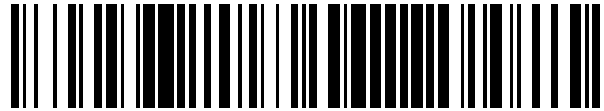


19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 635 437**

51 Int. Cl.:

**G06F 9/45**

(2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **11.03.2014 E 14158977 (0)**

97 Fecha y número de publicación de la concesión europea: **12.07.2017 EP 2778914**

54 Título: **Método y sistema para generar un analizador y análisis de los datos complejos**

30 Prioridad:

**15.03.2013 US 201361801432 P**  
**05.02.2014 US 201414173743**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**03.10.2017**

73 Titular/es:

**PALANTIR TECHNOLOGIES, INC. (100.0%)**  
**100 Hamilton Avenue Suite 300**  
**Palo Alto, California 94301, US**

72 Inventor/es:

**ELLIOT, MARK**

74 Agente/Representante:

**SÁEZ MAESO, Ana**

**ES 2 635 437 T3**

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

**DESCRIPCIÓN**

Método y sistema para generar un analizador y análisis de los datos complejos

5 Campo de la invención

Esto se refiere a un analizador y al análisis de los datos complejos.

Antecedentes

10

La cantidad de información que se procesa y se almacena aumenta rápidamente a medida que los avances tecnológicos presentan una capacidad cada vez mayor para generar y almacenar datos. Adicionalmente, los sistemas informáticos están cada vez más integrados por lo que existe una necesidad de integrar correctamente, pero con eficiencia los datos de un sistema en otro sistema. Garantizar la corrección de conversión puede requerir gastos considerables de desarrollo de software, lo cual está en tensión con la necesidad de realizar esta conversión con eficiencia comercial.

15

20

Un tipo común de conversión de formato de datos es la conversión de datos desde un primer formato textual a un segundo formato textual. Los ejemplos de tal conversión son evidentes fácilmente en todos los campos comerciales, educativos, políticos y técnicos. Por ejemplo, un registro electrónico para una compra con tarjeta de crédito puede componerse en parte de varios campos de texto, que incluyen el nombre del titular de la tarjeta, un número de identificación de la tarjeta de crédito usada en la transacción, e información comercial que identifica la naturaleza de la compra y que identifica al comerciante. Los consumidores frecuentemente dan seguimiento a sus compras con tarjeta de crédito a través de un pago de facturas en línea o software de banca en línea, pero el formato textual de los datos de transacciones con tarjetas de crédito dentro del entorno del pago de facturas en línea puede diferir del formato textual de los datos de transacciones con tarjetas de crédito dentro del entorno del procesador de tarjetas de crédito de origen. Por lo tanto, es necesaria la conversión de formato de datos para integrar los datos formateados de acuerdo con el entorno informático de una empresa de tarjetas de crédito con el entorno informático de la aplicación de pago de facturas en línea de un consumidor. Afortunadamente para los proveedores de software de pago de facturas, el formato de los datos de transacciones con tarjetas de crédito es relativamente sencillo y relativamente estable en comparación con otros entornos de conversión de datos.

25

30

35

Algunos entornos de conversión de datos tienen requisitos de conversión de datos muy complejos y estos requisitos de conversión de datos pueden estar sujetos a revisiones frecuentes. Los requisitos de conversión de datos complejos y los requisitos de conversión de datos que cambian con frecuencia pueden surgir en situaciones en las que un analizador debe procesar datos de numerosas fuentes independientes, cada una de las cuales puede dar formato a sus datos en formas arbitrariamente complejas y puede agregar nuevos formatos o cambiar formatos existentes con frecuencia arbitrariamente. A medida que aumenta el número de formatos diferentes que un analizador debe soportar, aumenta la complejidad del analizador. A medida que aumenta la complejidad del analizador, pueden aumentar drásticamente los recursos de desarrollo de software necesarios para actualizar y probar el analizador. Por lo tanto, el aumento de la complejidad del analizador está en tensión tanto con los objetivos de la conversión de datos fiable como comercialmente eficiente.

40

45

Las herramientas de análisis existentes no funcionan bien en los entornos de análisis complejos que cambian frecuentemente. Un enfoque tradicional para diseñar un analizador de texto es para un desarrollador de software escribir expresiones regulares que reconocerán cadenas o porciones de una cadena y modificar esas cadenas o porciones de una cadena mediante una transformación predefinida. Un problema con este enfoque es que las transformaciones de expresión regular pueden proporcionar soluciones eficientes y correctas para las conversiones de datos relativamente simples, pero las transformaciones complejas que usan la expresión regular pueden ser muy difíciles de escribir, probar, modificar y/o interpretar. Además, en algunos entornos de conversión de datos, un resultado generado por una transformación de expresión regular puede ser una entrada a otra transformación de expresión regular, lo que tiende a aumentar significativamente la complejidad conceptual y el gasto práctico de desarrollar y mantener un analizador basado en expresiones regulares.

50

55

La referencia se dirige al documento US 2010/0306285; Maggie Johnson: "Introduction to yacc and bison", 8 de julio de 2005, páginas 1-11; y US 7 089 541 B2. US 2010/0306285 se refieren a especificar un analizador mediante el uso de un archivo de propiedades.

60

La invención se expone en la reivindicación 1 y la reivindicación 5 y sus características subsidiarias se exponen en sus respectivas reivindicaciones dependientes.

Breve descripción de las figuras

65

Ahora se hará referencia a los dibujos acompañantes que muestran las modalidades ilustrativas de la presente solicitud, y en los cuales:

La Figura 1 ilustra un entorno informático para recibir una definición del analizador, generar un analizador, y analizar los datos de texto con el analizador generado, de acuerdo con algunas modalidades de la presente descripción.

5 La Figura 2 ilustra un diagrama de alto nivel de la estructura de una definición del analizador ilustrativo, de acuerdo con algunas modalidades de la presente descripción.

La Figura 3 ilustra un diagrama de alto nivel de la estructura de una definición del analizador ilustrativo que comprende las etiquetas de marcado, de acuerdo con algunas modalidades de la presente descripción.

10 La Figura 4 ilustra un diagrama de alto nivel de la estructura de un controlador ilustrativo, de acuerdo con algunas modalidades de la presente descripción.

15 La Figura 5 ilustra un diagrama de alto nivel de la estructura de un controlador ilustrativo, de acuerdo con algunas modalidades de la presente descripción.

La Figura 6 ilustra la estructura de un analizador construido ilustrativo, de acuerdo con algunas modalidades de la presente descripción.

20 La Figura 7 ilustra las etapas de un proceso ilustrativo, de acuerdo con algunas modalidades de la presente descripción.

Descripción detallada de las modalidades ilustrativas

25 Se hará referencia ahora en detalle a las modalidades, cuyos ejemplos se ilustran en los dibujos acompañantes. Donde quiera que sea posible, los mismos números de referencia se usarán a lo largo de todos los dibujos para referirse a las mismas partes o partes similares.

30 Las modalidades de la presente descripción pueden evitar las deficiencias de los analizadores de texto tradicionales al proporcionar sistemas y métodos para definir, construir, y utilizar un analizador de manera que permita una mayor complejidad y flexibilidad de análisis, lo que permite al mismo tiempo un diseño y modificación más eficiente del analizador. Las modalidades de la presente descripción abarcan técnicas para recibir una definición de análisis, generar un analizador de la definición de análisis, y utilizar el analizador generado para realizar transformaciones de texto de mayor complejidad de lo que habría sido práctico con los analizadores tradicionales. Aunque la siguiente descripción se refiere a ciertas modalidades ilustrativas, se apreciará que las modificaciones y variaciones de las modalidades descritas pueden hacerse de acuerdo con las enseñanzas de la presente descripción.

35 Las modalidades de acuerdo con la presente descripción pueden recibir como entrada y generar como información de salida en una o más formas. En algunas modalidades, un analizador de texto recibe datos de texto de cierta forma como entrada y genera como salida una estructura de datos, un informe de texto que documenta la estructura observada, y/o una imagen que representa un resultado del análisis. La estructura de datos de salida puede ser, por ejemplo, un árbol de análisis, un árbol de sintaxis abstracto, o alguna otra estructura de datos. En algunas modalidades, los datos de entrada pueden tomar la forma de un archivo que contiene datos textuales, un secuencia de datos textuales comunicados desde un programa o proceso en un ordenador a otro programa o proceso en el mismo ordenador o un ordenador diferente, o datos textuales almacenados en una base de datos. En algunas modalidades, los datos de salida emitidos desde un analizador de texto también pueden tomar una o más de las formas descritas anteriormente. En algunas modalidades, un analizador puede recibir los datos de entrada y emitir los datos de salida en el mismo formato mientras que en otras modalidades el formato de los datos de entrada puede diferir del formato de los datos de salida. Aunque los analizadores típicamente pueden recibir los datos en una sola forma y emitir los datos en una sola forma, un analizador de acuerdo con la presente descripción podría recibir los datos textuales a partir de una o más entradas y/o podría emitir los datos textuales en una o más formas como salidas.

40 En algunas modalidades, un analizador puede generar datos distintos de los datos textuales como una salida. Algunos analizadores se diseñan para reconocer patrones textuales y comunicar indicadores lógicos cuando uno o más patrones textuales están presentes dentro de una cadena de entrada textual. Por ejemplo, un analizador puede generar una indicación lógica que refleja que una cadena de texto predeterminada está presente o ausente de los datos de entrada. En otro ejemplo, un analizador puede generar una indicación lógica que refleja que una primera cadena precede a una segunda cadena dentro de los datos de entrada.

45 En algunas modalidades, puede generarse más de una indicación lógica desde un analizador. Por ejemplo, un analizador puede generar una primera indicación lógica que refleja que una primera cadena de texto predeterminada está presente o ausente de los datos de entrada y también puede generar una segunda indicación lógica que refleja que una segunda cadena de texto predeterminada está presente en una posición antes de una tercera cadena de texto predeterminada dentro de los datos de entrada. Algunas modalidades de la presente descripción son de este tipo, como se describe en más detalle a continuación.

50 En algunas modalidades, un analizador puede recibir una o más indicaciones lógicas como entradas a un analizador. Por ejemplo, un analizador puede configurarse para buscar una primera cadena de texto si una indicación lógica de

entrada es falsa pero el analizador puede configurarse para buscar una segunda cadena de texto si la indicación lógica de entrada es verdadera.

En algunas modalidades, un analizador puede configurarse para recibir una indicación de cadena de texto como una entrada. Por ejemplo, un analizador puede configurarse para buscar una cadena de texto que se recibe como una entrada al analizador - separada de la entrada de datos al analizador. En algunas modalidades, otras indicaciones de entrada tales como números enteros o de punto flotante pueden recibirse y utilizarse por un analizador para influir en las operaciones del analizador. En algunas modalidades, una combinación de dos o más indicaciones de entrada de uno o más tipos descritos anteriormente pueden recibirse por el analizador e influir en las operaciones del analizador.

La Figura 1 ilustra un entorno informático ilustrativo dentro del cual pueden implementarse las modalidades de la presente descripción.

El sistema informático 100 incluye un bus 102 u otro mecanismo de comunicación para comunicar la información, y uno o más procesadores de hardware 104 (denominado como el procesador 104 para fines de simplicidad) acoplados con el bus 102 para procesar la información. El procesador de hardware 104 puede ser, por ejemplo, uno o más microprocesadores de propósito general o puede ser un conjunto de instrucciones reducidas de uno o más microprocesadores.

El sistema informático 100 también incluye una memoria principal 106, tal como una memoria de acceso aleatorio (RAM) u otro dispositivo de almacenamiento dinámico, acoplado al bus 102 para almacenar la información y las instrucciones que se ejecutan por el procesador 104. La memoria principal 106 también puede usarse para almacenar variables temporales u otra información intermedia durante la ejecución de las instrucciones por el procesador 104. Tales instrucciones, cuando se almacenan en medios de almacenamiento no transitorios accesibles al procesador 104, hacen que el sistema informático 100 se convierta en una máquina de propósito especial que se personaliza para realizar las operaciones especificadas en las instrucciones.

El sistema informático 100 incluye además una memoria de sólo lectura (ROM) 108 u otro dispositivo de almacenamiento estático acoplado al bus 102 para almacenar la información estática y las instrucciones para el procesador 104. Se proporciona un dispositivo de almacenamiento 110, tal como un disco magnético, disco óptico, o unidad de memoria USB (unidad flash), etc. y se acopla al bus 102 para almacenar la información y las instrucciones.

El sistema informático 100 puede acoplarse a través del bus 102 a un visualizador 112, tal como un tubo de rayos catódicos (CRT), una pantalla LCD, o una pantalla táctil para visualizar la información a un usuario de ordenador. Un dispositivo de entrada 114, que incluye teclas alfanuméricas y otras teclas, se acopla al bus 102 para comunicar la información y las selecciones de comando al procesador 104. Otro tipo de dispositivo de entrada de usuario es el control del cursor 116, tal como un mouse, una bola de seguimiento, o teclas de dirección del cursor para comunicar la información de dirección y las selecciones de comando al procesador 104 y para controlar el movimiento del cursor en el visualizador 112. El dispositivo de entrada típicamente tiene dos grados de libertad en dos ejes, un primer eje (por ejemplo, x) y un segundo eje (por ejemplo, y), que permiten al dispositivo especificar las posiciones en un plano. En algunas modalidades, la misma información de dirección y las selecciones de comando como un control del cursor pueden implementarse a través de toques de recepción en una pantalla táctil sin un cursor.

El sistema informático 100 puede incluir un módulo de interfaz de usuario para implementar una GUI que puede almacenarse en un dispositivo de almacenamiento masivo como códigos de software ejecutables que se ejecutan por el uno o más dispositivos informáticos. Este y otros módulos pueden incluir, a manera de ejemplo, componentes de software, componentes de software orientados a objetos, componentes de clase y tarea, procesos, funciones, atributos, procedimientos, subrutinas, segmentos de código de programa, controladores, microprograma, microcódigo, circuitos, datos, bases de datos, estructuras de datos, tablas, matrices, y variables.

Generalmente, la palabra "módulo", como se usa en la presente descripción, se refiere a la lógica incorporada en el hardware o microprograma, o a una colección de instrucciones de software, que tiene posiblemente puntos de entrada y salida, escritos en un lenguaje de programación, tales como, por ejemplo, Java, Lua, C o C++. Un módulo de software puede compilarse y vincularse a un programa ejecutable, instalado en una biblioteca de vínculos dinámicos, o escrito en un lenguaje de programación interpretado tales como, por ejemplo, BASIC, Perl, o Python. Se apreciará que los módulos de software pueden llamarse de otros módulos o desde ellos mismos, y/o pueden invocarse en respuesta a eventos o interrupciones detectadas. Los módulos de software configurados para la ejecución en los dispositivos informáticos pueden proporcionarse en un medio legible por ordenador, tales como un disco compacto, disco de video digital, unidad flash, disco magnético, o cualquier otro medio tangible, o como una descarga digital (y pueden almacenarse originalmente en un formato comprimido o instalable que requiere instalación, descompresión, o descifrado antes de la ejecución). Tal código de software puede almacenarse, parcial o totalmente, en un dispositivo de memoria del dispositivo informático de ejecución, para la ejecución por el dispositivo informático. Las instrucciones de software pueden incorporarse en el microprograma, tal como una EPROM. Se apreciará además que los módulos de hardware pueden componerse de unidades lógicas conectadas, tales como compuertas, y/o pueden componerse de unidades programables, tales como matrices o procesadores de compuertas programables. Los módulos o la funcionalidad del dispositivo informático descritos en la presente descripción se implementan preferentemente como módulos de software,

pero pueden representarse en el hardware o microprograma. Generalmente, los módulos descritos en la presente descripción se refieren a los módulos lógicos que pueden combinarse con otros módulos o dividirse en submódulos a pesar de su organización física o almacenamiento.

5 El sistema informático 100 puede implementar los procesos y técnicas descritas en la presente descripción mediante el uso de lógica de cableado personalizada, uno o más ASIC o FPGA, microprograma y/o lógica de programa que en combinación con el sistema informático hace o programa el sistema informático 100 para que sea una máquina de propósito especial. De acuerdo con algunas modalidades, las técnicas y otras características descritas en la presente descripción se realizan por el sistema informático 100 en respuesta al procesador 104 que ejecuta una o más  
10 secuencias de una o más instrucciones contenidas en la memoria principal 106. Tales instrucciones pueden leerse en la memoria principal 106 desde otro medio de almacenamiento, tal como dispositivo de almacenamiento 110. La ejecución de las secuencias de las instrucciones contenidas en la memoria principal 106 hace que el procesador 104 realice las etapas del proceso descrito en la presente descripción. En las modalidades alternativas, los circuitos cableados pueden usarse en lugar de o en combinación con las instrucciones de software.

15 El término "medio no transitorio" como se usa en la presente descripción se refiere a cualquier medio no transitorio que almacena datos y/o instrucciones que hacen que una máquina funcione de una manera específica. Tales medios no transitorios pueden comprender medios no volátiles y/o medios volátiles. Los medios no volátiles incluyen, por ejemplo, disco ópticos o magnéticos, tal como el dispositivo de almacenamiento 110. Los medios volátiles incluyen una memoria  
20 dinámica, tal como la memoria principal 106. Las formas comunes de medios no transitorios incluyen, por ejemplo, un disquete, un disco flexible, disco duro, unidad de estado sólido, cinta magnética, o cualquier otro medio de almacenamiento de datos magnéticos, un CD-ROM, cualquier otro medio de almacenamiento de datos óptico, cualquier medio físico con patrones de orificios, una RAM, una PROM, y EPROM, una FLASH-EPROM, NVRAM, cualquier otro chip o cartucho de memoria, y versiones en red del mismo.

25 El medio no transitorio es distinto pero puede usarse junto con los medios de transmisión. Los medios de transmisión participan en la transferencia de información entre medios de almacenamiento. Por ejemplo, los medios de transmisión incluyen cables coaxiales, alambre de cobre y fibras ópticas, que incluyen los alambres que comprenden el bus 102. Los medios de transmisión pueden tener la forma de ondas acústicas o luminosas tales como las generadas durante las  
30 comunicaciones de datos de ondas de radio e infrarrojas.

Diversas formas de medios de comunicación pueden estar implicadas en el desempeño de una o más secuencias de una o más instrucciones al procesador 104 para la ejecución. Por ejemplo, las instrucciones pueden transportarse inicialmente en un disco magnético o unidad de estado sólido de un ordenador remoto. El ordenador remoto puede  
35 cargar las instrucciones en su memoria dinámica y enviar las instrucciones a través de una línea telefónica mediante el uso de un módem. Un módem localizado en el sistema informático 100 puede recibir los datos en la línea telefónica y usa un transmisor de infrarrojos para convertir los datos a una señal infrarroja. Un detector de infrarrojos puede recibir los datos transportados en la señal de infrarrojos y los circuitos apropiados pueden colocar los datos en el bus 102. El Bus 102 transporta los datos a la memoria principal 106, desde la cual el procesador 104 recupera y ejecuta las  
40 instrucciones. Las instrucciones recibidas por la memoria principal 106 pueden almacenarse opcionalmente en el dispositivo de almacenamiento 110 ya sea antes o después de la ejecución por el procesador 104.

45 El sistema informático 100 también incluye una interfaz de comunicación 118 acoplada al bus 102. La interfaz de comunicación 118 proporciona un acoplamiento de comunicación de datos de doble vía a un enlace de red 120 que se conecta a una red local 122. Por ejemplo, la interfaz de comunicación 118 puede ser una tarjeta de red digital de servicios integrados (ISDN), un módem de cable, módem satelital, o un módem para proporcionar una conexión de comunicación de datos a un tipo correspondiente de línea telefónica. Como otro ejemplo, la interfaz de comunicación 118 puede ser una tarjeta de red de área local (LAN) para proporcionar una conexión de comunicación de datos a una LAN compatible. También pueden implementarse enlaces inalámbricos. En cualquiera de tales implementaciones, la  
50 interfaz de comunicación 118 envía y recibe señales eléctricas, electromagnéticas u ópticas que transportan secuencias de datos digitales que representan varios tipos de información.

55 El enlace de red 120 típicamente proporciona comunicación de datos a través de una o más redes a otros dispositivos de datos. Por ejemplo, el enlace de red 120 puede proporcionar una conexión a través de la red local 122 a un ordenador central 124 o a un equipo de datos operado por un proveedor de servicios de Internet (ISP) 126. El ISP 126 a su vez proporciona servicios de comunicación de datos a través de la red mundial de comunicación de datos por paquetes ahora denominada comúnmente como "Internet" 128. Tanto la red local 122 como Internet 128 usan señales eléctricas, electromagnéticas u ópticas que transportan secuencias de datos digitales. Las señales a través de las  
60 diversas redes y las señales en el enlace de red 120 y a través de la interfaz de comunicación 118, que transportan los datos digitales hasta y desde el sistema informático 100, son formas ilustrativas de los medios de transmisión.

65 El sistema informático 100 puede enviar mensajes y recibir datos, que incluye un código de programa, a través de la(s) red(es), el enlace de red 120 y la interfaz de comunicación 118. En el ejemplo de Internet, un servidor 130 puede transmitir un código solicitado para un programa de aplicación a través de Internet 128, ISP 126, la red local 122 y la interfaz de comunicación 118. El código recibido puede ejecutarse por el procesador 104 cuando se recibe y/o se almacena en el dispositivo de almacenamiento 110, u otro almacenamiento no volátil para su posterior ejecución.

5 En algunas modalidades, el sistema informático 100 recibe una definición del analizador, genera un analizador, y analiza los datos textuales mediante el uso del analizador generado. Por ejemplo, un desarrollador de software puede definir un analizador de acuerdo con las modalidades de la presente descripción mediante la utilización del dispositivo de entrada 114, control del cursor 116, y/o visualizador 112 para introducir una definición del analizador al procesador 104.

10 En algunas modalidades, el procesador 104 almacena el analizador definido en el dispositivo de almacenamiento 110 o en la memoria principal 106. En otras modalidades, el procesador 104 puede almacenar una definición del analizador en una localización de almacenamiento acoplada al procesador 104 a través del enlace de red 120, tal como un disco duro en el ordenador central 124. Después de que se haya almacenado el analizador definido, el procesador 104 puede recibir la definición del analizador mediante la ejecución de instrucciones que leen el analizador definido desde su localización de almacenamiento. En algunas modalidades, la localización de almacenamiento para el analizador definido es uno o más del dispositivo de almacenamiento 110, ROM 108, o memoria principal 106. En otras modalidades, el procesador 104 puede leer el analizador definido desde otras localizaciones dentro del sistema informático 100. Por ejemplo, el procesador 104 puede leer el analizador definido desde un archivo almacenado en el ordenador central 124 o en el servidor 130 a través de un sistema de archivo de red tal como NFS. En otro ejemplo, el analizador definido puede leerse a través del enlace de red 120 mediante el uso de protocolos de comunicación tal como FTP o un protocolo asociado con un sistema de control de versión distribuida, tal como Git. En otros ejemplos, el procesador 104 puede ejecutar las instrucciones que recibe un analizador definido a través de la comunicación entre procesos desde otro programa que se ejecuta por el procesador 104 o a través de llamadas de procedimiento remotas con el ordenador central 124 o servidor 130.

25 Después que el procesador 104 recibe una definición del analizador, el procesador 104 puede usar la definición del analizador para generar un analizador. Los detalles relacionados con la generación de un analizador desde una definición del analizador se describen a continuación. En algunas modalidades, el analizador generado puede comprender las instrucciones que pueden ejecutarse directamente por uno o más procesadores, tal como el procesador 104. En otras modalidades, el procesador generado puede comprender el código de bytes ejecutable, que incluye pero sin limitarse a código de bytes Java que es ejecutable por una máquina virtual Java. En las modalidades adicionales, el procesador generado puede comprender instrucciones de programa en un lenguaje de alto nivel tal como C++ que pueden compilarse para formar un programa analizador ejecutable. En las modalidades alternativas, el procesador generado puede comprender instrucciones de programa en un lenguaje de programación interpretado tales como Python o en un lenguaje de procesamiento de texto tal como Sed. En algunas modalidades, el analizador generado se construye en un lenguaje orientado a objetos y el analizador generado comprende un objeto con funciones de miembro que realizan el procesamiento descrito más abajo. En otras modalidades, el analizador generado comprende una o más estructuras de datos, tal como una "estructura" en el lenguaje de programación C, además de soportar las funciones que realizan el procesamiento descrito más abajo.

40 Después que el procesador 104 genera el analizador, el procesador 104 puede analizar los datos textuales mediante el uso del analizador generado en la manera descrita en más detalle a continuación. El analizador generado, cuando se ejecuta por el procesador 104, puede recibir una entrada textual desde cualquiera de los mecanismos de E/S descritos anteriormente y puede generar, en algunas modalidades, la salida textual a través de cualquiera de los mecanismos de E/S descritos anteriormente. En algunas modalidades, además, el analizador generado puede generar uno o más indicadores lógicos, como se describió anteriormente. Un indicador lógico puede reflejar que existió una condición predeterminada durante el procesamiento de texto.

45 Un ejemplo de una de tales condiciones predeterminadas podría ser un valor lógico que refleja si un número de teléfono representaba un número internacional por un abonado dentro de los Estados Unidos. En este ejemplo, la condición podría tener un valor de "1" (es decir, VERDADERO) si los primeros tres dígitos de un número de teléfono eran "011" y de cualquier otra manera la condición podría tener un valor de "0" (es decir, FALSO).

50 En algunas modalidades, los valores lógicos reflejan la información acumulativa debido a que la comunicación del valor lógico, tal como el valor lógico de la llamada internacional, puede ser acumulativa a la comunicación de otra información que implica la condición, tal como el propio número de teléfono. En otras modalidades, sin embargo, los valores lógicos no son acumulativos a otra información que se comunica debido a que ya no existe la cadena de entrada que condujo al valor lógico que se genera. Por ejemplo, un valor lógico puede reflejar que una secuencia particular de caracteres de texto fue reconocida antes de que la secuencia de caracteres se reordenó o cambió por el analizador. Como se describe más abajo, algunas modalidades de la presente descripción comprueban una condición que refleja que una cadena que se analiza está dentro de una clase predeterminada de cadenas a pesar de que la cadena que se analiza puede modificarse durante el procesamiento.

60 En algunas modalidades, el procesador 104 genera el analizador antes de realizar una operación de análisis. Por ejemplo, la iniciación de una operación de análisis puede estar precedida por la generación del analizador. En otras modalidades, el procesador 104 genera el analizador como una etapa de procesamiento explícito separada de una operación de análisis. Por ejemplo, de acuerdo con esta modalidad, el procesador 104 puede dirigirse para generar el analizador como una operación de procesamiento y, después de eso, el procesador 104 puede dirigirse para usar el analizador generado para analizar la información textual.

- En algunas modalidades, una caché de analizadores generados se mantiene por el sistema informático 100. En estas modalidades, si la definición del analizador no ha cambiado desde que se generó por última vez el analizador, la regeneración del analizador de la generación del analizador puede ser ineficaz con relación a la lectura del analizador generado desde una caché. El procesador 104 puede leer el analizador generado desde la caché como una etapa explícita o como parte de una operación de análisis. En algunas modalidades, se usa un algoritmo de reemplazo usado menos recientemente para determinar cuáles analizadores generados almacenar en la caché en un momento determinado. En otras modalidades, otros algoritmos de sustitución pueden usarse para determinar cuáles analizadores generados almacenar en la caché del analizador en un momento determinado. Los mecanismos de almacenamiento en caché se conocen bien en la técnica por lo que los detalles estructurales de una caché y las etapas para identificar una entrada dentro de la caché y la lectura de y almacenamiento de esa entrada dentro de la caché no se describen en la presente descripción. A continuación una descripción de las modalidades ilustrativas relacionadas con una definición del analizador.
- En algunas modalidades, la estructura de un analizador se define por la información textual que refleja los componentes del analizador, el procesamiento de texto que realiza cada componente, y las salidas que generan cada componente. En algunas modalidades, como se describe anteriormente, un analizador puede generar salida textual y/o valores lógicos como salida. En algunas modalidades, se define una salida del analizador dentro de la definición del analizador.
- La Figura 2 ilustra un diagrama de alto nivel de la estructura de una definición del analizador ilustrativo 200, de acuerdo con algunas modalidades de la presente descripción. Como se ilustra, la estructura de definición del analizador 200 comprende una o más definiciones del controlador. Un controlador, como se describe en más detalle a continuación, puede configurarse para reconocer uno o más patrones de texto como entradas y generar uno o más patrones de texto como salidas. En algunas modalidades, el uno o más controladores realizan la funcionalidad del analizador como un todo.
- Como también se describe más abajo, en algunas modalidades, un controlador puede funcionar independientemente o puede funcionar regenerativamente. En el escenario anterior, un controlador puede generar al menos una salida de cadena de texto o una condición lógica que puede comunicarse a la salida del analizador como un todo sin influir en el comportamiento de otros controladores. En el último escenario, un controlador puede generar al menos una cadena de texto que se introduce en el proceso de análisis, que puede conducir a al menos un controlador que recibe la al menos una cadena de texto como una entrada. Por lo tanto, en este último escenario, la salida de un controlador influye en la entrada de al menos un controlador. Por ejemplo, como se describe más abajo, un controlador puede realizar una operación "dividir", que divide una cadena en una pluralidad de subcadenas. Cada subcadena generada por la operación de división puede conducir directa o indirectamente a una o más subcadenas que se introducen en el proceso para el análisis posterior.
- En la modalidad ilustrativa de la Figura 2, la definición del analizador 200 comprende n definiciones del controlador identificadas como la definición del controlador 202(o) a través de la definición del controlador 202(n-1). En algunas modalidades, la estructura de definición del analizador 200 y las definiciones del controlador 202(0) a 202(n-1) pueden expresarse en un lenguaje de marcado. En algunas modalidades, el lenguaje de marcado usado es XML mientras que en otras modalidades el lenguaje de marcado puede ser SGML, YAML, u otro lenguaje de marcado.
- La Figura 3 ilustra un diagrama de alto nivel de la estructura de una definición del analizador ilustrativo 300 que comprende las etiquetas de marcado, de acuerdo con las modalidades de la presente descripción. Como se ilustra, en algunas modalidades, la definición del analizador 300 comprende la etiqueta del analizador de apertura 306 ("") y etiqueta del analizador de cierre 308 ("</analizador>"). Estas etiquetas pueden encapsular la información que define la estructura del analizador.
- Como se ilustra, en algunas modalidades, la definición del analizador 300 también comprende la etiqueta de controladores de apertura 310 ("") y la etiqueta de controladores de cierre 312 ("</controlador>"). Tales etiquetas pueden encapsular la información que define cada uno de los controladores para el uno o más controladores dentro de la definición del analizador 300.
- En algunas modalidades, al menos dos tipos de controladores pueden definirse dentro de la definición del analizador 300. Un tipo de controlador es un "controlador predeterminado". En algunas modalidades, un controlador predeterminado realiza una descomposición inicial de una cadena en múltiples subcadenas constituyentes, cada una de las cuales se introduce en el canal de procesamiento para el procesamiento posterior como se describe más abajo. En otras modalidades, el controlador predeterminado realiza una modificación de cadena y la cadena modificada se introduce en el canal de procesamiento para el procesamiento posterior como se describe más abajo.
- Un ejemplo de una operación de descomposición de cadena es descomponer un número de teléfono de Estados Unidos de 10 dígitos en un código de área, una central telefónica, y cuatro dígitos para distinguir a los clientes dentro de la central telefónica. Un número de teléfono puede representarse como "XXX-XXX-XXXX", donde cada "X" representa un entero de cero a nueve. Además las subcadenas dentro del número de teléfono se delimitan por el carácter "-" en este ejemplo. Por lo tanto, en este ejemplo, los tres primeros enteros reflejan el código de área, los segundos tres enteros

reflejan la central telefónica, y los cuatro enteros restantes distinguen entre los usuarios dentro de la central telefónica. Cada uno de estos componentes podría ser la salida de un controlador predeterminado como entradas a uno o más controladores posteriores.

5 Un ejemplo de una operación de modificación de cadena es traducir una cadena que representa el texto en un primer lenguaje en una cadena que representa el texto en un segundo lenguaje. Por ejemplo, la palabra "teléfono" en inglés puede traducirse al checo como "telefon". En este ejemplo, la salida de un controlador que realiza esta traducción podría ser una entrada a uno o más controladores posteriores.

10 En algunas modalidades, un controlador predeterminado se configura para analizar una cadena si no se configura otro controlador para analizar esa cadena. Por ejemplo, si una cadena de entrada actual al canal de procesamiento era la cadena "Hola mundo" pero ninguno de los controladores 304(0) a 304(n-1) fueron configurados para analizar la cadena "Hola mundo", el controlador predeterminado podría invocarse para analizar esa cadena. En algunas modalidades, un controlador predeterminado puede invocarse como una etapa de análisis inicial para dividir una cadena de entrada inicial en varias subcadenas, cada una de las cuales coincide con que un controlador particular se configure para recibir y analizar aún más.

La Figura 4 ilustra un diagrama de alto nivel de la estructura de una definición del controlador ilustrativo 400, de acuerdo con algunas modalidades de la presente descripción. Como se ilustra, la definición del controlador 400 comprende el receptor del controlador 402 y la acción del controlador 404. Como se describe aún más junto con la Figura 5, el receptor del controlador 402 puede definir que una o más cadenas de entrada predeterminadas de un controlador se configure para recibir y analizar. Después de recibir la una o más cadenas de entrada predeterminadas como una entrada al controlador, como se determina por el receptor del controlador 402, el controlador puede analizar la cadena de entrada a través de las operaciones de análisis definidas dentro de la acción del controlador 404.

La Figura 5 ilustra un diagrama de alto nivel de la estructura del controlador ilustrativo 400, de acuerdo con las modalidades de la presente descripción. Como se ilustra en la Figura 5, la definición del receptor del controlador 402 comprende una etiqueta de detección de apertura ("") y una etiqueta de detección de cierre ("</detectar>"). En algunas modalidades, estas etiquetas encapsulan la información que define la estructura del receptor para un controlador. Como se describe más abajo, el receptor puede configurarse para que coincida con una o más cadenas de entrada predeterminadas. Se describen a continuación algunas cadenas para las cuales un receptor puede configurarse para detectar, de acuerdo con ciertas modalidades.

Como se muestra además en la Figura 5, la acción del controlador 404 comprende una etiqueta de acción de apertura ("") y una etiqueta de acción de cierre ("</acción>"). En esta modalidad, estas etiquetas pueden encapsular la información que define las etapas de análisis que un controlador realizará cuando el receptor del controlador determina que la una o más cadenas de entrada para las cuales se configura el receptor del controlador para reconocer que están presentes en la entrada del controlador. Algunas acciones que un analizador puede tomar en respuesta a la detección de una cadena de entrada, de acuerdo con ciertas modalidades, se describen a continuación.

En otras modalidades, los caracteres de texto empleados para formar las definiciones de detección, acciones, controlador y analizador de apertura y cierre pueden diferir de los caracteres de texto específicos descritos en la presente descripción. Por ejemplo, los caracteres reales de las etiquetas de "detección" de apertura y cierre podrían reemplazarse con los caracteres de una etiqueta de "recepción" de apertura y cierre ("" y "</receptor>") sin apartarse del espíritu de la descripción anterior. En otro ejemplo, la definición del controlador puede expresarse en una forma de expresión distinta de XML, tal como YAML u otro lenguaje de marcado. La importancia de la descripción anterior es que una o más definiciones del analizador, controlador, receptor, y acción pueden estar presentes en una definición del analizador pero las cadenas de texto real dentro de las etiquetas o identificadores para cada sección, para las modalidades que emplean las etiquetas o identificadores, respectivamente, es un detalle de implementación.

Se apreciará que otros mecanismos para dividir porciones de una definición del analizador se conocen y están dentro del espíritu de la descripción anterior. Por ejemplo, una o más porciones de una definición del analizador pueden expresarse en archivos separados de otras porciones de la definición del analizador. En estos ejemplos, una o más porciones de una definición del analizador pueden estar presentes en un archivo separado de un archivo que contiene parte o la totalidad del resto de la definición del analizador. Para tales ejemplos, pueden incorporarse uno o más archivos como referencia en la definición del analizador.

En algunas modalidades, una o más de las siguientes comparaciones se soportan por un receptor dentro de un controlador:

contiene
prefijo
finaliza con



5 Para la comparación "contiene", en algunas modalidades, un argumento para la comparación "contiene" puede ser una cadena predeterminada a escuchar; es decir, una cadena predeterminada para que el controlador compare con las cadenas de entrada recibidas por el controlador. En las modalidades que soportan la comparación "contiene", una definición del receptor para un controlador puede activar las acciones del controlador mediante el uso de la comparación "contiene" para reconocer que el argumento de cadena predeterminada a la comparación "contiene" está presente en algún lugar dentro de la cadena de entrada al controlador; es decir, el controlador ejecuta sus acciones de análisis si la cadena de entrada contiene el argumento de cadena predeterminada. En algunas modalidades, la comparación "contiene" es sensible a las mayúsculas mientras que en otras modalidades la comparación "contiene" no distingue entre mayúsculas y minúsculas.

15 Por ejemplo, una comparación "contiene" puede definir que un controlador debe escuchar cualquier cadena de entrada que contenga el texto "estado". Una definición del receptor que recibe cadenas de entrada "los Estados Unidos", "el presidente hizo una declaración" y "un estado de felicidad" ejecutaría una acción correspondiente al controlador con la comparación "contiene" al recibir cada una de esas cadenas.

20 Para la comparación "prefijo", en algunas modalidades, un argumento a la comparación "prefijo" puede ser una cadena predeterminada a escuchar. En las modalidades que soportan la comparación "prefijo", una definición del receptor para un controlador puede activar las acciones del controlador mediante el uso de la comparación "prefijo" para reconocer que el argumento de cadena predeterminada a la comparación "prefijo" está presente en el comienzo de la cadena de entrada al controlador; es decir, el controlador ejecuta sus acciones de análisis si la cadena de entrada comienza con el argumento de cadena predeterminada. En algunas modalidades, la comparación "prefijo" es sensible a las mayúsculas mientras que en otras modalidades la comparación "prefijo" no distingue entre mayúsculas y minúsculas.

25 Por ejemplo, una comparación "prefijo" puede definir que un controlador debe escuchar cualquier cadena de entrada que comienza con el texto "estado". Una definición del receptor que recibe cadenas de entrada "estado de California", "declaración del presidente" y "preferencia declarada" ejecutaría una acción correspondiente al controlador con la comparación "prefijo" al recibir cada una de esas cadenas.

30 Para la comparación "finaliza con", en algunas modalidades, un argumento a la comparación "finaliza con" puede ser una cadena predeterminada a escuchar. En las modalidades que soportan la comparación "finaliza con", una definición del receptor para un controlador puede activar las acciones del controlador mediante el uso de la comparación "finaliza con" para reconocer que el argumento de cadena predeterminada a la comparación "finaliza con" está presente en el final de la cadena de entrada al controlador; es decir, el controlador ejecuta sus acciones de análisis si la cadena de entrada termina con el argumento de cadena predeterminada. En algunas modalidades, la comparación "finaliza con" es sensible a las mayúsculas mientras que en otras modalidades la comparación "finaliza con" no distingue entre mayúsculas y minúsculas.

40 Por ejemplo, una comparación "finaliza con" puede definir que un controlador debe escuchar cualquier cadena de entrada que termina con el texto "estado". Una definición del receptor que recibe cadenas de entrada "ha abandonado el estado", "interesado", y "el difunto fue testado" ejecutaría una acción correspondiente al controlador con la comparación "finaliza con" al recibir cada una de esas cadenas.

45 Se apreciará que otras comparaciones u otros nombres de comparación también pueden usarse de acuerdo con las modalidades descritas en la presente descripción. Por ejemplo, la comparación "finaliza con" puede reflejarse en una definición del receptor del controlador a través del identificador "final" en lugar de "finaliza con".

En algunas modalidades, una o más de las siguientes acciones se soportan por un controlador:

50	chomp
	emitir
	formato
55	analizar
	reemplazar
	reemplazar-{primero, último, n}
60	dividir
	recortar

65 Para la operación "chomp", en algunas modalidades, un argumento a la operación "chomp" puede ser una cadena predeterminada o expresión regular que se hace coincidir. En las modalidades que soportan la operación "chomp", en una cadena de entrada a un receptor del controlador que coincide con la cadena predeterminada o expresión regular

que se detecta dentro de la entrada actual del controlador, la porción de la cadena de entrada que coincide con la cadena predeterminada o expresión regular puede ser la salida al inicio del proceso de análisis. La cadena resultante de esta operación es una entrada a las acciones posteriores para el controlador o puede ser una entrada a un controlador posterior.

5

Para la operación "emitir", en algunas modalidades, pueden existir dos formas de la operación de emisión. Una forma de la operación "emitir" puede generar una cadena generada por las operaciones de análisis anteriores dentro de las acciones del controlador actual. Por ejemplo, si las acciones A, B, y C son acciones dentro de un controlador y esas acciones modifican la cadena de entrada recibida por el controlador, una operación "emitir" posterior puede generar la cadena resultante de las operaciones de análisis anteriores del controlador.

10

Otra forma de la operación "emitir", en algunas modalidades, puede generar una condición lógica del controlador al analizador. Esta condición lógica puede reflejar un estado actual del analizador o que se realizó una secuencia predeterminada de las operaciones de análisis por el controlador. En esta forma de la operación "emitir", un argumento a la operación "emitir" puede ser una categoría lógica que refleja el tipo de condición lógica que debe indicarse en la ejecución de la operación de emisión. La categoría lógica puede definirse por una cadena de texto tal como "Encontrado: Hola mundo", que refleja que la subcadena "Hola mundo" se encontró dentro de la cadena de entrada al controlador.

15

Para la operación "formato", en algunas modalidades, un argumento a la operación "formato" puede ser una cadena de formato predeterminada que define cómo el controlador debe generar su texto de salida. En algunas modalidades, un formato de sustitución de cadena puede utilizarse para definir cómo deben generarse las cadenas. Por ejemplo, una cadena de formato para un controlador puede comprender la cadena "Encontrado este texto:%s", que indica que el controlador determinó que una cadena de texto predeterminada, localizada en una acción anterior, debería generarse pero precedida por un mensaje que refleje la importancia de este texto localizado. Como se apreciará, el operador "%s" dentro de esta cadena de formato refleja una operación de sustitución de cadena. Muchos otros operadores de sustitución se conocen en la técnica, tales como enteros ("%i"), números reales ("%d"), caracteres ("%c"), y horas y fechas ("%t").

20

25

La cadena resultante de esta operación es una entrada a las acciones posteriores para el controlador o puede ser una entrada a un controlador posterior.

Para la operación "analizar", en algunas modalidades, la(s) cadena(s) actual(es) que se procesan por el controlador se envían al proceso de análisis. Como se describe en otra parte, algunas modalidades generan cadenas al inicio del canal de procesamiento mientras que otras modalidades generan cadenas en una etapa predeterminada dentro del canal de procesamiento. En las modalidades que generan una cadena en una etapa predeterminada dentro del canal de procesamiento, la etapa del proceso de destino para recibir la salida del controlador puede definirse dentro de la definición del controlador como un argumento a la operación "analizar".

30

En algunas modalidades, la operación "analizar" puede ser apropiada para su inclusión dentro de un controlador predeterminado debido a que esta configuración puede condicionar y/o dividir una cadena recibida por un analizador de manera que permite a los otros controladores realizar eficientemente sus operaciones; es decir, una operación común a todas las operaciones de análisis puede incluirse de manera más eficiente en un controlador, tal como el controlador predeterminado, cuyas acciones se ejecutan antes de la ejecución de acciones por otros controladores.

35

Para la operación "reemplazar", en algunas modalidades, los argumentos a la operación "reemplazar" pueden ser una cadena predeterminada para que coincida con la cadena de entrada a la operación de reemplazo y una cadena predeterminada para reemplazar con la cadena que va a hacerse coincidir. Por ejemplo, una operación "reemplazar" puede tener una cadena predeterminada que coincida con "ABC" y una cadena predeterminada para reemplazar con "XYZ". Tras la ejecución de la operación de reemplazo, la cadena o subcadena "ABC" dentro de la cadena de entrada a la operación de reemplazo puede reemplazarse por la cadena "XYZ". En algunas modalidades, una operación "reemplazar" puede ejecutarse múltiples veces en la cadena de entrada. Por ejemplo, para una cadena de entrada "ABCABC", una operación "reemplazar" que dirige el reemplazo de "ABC" con "XYZ" puede generar "XYZXYZ". La cadena resultante de esta operación es una entrada a las acciones posteriores para el controlador o puede ser una entrada a un controlador posterior.

40

45

Para la operación "reemplazar el primero", en algunas modalidades, las operaciones descritas junto con la operación "reemplazar" solamente se realizan en la primera instancia de una cadena predeterminada dentro de la cadena de entrada. Otras variantes de esta idea son una operación "reemplazar el último", que refleja la operación en la última instancia de una subcadena dentro de una cadena, y "reemplazar-n", que refleja la operación en la n-sima instancia de una subcadena dentro de una cadena.

50

55

En algunas modalidades, todas las variantes de la operación "reemplazar" están disponibles mientras que en otras modalidades, solamente están disponibles algunas variantes de la operación "reemplazar".

Para la operación "dividir", en algunas modalidades, un argumento a la operación "dividir" puede ser un carácter o

- cadena delimitadora predeterminada. En las modalidades que soportan la operación "dividir", al detectarse el carácter o cadena delimitadora predeterminada dentro del entrada actual del controlador, la porción de la cadena de entrada que precede al carácter o cadena delimitadora predeterminada se separa de la porción de la cadena de entrada que sigue al carácter o cadena delimitadora predeterminada para formar dos subcadenas. En algunas modalidades, la operación "dividir" puede funcionar múltiples veces en la cadena de entrada. Por ejemplo, una operación "dividir" en un número de teléfono de Estados Unidos, que utiliza un carácter delimitador "-", puede dividir "123-456-7890" en subcadenas "123", "456", y "7890". La(s) cadena(s) resultante de esta operación puede ser una entrada a las acciones posteriores para el controlador o puede ser una entrada a un controlador posterior.
- En las modalidades que soportan la operación "recortar", esta operación elimina espacios delantero y final de la cadena de entrada a la acción. La cadena resultante de esta operación es una entrada a las acciones posteriores para el controlador o puede ser una entrada a un controlador posterior.
- En algunas modalidades, la estructura de la definición del analizador es paralela a la estructura de un analizador generado. La Figura 6 ilustra un analizador generado ilustrativo 600 de acuerdo con algunas modalidades. Los expertos en la técnica reconocerán que también pueden usarse otras operaciones de acuerdo con las modalidades descritas en la presente descripción.
- La Figura 6 ilustra la estructura de un analizador construido 600, de acuerdo con algunas modalidades de la presente descripción. Como se muestra, el analizador 600 comprende una serie de n controladores, identificados como 602(0) a 602(n-1) en la figura. En algunas modalidades, el analizador 600 puede organizarse como un proceso como se muestra en la Figura 6 mientras que en otras modalidades el analizador 600 puede organizarse en una estructura no procesada.
- En algunas modalidades, el analizador 600 recibe una entrada textual en el primer controlador 602(0). En algunas modalidades, el analizador 600 es un "objeto" en un sentido orientado a objetos como aquellos que están familiarizados con los lenguajes orientados a objetos tales como C++ o Java. Adicionalmente, en algunas modalidades, cada controlador también puede ser un objeto; es decir, el objeto del analizador puede componerse de uno o más objetos controladores. En estas modalidades orientadas a objetos, los controladores dentro del analizador pueden comunicarse entre sí mediante un controlador de transmisión que llama a una función miembro de un controlador de recepción. En otras modalidades, las estructuras de datos tales como FIFO o búferes pueden utilizarse para comunicar datos y/o comandos desde un controlador de transmisión hasta un controlador de recepción. En otras modalidades, el analizador puede ser un objeto pero los controladores pueden implementarse como funciones de miembro dentro del objeto analizador. Adicionalmente, en las modalidades no orientadas a objetos, el analizador puede implementarse como una función o una estructura de datos y los controladores también pueden implementarse como una o más funciones.
- En algunas modalidades, el analizador 600 genera una salida textual y una salida lógica, como se describe anteriormente mientras que en otras modalidades el analizador 600 puede o bien generar una salida textual o una salida lógica. Cada controlador dentro del analizador 600 recibe una entrada textual. La salida de cada controlador para una condición particular o entrada textual recibida se configura en la definición del controlador para ese controlador. Durante el análisis, un controlador determina si la entrada textual al controlador coincide con una o más cadenas de texto predeterminadas. Un controlador puede generar salidas textuales o lógicas si la entrada textual al controlador coincide con lo que el controlador está configurado para recibir.
- En algunas modalidades, puede usarse una sola cadena de texto predeterminada para fines de coincidencia dentro del controlador. En otras modalidades, pueden usarse dos o más cadenas de texto para fines de coincidencia dentro del controlador. En las modalidades que emplean dos o más cadenas de texto para fines de coincidencia, son posibles las operaciones booleanas sobre los resultados de comparación para las dos o más cadenas de texto. A manera de ejemplo, en una modalidad del controlador con tres cadenas de texto predeterminadas para fines de coincidencia, por ejemplo, las cadenas de texto A, B, y C, pueden realizarse las operaciones booleanas sobre los resultados de comparación para esas cadenas de texto predeterminadas tales como y, OR, XOR, XNOR, y NOT tales como ((A AND B) XOR (NOT(C))). En algunas modalidades puede comparaciones a realizar por un controlador y las operaciones booleanas sobre los resultados de comparación a generarse por un controlador, si es aplicable a un controlador particular, se definen en la porción del receptor de la definición del controlador para ese controlador.
- En cada una de las modalidades descritas anteriormente, la introducción de una cadena de texto en un controlador que coincida con una o más cadenas de texto predeterminadas del controlador puede invocar las funciones de análisis de una o más funciones del controlador correspondiente establecidas en la porción de acciones de la definición del controlador. De esta manera, un controlador puede configurarse para actuar en las cadenas de texto específicas dentro del canal de procesamiento pero también puede configurarse para ignorar las cadenas de texto distintas de aquellas para las que se configura el controlador para actuar. En algunas modalidades, cuando un controlador se configura para no analizar una cadena de texto particular que recibe como entrada, la cadena de texto se comunica desde el controlador a un controlador siguiente en el canal de procesamiento. Por ejemplo si el controlador 602(1) estaba configurado para no analizar la cadena "Hola mundo", pero recibe esa cadena como entrada, el controlador 602(1) genera la cadena "Hola mundo" al controlador 602(2) en esta modalidad.
- En otras modalidades, pueden usarse mecanismos de comunicaciones alternativos para hacer que la cadena actual del

proceso que se analiza esté disponible para los controladores dentro del canal de procesamiento. Por ejemplo, el analizador puede construirse para comunicar simultáneamente la cadena de texto actual que se procesa a todos los controladores y para dirigir un procesador particular para comparar la cadena de texto actual que se procesa a su una o más cadenas de texto predeterminadas a través de una señal lógica o una llamada de función generada por la lógica de control o código dentro del analizador.

En algunas modalidades, después que un controlador recibe como entrada una cadena que coincide con la una o más cadenas de texto predeterminadas del controlador, el controlador realiza una o más operaciones de análisis en la cadena. En algunas modalidades, la una o más operaciones de análisis a realizar en la cadena de entrada coincidente se define como una o más operaciones de procesamiento dentro de la porción de acción de la definición del controlador. Por ejemplo, un controlador puede configurarse para modificar la cadena "Hola mundo" para formar la cadena "Adiós mundo cruel".

Para las modalidades que implementan las diversas operaciones de análisis que incluyen una operación "reemplazar" descrita anteriormente, existen varias opciones para implementar esta modificación de cadena. Por ejemplo, la operación "reemplazar" podría emplearse para modificar el texto de entrada para formar el texto de salida. En algunas modalidades, las etapas de procesamiento definidas en la definición del controlador se ejecutan secuencialmente; es decir, la primera etapa de procesamiento (por ejemplo, una operación de reemplazo) puede preceder una segunda etapa de procesamiento (por ejemplo, una operación de división) que puede preceder una tercera etapa de procesamiento (por ejemplo, una etapa de generación de salida). En tales modalidades, el orden de las etapas de procesamiento ejecutado por un controlador se define por el orden de las etapas de procesamiento enumeradas en la definición del controlador.

En algunas modalidades, el orden de las definiciones del controlador presente en la definición del analizador determina el orden en que las cadenas de entrada se pasan de un controlador a otro. Por ejemplo, un primer controlador definido en una definición del analizador puede recibir la entrada al analizador mientras que un segundo controlador definido en la definición del analizador puede recibir como entrada la salida del primer controlador. En otras modalidades, todos los controladores podrían recibir simultáneamente todas las cadenas de entrada si las definiciones del receptor de los controladores contemplaban la recepción simultánea de todas las cadenas de entrada. En algunas modalidades donde todos los controladores reciben todas las cadenas de entrada, la lógica de control o software puede habilitar o deshabilitar la consideración de controladores particulares para cadenas de entrada particulares o en base al estado del analizador. En aún otras modalidades, la secuencia de controladores dentro de un proceso puede definirse como parte de cada controlador. Por ejemplo, una primera definición del controlador podría definir un segundo controlador desde el cual el primer controlador recibe su entrada. Alternativamente, un primer controlador podría definir un segundo controlador para el cual la salida del primer controlador es la entrada del segundo controlador.

En otras modalidades, como se muestra en la Figura 6, los controladores pueden generar salidas regenerativas. Como se describió anteriormente, un ejemplo de una operación de análisis que genera una salida regenerativa es una operación de análisis que divide una cadena en dos o más subcadenas que se analizan posteriormente. Por ejemplo, un número de teléfono de Estados Unidos de diez dígitos ("XXX-XXX-XXXX") puede descomponerse en un código de área, una central telefónica, y cuatro dígitos para distinguir los clientes dentro de la central telefónica. En este ejemplo, el número de teléfono puede dividirse en el carácter delimitador "-". Después de eso, en este ejemplo, cada subcadena puede volver a entrar al canal de procesamiento como se ilustra en la Figura 6. En otras modalidades de análisis, las cadenas pueden dividirse en otros caracteres delimitadores tales como espacios, barras ("/" o "\"), puntos ("."), comas, o cualquier carácter o caracteres arbitrarios que diferencian una porción de una cadena de otra porción de la cadena.

En algunas modalidades, las salidas regenerativas se introducen al inicio del canal de procesamiento; es decir, el primer controlador. En otras modalidades, las salidas regenerativas podrían introducirse en un controlador particular del canal de procesamiento. En las modalidades en las cuales una salida regenerativa de un controlador particular se introduce en un controlador distinto del primer controlador en el canal de procesamiento, la definición del controlador para el controlador particular puede incluir como un argumento una definición del controlador de destino que dirige el generador del analizador para comunicar la salida regenerativa del controlador particular a la entrada del controlador de destino (por ejemplo, el controlador 602(2)) en lugar de comunicar esa salida regenerativa a un controlador en el inicio del proceso de análisis (por ejemplo, el controlador 602(0)).

En algunas modalidades, la definición del analizador se usa para crear el analizador. Como se describió anteriormente, el analizador puede comprender instrucciones ejecutables, o código de bytes, o código interpretado. Estas implementaciones del analizador pueden crearse mediante el uso de un generador del analizador para recibir el analizador definido y generar las instrucciones, código de bytes, o código interpretado del analizador definido.

En algunas modalidades, la generación del analizador puede comenzar al leer una plantilla para un objeto del analizador desde un archivo. Esa plantilla puede contener la estructura del objeto del analizador específico a un lenguaje de destino predeterminado tal como Java. En otras palabras, la plantilla puede tener una definición de objetivo, un constructor de objeto, y una o más funciones miembro de objeto para el objeto analizador.

En algunas modalidades, el generador del analizador puede reconocer que los controladores se definen dentro de la

definición del analizador y puede añadir líneas de código de lenguaje de destino, tal como Java, para instanciar las instancias de un objeto controlador dentro del objeto analizador, tal como el constructor para el objeto analizador. El generador del analizador, en ciertas modalidades, también puede reconocer la secuencia de controladores definidos en la definición del analizador y puede secuenciar las entradas y salidas desde o a cada controlador en consecuencia. Por ejemplo, el generador del analizador puede reconocer que la definición del analizador contiene dos controladores y declara que el "controlador1" precede al "controlador2" en un proceso. En este ejemplo, el generador del analizador puede generar el código para el analizador que comunica la entrada al analizador como una entrada al controlador1, comunica la salida del controlador1 como una entrada al controlador2, y comunica la salida del controlador2 como una salida del analizador como un todo.

En algunas modalidades, el generador del analizador también puede leer plantillas escritas en el lenguaje de destino para los controladores dentro del analizador. Por ejemplo, puede existir una plantilla para un controlador que se dirige a un controlador escrito en Java. Esta plantilla puede tener una definición de objetivo para un controlador, una definición de constructor para el objeto controlador, y puede tener funciones de miembro que realizan operaciones de E/S para el objeto controlador.

En algunas modalidades, la plantilla para el controlador puede tener funciones de miembro para una capacidad del receptor como se describe anteriormente y funciones de miembro para realizar una o más acciones en respuesta a la capacidad del receptor. Las funciones de miembro para las capacidades del receptor y de acción pueden permitir al generador del analizador insertar un código específico a una o más cadenas de receptor y una o más acciones en las funciones del receptor y de acción mencionadas anteriormente.

Después que el generador del analizador ha analizado la definición del analizador y ha insertado un código para el objeto del analizador y los objetos controladores en sus plantillas respectivas, el generador del analizador puede, en algunas modalidades, compilar las plantillas antes mencionadas en código ejecutable. En otras modalidades, las plantillas antes mencionadas pueden convertirse en código de bytes o pueden estar disponibles para su interpretación por un entorno de programación que interpreta instrucciones de código (por ejemplo, Python o Perl) en lugar de ejecutar las instrucciones. Otros tipos de plantillas también pueden usarse junto con las modalidades descritas en la presente descripción.

La Figura 7 ilustra las etapas de un proceso ilustrativo 700, de acuerdo con algunas modalidades de la presente descripción. Como se muestra, el proceso ilustrativo de la Figura 7 incluye las etapas de recibir una definición del analizador (702), generar un analizador (704), y ejecutar el analizador generado (706), de acuerdo con las modalidades descritas en la presente descripción. Cada una de estas etapas se describieron en las secciones anteriores de esta descripción y en consecuencia se llevaron a cabo.

En la descripción anterior, se han descrito las modalidades con referencia a numerosos detalles específicos que pueden variar desde la implementación hasta la implementación. Pueden realizarse ciertas adaptaciones y modificaciones de las modalidades descritas. Por lo tanto, las modalidades descritas anteriormente se consideran ilustrativas y no restrictivas.

Reivindicaciones

1. Un sistema configurado para generar un analizador, el sistema que comprende:  
 un dispositivo de memoria (106) configurado para almacenar un conjunto de instrucciones; y  
 uno o más procesadores (104) configurados para ejecutar el conjunto de instrucciones para realizar las siguientes operaciones:  
 recibir una definición del analizador como una entrada a un generador del analizador en donde la definición del analizador comprende una o más definiciones del controlador, en donde:  
 cada definición del controlador de la una o más definiciones del controlador comprende una definición de acción del controlador que comprende las definiciones de una o más acciones,  
 al menos una definición del controlador de la una o más definiciones del controlador comprende una definición del receptor del controlador que comprende una o más comparaciones de cadena, y  
 al menos una definición del controlador de la una o más definiciones del controlador define un controlador de destino para recibir una salida; y  
 generar el analizador mediante el uso de la definición del analizador, en donde el analizador comprende un proceso de análisis que comprende dos o más controladores en base a la una o más definiciones del controlador, en donde la salida de al menos uno de los dos o más controladores que define el controlador de destino se proporciona como entrada a un controlador predeterminado dentro del canal de procesamiento en base al controlador de destino definido.
2. El sistema de acuerdo con la reivindicación 1, en donde dos o más comparaciones de cadena se asocian a través de una expresión booleana.
3. El sistema de cualquiera de las reivindicaciones anteriores, en donde el uno o más procesadores se configuran para ejecutar instrucciones adicionales para realizar la siguiente operación:  
 almacenar el analizador generado en una caché del analizador.
4. El sistema de acuerdo con la reivindicación 3, en donde el uno o más procesadores se configuran para ejecutar instrucciones adicionales para realizar las siguientes operaciones:  
 recibir una solicitud para ejecutar el analizador generado;  
 recuperar el analizador generado desde la caché del analizador; y  
 ejecutar el analizador generado recuperado desde la caché del analizador, en donde el analizador ejecutado actúa sobre los datos para generar una salida.
5. Un método para generar un analizador, el método que se realiza por uno o más procesadores y que comprende:  
 recibir una definición del analizador como una entrada a un generador del analizador en donde la definición del analizador comprende una o más definiciones del controlador, en donde:  
 cada definición del controlador de la una o más definiciones del controlador comprende una definición de acción del controlador que comprende las definiciones de una o más acciones,  
 al menos una definición del controlador de la una o más definiciones del controlador comprende una definición del receptor del controlador que comprende una o más comparaciones de cadena, y  
 al menos una definición del controlador de la una o más definiciones del controlador define un controlador de destino para recibir una salida; y  
 generar, mediante el uso de uno o más procesadores, un analizador mediante el uso de la definición del analizador, en donde el analizador comprende un proceso de análisis que comprende dos o más controladores en base a la una o más definiciones del controlador, en donde la salida de al menos uno de los dos o más controladores que define el controlador de destino se proporciona como entrada a un controlador predeterminado dentro del canal de procesamiento en base al controlador de destino definido.
6. El método de acuerdo con la reivindicación 5, en donde dos o más comparaciones de cadena se asocian a través de una expresión booleana.
7. El método de acuerdo con la reivindicación 5 o 6, que comprende además almacenar, con uno o más procesadores, el analizador generado en una caché del analizador.
8. El método de la reivindicación 7, que comprende además:  
 recibir una solicitud para ejecutar el analizador generado;  
 recuperar el analizador generado desde la caché del analizador; y  
 ejecutar el analizador generado recuperado desde la caché del analizador, en donde el analizador ejecutado actúa sobre los datos para generar una salida.
9. El método de acuerdo con la reivindicación 7, en donde el almacenamiento comprende además seleccionar una entrada en la que el analizador construido se almacenará dentro de la caché del analizador, y opcionalmente en donde se determina la entrada seleccionada al menos en parte mediante la identificación de la entrada de caché que se usó menos recientemente.

10. Un programa de computadora que comprende las instrucciones legibles por máquina que cuando se ejecutan por un aparato informático que hace que realice el método de cualquiera de las reivindicaciones 5 a 9.

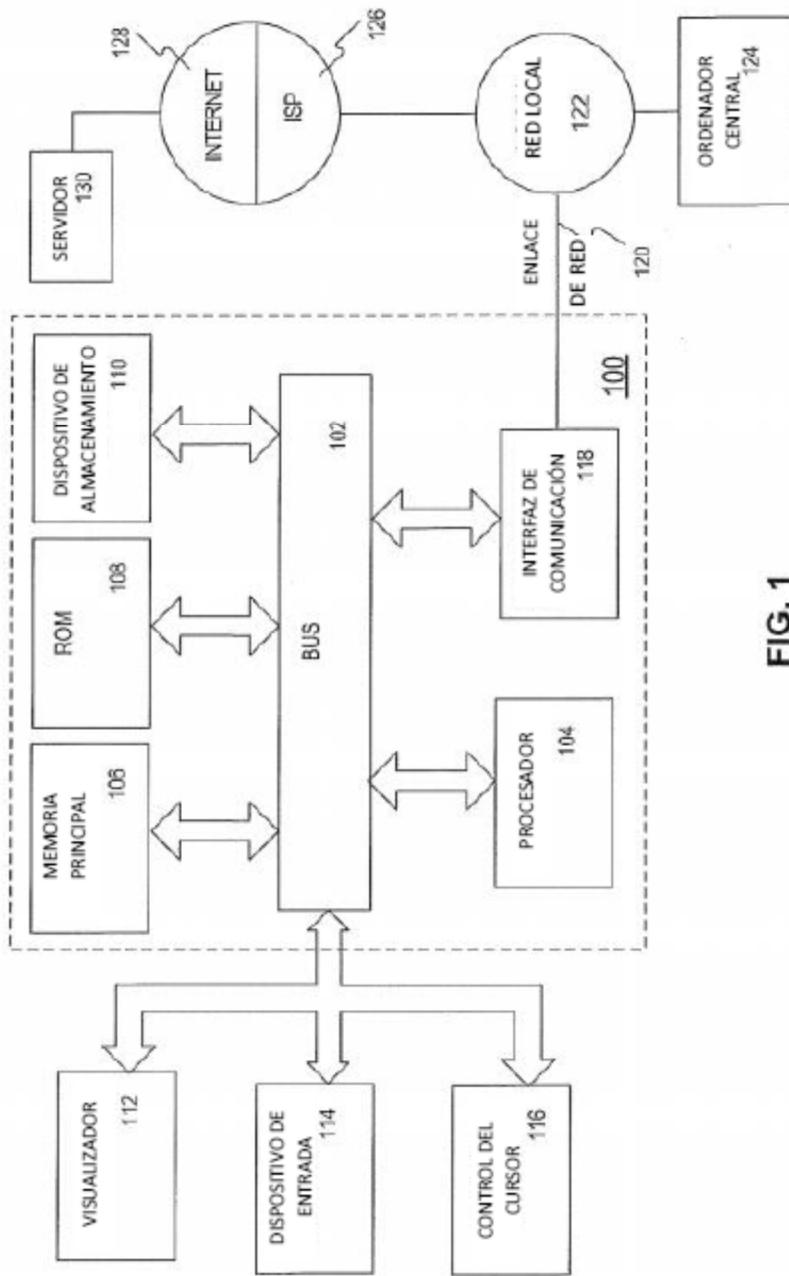


FIG. 1



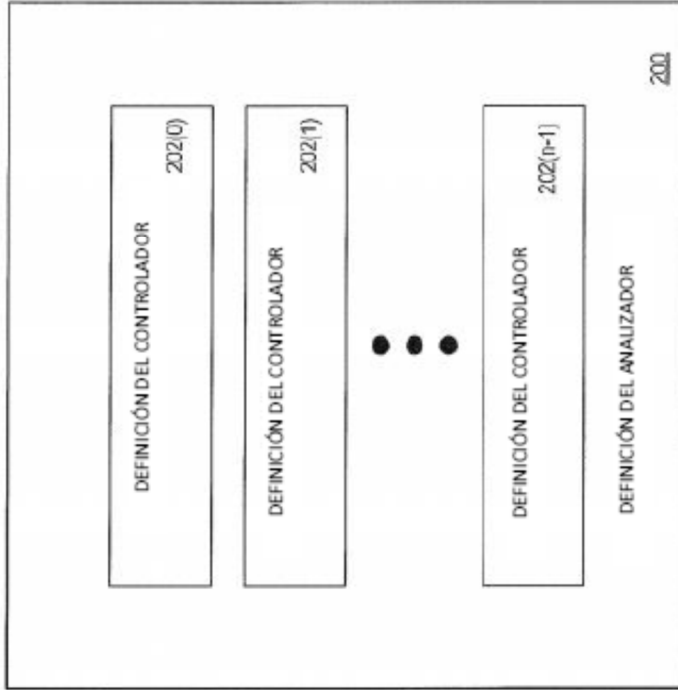


FIG. 2

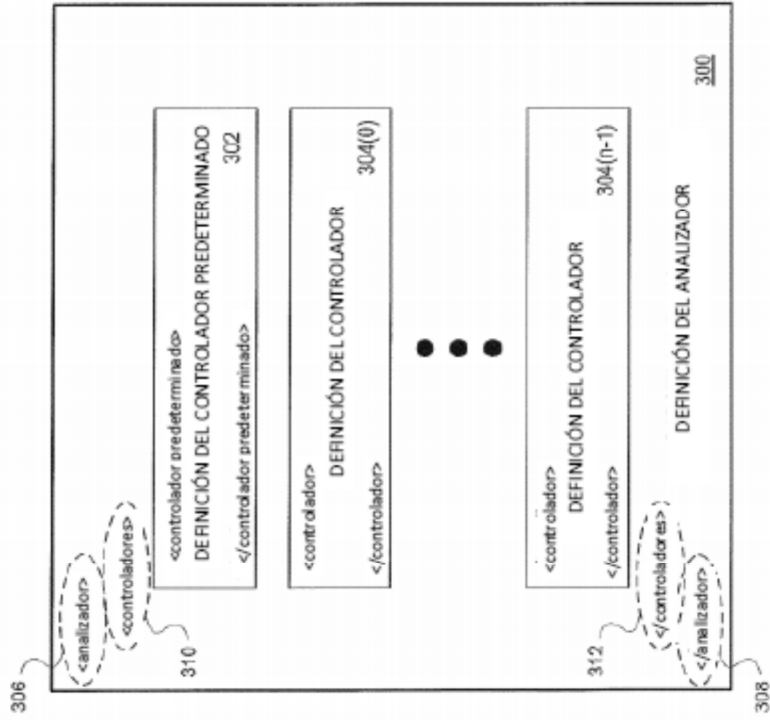


FIG. 3

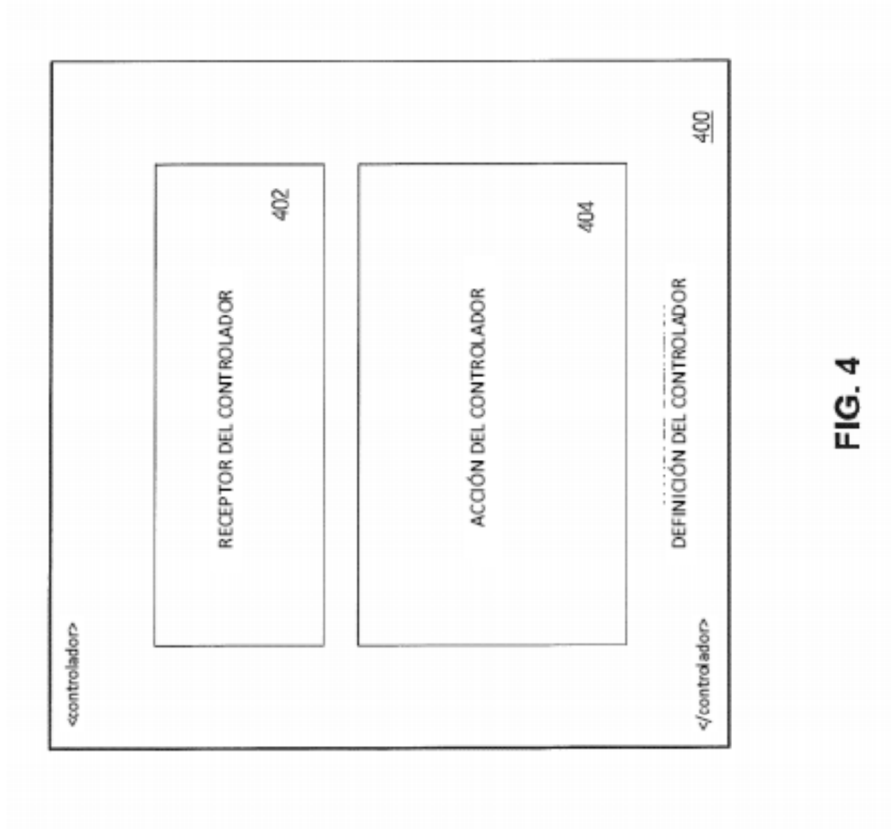


FIG. 4

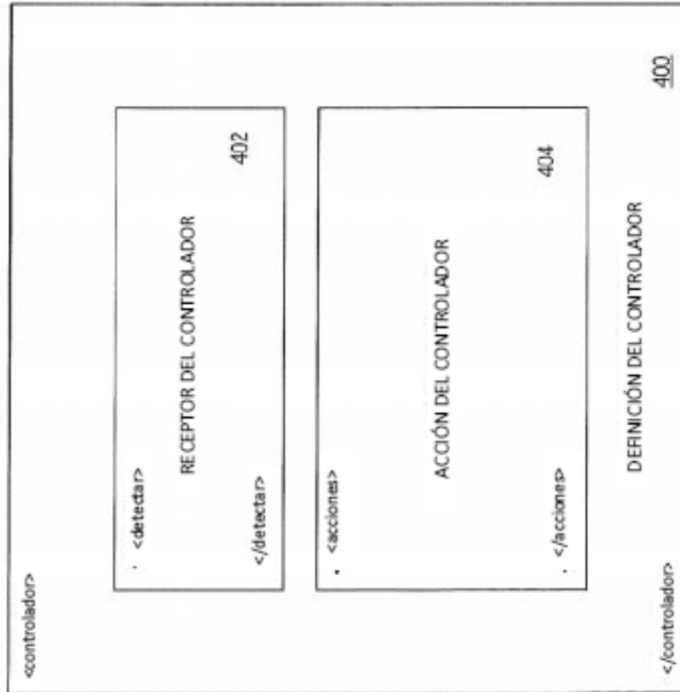


FIG. 5

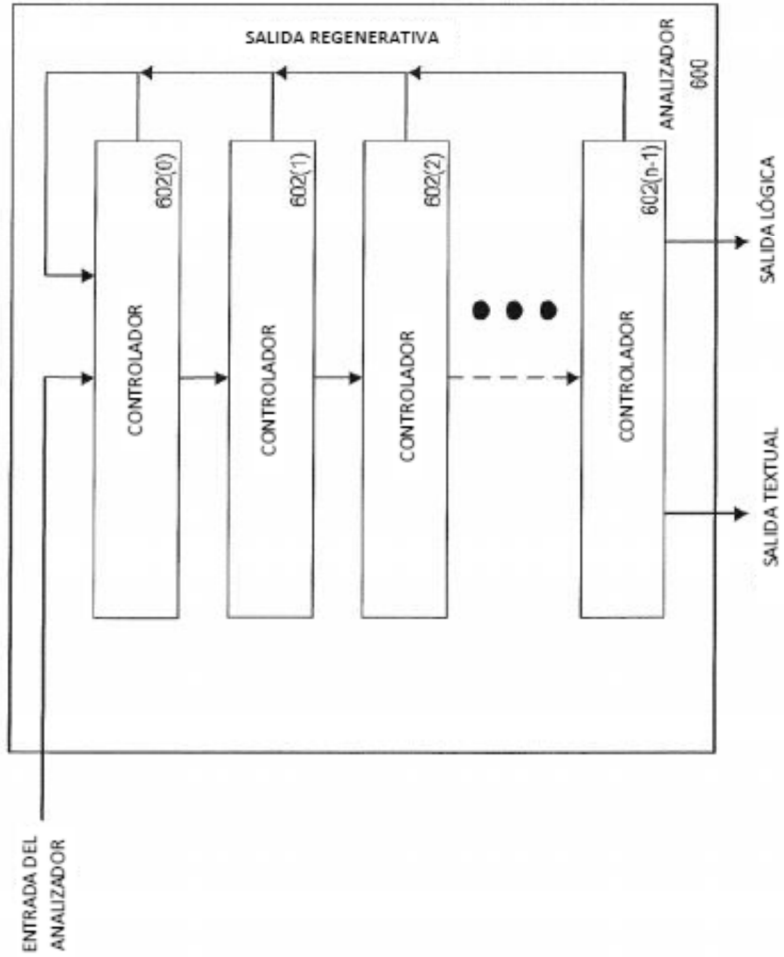
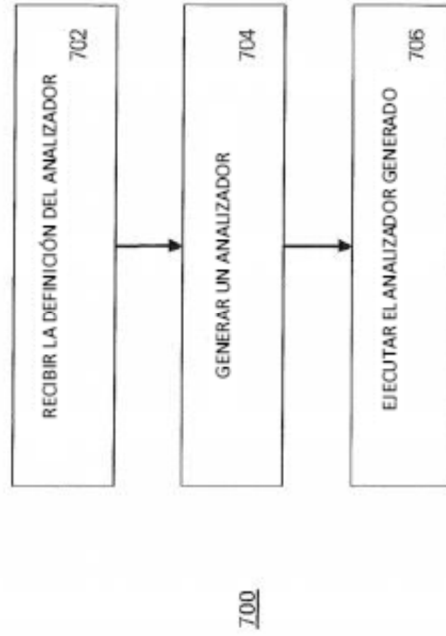


FIG. 6



**FIG. 7**