El objeto tipeado está anotado con uno o más atributos de declaración del mensaje de un modelo de declaración de mensajes. El uno o más atributos de declaración del mensaje definen un mapeo entre los parámetros del objeto tipeado y los elementos del SOAP correspondientes.

10 El sistema de ordenador mapea los parámetros del objeto tipeado accedido en elementos del SOAP de acuerdo con los uno o más atributos de declaración del mensaje. El sistema de ordenador se refiere al uno o más atributos de declaración del mensaje para identificar una localización (por ejemplo, una cabecera del mensaje o el cuerpo del mensaje) dentro de una envolvente del SOAP. El sistema de ordenador inserta el elemento del SOAP en la localización identificada dentro de la envolvente del SOAP. La envolvente del SOAP puede situarse a continuación en un canal, codificarse dentro de los octetos puros correspondientes, y transferirse los octetos puros correspondientes a otro sistema de ordenador.

15 En otras realizaciones, un sistema de ordenador recupera los datos tipeados a partir del mensaje del SOAP. El sistema de ordenador puede recibir octetos puros desde otro sistema de ordenador, decodificar los octetos puros en una envolvente del SOAP correspondiente y colocar la envolvente del SOAP en un canal. El sistema de ordenador accede a un elemento del SOAP incluido en la envolvente del SOAP.

20 El sistema de ordenador identifica el objeto tipeado que corresponde al elemento del SOAP. El objeto tipeado se anota con uno o más atributos de declaración del mensaje de un modelo de declaración de mensajes. El uno o más atributos de declaración del mensaje definen un mapeo entre los elementos del SOAP y los parámetros del objeto tipeado correspondientes. El sistema de ordenador mapea el elemento del SOAP accedido en parámetros del objeto tipeado de acuerdo con el uno o más atributos de declaración del mensaje. Los parámetros del objeto tipeado se presentan al código de la aplicación (por ejemplo, un servicio) que incluye el objeto tipeado identificado.

25 Las realizaciones dentro del alcance de la presente invención incluyen medios legibles por el ordenador para transportar o tener las instrucciones ejecutables por el ordenador o las estructuras de datos almacenadas en los mismos. Tales medios legibles por el ordenador pueden ser cualesquiera medios disponibles, que son accesibles por un sistema de ordenador de propósito general o de propósito especial. A modo de ejemplo, y no de limitación, tales medios legibles por ordenador pueden comprender medios de almacenamiento físico tales como una RAM, ROM, EPROM, CD-ROM u otro almacenamiento de disco óptico, almacenamiento de disco magnético u otros dispositivos de almacenamiento magnético, o cualquier otro medio que pueda usarse para transportar o almacenar los medios del código de programa deseado en la forma de instrucciones ejecutables por el ordenador, instrucciones legibles por el ordenador, o estructuras de datos y que pueden accederse por un sistema de ordenador de propósito general o de propósito especial.

30 En esta descripción y en las siguientes reivindicaciones, se define una "red" como uno o más enlaces de datos que posibilitan el transporte de datos electrónicos entre sistemas de ordenadores y/o módulos. Cuando se transfiere la información o se proporciona sobre una red u otra conexión de comunicaciones (bien cableada, inalámbrica o una combinación de cableada o inalámbrica) a un sistema de ordenadores, la conexión se ve propiamente como el medio legible por el ordenador. De este modo, cualquiera de tales conexiones se denomina adecuadamente como un medio legible por ordenador. También deberían incluirse combinaciones de las anteriores dentro del alcance de los medios legibles por ordenador. Las instrucciones ejecutables por el ordenador comprenden, por ejemplo, instrucciones y datos que causan que un sistema de ordenador de propósito general o un sistema de ordenador de propósito especial realice cierta función o grupo de funciones. Las instrucciones ejecutables por el ordenador pueden ser por ejemplo, binarias, instrucciones de formato intermedio, tal como un lenguaje ensamblador, o incluso un código fuente.

35 En esta descripción y en las siguientes reivindicaciones, se define un "sistema de ordenador" como uno o más módulos software, uno o más módulos hardware, o combinaciones de los mismos, que funcionan juntos para realizar operaciones sobre datos electrónicos. Por ejemplo, la definición de un sistema de ordenador incluye los componentes hardware del ordenador personal, así como los módulos software, tales como el sistema operativo del ordenador personal. La capa física de los módulos no es importante. Un sistema de ordenador puede incluir uno o más ordenadores acoplados a través de una red. Del mismo modo, un sistema de ordenador puede incluir un dispositivo físico único (tal como un teléfono móvil o un Asistente Digital Personal "PDA") donde módulos internos

(tales como una memoria y un procesador) funcionan juntos para realizar operaciones sobre datos electrónicos.

En esta descripción y en las siguientes reivindicaciones, se define un "objeto" como cualquier elemento que puede seleccionarse y manipularse de forma individual. El objeto se define para incluir los objetos software, tales como por ejemplo, los procedimientos y otros objetos de software, que incluyen datos y procedimientos para manipular los datos. El objeto también se define para incluir parámetros y devolver valores (por ejemplo, los parámetros del procedimiento y los valores devueltos por el procedimiento) que pueden intercambiarse entre objetos software (por ejemplo, proporcionarse como entrada o salida de).

En esta descripción y en las siguientes reivindicaciones, se define una "declaración de mensaje" como una declaración, tal como, por ejemplo un modelo de programación declarativo, que describe un formato del Protocolo de Acceso a Objetos Simples ("SOAP"). Una declaración de un mensaje puede estar basada en los atributos del Lenguaje Común del Tiempo de ejecución que define las porciones de acción, cabeceras, y cuerpo de una envolvente del SOAP. Los atributos pueden usarse sobre un tipo para controlar el modo en el cual las envolventes del SOAP se construyen a partir de, por ejemplo, procedimientos, parámetros de procedimiento, y valores devueltos por el procedimiento. Los campos y/o propiedades a incluir en una declaración de mensaje pueden anotarse, por ejemplo como, [CabeceraMensaje] o [CuerpoMensaje].

Los especialistas en la técnica apreciarán que la invención puede ponerse en práctica en entornos de computación de red con muchos tipos de configuraciones del sistema de ordenador, incluyendo, ordenadores personales, ordenadores portátiles, dispositivos de mano, sistemas de multi-procesador, dispositivos electrónicos de usuario basados en microprocesadores o programables, PC en red, miniordenadores, ordenadores centrales, teléfonos móviles, PDA, buscapersonas, y similares. La invención también puede ponerse en práctica en entornos de sistemas distribuidos donde los sistemas de ordenadores locales y remotos, que están enlazados (bien por enlaces de datos cableados, enlaces de datos inalámbricos, o por una combinación de enlaces inalámbricos y cableados) a través de una red, realizan ambos tareas. En un entorno de sistema distribuido, los módulos de programa pueden localizarse tanto en dispositivos de almacenamiento de memoria locales como remotos.

Como se ha representado en la arquitectura de ordenadores 100, los sistemas de ordenadores 101 y 111 están conectados a la red 161. La red 161 puede ser una Red de Área Local ("LAN"), una Red de Área Ancha ("WAN"), o incluso la Internet. Los sistemas de ordenadores conectados a la red 161 pueden recibir datos de otros sistemas de ordenadores conectados a la red 161 y enviar datos a los mismos. Por consiguiente, los sistemas de ordenadores 101 y 111, así como otros sistemas de ordenadores conectados (no mostrados), pueden crear datos relacionados con el mensaje e intercambiar datos relacionados con el mensaje (por ejemplo, datagramas del Protocolo de Internet ("IP") y otros protocolos de capa superior que utilizan los datagramas de IP, tales como, el Protocolo de Control de Transmisión ("TCP"), el Protocolo de Transferencia de Hipertexto ("HTTP"), el Protocolo de Transferencia de Correo Simple ("SMTP"), etc.) sobre la red 101. Por ejemplo, los sistemas de ordenadores 101 y 111 pueden crear envolventes del SOAP e intercambiar envolventes del SOAP sobre la red 161.

El sistema de ordenador 101 incluye un código de aplicación 102, un convertidor de tipos 103, canales 106, codificadores 107, y almacenamiento 163. El código de aplicación 102 puede ser una porción de una aplicación distribuida, tal como, por ejemplo, un cliente proxy correspondiente al código de aplicación 112. El código de aplicación 102 puede ser un código que se desarrolló de acuerdo con un procedimiento o modelo de programación orientado a objetos. De este modo, los datos intercambiados entre los módulos de código de aplicación 102, entre los módulos de código de aplicación 102 y los módulos del código de aplicación 112, y entre los módulos del código de aplicación 102 y otros códigos de aplicación (por ejemplo, una porción diferente de la aplicación distribuida) pueden ser datos tipeados, tales como, por ejemplo, datos lógicos, booleanos, binarios, octales, decimales, hexadecimales, enteros, de punto flotante, caracteres, cadenas de caracteres, tipos de datos definidos por el usuario, y combinaciones de estos tipos de datos usados para definir estructuras de datos.

El convertidor de tipos 103 está configurado para convertir (o serializar) datos tipeados correspondientes a un objeto tipeado, tal como, por ejemplo, los parámetros para una invocación de procedimiento, en un elemento del SOAP correspondiente y para convertir (o des-serializar) los elementos del SOAP en el tipo de datos correspondiente que corresponden al objeto tipeado. Los objetos tipeados, por ejemplo, los procedimientos incluidos en el código de aplicación 102 (por ejemplo de una clase), pueden anotarse con uno o más atributos de declaración del mensaje (por ejemplo, los atributos del Lenguaje Común del Tiempo de ejecución ("CLR") de un modelo de declaración del mensaje, tal como, por ejemplo, el modelo de declaración de mensajes 191. El convertidor de tipos 103 puede procesar atributos de declaración del mensaje para mapear los datos tipeados para el objeto tipeado a un elemento del SOAP correspondiente, tal como, por ejemplo, a instrucciones del Lenguaje de Marcado ("XML"). En algunas realizaciones, el convertidor de tipos 103 es un componente del tiempo de ejecución, configurado para convertir parámetros para un procedimiento de CLR en uno o más elementos del SOAP y para convertir un elemento del SOAP en parámetros para un procedimiento de CLR de acuerdo con un modelo de declaración del mensaje. Por consiguiente, el convertidor de tipos 103 puede usar el modelo de declaración del mensaje 191 para convertir un objeto de CLR tipeado en una representación de CLR de un elemento del SOAP y para convertir una representación de CLR de un elemento del SOAP en un objeto de CLR tipeado.

Los canales 106 proporcionan la abstracción de I/O para enviar y recibir envolventes del SOAP. Los canales 106

pueden proporcionar abstracciones para usar cualquiera de una diversidad de esquemas de protocolos de transporte (incluyendo HTTP, TCP, y el Protocolo de Datagramas de Usuario ("UDP")), esquemas de interfaz, y esquemas de direccionamiento. Generalmente, un canal también puede denominarse como una conexión.

5 Los codificadores 107 incluyen uno o más codificadores para codificar un elemento del SOAP en octetos puros y para decodificar los octetos puros en elementos del SOAP. De este modo, aunque los codificadores 107 se denominan como codificadores, los codificadores 107 también están configurados para decodificar. En la arquitectura de ordenadores 100, el codificador 107 incluye los codificadores 127, 128, y 129. Cada uno de los codificadores 127, 128, y 129 pueden estar configurados para codificar/decodificar de forma diferente. Por ejemplo, el codificador 127 puede estar configurado para soportar el SOAP sobre XML de texto, el codificador 128 puede estar configurado para soportar el SOAP sobre XML binario, y el codificador 129 puede estar configurado para soportar los Mecanismos de Optimización de la Transmisión de Mensajes (MTOM). Los codificadores también pueden realizar esquemas de compresión, incluyendo pero sin limitarse a esquemas de compresión genéricos como LZ o GZIP, que son codificadores por capas (codificadores construidos sobre otros codificadores más primitivos).

15 La codificación basada en el texto proporciona un mecanismo para aumentar la interoperabilidad entre aplicaciones distribuidas. Por ejemplo, la codificación basada en el texto puede soportar conjuntos de caracteres UTF8, UTF16 Big Endian, y UTF16 Little Endian. La codificación basada en el texto puede soportar diversos Tipos de Contenidos, tal como, por ejemplo, "texto/xml" y "aplicaciones/soap + xml" (por ejemplo, correspondientes bien a SOAP 1.1 o SOAP 1.2).

20 La codificación XML binaria proporciona un mecanismo para la codificación XML en una representación binaria que puede ser más compacta y eficiente de analizar. La codificación basada en binario puede soportar TiposContenidos, tales como, por ejemplo, "aplicación/soap + msbin1" y "aplicación/soap + msbinsesión1".

25 La codificación MTOM proporciona un mecanismo para codificar de forma eficaz elementos binarios XML dentro del contexto de una envolvente SOAP. MTOM representa una envolvente del SOAP como una parte (por ejemplo, una de muchas partes) de un contenedor MIME multi-parte. La localización de la envolvente del SOAP puede variarse para hacer que bien el transmisor o el receptor almacenen cualesquiera acoplamientos correspondientes. Por ejemplo un contenedor MIME que tiene una envolvente del SOAP como el primer elemento (por ejemplo, la primera de una o más partes) puede causar que el escritor almacene cualquier acoplamiento correspondiente posterior. Por otra parte, un contenedor MIME que tiene un SOAP como el último elemento (por ejemplo, el último de una o más partes) puede causar que el lector almacene cualquier acoplamiento correspondiente anterior.

30 La codificación MTOM incluye lógicamente los acoplamientos binarios dentro de la envolvente del SOAP, La codificación MTOM es polimórfica con otros codificadores de mensajes y no necesita cambios para un modelo de programación del usuario final. Además, la codificación MTOM facilita el acoplamiento de datos binarios en su formato natural (sin codificar) conservando por lo tanto el espacio con un mensaje del SOAP (en comparación con los formatos codificados) y liberando recursos que de otro modo se usarían para implementar algoritmos de codificación (potencialmente caros desde el punto de vista computacional).

35 El almacenamiento 163 puede ser un dispositivo de almacenamiento masivo remoto o local, tal como, por ejemplo, un controlador de disco duro magnético. El almacenamiento 163 almacena el diccionario 162. El diccionario 162 puede contener uno o más términos del SOAP y/o XML y los términos sustitutos de longitud reducida correspondientes. Por ejemplo, el diccionario 162 puede almacenar el valor "1" como una sustitución para "Envolvente" o "<http://www.w3.org/2003/05/soap-envelope>". Por consiguiente, puede usarse la sustitución para reducir el tamaño de la envolvente del SOAP antes de codificarlo para el transporte sobre la red 161.

40 El diccionario 162 puede incluir una porción estática de diccionario y una porción dinámica de diccionario. La porción estática del diccionario 162 puede incluir términos del SOAP y/o XML que tienen un mayor grado de concordancia entre las diversas y diferentes aplicaciones. Por ejemplo, hay un alto grado de probabilidad de que muchas aplicaciones diferentes (independientemente de la funcionalidad) usen los términos "Envolvente" y "<http://www.w3.org/2003/05/soap-envelope>". De este modo, estos términos pueden incluirse en la porción estática del diccionario 162.

45 La porción dinámica del diccionario 162 puede incluir términos del SOAP y/o XML que son comunes entre una pluralidad más pequeña de aplicaciones pero que no son necesariamente comunes entre todas las aplicaciones o una mayor pluralidad de aplicaciones. Por ejemplo, los términos comunes para aplicaciones bancarias pueden incluirse en una porción dinámica de los diccionarios en los sistemas de ordenadores que utilizan aplicaciones bancarias. Sin embargo, los términos bancarios pueden no incluirse en las porciones dinámicas de los diccionarios en los sistemas de ordenadores que no usan aplicaciones bancarias. Los términos pueden incluirse en la porción dinámica de un diccionario cuando los términos se identifican como comunes entre aplicaciones. De vez en cuando (por ejemplo, cuando expira un periodo sin uso), los términos pueden eliminarse de la porción dinámica del diccionario. Los términos del diccionario dinámico difieren de los términos del diccionario estático en que los términos del diccionario dinámico pueden transmitirse de forma incremental sobre una conexión junto con el primer mensaje enviado sobre la conexión que los referencia. Este muestreo puede limitarse en alcance a partes del mensaje que es probable que se repitan, tales como los nombres de XML y los nombres de espacios relacionados

con una estructura de datos tipeados particular o procedimiento, parámetro o valor devuelto.

El sistema de ordenador 111 incluye un código de aplicación 112, el convertidor de tipos 113, los canales 116, los codificadores 117, y el almacenamiento 143. El código de aplicación 112 puede ser una porción de una aplicación distribuida, tal como, por ejemplo un servicio correspondiente a un código de aplicación 102. El código de aplicación 112 puede ser un código que se desarrolló de acuerdo con un procedimiento o modelo de programación orientado a objetos. De este modo, los datos intercambiados entre los módulos del código de aplicación 112, entre los módulos del código de aplicación 112 y módulos del código de aplicación 102, y entre módulos del código de aplicación 112 y otros códigos de aplicación (por ejemplo, una porción diferente de la aplicación distribuida) pueden ser datos tipeados, tales como, por ejemplo, datos lógicos, binarios, octales, decimales, hexadecimales, enteros, de punto flotante, caracteres, cadenas de caracteres, tipos de datos definidos por el usuario, y combinaciones de estos tipos de datos para estructuras de datos definidas.

El convertidor de tipos 113 está configurado para convertir (o serializar) los datos tipeados correspondientes a un objeto tipeado, tales como, parámetros para una invocación de procedimiento, en un elemento del SOAP correspondiente y convertir (o des-serializar) los elementos del SOAP en los datos del tipo correspondiente que correspondan a un objeto tipeado. Los objetos tipeados, por ejemplo, los procedimientos incluidos en el código de aplicación 112 (por ejemplo de una clase sellada pública), pueden anotarse con uno o más atributos de declaración del mensaje (por ejemplo, los atributos del Lenguaje Común del Tiempo de ejecución ("CLR")) de un modelo de declaración de mensaje, tal como, por ejemplo, el modelo de declaración del mensaje 191. El convertidor de tipos 113 puede procesar los atributos de declaración del mensaje para mapear los datos tipeados para un objeto tipeado para un elemento del SOAP correspondiente, tal como, por ejemplo, en elementos de XML. En algunas realizaciones, el convertidor de tipos 113 es una componente del tiempo de ejecución, configurada para convertir los parámetros para un procedimiento de CLR en un elemento del SOAP y para convertir un elemento del SOAP en parámetros para un procedimiento de CLR de acuerdo con un modelo de declaración del mensaje. Por consiguiente, el convertidor de tipos 113 puede usar el modelo de declaración del mensaje 191 para convertir un objeto de CLR tipeado en una representación de CLR de un elemento del SOAP y para convertir una representación de CLR de un elemento del SOAP en un objeto de CLR tipeado.

El despachador 114 recibe los elementos del SOAP (por ejemplo, desde un canal apropiado) y despacha los elementos del SOAP a los objetos tipeados apropiados (por ejemplo, el procedimiento). Cuando el despachador 114 recibe un elemento del SOAP, el despachador 114 intenta compaginar el elemento del SOAP con un objeto tipeado apropiado. En algunas realizaciones, el despachador 114 intenta compaginar las porciones de los elementos de XML (en un elemento del SOAP) con un objeto tipeado apropiado. El despachador 114 puede comparar porciones de los elementos de XML con los atributos de declaración del mensaje que anotan objetos en objetos 141. Cuando se identifica una coincidencia, el despachador 114 despacha el elemento del SOAP para el objeto coincidente. Los mensajes pueden despacharse sobre el valor de una Acción propiedad de un atributo de declaración del mensaje que anota el objeto coincidente. Cuando no se identifica ninguna coincidencia, el despachador 114 puede despachar el elemento del SOAP para un objeto por defecto.

Los canales 116 proporcionan la abstracción de I/O para la transmisión y recepción de envoltorios del SOAP. Los canales 116 pueden proporcionar abstracciones para usar cualquiera de una diversidad de esquemas de protocolo de transporte (incluyendo HTTP, TCP, y el Protocolo de Datagramas de Usuario ("UDP")), esquemas de interfaces, y esquemas de direccionamiento. Los canales 116 pueden proporcionar una abstracción para los esquemas del protocolo de transporte, esquemas de la interfaz, y esquemas de direccionamiento que también se proporcionan en los canales 106.

Los codificadores 117 incluyen uno o más codificadores para codificar un elemento del SOAP en octetos puros y para decodificar los octetos puros en un elemento del SOAP. De este modo, aunque los codificadores 117 se denominen como decodificadores, los codificadores 117 están también configurados para decodificar. En la arquitectura de ordenadores 100, el codificador 117 incluye los codificadores 137, 138 y 139. Cada uno de los codificadores 137, 138 y 139 pueden estar configurados para codificar/decodificar de forma diferente. Por ejemplo, el codificador 137 puede configurarse para soportar SOAP sobre XML de texto, el codificador 138 puede configurarse para soportar SOAP sobre XML binario, y el codificador 139 puede configurarse para soportar los Mecanismos de Optimización de la Transmisión de Mensajes (MTOM).

El almacenamiento 143 puede ser un dispositivo de almacenamiento masivo local o remoto, tal como, por ejemplo, un controlador de disco duro magnético. El almacenamiento 143 almacena el diccionario 162 y los objetos 141. El diccionario 162 puede contener uno o más términos del SOAP y/o XML y las substituciones correspondientes que son de longitud reducida. Por ejemplo, el diccionario 162 puede almacenar el valor "2" como una substitución para "wsa". Por consiguiente, una substitución puede usarse para reducir el tamaño de las envoltorios del SOAP antes de la codificación para el transporte sobre la red 161.

El diccionario 142 puede incluir una porción estática del diccionario y una porción dinámica del diccionario. La porción estática puede incluir términos del SOAP y/o de XML que tienen un mayor grado de concordancia entre las diversas y diferentes aplicaciones. La porción dinámica del diccionario 162 puede incluir términos del SOAP y/o XML que son comunes entre una pluralidad más pequeña de aplicaciones pero que no son necesariamente comunes

entre todas las aplicaciones o una mayor pluralidad de aplicaciones.

La Figura 2 ilustra un diagrama de flujo de ejemplo de un procedimiento 200 de envío de modo flexible de datos de aplicación tipeados. El procedimiento 200 se describirá con respecto a los módulos y los datos en la arquitectura de ordenador 100.

- 5 El código de aplicación 102 puede generar datos tipeados 131 que están, por ejemplo, incluidos en un mensaje tipeado. El mensaje tipeado puede configurarse de acuerdo con el modelo de declaración de mensajes 191 para representar una envolvente SOAP como un objeto tipeado (CLR). Los datos tipeados 131 pueden incluir parámetros y/o un valor de retorno para un procedimiento. Por ejemplo, el código de aplicación 102 puede llamar a un procedimiento con parámetros que causan que se invoque un procedimiento.
- 10 El procedimiento 200 incluye un acto de acceso a los parámetros del objeto tipeado que corresponden a un objeto tipeado (acto 210). Por ejemplo, el convertidor de tipos 103 puede acceder a los datos tipeados 131. Los datos tipeados 131 pueden incluir parámetros que corresponden al objeto tipeado (por ejemplo, parámetros para invocar a un procedimiento. El objeto tipeado puede anotarse con uno o más atributos de declaración del mensaje del modelo de declaración de mensajes 191, tales como, por ejemplo, los atributos 171. Por consiguiente, los atributos 171 definen un mapeo entre los datos tipeados 131 y un elemento del SOAP correspondiente (por ejemplo, el elemento del SOAP 172).

- 20 El procedimiento 200 incluye el acto de mapear los parámetros del objeto tipeado accedido en un elemento del SOAP de acuerdo con el uno o más atributos de declaración del mensaje (acto 202). Por ejemplo, el convertidor de tipos 103 puede mapear datos tipeados 131 (por ejemplo parámetros para la invocación de un procedimiento) en un elemento del SOAP 172 (por ejemplo, elementos de XML) de acuerdo con los atributos 171.

- 25 El procedimiento 200 incluye el acto de referirse a uno o más atributos de declaración del mensaje para identificar una localización dentro de una envolvente del SOAP (acto 203). Por ejemplo, el convertidor de tipos 103 puede referirse a atributos 171 para identificar una localización dentro de la envolvente 132A (creada en el sistema de ordenador 101). Los atributos 171 pueden identificar, por ejemplo, una cabecera de mensaje o un cuerpo de mensaje dentro de la envolvente 132A. Atributos adicionales 171 pueden identificar posiciones específicas dentro de una cabecera de mensaje o cuerpo de mensaje. De este modo, los elementos del SOAP correspondientes pueden ordenarse dentro de una cabecera del mensaje o un cuerpo de mensaje. De este modo, los elementos del SOAP correspondientes pueden ordenarse dentro de una cabecera de mensaje o dentro de un cuerpo de mensaje.

- 30 El procedimiento 200 incluye el acto de insertar el elemento del SOAP en la localización identificada dentro de la envolvente del SOAP (acto 204). Por ejemplo, el convertidor de tipos 103 puede insertar el elemento del SOAP 172 en (por ejemplo, una posición especificada) una porción de la cabecera o una porción del cuerpo de la envolvente 132A. La envolvente 132A puede transferirse a través de un canal de entre los canales 106, por ejemplo, de acuerdo con un transporte seleccionado, interfaz y esquema de direccionamiento. Por ejemplo, la envolvente 132A puede encapsularse en el transporte 133.

- 35 Un codificador de los codificadores 107, tal como, por ejemplo, el codificador 128 puede codificar la envolvente encapsulada 132A en octetos puros 134. Cuando sea apropiado, el codificador 128 se refiere al diccionario 162 para substituir términos usados comúnmente. El sistema de ordenador 101 envía octetos puros 134, a través de la red 161, a un sistema de ordenador 111. El sistema de ordenador 111 recibe los octetos puros 134 a través de la red 161, desde el sistema de ordenador 101.

- 40 La Figura 3 ilustra un diagrama de flujo de ejemplo de un procedimiento 300 para la recepción de modo flexible de los datos de aplicación tipeados. El procedimiento 300 se describirá con respecto a los módulos y los datos en la arquitectura de ordenador 100.

- 45 Un codificador de los codificadores 117, tal como, por ejemplo, el codificador 138 puede recibir octetos puros 134. El codificador 138 puede decodificar los octetos puros 134 para revelar la envolvente 132B. Cuando sea apropiado, el codificador 138 se refiere al diccionario 162 para reemplazar los términos que se substituyeron antes de la codificación. La decodificación de los octetos puros 134, puede revelar que la envolvente 132B se transfirió, por ejemplo, de acuerdo con un transporte seleccionado, interfaz y esquema de direccionamiento. Por ejemplo, la decodificación de los octetos puros 134 puede revelar que la envolvente 132B está encapsulada en el transporte 133.

- 50 En base al transporte, las interfaces, y/o el esquema de direccionamiento, la envolvente encapsulada 132B puede transferirse a través de un canal apropiado. Como es posible para los intermediarios en la red 161 alterar las porciones de los datos en la envolvente 132A (por ejemplo, las cabeceras de encaminamiento) durante la transferencia, la envolvente 132A y la envolvente 132B pueden incluir datos diferentes. El despachador 114 recibe la envolvente 132B incluyendo el elemento del SOAP 172.

- 55 El procedimiento 300 incluye el acto de acceder al elemento del SOAP incluido en una envolvente del SOAP (acto 301). Por ejemplo el despachador 114 puede acceder al elemento del SOAP 172.

El procedimiento 300 incluye el acto de identificar un objeto tipeado que corresponde al elemento del SOAP (acto 302). Por ejemplo, el despachador 114 puede identificar el objeto 151 a partir de los objetos 141 (por ejemplo, una colección de objetos incluidos en diversos componentes de aplicación distribuidos en el sistema de ordenador 111). El objeto 151 puede ser un objeto tipeado (por ejemplo, un procedimiento del código de aplicación 112) que está anotado con los atributos 181 del modelo de declaración del mensaje 191. Los atributos 181 pueden definir un mapeo entre el elemento del SOAP 172 y los parámetros del objeto tipeado correspondiente (por ejemplo, los parámetros para la invocación de un procedimiento). El despachador 114 puede compaginar porciones del elemento del SOAP 172 (por ejemplo, porciones de elementos de XML) con atributos 181 para identificar que el elemento del SOAP 172 corresponde al objeto 151. En algunas aplicaciones, esto incluye la utilización de un URI (por ejemplo, un URI de Acción de Direccionamiento WS) para mapear el elemento del SOAP 172 al objeto 151.

El procedimiento 300 incluye el acto de mapear el elemento del SOAP accedido en parámetros del objeto tipeado de acuerdo con uno o más atributos de declaración del mensaje (acto 303). Por ejemplo, el convertidor de tipos 113 puede mapear el elemento del SOAP 172 a un mensaje tipeado que incluye datos tipeados 131 (de acuerdo con los atributos 181). El mensaje tipeado puede configurarse de acuerdo con el modelo de declaración de mensajes 191 para la representación de una envolvente del SOAP como un objeto tipeado (CLR). Los datos tipeados 131 pueden incluir parámetros y/o un valor de retorno para un procedimiento. Por ejemplo, el código de aplicación 112 puede recibir una llamada a un procedimiento (iniciado en la aplicación 102) con parámetros que causan la invocación del procedimiento.

De este modo, las realizaciones de la presente invención posibilitan al desarrollador controlar el modo en el que los parámetros del procedimiento y los valores de retorno se serializan para las cabeceras del mensaje y los cuerpos del mensaje. Del mismo modo, un desarrollador puede controlar el modo en el cual los parámetros del procedimiento y los valores de retorno se des-serIALIZAN a partir de las cabeceras del mensaje y los cuerpos del mensaje. Por consiguiente, las aplicaciones desarrolladas pueden retener tanto los beneficios de los datos tipeados (por ejemplo, los objetos tipeados de CLR) como la flexibilidad de la mensajería del SOAP.

Las siguientes instrucciones de código de ejemplo representan una clase anotada en la que las propiedades y los campos de la clase anotada se anotan con atributos de declaración del mensaje:

```

Línea 1:      [DeclaraciónMensaje (Acción = "http://createpersonrequest", Nombre = " CreatePersonRequest")]
Línea 2:      clase pública CreatePersonRequest {
Línea 3:      [CuerpoMensaje (Namespace = "http:// namespace ", Nombre = "MisDatos", Posición = 2)]
Línea 4:      cadena de datos pública DatosOpcionales;
Línea 5:      [CuerpoMensaje (Namespaceuri = "http:// namespace", Nombre = LaLista")]
Línea 6:      publica llist<int> lds;
Línea 7:      [CabeceraMensaje (Namespaceuri = "http://namespace", Nombre = "Petición")]
Línea 8:      Internal Guid PeticiónId;
Línea 9:      [CabeceraMensaje (Namespaceuri = "http://namespace", Nombre = "Session", DebeEntender=
cierto, Relay = falso, Actor = http://actor)]
Línea 10:     Internal Guid SessionId;
    
```

El atributo [DeclaraciónMensaje] en la Línea 1 anota la clase pública CreatePersonRequest en la Línea 2 y define una declaración del mensaje "CreatePersonRequest" (por ejemplo, el modelo de declaración del mensaje 191), para controlar el procesamiento y creación de las envolventes del SOAP. Las líneas 3 y 5 anotan las líneas 4 y 6 respectivamente con [CuerpoMensaje] indicando la inclusión en un cuerpo de la envolvente SOAP. Las líneas 7 y 8 anotan las líneas 9 y 10 respectivamente con [CabeceraMensaje] indicando la inclusión en una cabecera de la envolvente del SOAP. El atributo "Posición = 2" sobre la línea 3 indica que los datos tipeados anotados se incluirán en el segundo elemento en un cuerpo del mensaje.

Un convertidor de tipos, tal como, por ejemplo, el convertidor de tipos 103 o el convertidor de tipos 113 puede usar un modelo de declaración del mensaje, tal como, por ejemplo, el modelo de declaración del mensaje 191, para convertir las instrucciones del código de ejemplo en la siguiente envolvente del SOAP de ejemplo:

```

<soap: Envolvente>
xmlns: soap = "http://schemas.xmlsoap.org/soap/envelope/"
xmlns: wsa = "http://schemas.xmlsoap.org/ws/2004/03/addressing"
<soap: Cabecera xmlns : x = "http://namespace">
5     <x: Request>xc6gf83</Request>
     <x: Session DebeEntender = "cierto", Relay = "falso", Actor = "http://actor">
         gg57hjfg
     </x: Session>
     <wsa: Action> http://createpersonrequest </wsa: Action>
10 </soap: Cabecera>
<soap: Cuerpo xmlns: x= 'http:// namespace'>
     <x: LaLista>
         <x: Elemento> 5 </x: Elemento >
         <x: Elemento > 10 </x: Elemento >
15     <x: LaLista>
         <x: MisDatos> aquí algunos datos </x: MisDatos>
</soap: Cuerpo>
</soap: Envolvente>

```

20 Un convertidor de tipos, tal como, por ejemplo, el convertidor de tipos 103 o el convertidor de tipos 113, también pueden usar un modelo de declaración del mensaje, tal como, el modelo de declaración de mensajes 191, para convertir la envoltura del SOAP de ejemplo en las instrucciones de código de ejemplo. En el ejemplo de envoltura del SOAP, el elemento "MisDatos" está en la segunda posición dentro del cuerpo de la envoltura del SOAP.

25 En algunas realizaciones de la presente invención, el transporte de reproducción directa se utiliza para compensar los mensajes más largos que no pueden representarse completamente en la memoria en un instante en el tiempo. Esto es, el transporte de reproducción directa es ventajoso para mensajes que son más largos que la memoria disponible (por ejemplo, debido a limitaciones físicas o limitaciones de la memoria asignada) para el servicio de procesamiento del mensaje. El transporte de reproducción directa puede incluir una conexión de transporte dedicada (base TCP, canalizaciones con nombre (named pipe), conexión HTTP, etc.) para difundir el mensaje. Los sistemas de ordenadores que son para el uso del transporte de reproducción directa pueden configurarse de acuerdo con una propiedad del Modo de Transferencia de una fábrica de transporte adecuado (por ejemplo FabricaCanalSuperior / FabricaOyenteSuperior, FábricaCanalCanalizaciónConNombre / FábricaOyenteCanalCanalizaciónConNombre y HttpFábricaCanal/ HttpFábricaOyente.

35 El transporte de reproducción directa puede posibilitarse a través de los elementos obligatorios de transporte correspondientes (por ejemplo, ElementoObligatorioTransporte, ElementoObligatorioTransporteCanalConNombre y HttpElementoObligatorioTransporte) usados para anotar un objeto tipeado. Las propiedades de transporte pueden variarse en base al tipo de canal. Por ejemplo, para el TCP o CanalizaciónConNombre, puede usarse ICanalSalida / ICanalEntrada o ICanalSesiónDúplex (que usa un par de ICanalSalida / ICanalEntrada). Para HTTP, pueden usarse ICanalPetición / ICanalRespuesta, ICanalSesiónPetición / ICanalSesiónRespuesta, ICanalSalida / ICanalEntrada o ICanalFullDuplex. Cuando está configurada la reproducción directa para una declaración de datos determinada, un canal apropiado puede facilitar la petición y creación de un tipo de canal adecuado (por ejemplo, en los canales 106 y/o 116).

45 El transporte de reproducción directa es transparente para el lector del mensaje. Esto es, la aplicación no necesita escribirse de forma diferente para tratar con el mensaje. La aplicación simplemente puede leer los valores de forma secuencial usando un elemento de XML común API (por ejemplo, XmlReader). Un elemento obligatorio de transporte puede configurarse por la instanciación del elemento obligatorio de transporte y la asignación de un valor a la propiedad de ese elemento. Los elementos obligatorios de transporte pueden también configurarse a través de la configuración de CLR (por ejemplo, un archivo myapp.config).

50 La Figura 4 ilustra un entorno adecuado de operación para los principios de la presente invención. La Figura 4 y la siguiente discusión pretenden proporcionar una breve descripción general de un entorno de computación adecuado en el cual puede implementarse la invención. Aunque no se requiere, la invención se describirá en el contexto general de instrucciones ejecutables por ordenador, tal como los módulos de programas, que se ejecutan por los sistemas de ordenadores. De forma general, los módulos de programas, incluyen rutinas, programas, objetos, componentes, estructuras de datos, y similares, que realizan las tareas particulares o implementan los tipos de datos abstractos particulares. Las instrucciones ejecutables por ordenador, estructuras de datos asociadas, y módulos de programas representan ejemplos de un medio de código de programa para la ejecución de actos de los procedimientos desvelados en el presente documento.

60 Con referencia a la Figura 4, un sistema de ejemplo para la implementación de la invención incluye un dispositivo de computación de propósito general en la forma de un sistema de ordenador 420, incluyendo una unidad de procesamiento 421, una memoria del sistema 422 y un bus del sistema 423 que acopla los diversos componentes del sistema incluyendo la memoria del sistema 422 a la unidad de procesamiento 421. La unidad de procesamiento 421 puede ejecutar instrucciones ejecutables por ordenador diseñadas para implementar características del sistema

de ordenador 420, incluyendo las características de la presente invención. El bus del sistema 423 puede ser cualquiera de varios tipos de estructuras de buses incluyendo un bus de memoria o controlador de memoria, un bus periférico, y un bus local usando cualquiera de una diversidad de arquitecturas de bus. La memoria del sistema incluye memoria de sólo lectura ("ROM") 424 y memoria de acceso aleatorio ("RAM") 425. Un sistema de entrada/salida básico ("BIOS") 426, que contiene las rutinas básicas que ayudan en la transferencia de información entre elementos dentro de un sistema de ordenador 420, tal como durante el establecimiento, puede almacenarse en la ROM 424.

El sistema de ordenador 420, puede incluir también un controlador de disco duro magnético 427 para la lectura y escritura sobre el disco duro magnético 439, el controlador de disco magnético 428 para la lectura o escritura sobre el disco magnético extraíble 429, un controlador de disco óptico 430 para la lectura o escritura sobre el disco óptico extraíble 431, tal como, por ejemplo, un CD-ROM u otro medio óptico. El controlador de disco duro magnético 427, el controlador de disco magnético 428, y el controlador de disco óptico 430 se conectan al bus del sistema 423 por la interfaz de control del disco duro 432, la interfaz del control del disco magnético 433, y la interfaz de control del disco óptico 434, respectivamente. Los controladores y sus medios asociados legibles por el ordenador proporcionan un almacenamiento no volátil de las instrucciones ejecutables por el ordenador, estructuras de datos, módulos de programas, y otros datos para el sistema de ordenador 420. Aunque el entorno de ejemplo descrito en el presente documento emplea un disco duro magnético 439, un disco magnético extraíble 429 y un disco óptico extraíble 431, pueden usarse otros tipos de medios legibles por el ordenador para el almacenamiento de datos, incluyendo casetes magnéticas, tarjetas de memoria flash, discos versátiles digitales, cartuchos Bernoulli, RAM, ROM, y similares.

El medio del código de programación que comprende uno o más módulos de programas puede almacenarse sobre el disco duro 439, el disco magnético 429, el disco óptico 431, ROM 424, o RAM 425, incluyendo un sistema operativo 435, uno o más programas de aplicación 436, otros módulos de programa 437, y datos de programa 438. Un usuario puede introducir comandos e información dentro del sistema de ordenador 420 a través del teclado 440, un dispositivo de apuntamiento 442, u otros dispositivos de entrada (no mostrados), tales como, por ejemplo, un micrófono, una palanca de mando, un mando de juegos, escáner, o similares. Estos y otros dispositivos de entrada pueden conectarse a la unidad de procesamiento 421 a través de la interfaz de entrada/salida 446 acoplada al bus del sistema 423. La interfaz de entrada/salida 446 desde el punto de vista lógico representa cualquiera de una amplia variedad de interfaces diferentes, tales como, por ejemplo, una interfaz de puerto serie, una interfaz de PS/2, una interfaz de puerto paralelo, una interfaz de Bus Serie Universal ("USB"), o una interfaz del Instituto de Ingenieros Eléctricos y Electrónicos ("IEEE") 1394 (es decir una interfaz Cortafuegos), o puede incluso representar de forma lógica una combinación de las diferentes interfaces.

Un monitor 447 u otro dispositivo de pantalla también está conectado al bus del sistema 423 a través de la interfaz de video 448. Otros dispositivos de salida periféricos (no mostrados) tales como, por ejemplo, altavoces e impresoras, pueden también conectarse al sistema de ordenadores 420.

El sistema de ordenadores 420 es conectable a redes, tales como, por ejemplo, una red de ordenadores del ámbito de la oficina, o del ámbito de la compañía, una red doméstica, una intranet, y/o la Internet. El sistema de ordenadores 420 puede intercambiar datos con fuentes externas, tales como, por ejemplo, sistemas de ordenadores remotos, aplicaciones remotas, y/o bases de datos remotas sobre tales redes.

El sistema de ordenador 420 incluye una interfaz de red 453, a través de la cual el sistema de ordenador 420 recibe datos desde fuentes externas y/o transmite datos a fuentes externas. Como se representa en la Figura 4, la interfaz de red 453 facilita el intercambio de datos con el sistema de ordenador remoto 483 a través del enlace 451. La interfaz de red 453 puede representar de forma lógica uno o más módulos software y/o software, tales como, por ejemplo, una tarjeta de interfaz de red y la correspondiente pila de Especificaciones de Interfaz de Control de Red ("NDIS"). El enlace 451 representa una porción de la red (por ejemplo, un segmento de Ethernet), y el sistema de ordenadores remoto 483 representa un nodo de la red.

Del mismo modo, el sistema de ordenadores 420 incluye una interfaz de entrada/salida 446, a través de la cual el sistema de ordenador 420 recibe datos desde fuentes externas y/o transmite datos a fuentes externas. La interfaz de entrada/salida 446 está acoplada al módem 454 (por ejemplo, un módem normalizado, un módem de cable, o un módem de la línea de abonado digital ("DSL")) a través del enlace 459, a través del cual el sistema de ordenador 420 recibe datos desde las fuentes externas y/o transmite datos a las mismas. Como se representa en la Figura 4, la interfaz de entrada/salida 446 y el módem 454 facilitan el intercambio de datos con el sistema de ordenador remoto 493 a través del enlace 452. El enlace 452 representa una porción de una red y el sistema de ordenador remoto 493 representa un nodo de la red.

Aunque la Figura 4 representa un entorno de operación adecuado para la presente invención, los principios de la presente invención pueden emplearse en cualquier sistema que sea capaz, con las modificaciones adecuadas si es necesario, de implementar los principios de la presente invención. El entorno ilustrado en la Figura 4 es sólo ilustrativo y de ninguna manera representa incluso una pequeña porción de la amplia variedad de entornos en los cuales pueden implementarse los principios de la presente invención.

De acuerdo con la presente invención, los módulos que incluyen el código de aplicación, los proxy de cliente, los

servicios, los convertidores de tipos, los canales, los codificadores/decodificadores de mensajes, y los despachadores así como los datos asociados, incluyendo los datos tipeados, las envolventes del SOAP, los transportes, los octetos puros, los diccionarios y objetos, pueden almacenarse y accederse desde cualquiera de los medios legibles por ordenador asociados con el sistema de ordenador 420. Por ejemplo, las porciones de tales módulos y porciones de los datos de programa asociados pueden incluirse en el sistema operativo 435, los programas de aplicación 436, los módulos de programa 437, y/o los datos de programa 438, para el almacenamiento en la memoria del sistema 422.

10 Cuando un dispositivo de almacenamiento masivo, tal como, por ejemplo, el disco duro magnético 439, está acoplado al sistema de ordenador 420, tales módulos y los datos de programa asociados también pueden almacenarse en el dispositivo de almacenamiento masivo. En un entorno de red, los módulos de programa representados con relación al sistema de ordenador 420, o porciones del mismo, pueden almacenarse en los dispositivos de almacenamiento de memoria, tales como, la memoria del sistema y/o los dispositivos de almacenamiento masivo asociados con el sistema de ordenador remoto 483 y/o el sistema de ordenador remoto 493. La ejecución de tales módulos puede realizarse en un entorno distribuido como se ha descrito anteriormente.

15 La presente invención puede realizarse de otras formas específicas sin apartarse de sus características esenciales. Las realizaciones descritas se considerarán en todos los aspectos sólo como ilustrativas y no como restrictivas. El alcance de la invención está, por lo tanto indicado por las reivindicaciones adjuntas más que por la descripción anterior. Todos los cambios que caen dentro del significado y el rango de equivalencia de las reivindicaciones están incluidas dentro de su alcance.

**REIVINDICACIONES**

1. Un procedimiento de envío de forma flexible de datos de aplicación tipeados (131) para un sistema de ordenador (101) que es una red (161) conectable junto con uno o más sistemas de ordenador a una red, comprendiendo el procedimiento:
  - 5 un acto (201) de acceder a los parámetros del objeto tipeado correspondiente a un objeto tipeado, estando anotado el objeto tipeado con uno o más atributos de declaración del mensaje (171) de un modelo de declaración del mensaje (191), definiendo el uno o más atributos de declaración del mensaje un mapeo entre los parámetros del objeto tipeado y los elementos del SOAP correspondientes;
  - 10 un acto (202) del mapeo de los parámetros del objeto tipeado accedido en un elemento del SOAP de acuerdo con el uno o más atributos de declaración del mensaje; y
  - un acto (203) de referirse a uno o más atributos de declaración del mensaje para identificar una localización dentro de la envolvente del SOAP;
  - un acto (204) que inserta el elemento del SOAP en la localización identificada dentro de la envolvente del SOAP (132).
- 15 2. El procedimiento según la reivindicación 1, en el que el acto (201) de acceder a los parámetros del objeto tipeado correspondiente a un objeto tipeado comprende un acto de acceder a la clase que representa parámetros para el procedimiento.
3. El procedimiento según la reivindicación 2, en el que el acto (201) de acceder a una clase que representa parámetros para un procedimiento comprende un acto de acceder a una clase pública que representa parámetros para un procedimiento del Lenguaje Común del Tiempo de funcionamiento.
- 20 4. El procedimiento según la reivindicación 1, en el que el acto (201) de acceder a los parámetros del objeto tipeado correspondiente a un objeto tipeado comprende un acto de acceder a un objeto tipeado que está anotado con una Acción URI.
5. El procedimiento según la reivindicación 1, en el que el acto (201) de acceder a los parámetros del objeto tipeado correspondiente a un objeto tipeado comprende un acto de acceder a una clase que representa un valor de retorno del procedimiento.
- 25 6. El procedimiento según la reivindicación 1, en el que el acto (201) de acceder a los parámetros del objeto tipeado correspondiente a un objeto tipeado comprende un acto de acceder a una representación del Lenguaje Común del Tiempo de funcionamiento de un elemento del SOAP.
- 30 7. El procedimiento según la reivindicación 1, en el que el acto (201) de acceder a los parámetros del objeto tipeado correspondiente a un objeto tipeado comprende un acto de acceder a un mensaje tipeado.
8. El procedimiento según la reivindicación 1, en el que el acto (201) de acceder a los parámetros del objeto tipeado correspondiente a un objeto tipeado comprende un acto de acceder a un parámetro del objeto tipeado anotado que está anotado con un atributo de declaración del mensaje indicando que la representación XML del parámetro del objeto tipeado se incluirá en una cabecera de la envolvente (132) del SOAP.
- 35 9. El procedimiento según la reivindicación 1, en el que el acto (201) de acceder a los parámetros del objeto tipeado correspondiente a un objeto tipeado comprende un acto de acceder a un parámetro del objeto tipeado que está anotado con un atributo de declaración del mensaje (171) indicando que una representación XML del parámetro del objeto tipeado se incluirá en el cuerpo de la envolvente del SOAP.
- 40 10. El procedimiento según la reivindicación 1, en el que el acto (201) de acceder a los parámetros del objeto tipeado correspondiente a un objeto tipeado comprende un acto de acceder a los parámetros del objeto tipeado anotado, estando anotados los parámetros del objeto tipeado anotado con uno o más atributos de declaración del mensaje (171), estando seleccionados el uno o más atributos de declaración del mensaje de entre los atributos que indican una posición dentro de una envolvente del SOAP (132), un nombre, un espacio de nombre URI, si debe entenderse un parámetro del objeto tipeado, si puede retransmitirse el parámetro del objeto tipeado, y un actor para el parámetro del objeto tipeado.
- 45 11. El procedimiento según la reivindicación 1, en el que el acto (202) de mapear los parámetros del objeto tipeado accedido en un elemento del SOAP de acuerdo con el uno o más de los atributos de declaración del mensaje (171) comprende un acto de serializar los parámetros del objeto tipeado accedido a un elemento del SOAP.
- 50 12. El procedimiento según la reivindicación 1, en el que el acto (202) de mapear los parámetros del objeto tipeado accedido en un elemento del SOAP de acuerdo con uno o más atributos de declaración del mensaje (171) comprende un acto de mapear los parámetros del objeto tipeado accedido en los elementos de XML correspondientes.

13. El procedimiento según la reivindicación 1, en el que el acto (203) de referirse a uno o más atributos de declaración del mensaje (171) para identificar una localización dentro de una envolvente del SOAP (132) comprende un acto de referirse a un atributo de declaración del mensaje que identifica una porción de cabecera de una envolvente del SOAP.
- 5 14. El procedimiento según la reivindicación 1, en el que el acto (203) de referirse a uno o más atributos de declaración del mensaje (171) para identificar una localización dentro de una envolvente del SOAP (132) comprende un acto de referirse a un atributo de declaración del mensaje que identifica una porción del cuerpo de una envolvente del SOAP.
- 10 15. El procedimiento según la reivindicación 1, en el que el acto (204) de insertar el elemento del SOAP en la localización identificada dentro de la envolvente del SOAP (132) comprende un acto de insertar las instrucciones XML correspondientes dentro de una porción de cabecera de la envolvente del SOAP.
16. El procedimiento según la reivindicación 1, en el que el acto (204) de insertar el elemento del SOAP en la localización identificada dentro de la envolvente del SOAP (132) comprende un acto de insertar elementos de XML dentro de una porción del cuerpo de la envolvente del SOAP.
- 15 17. El procedimiento según la reivindicación 1, que comprende además:  
 un acto de codificación de la envolvente del SOAP (132) en octetos puros (134); y  
 un acto de enviar los octetos puros a un sistema de ordenador receptor (111).
- 20 18. El procedimiento según la reivindicación 17, en el que el acto de codificar la envolvente del SOAP (132) en octetos puros (134) comprende un acto de codificar la envolvente del SOAP en octetos puros en el que el tamaño de los octetos puros no tiene un límite predefinido.
19. El procedimiento según la reivindicación 17, en el que el acto de codificar la envolvente del SOAP (132) en octetos puros (134) comprende un acto de generar una codificación basada en el texto de la envolvente del SOAP.
- 25 20. El procedimiento según la reivindicación 17, en el que el acto de codificar la envolvente del SOAP (132) en octetos puros (134) comprende un acto de usar un Mecanismo de Optimización de la Transmisión de Mensajes para codificar la envolvente del SOAP.
21. El procedimiento según la reivindicación 17, en el que el acto de codificar la envolvente del SOAP (132) en octetos puros (134) comprende el acto de generar una codificación basada en binario de la envolvente del SOAP.
- 30 22. El procedimiento según la reivindicación 21, en el que el acto de generar una codificación basada en binario de la envolvente del SOAP (132) comprende el acto de sustituir uno o más términos del SOAP y/o términos de XML con términos sustitutos de longitud reducida de modo que la envolvente del SOAP puede transferirse de forma más eficaz.
23. El procedimiento según la reivindicación 17, en el que el acto de enviar los octetos puros (134) a un sistema de ordenador de recepción comprende el acto de reproducción directa de los octetos puros para el sistema de ordenador receptor (111).
- 35 24. Un procedimiento de recibir de modo flexible datos de aplicación tipeados (131), para un sistema de ordenador (111) que es la red (161) conectable junto con uno o más sistemas de ordenadores a una red, comprendiendo el procedimiento:  
 un acto (301) de acceso a un elemento del SOAP incluido en una envolvente del SOAP (132);  
 un acto (302) de identificar un objeto tipeado que corresponde al elemento del SOAP, estando anotado el elemento tipeado con uno o más atributos de declaración del mensaje (171) de un modelo de declaración del mensaje, definiendo el uno o más atributos de declaración del mensaje un mapeo entre los elementos del SOAP y los parámetros del objeto tipeado correspondiente; y  
 un acto (303) de mapeo del elemento del SOAP accedido en parámetros del objeto tipeado de acuerdo con el uno o más atributos de declaración del mensaje, presentándose los parámetros del objeto tipeado al objeto tipeado identificado.
- 40 45 25. El procedimiento según la reivindicación 24, que comprende además:  
 un acto de recibir octetos puros (134) desde un sistema de ordenador transmisor (101); y  
 un acto de decodificar los octetos puros en el interior de la envolvente del SOAP (132).
- 50 26. El procedimiento según la reivindicación 25, en el que el acto de recibir los octetos puros (134) desde un sistema de ordenador transmisor (101) comprende el acto de recibir octetos puros de reproducción directa.
27. El procedimiento según la reivindicación 25, en el que el acto de decodificar los octetos puros (134) en el interior de la envolvente del SOAP (132) comprende un acto de decodificar la codificación basada en texto de la envolvente

del SOAP.

28. El procedimiento según la reivindicación 25, en el que el acto de decodificar los octetos puros (134) en el interior de la envolvente del SOAP (132) comprende un acto de decodificar la codificación basada en binario de la envolvente del SOAP.

5 29. El procedimiento según la reivindicación 28, en el que el acto de decodificar la codificación basada en binario de la envolvente del SOAP (132) comprende el acto de reemplazar términos sustitutos de longitud reducida con los términos originales del SOAP y/o términos de XML de modo que la envolvente del SOAP se devuelve a la configuración anterior.

10 30. El procedimiento según la reivindicación 25, en el que el acto de decodificar los octetos puros (134) en el interior de la envolvente del SOAP (132) comprende un acto de usar un Mecanismo de Optimización de la Transmisión de Mensajes para decodificar los octetos puros.

31. El procedimiento según la reivindicación 24, en el que el acto (301) de acceder a un elemento del SOAP incluido en una envolvente del SOAP comprende un acto de acceder a elementos de XML desde una porción de la cabecera de la envolvente del SOAP.

15 32. El procedimiento según la reivindicación 24, en el que el acto (301) de acceder a un elemento del SOAP incluido en una envolvente del SOAP comprende un acto de acceder a elementos XML desde una porción del cuerpo de la envolvente del SOAP.

20 33. El procedimiento según la reivindicación 24, en el que el acto (302) de identificar un objeto tipeado que corresponde al elemento del SOAP comprende un acto de comparar los elementos de XML en la envolvente del SOAP a atributos de la declaración del mensaje (171) de una clase anotada.

34. El procedimiento según la reivindicación 24, en el que el acto (302) de identificar un objeto tipeado que corresponde al elemento del SOAP comprende un acto de acceder un URI correspondiente al objeto tipeado.

25 35. El procedimiento según la reivindicación 24, en el que el acto (302) de identificar un objeto tipeado que corresponde al elemento del SOAP comprende un acto de identificar una clase que representa los parámetros para un procedimiento.

36. El procedimiento según la reivindicación 35, en el que el acto (302) de identificar una clase que representa los parámetros para un procedimiento comprende un acto de acceder una clase pública que representa parámetros para un procedimiento de Lenguaje Común del Tiempo de ejecución.

30 37. El procedimiento según la reivindicación 35, en el que el acto (302) de identificar un objeto tipeado que corresponde al elemento del SOAP comprende un acto de despachar la envolvente del SOAP (132) a un procedimiento apropiado.

38. El procedimiento según la reivindicación 24, en el que el acto (303) de mapear el elemento del SOAP accedido en parámetros del objeto tipeado de acuerdo con el uno o más atributos de declaración del mensaje (171) comprende un acto de des-serializar el elemento del SOAP en parámetros del objeto tipeado.

35 39. El procedimiento según la reivindicación 24, en el que el acto (303) de mapear el elemento del SOAP accedido en parámetros del objeto tipeado de acuerdo con uno o más atributos de declaración del mensaje (171) comprende un acto de mapear los elementos de XML dentro de los parámetros del objeto tipeado correspondiente.

40. El procedimiento según la reivindicación 24, que comprende además:

un acto de invocar un procedimiento con los parámetros del objeto tipeado.

40 41. Un producto de programa de ordenador para su uso en un sistema de ordenador (101) que es la red (161) conectable junto con uno o más sistemas de ordenador (111) a una red, el producto de programa de ordenador para la implementación de un procedimiento para el envío de modo flexible de datos de la aplicación tipeados (131), comprendiendo el producto de programa de ordenador uno o más medios legibles por ordenador que tienen almacenadas sobre el mismo instrucciones ejecutables por ordenador, que cuando se ejecutan por un procesador 45 (421) causan que el sistema de ordenador realice lo siguiente:

acceder (201) a parámetros del objeto tipeado correspondientes al objeto tipeado, estando anotado el objeto tipeado con uno o más atributos de declaración del mensaje de un modelo de declaración del mensaje, definiendo el uno o más atributos de declaración del mensaje un mapeo entre los parámetros del objeto tipeado y los elementos del SOAP correspondientes;

50 mapear (202) los parámetros del objeto tipeado accedido en un elemento del SOAP de acuerdo con el uno o más atributos de declaración del mensaje; y referirse (203) a uno o más atributos de declaración del mensaje para identificar una localización dentro de una

envolvente del SOAP;  
 insertar (204) el elemento del SOAP en la localización identificada dentro de la envolvente del SOAP.

- 5 42. El producto de programa de ordenador según la reivindicación 41, en el que las instrucciones ejecutables por ordenador, que cuando se ejecutan, causan que el sistema de ordenador acceda (201) a los parámetros del objeto tipeado correspondiente a un objeto tipeado comprende instrucciones ejecutables por ordenador, que cuando se ejecutan, causan que el sistema de ordenador acceda a una clase pública que representa parámetros para un procedimiento de Lenguaje Común del Tiempo de ejecución.
- 10 43. El producto de programa de ordenador según la reivindicación 41, en el que las instrucciones ejecutables por ordenador, que cuando se ejecutan, causan que el sistema de ordenador acceda (201) a los parámetros del objeto tipeado correspondiente a un objeto tipeado comprende instrucciones ejecutables por ordenador, que cuando se ejecutan, causan que el sistema de ordenador acceda a una clase pública que representa un valor de retorno del procedimiento de Lenguaje Común del Tiempo de ejecución.
- 15 44. El producto de programa de ordenador según la reivindicación 41, en el que las instrucciones ejecutables por ordenador, que cuando se ejecutan, causan que el sistema de ordenador mapee (202) los parámetros del objeto tipeado accedido en un elemento del SOAP de acuerdo con uno o más atributos de declaración del mensaje comprende instrucciones ejecutables por ordenador, que cuando se ejecutan, causan que el sistema de ordenador serialice el acceso a los parámetros del objeto tipeado para un elemento del SOAP.
- 20 45. El producto de programa de ordenador según la reivindicación 41, en el que las instrucciones ejecutables por ordenador, que cuando se ejecutan, causan que el sistema de ordenador se refiera (203) a uno o más atributos de declaración del mensaje para identificar una localización dentro de una envolvente del SOAP comprenden instrucciones ejecutables por el ordenador, que cuando se ejecutan, causan que el sistema de ordenador se refiera a un atributo de declaración del mensaje que identifica una porción de cabecera de la envolvente del SOAP.
- 25 46. El producto de programa de ordenador según la reivindicación 41, en el que las instrucciones ejecutables por ordenador, que cuando se ejecutan, causan que el sistema de ordenador se refiera (203) a uno o más atributos de declaración del mensaje para identificar una localización dentro de una envolvente del SOAP comprenden instrucciones ejecutables por el ordenador, que cuando se ejecutan, causan que el sistema de ordenador se refiera a un atributo de declaración del mensaje que identifica una porción del cuerpo de la envolvente del SOAP.
- 30 47. El producto de programa de ordenador según la reivindicación 41, en el que las instrucciones ejecutables por ordenador, que cuando se ejecutan, causan que el sistema de ordenador se refiera a insertar (204) el elemento del SOAP en la localización identificada dentro de la envolvente del SOAP comprenden instrucciones ejecutables por el ordenador, que cuando se ejecutan, causan que el sistema de ordenador inserte el elemento del SOAP en una porción de cabecera de la envolvente del SOAP.
- 35 48. El producto de programa de ordenador según la reivindicación 41 en el que las instrucciones ejecutables por ordenador, que cuando se ejecutan, causan que el sistema de ordenador se refiera a insertar (204) el elemento del SOAP en la localización identificada dentro de la envolvente del SOAP comprenden instrucciones ejecutables por el ordenador, que cuando se ejecutan, causan que el sistema de ordenador inserte el elemento del SOAP en una porción del cuerpo de la envolvente del SOAP.
- 40 49. Un producto de programa de ordenador para su uso en un sistema de ordenador (111) que es la red (161) conectable junto con otros uno o más sistemas de ordenador (101) a una red, el producto de programa de ordenador para implementar un procedimiento para recibir de modo flexible datos de aplicación tipeados (131), comprendiendo el producto de programa de ordenador uno o más medios legibles por el ordenador que tienen almacenados en el mismo instrucciones ejecutables por el ordenador, que cuando se ejecutan por un procesador (421), causan que el sistema de ordenador realice lo siguiente:
- 45 acceder (301) a un elemento del SOAP incluido en una envolvente del SOAP;  
 identificar (302) un objeto tipeado que corresponde al elemento del SOAP, estando anotado el objeto tipeado con uno o más atributos de declaración del mensaje de un modelo de declaración del mensaje, definiendo el uno o más atributos de declaración del mensaje un mapeo entre elementos del SOAP y los parámetros del objeto tipeado correspondiente; y
- 50 mapear (303) el elemento del SOAP accedido en parámetros del objeto tipeado de acuerdo con uno o más atributos de declaración del mensaje, presentándose los parámetros del objeto tipeado al objeto tipeado identificado.
- 55 50. El producto de programa de ordenador según la reivindicación 49, en el que las instrucciones ejecutables por ordenador, que cuando se ejecutan causan que el sistema de ordenador se refiera al acceso (301) de un elemento del SOAP incluido en una envolvente del SOAP comprenden instrucciones ejecutables por el ordenador, que cuando se ejecutan, causan que el sistema de ordenador acceda a elementos de XML de una porción de la cabecera de la envolvente del SOAP.

51. El producto de programa de ordenador según la reivindicación 49, en el que las instrucciones ejecutables por ordenador, que cuando se ejecutan causan que el sistema de ordenador se refiera al acceso (301) de un elemento del SOAP incluido en una envoltente del SOAP comprenden instrucciones ejecutables por el ordenador, que cuando se ejecutan, causan que el sistema de ordenador acceda a elementos de XML de una porción del cuerpo de la envoltente del SOAP.
52. El producto de programa de ordenador según la reivindicación 49, en el que las instrucciones ejecutables por ordenador, que cuando se ejecutan causan que el sistema de ordenador se refiera a la identificación (302) de un objeto tipeado que corresponde al elemento del SOAP comprenden instrucciones ejecutables por ordenador, que cuando se ejecutan, causan que el sistema de ordenador acceda a un URI correspondiente al objeto tipeado.
- 10 53. El producto de programa de ordenador según la reivindicación 49, en el que las instrucciones ejecutables por ordenador, que cuando se ejecutan causan que el sistema de ordenador mapee (303) el elemento del SOAP accedido en parámetros del objeto tipeado de acuerdo con el uno o más atributos de declaración del mensaje comprenden instrucciones ejecutables por ordenador, que cuando se ejecutan, causan que el sistema de ordenador des-serialice el elemento del SOAP en los parámetros del objeto tipeado.

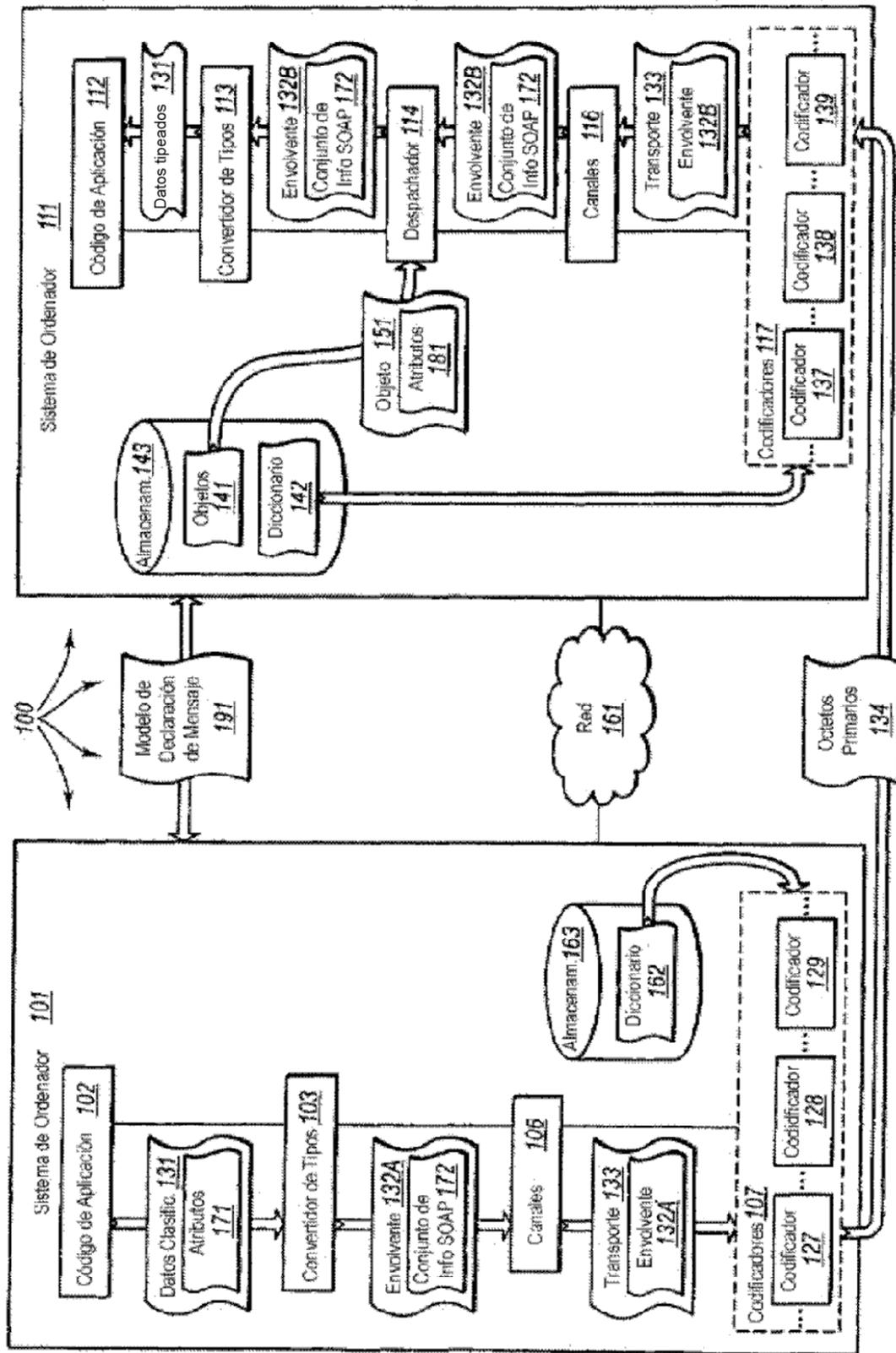
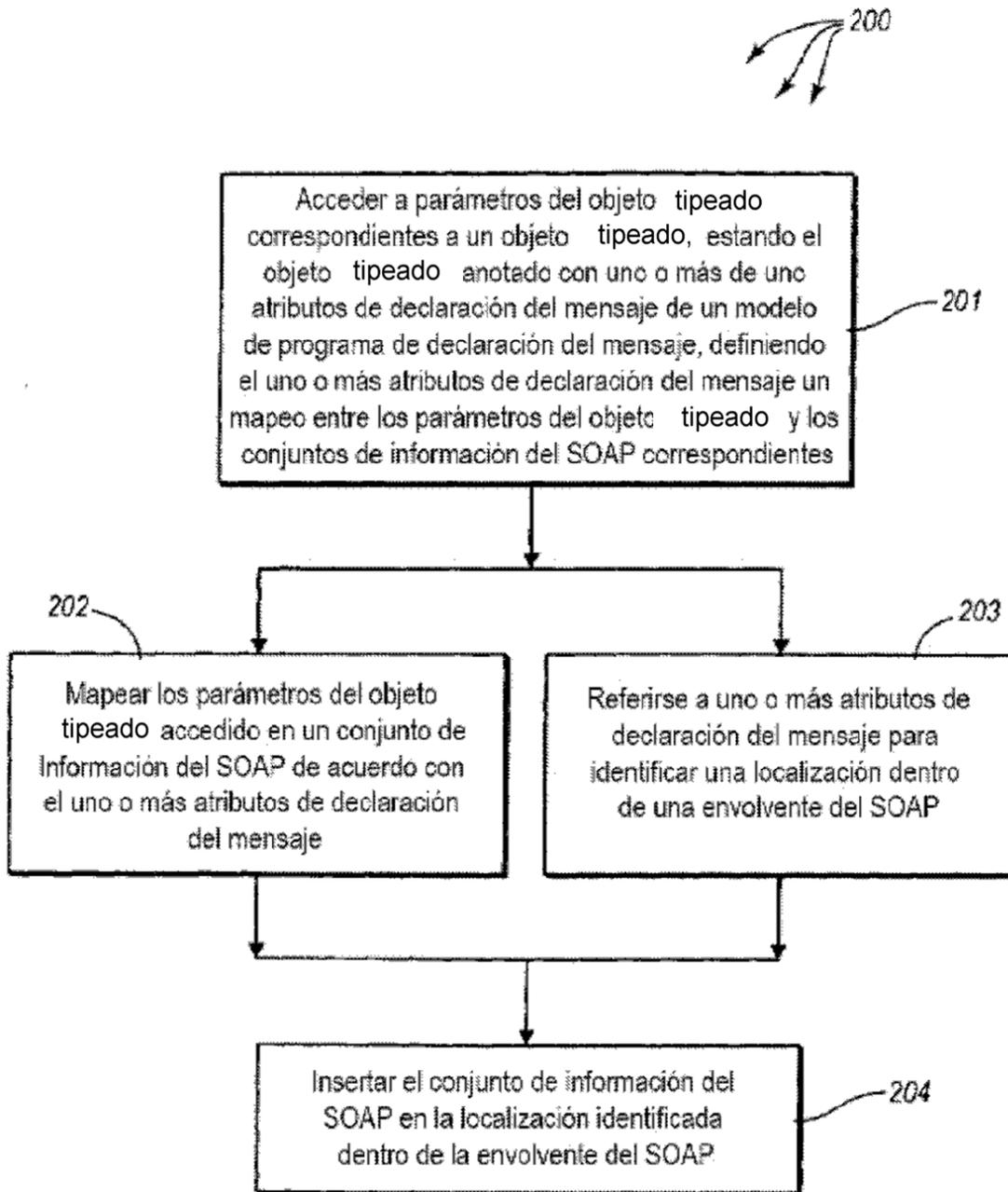
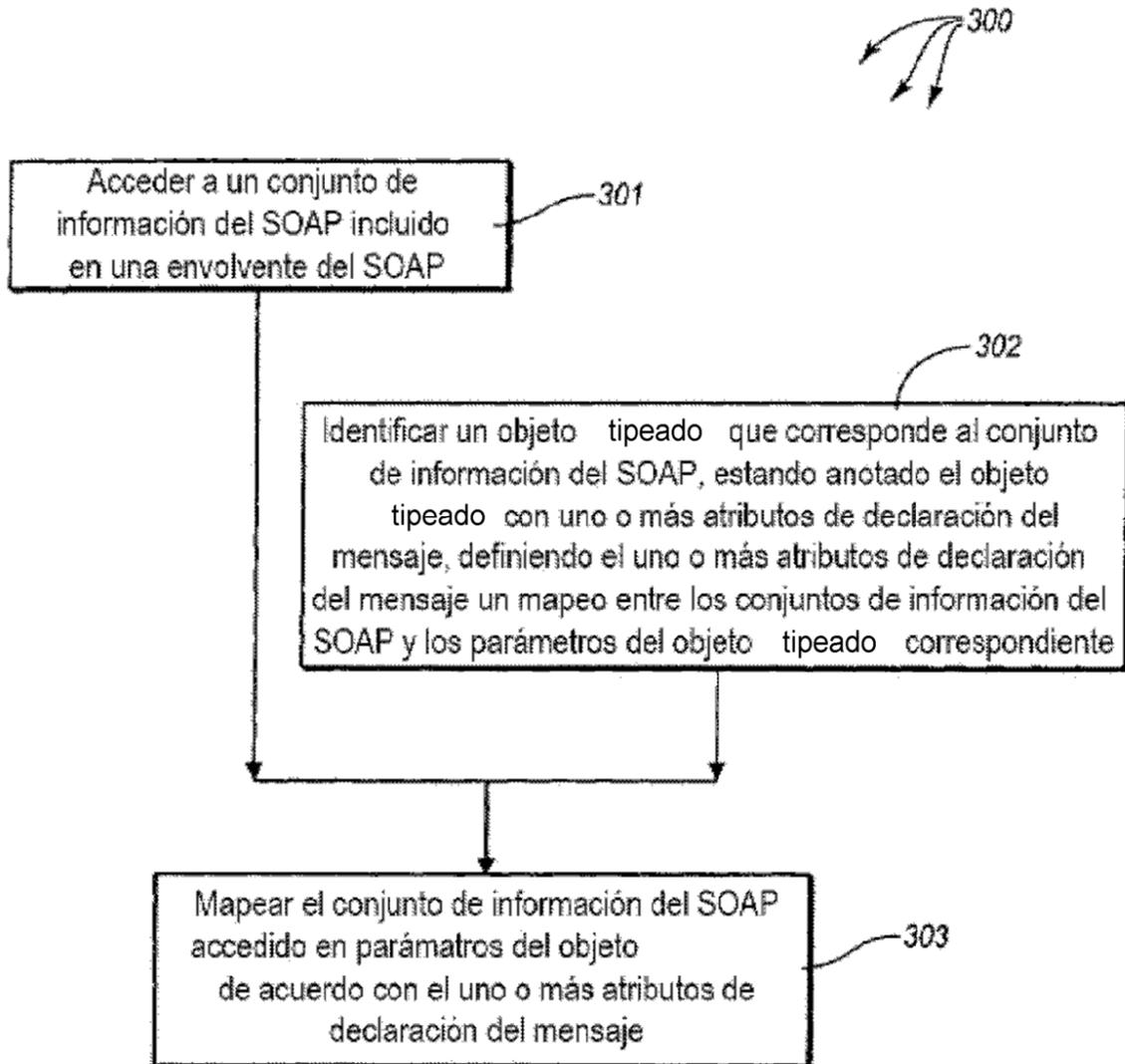


Fig. 1



**Fig. 2**



**Fig. 3**

