



- (51) Clasificación Internacional de Patentes:
G06F 9/45 (2006.01)
- (21) Número de la solicitud internacional:
PCT/IB2016/052465
- (22) Fecha de presentación internacional:
29 de abril de 2016 (29.04.2016)
- (25) Idioma de presentación: español
- (26) Idioma de publicación: español
- (30) Datos relativos a la prioridad:
62/154,093 28 de abril de 2015 (28.04.2015) US
- (72) Inventor; e
- (71) Solicitante : **HUEBRA, Nadia Analía** [AR/CO]; Calle 110 No. 7C - 46, Bogotá (CO).
- (74) Mandatario: **OLARTE, Carlos R.**; Carrera 5 No. 34-03, La Merced, Bogotá, 110311 (CO).
- (81) Estados designados (*a menos que se indique otra cosa, para toda clase de protección nacional admisible*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, BR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA,

MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) Estados designados (*a menos que se indique otra cosa, para toda clase de protección regional admisible*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), euroasiática (AM, AZ, BY, KG, KZ, RU, TJ, TM), europea (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declaraciones según la Regla 4.17:

- sobre la identidad del inventor (Regla 4.17(i))

Publicada:

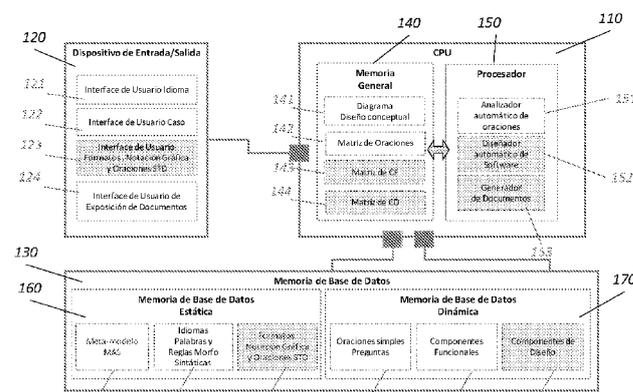
- con informe de búsqueda internacional (Art. 21(3))
- antes de la expiración del plazo para modificar las reivindicaciones y para ser republicada si se reciben modificaciones (Regla 48.2(h))

[Continúa en la página siguiente]

(54) Title: METHOD AND SYSTEM FOR AUTOMATICALLY GENERATING FUNCTIONAL ARCHITECTURE DOCUMENTS AND DOCUMENTS FOR SOFTWARE DESIGN AND ANALYSIS SPECIFICATION

(54) Título : PROCESO Y SISTEMA PARA GENERAR DOCUMENTOS DE ARQUITECTURA FUNCIONAL Y DOCUMENTOS DE ESPECIFICACIÓN DE ANÁLISIS Y DE DISEÑO DE SOFTWARE DE MANERA AUTOMÁTICA

FIG. 1



- 120 Input/output device
- 121 User interface for language
- 122 User interface for case
- 123 User interface for format, graphical notation and STD sentence
- 124 User interface for document display
- 140 General memory
- 141 Conceptual design diagram
- 142 Sentence matrix
- 143 CF matrix
- 144 CD matrix
- 150 Processor
- 151 Automatic sentence analyser
- 152 Automatic software designer
- 153 Document generator
- 130 Database memory
- 161 MAS Metamodel
- 162 Languages, words and morphosyntactic rules
- 163 Formats, graphical notation and STD sentences
- 171 Simple question sentences
- 172 Functional components
- 173 Design components

(57) Abstract: The invention relates to a computer-implemented method and system for automatically generating functional architecture documents, business documents, analysis documents and software design documents from phrases expressed in natural language. The system allows the input of selectable languages and their syntactic and grammatical characteristics. The system receives the description of a case in natural language, in any one of the selectable languages, via an input/output device. The processor automatically analyses the sentences of the description, automatically extracts the functional components derived from its morphosyntactic structure and automatically generates functional architecture documents for industrial application. On the basis of the functional components of the text, the processor automatically generates design components so as to subsequently produce business documents, analysis documents and software design specification documents.

(57) Resumen:

[Continúa en la página siguiente]

WO 2016/174638 A1



— *con información relativa a una petición de restablecimiento del derecho de prioridad sobre una o más reivindicaciones de prioridad; la decisión de la Oficina receptora sobre una petición de restablecimiento*

está en curso y será publicada por separado una vez disponible (Reglas 26bis.3 y 48.2(j))

El presente invento es un sistema y un método implementado por computador para generar automáticamente documentos de Arquitectura Funcional, documentos de negocio, de análisis y de diseño de software, a partir de frases expresadas en lenguaje natural. El sistema permite ingresar idiomas elegibles y sus características sintácticas y gramaticales. A través de un dispositivo de entrada/salida recibe la descripción de un caso en lenguaje natural, en alguno de los idiomas elegibles. El procesador analiza automáticamente las oraciones de la descripción, extrae automáticamente los componentes funcionales derivados de su estructura morfo-sintáctica y genera automáticamente documentos de arquitectura funcional para aplicación industrial. A partir de los componentes funcionales del texto, el procesador genera automáticamente componentes de diseño para luego producir documentos de negocio, análisis y especificación de diseño de software.

PROCESO Y SISTEMA PARA GENERAR DOCUMENTOS DE ARQUITECTURA
FUNCIONAL Y DOCUMENTOS DE ESPECIFICACIÓN DE ANÁLISIS Y DE DISEÑO DE
SOFTWARE DE MANERA AUTOMÁTICA

5

1. Referencia cruzada a solicitudes relacionadas

Esta solicitud reivindica los beneficios de la fecha de presentación de la solicitud de
patente provisional U.S. N° 62/154,093 titulada " Proceso para generar documentos de
10 especificación de análisis y de diseño", que fue presentada el 28 de abril de 2015, por el
mismo inventor de esta solicitud. Que la aplicación provisional se incorpora a la
presente por referencia como si se expusiera en el presente documento.

2. Campo de la invención

15 El presente invento se refiere en general al proceso de modelado del razonamiento
mediante una lógica formal análoga al pensamiento abstracto que el ser humano
desarrolla frente al lenguaje natural, y más particularmente a derivar modelos de
proceso de las especificaciones de casos a partir del lenguaje natural.

20 **3. Descripción del estado del arte**

En el estado del arte se encuentran divulgados dispositivos para generar
automáticamente documentos de diseño de software a partir de requerimientos
expresados en frases que son almacenadas en bases de conocimiento y que a partir de
25 éstas modelan 'Clases' (en el contexto de la Programación Orientada a Objetos) que
posteriormente son visualizadas en forma de diagramas UML. También se encuentran
sistemas y métodos para la formalización del lenguaje natural de tal manera que puedan
ser llevados a lenguajes susceptibles de ser procesados por un computador.

30 La patente U.S. 2013/0097583 divulga un sistema y un método que proporciona una
herramienta de automatización de soporte para guiar/ayudar a un ingeniero de software
a crear un producto de software. El sistema incluye dispositivos de entrada-salida, de

almacenamiento, de procesamiento y de comunicación. El sistema ayuda al ingeniero de software en la identificación y la representación gráfica de los casos de uso derivados de los requerimientos, actores o entidades, sistemas y subsistemas a través de una o más frases que describan los requerimientos y los actores o entidades involucrados, así como

5 de generar un modelo de dominio y un diagrama de clases UML para visualizar el modelo de dominio. El sistema también permite generar casos de uso expandidos, generar un modelo de interacción de objetos, generar los diagramas de secuencia y a crear un diagrama de clases de diseño. Desde el diagrama de clases de diseño, el ingeniero de software puede producir programas de computador de alta calidad. El

10 sistema puede incluir o puede conectarse a un generador de diagramas para la generación automática de diagramas de clases UML (Unified Modeling Language). A diferencia de la presente invención, el input de esta patente son requerimientos de software escritos en lenguaje natural. En nuestro invento son oraciones en lenguaje natural que describen “el caso” de negocio. El input de esta patente es uno de los casos

15 particulares que podría resolver nuestro invento. El parser (técnica de identificación de las palabras) es tradicional, está basado en un diccionario. El parser de nuestra invención es automático y está basado en reglas que no usan diccionario. El presente invento aumenta la precisión en un tiempo de procesamiento mucho menor. En la patente U.S. 2013/0097583, para lograr el diseño completo del software hace falta la

20 participación del ingeniero de software para diseño y desarrollo de código que se apoya en los diagramas generados por el invento de la patente. En la presente invención, el diseño resultante es completo y automático, abarcando el 100% del diseño enunciado en el texto descriptivo. La abstracción del modelo de dominio conceptual en la presente invención es formal, es decir que el diseño siempre responde a las mismas reglas y por

25 lo tanto arroja resultados previsibles. En la patente U.S. 2013/0097583 el modelo conceptual tiene un 20% de composición artesanal y depende del caso.

La patente U.S. 2011/0239183 divulga un método en donde uno o más modelos de

30 proceso se derivan desde modelos de casos de uso del lenguaje natural, por ejemplo, el uso de un procesador y el acceso a un modelo almacenado en una memoria de un caso de uso que proviene de información de un texto en lenguaje natural descrito a partir de

una lista taxativa de palabras predefinidas con significado preasignado; transformando el modelo almacenado en memoria en un modelo de proceso, en una determinada notación de modelado. En el presente invento, el diseño se realiza a nivel de Documento de Arquitectura funcional, a partir del lenguaje natural ilimitado, a partir de los cuales se pueden obtener los diferentes tipos de diagramas de diseño de software.

Todas las industrias se enfrentan al desafío de mejorar los procesos de construcción, cuyo principal factor de error reside en la diferencia que existe entre lo que un solicitante describe (“el caso”) y lo que la industria resuelve (la solución).

10 En la actualidad, alguien que desea describir “el caso” para, a partir de dicha descripción construir una solución, necesita:

- Un analista de la industria que interprete y documente. Esta documentación se realiza a través de notas donde el analista registra lo que el solicitante le cuenta acerca de “el caso”, luego lo analiza y realiza unos diagramas gráficos aplicando alguna de las técnicas conocidas en la industria y cuando los documentos han sido lo suficientemente revisados se derivan a la etapa de diseño de “el caso”. El contenido de estos diagramas depende exclusivamente de la interpretación del especialista y su capacidad de traducción a los mencionados diagramas, propios de cada industria.

20 - un diseñador de la industria que diseñe la solución deseada, partiendo del análisis y documentación referida al entendimiento de “el caso” descripto. El diseñador toma los diagramas de análisis, se reúne con el analista para obtener una explicación que le permita maximizar la correcta comprensión de los diagramas y partir de allí diseña una solución para “el caso”. Este diseño se lleva a cabo creando otros diagramas, que también dependen de la industria. Estos diagramas constituyen nuevas traducciones de la lógica descripta por el solicitante al lenguaje técnico necesario para construir la solución de “el caso”.

Tanto el analista como el diseñador son profesionales que han sido formados con el objetivo de comprender la descripción de “el caso” y traducirla a lenguajes técnicos de análisis y diseño que luego puedan ser interpretados por los constructores de la solución. Es decir que existen varios lenguajes que son utilizados desde que se describe “el caso”

hasta que se inicia la construcción de la solución asociada al mismo (ver **¡Error! No se encuentra el origen de la referencia.**):

- A. Lenguaje de la descripción de “el caso” (rol interviniente: solicitante, quien plantea la necesidad): lenguaje natural
- 5 B. Lenguaje del análisis de “el caso” (rol interviniente: analista de la industria, quien interpreta al solicitante y realiza la traducción a diagramas de análisis): gráficas que utilizan notación estándar de la industria, bocetos artesanales, acompañados por descripciones complementarias expresadas en lenguaje natural con un enfoque orientado al diseño.
- 10 C. Lenguaje de diseño de “el caso” (rol interviniente: diseñador, arquitecto quien interpreta al analista y realiza la traducción a diagramas de diseño): gráficas, planos, diagramas de arquitectura propios de la industria de aplicación, que representan el diseño de la solución.

Quien tiene la necesidad de que se diseñe una solución determinada (el solicitante),
15 corrientemente no es un diseñador experto de la industria específica, por lo tanto, no podría construir la solución por sus propios medios. Él necesita que el analista de la industria y el diseñador de la industria puedan interpretar adecuadamente su requerimiento y reflejarlo en el diseño de solución para el “el caso”. Sucede que esto resulta imposible en la práctica porque el solicitante no entiende el lenguaje de diseño
20 (arquitecto, ingeniero civil, diseñador de libros, diseñador de software, entre otras aplicaciones), con lo cual tampoco puede controlar el diseño resultante para asegurar que se construirá una solución para “el caso” que describió. Aquí radica el mayor porcentaje de desvío que se presenta entre lo descrito (“el caso”) y lo efectivamente construido (la solución) porque normalmente se produce una distorsión en la
25 concepción del significado producto del cambio del lenguaje, como se puede observar en la Figura I. Como consecuencia, el solicitante va recibirá un diseño y un producto resultante (una casa, un software, un mueble, un libro, ...) que no se ajusta perfectamente a su necesidad.

En la industria se realizan esfuerzos permanentes para mejorar la aproximación de las
30 soluciones con las descripciones de necesidad de los solicitantes. Estos esfuerzos están enfocados en la mejora de los métodos actualmente utilizados, mediante capacitación de los especialistas o mejora de herramientas que dichas personas utilizan en sus labores

sin reemplazar su intervención en el proceso artesanal de interpretación y estructuración de las mencionadas descripciones.

Este modo de abordar el problema técnico, necesariamente genera interpretaciones múltiples de las necesidades, porque la traducción que se produce entre el lenguaje utilizado por el solicitante (lenguaje natural) y el lenguaje utilizado por analistas y diseñadores (lenguajes técnicos) implica una transformación durante la cual existe una fuga de conocimiento que nadie puede controlar de manera precisa.

En el presente invento, para describir de manera completa “el caso”, es suficiente con utilizar el lenguaje natural estructurando oraciones. No es necesario traducir de manera artesanal dichas oraciones en diagramas con notaciones gráficas técnicas para poder comprender y transmitir el conocimiento inherente a la descripción, dichos diagramas son generados de manera automática por la herramienta a partir del lenguaje natural.

La diferencia entre lo que existe y el invento, se representa en la siguiente tabla (**¡Error! No se encuentra el origen de la referencia.**).

Etapa	Subetapa	Lo que existe actualmente		El invento	
		Lenguaje	Método	Lenguaje	Método
Descripción (1.1)	Interpretación Intelectual (1.8)	Semántica del Lenguaje natural	Interpretación de la realidad del problema por parte del solicitante.	Lenguaje natural	Interpretación y análisis de la realidad del problema por parte del solicitante.
	Descripción Lenguaje natural (1.10)	Gramática y Sintaxis del Lenguaje natural	Redacción y/o relato del problema por parte del solicitante.	Lenguaje natural	Redacción y/o relato del problema por parte del analista de la industria.
Análisis (1.2)	Interpretación Intelectual (1.11.)	Semántica del Lenguaje natural	Interpretación del análisis del problema por parte del analista de la industria, en base a la descripción previa.		
	Análisis Intelectual (1.13)	Semántica y Semiótica del Lenguaje industrial basado en estándares de análisis de la industria	Análisis por parte del analista de la industria y representación gráfica y/o textual del resultado.	Etapa resuelta automáticamente por la máquina.	

Diseño (1.3)	Creación Intelectual del diseño conceptual (1.15.)	Semántica del Lenguaje natural y semiótica del diseño conceptual.	Creación de un diseño conceptual por parte del diseñador de la industria. Este diseño representa el significado del análisis.	Lenguaje natural	Inferencia directa y automática realizada por la máquina diseñadora conceptual (1.21.) a partir de la Descripción Lenguaje natural (1.10).
	Creación Intelectual del diseño industrial (1.17.)	Semántica y Semiótica del Lenguaje industrial basado en estándares de diseño de la industria.	Creación de un diseño industrial por parte del diseñador de la industria, con base en el diseño conceptual, utilizando los símbolos de estándares definidos por la industria de aplicación.	Lenguaje natural	Inferencia directa y automática realizada por la máquina diseñadora industrial (1.22.) a partir de la Descripción Lenguaje natural (1.10).

Dada una descripción de la realidad para la cual se desea obtener un diseño industrial particular en base a lo que dicha descripción representa, el sistema estructura las descripciones en lenguaje natural garantizando el almacenamiento de la totalidad de los componentes lingüísticos de “el caso” de manera relacionada y coherente desde el punto de vista semántico, y construyendo de forma automática Documentos de Arquitectura Funcional que son útiles en múltiples aplicaciones industriales.

En el presente invento se presenta el uso de los Documentos de Arquitectura Funcional y sus componentes para construir aplicaciones de Diseño de Software.

10

En la presente invención se hará referencia a “el caso” cuando nos encontramos frente a una problemática de la realidad para la cual queremos construir una aplicación de software que permita procesar su lógica a través de un computador. Así entonces, en la industria de software de “el caso” se enfrenta al desafío de mejorar los proyectos de construcción de software, cuyo principal factor de error reside en la diferencia que existe entre lo que el cliente solicita (el problema) y lo que “el caso” resuelve (la solución). En la actualidad, alguien que desea describir “el caso” para, a partir de dicha descripción construir una aplicación de software, necesita:

- un analista de sistemas que lo interprete y documente. Esta documentación se realiza a través de notas de relevamiento donde el analista registra lo que el cliente le cuenta acerca de “el caso”, luego lo analiza y realiza unos

20

- diagramas gráficos aplicando alguna de las técnicas conocidas en la industria (diagramas UML, diagramas de flujo, diagramas de proceso) y cuando los documentos han sido lo suficientemente revisados se derivan a la etapa de diseño de “el caso”. El contenido de estos diagramas depende
- 5 exclusivamente de la interpretación del analista y su capacidad de traducción a los mencionados diagramas.
- un diseñador de software que diseñe dicho software, el cual debe permitir automatizar la captura de datos que se generan a través de “el caso” descripto. El diseñador toma los diagramas de análisis, se reúne con el
- 10 analista para obtener una explicación que le permita maximizar la correcta comprensión de los diagramas y partir de allí diseña “el caso”. Este diseño se lleva a cabo creando otros diagramas tales como diagramas de bases de datos (para almacenar los datos del sistema), diagramas de pantallas (donde se realizarán las cargas de datos, consultas y procesamientos), diagramas de
- 15 lógica funcional de “el caso” (representa la lógica del problema a resolver), diagramas de arquitectura (que definen cómo estará organizado “el caso” internamente para lograr un funcionamiento efectivo, estable y mantenible: cliente servidor, SOA: services oriented architecture, entre otras). Estos diagramas constituyen nuevas traducciones de la lógica descripta por el
- 20 cliente al lenguaje técnico necesario para construir la solución de “el caso”.

Tanto el analista como el diseñador son profesionales que han sido formados con el objetivo de comprender la descripción de “el caso” y traducirla a lenguajes técnicos de análisis y diseño de software que luego puedan ser interpretados por los desarrolladores

25 de la aplicación. La Figura 1 muestra un esquema comparativo entre el estado del arte y el presente invento, respecto del lenguaje que se utiliza en cada una de las etapas de la comprensión y traducción de “el caso” desde el inicio de la descripción del caso hasta el diseño del software. En el estado del arte, se utilizan los siguientes lenguajes:

- 30 A. Lenguaje de la descripción de “el caso” (rol interviniente: usuario, quien plantea la necesidad): lenguaje natural

- 5 B. Lenguaje del análisis de “el caso” (rol interviniente: analista de sistemas, quien interpreta al usuario y realiza la traducción a diagramas de análisis): gráficas de flujo de datos, casos de uso de análisis, acompañadas por descripciones complementarias expresadas en lenguaje natural con un enfoque sistémico orientado al diseño.
- 10 C. Lenguaje de diseño de “el caso” (rol interviniente: diseñador, arquitecto quien interpreta al analista y realiza la traducción a diagramas de diseño): gráficas (generalmente UML en la actualidad) que representan el diseño funcional de la aplicación; diagramas de bases de datos que representan el diseño de la base de datos que almacenará los datos del usuario; diagramas de arquitectura que representan la arquitectura de capas de la aplicación a construir.

15 El usuario que tiene la necesidad de que se le diseñe un software a la medida, corrientemente no es un diseñador de software o ingeniero desarrollador de software, por lo tanto, no podría construirlo por sus propios medios. Él necesita que el analista y el diseñador puedan interpretar adecuadamente sus requerimientos y reflejarlo en el diseño de “el caso” resultante.

20 Sucede que esto resulta imposible en la práctica porque el usuario no entiende el lenguaje de diseño, con lo cual tampoco puede controlar el diseño resultante para asegurar que se construirá “el caso” que pidió. En este punto radica el mayor porcentaje de error que se presenta entre lo solicitado (el problema) y lo efectivamente construido (la solución). Es decir, como consecuencia, el usuario o cliente va a recibir un diseño y un producto resultante (software) que no se necesariamente se ajusta perfectamente a sus necesidades, modificaciones o cambios posteriores (que en ocasiones pueden demorar más o ser más complejos y también más costosos que lo previsto inicialmente).

30 En la industria se está realizando un esfuerzo importante para aproximarse a solucionar este problema con CASE Tools (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora), sin embargo, estas herramientas no trabajan con lenguaje natural con lo cual presentan varias de las deficiencias descritas previamente.

Este modo de abordar el problema técnico, necesariamente genera interpretaciones múltiples del problema porque la traducción que se produce entre el lenguaje utilizado por el usuario o cliente conocedor del proceso (lenguaje natural) y el lenguaje utilizado por analistas y diseñadores (lenguajes técnicos) implica una transformación durante la cual existe una fuga de conocimiento que nadie puede controlar de manera precisa.

El presente invento hace uso del lenguaje natural como único lenguaje para la descripción, el análisis y el diseño de la solución de software minimizando así, la pérdida de conocimiento que implicaría una distorsión en el resultado.

10

A diferencia del estado del arte, el presente invento permite describir de manera completa “el caso”, utilizando el lenguaje natural estructurando oraciones simples. A diferencia del estado del arte, el presente invento no necesita traducir dichas oraciones en diagramas con notaciones gráficas técnicas para poder comprender y transmitir el conocimiento del proceso, sino que dichos diagramas son generados de manera automática por el presente invento a partir del lenguaje natural, utilizando un único modelo determinístico que permite la identificación de los componentes conceptuales del lenguaje, mediante la aplicación de reglas analíticas que implementan la lógica de funcionamiento de dicho modelo

20

La descripción en oraciones simples, estructuradas a partir de una jerarquía de conceptos predefinida, garantiza la completitud de la descripción y permite inferir diseños de software de gran calidad, de manera automática.

25 Con el ánimo de facilitar la comprensión de la presente invención y que sea fácil distinguir el presente invento de lo disponible en el estado del arte, a continuación, se presenta una tabla comparativa entre el estado del arte y el presente invento:

30

Etapa	Estado del arte		El presente invento	
	Lenguaje	Método	Lenguaje	Método
Descripción del caso	Lenguaje natural	Redacción libre a cargo del analista	Lenguaje natural	Redacción inducida a lograr completitud y calidad a través de preguntas
Análisis del caso	Análisis de requisitos Diagramas de Flujo Diagramas UML (casos de uso de análisis) Diagramas de Proceso	Interpretación y confección artesanal a cargo del criterio del analista	Lenguaje natural	Redacción inducida a través de pautas precisas para identificación de requisitos
Diseño del software	Diagramas de Bases de Datos Diagramas UML (diagramas de estados, diagramas de clases, casos de uso de realización) Diagramas de Pantallas Diagramas de arquitectura	Interpretación y confección libre a cargo del criterio del diseñador	Lenguaje natural	Inferencia directa y automática de los diseños de “el caso” a partir de los componentes de las oraciones de los pasos

El estado del arte actual determina que el software se diseña mediante un proceso artesanal o semi-artesanal que por sus características presenta los siguientes problemas: Ambigüedad en la interpretación de requerimientos del sistema, alto costo derivado de errores de diseño, dificultades para escalar y evolucionar los diseños y documentación desactualizada entre otros. El presente invento supera estas dificultades puesto que se basa en un proceso totalmente automatizado e industrializable, logrando precisión conceptual y produciendo diseños evolutivos escalables, documentación actualizada automáticamente y disminuyendo de manera significativa los costos del diseño de software.

4. Glosario

Oración: el constituyente sintáctico más pequeño posible, capaz de realizar un enunciado o expresar el contenido de una proposición lógica.

5

Oración simple: aquella que contiene un solo verbo.

Oración simple completa: desde el punto de vista del invento es aquella que responde de manera explícita a todas las preguntas enunciadas.

10

Lenguaje natural: Conocemos por lenguaje natural a la lengua o idioma hablado o escrito por humanos para propósitos generales de comunicación.

Requerimiento funcional: Como se define en la ingeniería de requisitos, los requisitos funcionales establecen los comportamientos del sistema. Un requerimiento funcional Global agrupa otros requerimientos funcionales de mayor detalle relacionados con el mismo comportamiento del sistema.

Diagrama de clases: estos diagramas de clases muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras. También muestran las clases, junto con sus métodos y atributos, así como las relaciones entre ellas.

Universo de “el caso”: Es el conjunto de definiciones y conceptos que rodean a “el caso”.

25

Sustantivos agrupados: Sustantivos que se conforman con más de una palabra. Por ejemplo: lista de precios, número de documento.

+: en el presente documento significa concatenar, unir o adicionar texto.

30

Idioma elegible: dado que se pueden utilizar diferentes idiomas para describir “el caso”, es posible declararlos a cada uno de ellos con sus respectivas estructuras sintácticas y gramaticales.

- 5 **Concatenar:** dadas una o más palabras, concatenar implica unir las en una sola frase, separadas por espacios. Ejemplo: “la”, “casa”. Concatenadas ambas palabras resulta: “la casa”.

Regla Base: en la presente invención, regla base es una estructura de comportamiento que se define por pregunta, y tiene como objetivo describir la secuencia morfosintáctica de las palabras que componen la respuesta a dicha pregunta.

10

OD: significa objeto directo. En la presente invención, se refiere a las palabras de tipo sustantivo que se encuentran en la pregunta Qué y son consecutivas a un verbo transitivo.

15

5. Breve descripción de los gráficos

- La FIG. I. ilustra el proceso de transformación del lenguaje natural del presente invento.
- 20 La FIG. II. Ilustra las etapas del proceso del presente invento, para la obtención de Documentos de Arquitectura Funcional.
- La FIG. 1. ilustra el sistema del presente invento.
- La FIG. 2. ilustra las etapas del proceso del presente invento, para la obtención de Documentos de Diseño de Software.
- 25 La FIG. 2A. ilustra la etapa A
- La FIG. 2A1. ilustra el meta-modelo MAS
- La FIG. 2A2. ilustra una Regla Morfo-Sintáctica
- La FIG. 2A3. ilustra un caso de aplicación del Meta-Modelo MAS
- La FIG. 2B. ilustra la etapa B
- 30 La FIG. 2C. ilustra la etapa C
- La FIG. 2D. ilustra la etapa D
- La FIG. 2E. ilustra la etapa E

- La FIG. 2E1. Ilustra la lógica funcional del Procesador 150, configurado como Analizador Automático de Oraciones 151
- La FIG. 2F. ilustra la etapa F
- La FIG. 2F1. ilustra un diagrama de Arquitectura Funcional
- 5 La FIG. 2G. ilustra la etapa G
- La FIG. 2G1. Ilustra la lógica funcional del Procesador 150, configurado como Diseñador Automático de Software 152
- La FIG. 2H. ilustra la etapa H
- La FIG. 2I. ilustra la etapa I
- 10 La FIG. 2I1. Ilustra un diagrama de diseño de software
- La FIG. 2I2. Ilustra la estructura de un documento de negocio
- La FIG. 2I3. Ilustra la estructura de un documento de análisis
- La FIG. 3. Ilustra el ejemplo de diseño de clases

6. Descripción detallada de la invención

15

El presente invento proporciona una solución a los problemas antes citados, mediante un método implementado por computador y un dispositivo visual para generar documentos de especificación de diseño de software a partir de requerimientos expresados en frases en lenguaje natural para describir un caso, que son almacenadas en

20 bases de conocimiento y que a partir de éstas permiten modelar componentes de diseño que posteriormente son transformados y visualizados en forma de diagramas de clases para diseño de software. El dispositivo es adaptable para interactuar con un procesador.

Inicialmente, como se describe en la Figura 2, el proceso inicia con la descripción de “el

25 caso” en lenguaje natural. En el presente invento, para poder realizar dicha descripción, es necesario describir previamente en qué idioma se realizará dicha descripción y cuáles son sus características gramaticales y sintácticas, tales como la secuencia de adjetivación que describe si el adjetivo se localiza antes o después del sustantivo y la lista de palabras que el idioma reconoce como artículos, entre otras.

30

Para describir “el caso”, en cualquier idioma que posea una estructura sintáctica y reglas gramaticales definidas, se utiliza un sistema como el descrito en la Figura 1, para

ingresar la descripción de “el caso” en lenguaje natural a través de un Dispositivo de Entrada/Salida **120**. Los textos ingresados se transfieren a la memoria general para luego ser procesados por el Procesador **150**. El procesador se configura para analizar, diseñar y generar documentos de manera automática. Los resultados de cada función de
5 procesamiento se almacenan en la memoria de Base de Datos **130**.

A continuación, se describen los componentes del sistema **100** de la Figura 1:

1. Dispositivo de Entrada/Salida **120**: es el dispositivo mediante el cual se ingresan los textos en lenguaje natural. Permite a un procesador presentar en medios de
10 visualización (pantallas, proyectores, televisores, impresoras, monitores, dispositivos móviles, entre otros) estructuras donde el usuario puede ingresar los datos de la descripción de “el caso” y visualizar los documentos obtenidos, utilizando las siguientes configuraciones:
 - a. Interface de Usuario Idioma **121**: es una estructura visual que permite al
15 usuario elegir el idioma y cargar sus características gramaticales y sintácticas que se almacenarán en la Memoria de Base de Datos **130**.
 - b. Interface de Usuario Caso **122**: es una estructura visual que permite al usuario cargar las oraciones simples que describen “el caso” que serán almacenadas en la Memoria de Base de Datos **130**. Esta estructura
20 también permite al usuario interactuar con las funciones del Procesador **150** configurado como Analizador Automático de Oraciones **151** y como Diseñador Automático de Software **152**.
 - c. Interface de Usuario Formatos, Notación Gráfica y Oraciones STD **123**:
25 en una modalidad de la invención se presenta esta estructura visual que permite al usuario agregar o modificar los formatos, notaciones gráficas y oraciones STD disponibles en la Memoria de Base de Datos Estática **160**, para la confección de documentos de negocio, análisis y diseño que el Procesador **150** construye configurado como Generador Automático de Documentos **153**.
 - d. Interface de Usuario de Exposición de Documentos **124**: es una
30 estructura que permite al usuario acceder a los documentos creados por

el Procesador **150** configurado como Generador Automático de Documentos **153**.

2. CPU **110**: es el dispositivo de procesamiento del sistema 100. Este dispositivo está estructurado para realizar todas las funciones de procesamiento de lenguaje natural para la obtención automática de diseños y contiene la memoria general que permite el intercambio entre dichas funciones y el resto de los componentes del sistema.
- 5
- a. Memoria General **140**: es un almacenamiento volátil que se utiliza como dispositivo de intercambio entre el dispositivo de entrada/salida, la memoria de base de datos y el procesador. Realiza las siguientes funciones según su configuración:
- 10
- i. Diagramas **141**: es la configuración de la Memoria General que posibilita el procesamiento de los documentos mediante el Procesador **150** configurado como Generador Automático de Documentos **153**, haciendo uso de los Formatos, notaciones gráficas y oraciones STD **163** almacenadas en la Memoria de Base de Datos estática **160**.
- 15
- ii. Matriz de Oraciones **142**: es la configuración de la Memoria General que posibilita el procesamiento las Oraciones simples mediante el Procesador **150** configurado como Analizador Automático de Oraciones **151**, haciendo uso del Meta-Modelo MAS **161**, los Idiomas, Palabras y Reglas Morfo Sintácticas **162** almacenados en la Memoria de Base de Datos estática **160**.
- 20
- iii. Matriz de CF **143**: es la configuración de la Memoria General que posibilita el procesamiento los componentes funcionales mediante el Procesador **150** configurado como Analizador Automático de Oraciones **151**, haciendo uso del Meta-Modelo MAS **161**, los Idiomas, Palabras y Reglas Morfo Sintácticas **162** almacenados en la Memoria de Base de Datos estática **160**.
- 25
- iv. Matriz de CD **144**: es la configuración de la Memoria General que posibilita el procesamiento los componentes de diseño mediante el Procesador **150** configurado como Diseñador
- 30

Automático de Software **152**, haciendo uso del Meta-Modelo MAS **161**, los Idiomas, Palabras y Reglas Morfo Sintácticas **162** almacenados en la Memoria de Base de Datos estática **160**.

- 5 b. Procesador **150**: es el dispositivo en el cual se llevan a cabo las tareas de procesamiento e intercambio. Realiza las siguientes funciones según su configuración:
- 10 i. Analizador automático de oraciones **151**: es la configuración del Procesador que se encarga principalmente de generar automáticamente los componentes funcionales a partir de las oraciones simples y enviarlos a la Memoria de Base de Datos Dinámica **170** para su almacenamiento.
- 15 ii. Diseñador automático de Software **152**: es la configuración del Procesador que se encarga principalmente de generar automáticamente los componentes de diseño a partir de las oraciones simples y enviarlos a la Memoria de Base de Datos Dinámica **170** para su almacenamiento.
- 20 iii. Generador de Documentos **153**: es la configuración del Procesador que se encarga principalmente de generar automáticamente los documentos de análisis y diseño a partir de los componentes funcionales y componentes de diseño, y luego enviarlos a la Memoria de Base de Datos Dinámica **170** para su almacenamiento.
- 25 3. Memoria de Base de Datos **130**: es una memoria permanente que alberga los datos cargados por el usuario y generados por el Procesador **150** en sus diferentes configuraciones. Esta memoria contiene dos configuraciones de almacenamiento, una configuración estática y una configuración dinámica. La configuración estática almacena aquellos datos fijos necesarios que se cargan por única vez para el procesamiento que no son propios de “el caso”. La configuración dinámica almacena los datos propios de “el caso” que se cargan
- 30 a. Memoria de Base de Datos Estática **160**:

- 5
- 10
- 15
- 20
- 25
- 30
- i. Meta-Modelo MAS **161**: configuración de la Memoria de Base de Datos **130** que contiene las reglas de una lógica formal de alto nivel de abstracción que da origen a las Reglas Morfo-Sintácticas que se almacenan en la estructura Idiomas, Palabras y Reglas Morfo Sintácticas **162** de la Memoria de Base de Datos **130**.
 - ii. Idiomas, Palabras y Reglas Morfo-Sintácticas **162**: configuración de la Memoria de Base de Datos **130** que contiene las características de los idiomas elegibles, las palabras especiales correspondientes a cada idioma elegible y sus Reglas Morfo-Sintácticas.
 - iii. Formatos, Notación Gráfica y Oraciones STD **163**: configuración de la Memoria de Base de Datos **130** que contiene los Formatos que aplican a los documentos, la notación gráfica que aplica a los diagramas y las Oraciones STD necesarias para la confección de documentos de análisis.
- b. Memoria de Base de Datos Dinámica **170**
- i. Oraciones simples Preguntas **171**: configuración de la Memoria de Base de Datos **130** que contiene las Oraciones simples cargadas por el usuario y las respuestas a las preguntas Cuándo, Quién, Qué, Cómo y Dónde relacionadas con cada Oración simple.
 - ii. Componentes Funcionales **172**: configuración de la Memoria de Base de Datos **130** que contiene los Componentes Funcionales generados automáticamente por el Procesador **150** configurado como Analizador Automático de Oraciones **151**.
 - iii. Componentes de Diseño **173**: configuración de la Memoria de Base de Datos **130** que contiene los Componentes de Diseño generados automáticamente por el Procesador **150** configurado como Diseñador Automático de Software **152**.

En uso del sistema descrito, el presente invento realiza la secuencia de etapas que se describen en la Figura 2, y se describen a continuación:

- Etapa A. Cargar a través del Dispositivo de Entrada/Salida **120** el Meta-Modelo MAS en la estructura lógica del meta-modelo MAS **161** de la Memoria de Base de Datos estática **160**.
- 5
- Etapa B. Ingresar a través de la Interface de Usuario Idioma **121**, del dispositivo de visualización **120** los idiomas elegibles, las características sintácticas y gramaticales de cada uno de los idiomas elegibles y almacenar estos datos en la estructura lógica de Idiomas, Palabras y Reglas Morfo Sintácticas **162** de la memoria de base de datos
- 10 estática **160**;
- Etapa C. Ingresar a través de la Interface de Usuario Caso **122**, del dispositivo de visualización **120** la descripción de “el caso” como texto en lenguaje natural, en uno de los escogidos de la lista de idiomas elegibles de la Etapa B. Identificar los pasos de “el caso” y luego almacenarlos en la estructura lógica de Oraciones Simples y Preguntas
- 15 **171** de la memoria de base de datos dinámica **170**;
- Etapa D. Identificar las oraciones simples de “el caso” correspondientes a los pasos de la Etapa C y cargarlas a través de la Interface de Usuario Caso **122**, del dispositivo de visualización **120** para luego almacenarlas en la estructura lógica de Oraciones Simples y Preguntas **171** de la memoria de base de datos dinámica **170**.
- 20
- Etapa E. Identificar automáticamente los componentes funcionales (se debe entender que un componente funcional corresponde a cada palabra de la oración extraída y caracterizada automáticamente por el Procesador **150**) de las oraciones simples de la Etapa D, con un Procesador **150** configurado como Analizador automático de Oraciones **151**, que opera en función de la estructura lógica residente en el meta-
- 25 modelo MAS **161** cargado en la Etapa A. El Procesador **150** va almacenando temporalmente dichos componentes funcionales en la estructura lógica Matriz de CF **143** de la Memoria General **140**. Estos componentes finalmente son almacenados en la estructura lógica de Componentes Funcionales **172** de la memoria de base de datos dinámica **170**;
- 30
- Etapa F. Identificar automáticamente los componentes de diseño de “el caso” a partir de los componentes funcionales de la Etapa E. Se debe entender que un componente de diseño se refiere a cada modelo de diseño de software que se

corresponde con un componente funcional creado automáticamente por el Procesador **150**. Los componentes de diseño son creados con un Procesador **150** configurado como Diseñador automático de Software **152**, que en una modalidad de la invención, opera en función de la estructura lógica residente en el meta-modelo MAS **161** cargado en la

5 Etapa A. El Procesador **150** va almacenando temporalmente dichos componentes de diseño en la estructura lógica Matriz de CD **144** de la Memoria General **140**. Estos componentes son finalmente almacenados en la estructura lógica de Componentes de Diseño **173** de la memoria de base de datos dinámica **170**;

Etapa G. Ingresar a través de la Interface de Usuario Formatos, Notación Gráfica y

10 Oraciones STD **123**, del dispositivo de visualización **120** los formatos de salida para documentos de negocio, análisis y diseño, los parámetros de oraciones estándares para requerimientos y la notación gráfica para diagramas de diseño y almacenarlos en la estructura lógica de Formatos, Notación Gráfica y Oraciones STD **163** de la memoria de base de datos estática **160**;

15 Etapa H. Generar automáticamente los documentos de negocio (se entiende por documento de negocio una descripción paso a paso, en lenguaje natural, del problema a resolver) a partir de las oraciones simples de la Etapa D, los documentos de análisis (se entiende por documento de análisis la descripción de requerimientos funcionales del software a construir) a partir de los componentes funcionales de la Etapa E y los

20 documentos de diseño (se entiende por documento de diseño el que contiene las indicaciones de cómo construir el software) a partir de los componentes de diseño de la Etapa F, con un Procesador **160** configurado como Generador de Documentos **163**, utilizando los Formatos, Notaciones gráficas y Oraciones STD definidas en la Etapa G;

Etapa I. Generar automáticamente los documentos de Arquitectura Funcional (se

25 entiende por documento de arquitectura funcional un diagrama gráfico que estructura el significado del texto en lenguaje natural de manera resumida representando los conceptos esenciales del significado) a partir de los componentes funcionales de la Etapa E con un Procesador **160** configurado como Generador de Documentos **163**, utilizando la notación gráfica definida específicamente para este tipo de Documentos.

30 A continuación, se describen en detalle las etapas del proceso:

Etapa A. Cargar el meta-modelo MAS

En esta etapa, se describe detalladamente el Meta-Modelo MAS y sus principios axiomáticos. Este Meta-Modelo se utiliza para establecer el comportamiento del Procesador **150** en su configuración de Analizador Automático de Oraciones **151**.

- 5 La Figura 2A1 muestra un modelo de clases, según el paradigma de Orientación a Objetos OO, que representa las reglas que este Meta-Modelo define para modelar la lógica de interpretación del lenguaje natural. Estas reglas se utilizan como base de las Reglas Morfo-Sintácticas que el Procesador **150**, configurado como Analizador Automático de Oraciones **151**, utiliza para generar los componentes funcionales en la
- 10 Etapa D.

En esta etapa, haciendo referencia a la Figura 2A se crea el Meta-Modelo MAS mediante las siguientes subetapas:

Subetapa A1. Almacenar Meta-Modelo MAS

- 15 En esta subetapa se carga en la Memoria de Base de datos estática **161** la lógica del Meta-Modelo MAS.

Esta lógica define que toda palabra perteneciente a una descripción en lenguaje natural, es una Palabra **161-1**. Cuando la palabra es un verbo transitivo, se definen dos palabras nuevas de forma automática, una Palabra Factor **161-2** cuyo nombre es el verbo transitivo más el sufijo “OR” y una Palabra Factible **161-3** cuyo nombre es el verbo transitivo más el sufijo “BLE”. El verbo transitivo es una Palabra Factibilidad **161-4** que tiene una relación **161-5** con la Palabra Factor **161-2** y otra relación **161-6** con la Palabra Factible **161-3**.

20

Las Palabras **161-1** se relacionan entre ellas de tal modo que una palabra que no es verbo transitivo puede establecer una relación de herencia **161-9** con una Palabra Factor **161-2** y/o una relación de herencia **161-8** con una Palabra Factible **161-3**.

25

Cada Palabra **161-1** puede definir una relación de herencia **161-0** con otra Palabra **161-1**. Cada Palabra **161-1** puede definir una relación de asociación **161-7** con otra Palabra **161-1**.

- 30 El uso de las opciones que presenta el Meta-Modelo MAS depende de la morfosintaxis de la oración que se esté procesando en la Etapa D.

De este modo, todas las palabras de una descripción, en conjunto, pueden ser representadas por este meta-modelo, que define una regla axiomática para el lenguaje, dado que se cumple en todos los casos y constituye la base para modelar cualquier regla Morfo-Sintáctica de cualquier idioma.

5

Subetapa A2. Almacenar lógica MAS para reglas Morfo-Sintácticas

En esta subetapa se carga en la Memoria de Base de datos estática **161** la lógica del Meta-Modelo MAS que define un modo de caracterizar las palabras del lenguaje natural para determinar su significado en diferentes contextos de aplicación funcional. Las estructuras que tratan la forma y la sintaxis del lenguaje natural, se llaman estructuras morfo-sintácticas.

En la presente invención se considera regla Morfo-Sintáctica a un conjunto de pautas para dar tratamiento al lenguaje natural. Una regla Morfo-Sintáctica está compuesta por las pautas de tratamiento del texto que definen: a) la lógica de preguntas: define a qué pregunta (Cuándo, Quién, Qué, Cómo, Dónde) pertenece la palabra dentro de una oración simple, b) la lógica de tipo de paso: define qué tipo de verbo compone la pregunta Qué (FX,VC, CB, R, F) y c) la lógica de palabras: define de qué tipo de palabra se trata. El tipo de palabra se determina a través de su caracterización gramatical verbo (VERB), sustantivo (SUST), preposición (PREP), artículo (ART), conjunción (CONJ), adverbio (ADV).

Para cada regla Morfo-Sintáctica se define un identificador. El identificador de una regla Morfo-Sintáctica se conforma con la concatenación de los tipos de palabras que componen la oración, exceptuando los tipos de palabras Factor, Factible y Factibilidad. Puede ser, por ejemplo, ART-SUST-VERB-ART-PREP-SUST, para una oración que está formada por una secuencia de palabras de tipo artículo, sustantivo, verbo, artículo, preposición, sustantivo. Estos tipos de palabra pueden ser tratados en general, o referirse a palabras específicas de su tipo. Por ejemplo, en la Regla VERB(es)-PREP-SUST se indica que la oración a la que aplica esta regla contiene un verbo, y se trata de la palabra específica “es” dentro del tipo de palabra VERB, seguido de una preposición cualquiera

PREP y seguido de un sustantivo SUST cualquiera. Esta regla aplicaría a las siguientes oraciones: es un mono, es una luna.

Es decir, que a toda oración simple se le aplica una Regla Morfo-Sintáctica que define su estructura en función de los tipos de palabras que la componen.

Dada una oración simple, antes de componer el identificador de Regla Morfo-Sintáctica que le corresponde, el Procesador **150** configurado como Analizador Automático de Oraciones **151** excluye de la oración, las palabras especiales cargadas en la Subetapa B2, artículos, conjunciones y preposiciones. Luego busca en la Memoria de Base de Datos estática **160** las reglas Morfo-Sintácticas residentes en la configuración Idiomas Palabras y Reglas Morfo-Sintácticas **162**. Una vez localizada la regla con identificador coincidente, si existe más de una regla con el mismo identificador coincidente, busca dentro de ese conjunto la coincidencia de palabras específicas del tipo de palabra que fueron excluidas para la composición del identificador de regla de la oración. Si existe dicha coincidencia, asigna la regla que presenta la coincidencia con la o las palabras específicas excluidas. Si no existe, le asigna la regla general.

El Meta-Modelo MAS define las siguientes reglas base a partir de las cuales se derivan reglas específicas para diferentes casos.

Reglas Base

Asociadas a la pregunta Qué, existen dos reglas base, con sus respectivas especialidades:

- 25 - Regla VERB-SUST:
 - Caso general; Regla VERB-SUST **161-12**: permite modelar oraciones que tienen la estructura verbo, sustantivo, sustantivo.
 - Caso específico; VERB(es)-SUST **161-10**: permite modelar oraciones que tienen la estructura verbo, sustantivo, cuando el verbo es el verbo “ser”. Este verbo es un verbo especial desde el punto de vista de la aplicación del Meta-Modelo MAS.
- 30 - Regla VERB-SUST-PREP-SUST:

- Caso general: Regla VERB-SUST-PREP-SUST **161-11**: permite modelar oraciones que tienen la estructura verbo, sustantivo, preposición, sustantivo.
- 5 ○ Caso específico: Regla VERB(responsabiliza)-SUST-PREP-SUST **161-13**: permite modelar oraciones que tienen la estructura verbo, sustantivo, cuando el verbo es el verbo “responsabilizar”. Este verbo es un verbo especial, desde el punto de vista de la aplicación del Meta-Modelo MAS.
- 10 ○ Caso específico: Regla VERB(relaciona)-SUST-PREP-SUST **161-14**: permite modelar oraciones que tienen la estructura verbo, sustantivo, cuando el verbo es el verbo “relacionar”. Este verbo es un verbo especial, desde el punto de vista de la aplicación del Meta-Modelo MAS.

Asimismo, existen una regla base para modelar el Quién de la oración simple:

- Regla SUST: está presente en todas las oraciones simples y permite modelar el comportamiento del sustantivo principal que hace referencia al sujeto.
- 15 Para las preguntas Cuándo, Cómo, Dónde, que no son de completitud obligatoria, se identifican los tipos de palabras que aparecen en la respuesta y se generan los identificadores de regla necesarios.

20 En una modalidad de la invención, el usuario puede agregar reglas base y reglas Morfo-Sintácticas, siempre dependientes de la lógica del Meta-Modelo MAS.

Dada una oración simple en lenguaje natural, y una vez localizada la regla Morfo-Sintáctica, el Procesador **150** configurado como Analizador Automático de Oraciones **151**, genera los componentes funcionales aplicando las pautas de tratamiento del texto definidas por dicha regla: a) una lógica de preguntas; b) la lógica de tipo de paso y c) la lógica de palabras, 25 como se observa en la Figura 2A2.

(a) Lógica de Preguntas: determina la pregunta a la cual deben pertenecer las palabras a las que se aplique la regla

30 Se debe considerar que una descripción es un conjunto de oraciones simples que describen un proceso. A cada paso del proceso le corresponde una oración simple que se deriva de la respuesta a las preguntas: Quién, Qué, Dónde, Cómo, Cuándo.

(b) *Lógica Tipo de Paso: determina cuál es el tipo de paso al cual deben pertenecer las palabras a las que aplique la regla*

Cada paso, se puede clasificar mediante un Tipo de Paso. El Tipo de Paso está determinado en una regla, por la palabra específica asociada al Verbo de la pregunta
5 Qué. Toda Regla Morfo-Sintáctica está basada en el Meta-Modelo MAS, y dependiendo del verbo específico vinculado a la pregunta Qué puede ser clasificada en los siguientes tipos: Validación / Control; Consulta /Búsqueda; Cálculo; Relación; Funcional; Notificación/Alerta.

10

Paso VC: (Paso que determina acciones de validación/control)

Son pasos donde el verbo determina acciones de validación o control, tales como: validar, controlar, restringir y otros similares. En este caso, el verbo es transitivo y el sustantivo que le sigue, siempre está acompañado de una oración que representará para
15 el sistema una lógica matemática.

A este tipo de paso le corresponde un identificador de Regla Morfo-Sintáctica VERB-SUST y agrega la descripción matemática de la lógica de validación/control.

Ejemplos de este tipo de oraciones son: validar que los artículos tengan stock positivo (VERB-SUST; “tengan stock positivo” es la lógica matemática de la validación),
20 controlar que el cliente tenga número fiscal (VERB-SUST; “tenga número fiscal” es la lógica matemática de validación), restringir el ingreso de mercaderías vencidas (VERB-SUST; “mercaderías vencidas” es la lógica matemática de la validación).

Paso CB (Paso que contiene acciones de consulta/búsqueda)

25 Son pasos donde el verbo determina acciones de consulta o búsqueda, tales como: buscar, localizar, seleccionar, consultar, mostrar, visualizar, exponer y otros similares. En este caso, el verbo es transitivo y el sustantivo que le sigue señala el objeto de la búsqueda/consulta.

A este tipo de paso le corresponde un identificador de Regla Morfo-Sintáctica VERB-
30 SUST, donde el sustantivo describe el objeto de la búsqueda, lo cual implica que el verbo se ejerce sobre un conjunto de instancias del sustantivo.

Ejemplos de este tipo de oraciones son: consultar los artículos (VERB-SUST), mostrar los saldos (VERB-SUST), visualizar los datos (VERB-SUST), exponer los resultados (VERB-SUST), buscar los artículos (VERB-SUST), localizar los repuestos (VERB-SUST), seleccionar los clientes (VERB-SUST).

5

Paso FX (Paso que contiene acciones de cálculo explícitas)

Son pasos donde el verbo determina acciones de cálculo de manera explícita, tales como: calcular, agrupar, promediar, sumar, y otros similares. En este caso, el verbo siempre está acompañado de una oración que representa para el sistema una lógica matemática.

10

A este tipo de paso le corresponde un identificador de Regla Morfo-Sintáctica VERB-SUST, donde el sustantivo describe la lógica de cálculo que indica el verbo.

Ejemplos de este tipo de oraciones son: calcular el total de ventas (VERB-SUST), agrupar los artículos por color (VERB-SUST), promediar el costo del último mes (VERB-SUST).

15

Paso R (Paso que contiene acciones de relación)

Son pasos donde el verbo determina acciones de relación, tales como: relacionar, vincular, asociar, componer, y otros similares. En este caso, el verbo siempre tiene al menos dos sustantivos a continuación.

20

El verbo puede presentar cualquiera de los modelos siguientes:

Modelo 1:

A este tipo de paso le corresponde un identificador de Regla Morfo-Sintáctica VERB(relaciona)-SUST-PREP-SUST, donde el verbo específico puede ser relaciona o cualquier sinónimo y la preposición, generalmente es “con”, aunque podría ser otra.

25

Ejemplos de este tipo de oraciones son:

relacionar los artículos con los precios (VERB(relaciona)-SUST-PREP-SUST), vincular las maquinarias con los repuestos (VERB(relaciona)-SUST-PREP-SUST), asociar los impuestos con las alícuotas (VERB(relaciona)-SUST-PREP-SUST), componer el producto con los insumos (VERB(relaciona)-SUST-PREP-SUST).

30

Modelo 2:

A este tipo de paso le corresponde un identificador de Regla Morfo-Sintáctica VERB(responsabiliza)-SUST-PREP-SUST, donde el verbo específico puede ser responsabiliza o cualquier sinónimo y la preposición, generalmente es “sobre”, aunque
5 podría ser otra.

Ejemplos de este tipo de oraciones son: responsabiliza al cliente sobre el pago (VERB(responsabiliza)-SUST-PREP-SUST), responsabiliza al empleado sobre el control de stock (VERB(responsabiliza)-SUST-PREP-SUST).

10 Modelo 3:

A este tipo de paso le corresponde un identificador de Regla Morfo-Sintáctica VERB-SUST-PREP(una)-SUST, donde la preposición específica puede ser un, una o cualquier otra.

Ejemplos de este tipo de oraciones son: mide el artículo en una unidad de medida
15 (VERB-SUST-PREP(una)-SUST), valoriza el artículo con una lista de precios (VERB-SUST-PREP(una)-SUST).

Paso F (Paso que no pertenece a ninguno de los tipos anteriores)

20 Son pasos donde el verbo no pertenece a ninguno de los tipos anteriores porque se refiere a una acción específica propia de “el caso”.

El verbo puede presentar cualquiera de los modelos siguientes:

Modelo 1:

25 A este tipo de paso le corresponde un identificador de Regla Morfo-Sintáctica VERB-SUST y se utiliza para todos los verbos que no están listados como palabra específica en los tipos de paso VC, CB o FX, o en alguno de los modelos de este mismo tipo.

Verbo transitivo seguido de uno o más sustantivos. Por ejemplo: Comprar un artículo (VERB-SUST), Vender frutas (VERB-SUST), Arreglar las herramientas (VERB-
30 SUST), Solucionar problemas (VERB-SUST).

Modelo 2:

A este tipo de paso le corresponde un identificador de Regla Morfo-Sintáctica VERB(es)-SUST, donde el verbo específico puede ser "es". Este es un verbo muy especial que implica la naturaleza del sustantivo.

Ejemplos de este tipo de oraciones son: es fruta (VERB(es)-SUST), es una empresa
5 (VERB(es)-SUST).

(a) *Lógica de Palabras: determina cuál es la estructura de tipos de palabras a la cual pertenece la oración y pregunta analizada*

El Procesador **150**, configurado como Analizador Automático de Oraciones **151**, en
10 referencia a la Figura 2E1, realiza las siguientes acciones para obtener de cada oración simple sus componentes funcionales asociados a partir de las palabras:

- Para el idioma de "el caso" toma de la Memoria de Base de Datos estática **160**, de la estructura lógica de Idiomas, Palabras y Reglas Morfo Sintácticas **162**, las terminaciones verbales (TV), la lista de palabras especiales (PE) y dentro de ellas, la
15 lista de palabras agrupadoras (PAG), la lista de palabras a excluir (PEX), y también toma las reglas Morfo-Sintácticas que definen las pautas a aplicar.

- Toma de la Memoria de Base de Datos dinámica **170**, de la estructura lógica Oraciones Simples, Preguntas **171**, las oraciones simples de "el caso" estructuradas en las correspondientes preguntas.

20 - A cada oración simple de "el caso" le asigna un número secuencial.

- Por cada oración simple de "el caso", por cada pregunta, divide una a una las palabras y las caracteriza según su tipo de palabra, así:

. si la pregunta es Cómo, comparar las terminaciones de cada palabra que la componen con las terminaciones verbales TV para identificar los verbos. Una vez identificados los
25 verbos, comparar el resto de las palabras con la lista de palabras especiales PE para identificar preposiciones, artículos, conjunciones y adverbios. Las palabras consecutivas a los verbos, que no son PE, son sustantivos. Las palabras que son consecutivas a los sustantivos, pueden ser PE que forman parte de la lista de palabras agrupadoras PAG, en tal caso, la palabra PAG más su anterior y su consecutiva conforman un sustantivo
30 agrupado. Las palabras restantes que no son PE, son adjetivos.

. si la pregunta es Qué, la primera palabra es un verbo. Una vez identificados los verbos, comparar el resto de las palabras con la lista de palabras especiales para identificar

preposiciones, artículos, conjunciones y adverbios. Si el verbo es transitivo, la palabra consecutiva al verbo que no es PE, es un sustantivo que se comporta como objeto directo (OD). Las palabras que son consecutivas a los sustantivos, pueden ser PE que forman parte de la lista de palabras PAG, en tal caso, la palabra PAG más su anterior y su consecutiva conforman un sustantivo agrupado. Las palabras restantes que no son PE, son adjetivos.

. si la pregunta es Quién, comparar el resto de las palabras con la lista de palabras especiales para identificar preposiciones, artículos y conjunciones. Las palabras que son consecutivas a los sustantivos, pueden ser PE que forman parte de la lista de palabras PAG, en tal caso, la palabra PAG más su anterior y su consecutiva conforman un sustantivo agrupado. Las palabras restantes que no son PE, son adjetivos.

El Meta-Modelo MAS y la estructura de Reglas Morfo-Sintácticas se aplica en la Subetapa E4 para Identificar los componentes funcionales útiles para construir documentos de Arquitectura Funcional, en la Subetapa G6 para generar los componentes de diseño de software, y en la Subetapa H4 para definir la notación gráfica asociada al Meta-Modelo MAS de diseños de software. Estas subetapas representan modalidades preferidas de la invención donde los verbos transitivos del texto de “el caso” se tratan según lo define la Subetapa A1.

20 **Etapa B. Ingresar las características del idioma y su estructura**

En esta etapa, haciendo referencia a la Figura 2B se define el o los idiomas que se utilizarán para la descripción de “el caso” en lenguaje natural y sus características morfo-sintácticas, mediante las siguientes subetapas:

25 **Subetapa B1. Definir secuencia de adjetivación**

La secuencia de adjetivación se refiere al orden que ocupan los adjetivos respecto del sustantivo de un idioma.

En el dispositivo de entrada/salida **120**, configurado como Interface de Usuario Idioma **121**, definir la secuencia de adjetivación correspondiente al idioma, de la lista de secuencias de adjetivación disponibles: Secuencia 1: sustantivo+adjetivo; Secuencia 2: adjetivo+sustantivo; Secuencia 3: adjetivo modificado+ sustantivo modificado; Secuencia 4: adjetivo + sustantivo modificado; Secuencia 5: adjetivo

modificado+sustantivo. Por ejemplo, para el idioma castellano, la secuencia de adjetivación correspondiente es sustantivo+adjetivo.

Subetapa B2. Cargar las palabras especiales

En el dispositivo de entrada/salida **120**, configurado como Interface de Usuario Idioma
5 **121**, cargar las palabras especiales del idioma definido en la Subetapa B1. Las palabras especiales son utilizadas por el Procesador **150** configurado como Analizador Automático de Oraciones **151** para excluirlas del análisis de oraciones. En el presente invento, se consideran palabras especiales de un idioma a los artículos (ART), preposiciones (PREP), conjunciones (CONJ) y adverbios (ADV). En el idioma
10 castellano, son ejemplo de artículos, las palabras “el” y “los”, son ejemplo de preposiciones las palabras “para” y “con”, son ejemplo de conjunciones las palabras “y” u “o” y son ejemplo de adverbios las palabras “como” y “cuando”.

Subetapa B3. Cargar las palabras agrupadoras

En el dispositivo de entrada/salida **120**, configurado como Interface de Usuario Idioma
15 **121**, cargar las palabras agrupadoras del idioma definido en la Subetapa B1. En el presente invento, palabras agrupadoras son palabras especiales de las definidas en la Subetapa B2 que pueden unir a otras dos palabras para componer una palabra agrupada. Tal es el caso de la palabra “lista de precios”, agrupada por la palabra especial “la”.

Subetapa B4. Cargar las terminaciones verbales

20 En el dispositivo de entrada/salida **120**, configurado como Interface de Usuario Idioma **121**, cargar las terminaciones verbales correspondientes al idioma de la Subetapa B1, correspondientes a la sílaba final de las conjugaciones verbales regulares. Son ejemplo para el caso del castellano, las siguientes terminaciones verbales: ar, er, ir, ando, iendo.

25 **Etapa C. Ingresar la descripción de “el caso”**

En esta etapa, haciendo referencia a la **Figura 2C**, se describe el caso en lenguaje natural mediante las siguientes subetapas:

Subetapa C1. Escoger el idioma para la descripción de “el caso”

30 En el dispositivo de entrada/salida **120**, configurado como Interface de Usuario del Caso **122**, elegir el idioma en el cual se describe “el caso”. Este idioma se elige de la lista de idiomas elegibles cargados en la Etapa B.

Subetapa C2. Identificar los componentes que estructuran el contexto de “el caso”

5 En una modalidad de la invención se identifican los componentes que estructuran el contexto de “el caso”.

En esta etapa se trabaja sobre la visión estática y la visión dinámica temporal del Universo de “el caso”. En la visión estática se observa desde una perspectiva global la estructura conceptual de “el caso” sin tomar en cuenta lo que sucede con el paso del tiempo, dividiendo los conceptos en dimensiones o grandes aspectos que lo componen: 10 capas y recursos. En la visión dinámica temporal se observa desde la perspectiva de los acontecimientos que suceden con el paso del tiempo (actividades temporales) y en el orden en que suceden: procesos, momentos y pasos.

15 Para estructurar la descripción de “el caso” en una modalidad sugerida de la invención se identifican las capas, los recursos, los procesos, los subprocesos y los momentos.

(a) Identificar las capas de “el caso”

20 Se entiende por capa una jerarquía de información que participa en un proceso y podría ser tratada de manera independiente de las otras, que podrían funcionar de manera aislada con entradas y salidas bien diferenciadas. Por ejemplo, podría identificar dos capas: la capa de datos (donde estructuro y accedo a los datos de entrada de un proceso) y la capa de cálculo (donde realizo cálculos en base a dichos datos).

(b) Identificar los recursos utilizados en “el caso”

25 Los recursos son aquellos componentes que son sometidos a transformación a lo largo del proceso y permiten obtener otros recursos como resultado del proceso de transformación. Por ejemplo, los bienes de cambio son recursos que se someten al proceso de venta.

30 *(c) Identificar los procesos contenidos en “el caso”*

Los procesos son aquellas acciones o grupos de acciones que transforman a los recursos. Por ejemplo, el proceso de “comercializar bienes de cambio”.

(d) Identificar los subprocesos de cada proceso de “el caso”

Los subprocesos son partes de un proceso que permiten analizarlo de modo fragmentado. Por ejemplo, en el proceso de “comercializar bienes de cambio” puedo referirme a los subprocesos de “cotización” y “venta”.

(e) Identificar los Momentos de “el caso”

En la presente invención, los momentos son divisiones de un subproceso que enuncian cuándo sucede algo que constituye un hecho relevante para el subproceso. Por ejemplo, “Cuando el cliente tiene interés” es un momento que se produce dentro del subproceso de “cotización”.

Subetapa C3. Identificar los Pasos de “el caso”

Los pasos son las actividades que se llevan a cabo en un determinado momento. Ejemplo: las actividades del momento “Al ingresar la persona a la oficina del test”, podrían ser: Solicitar su documento de identidad, Registrar los datos de su documento, Realizar consultas de su grupo familiar, Consultar acerca de su trabajo actual.

Para asegurar que la descripción de “el caso” sea completa es necesario incluir todos los pasos, aún aquellos que generalmente se omiten porque parecen obvios.

Los pasos, dado que son oraciones y determinan la acción a realizar a través del verbo, se pueden clasificar según el tipo de acción que el verbo determina, y analizando los sustantivos que siguen al verbo (sin considerar las palabras especiales definidas en la Subetapa B2 artículos, conjunciones, proposiciones y preposiciones).

Las pautas para clasificar las oraciones por Tipo de paso, se denomina Lógica de Tipo de Paso, como parte de las Reglas Morfo-Sintácticas definidas en la Subetapa A2 y se describen a continuación:

Paso VC: (Paso que contiene acciones de validación/control)

Son pasos donde el verbo determina acciones de validación o control, tales como: Validar, Controlar, Restringir y otros similares. En este caso, el verbo siempre está acompañado de una oración que representará para el sistema una lógica matemática. Ejemplos de este tipo de oraciones son: Validar que los artículos tengan stock positivo, Controlar que el cliente tenga número fiscal, Restringir el ingreso de mercaderías vencidas.

Paso CB (Paso que contiene acciones de consulta/búsqueda)

Son pasos donde el verbo determina acciones de consulta o búsqueda, tales como: Buscar, Localizar, Seleccionar, Consultar, Mostrar, Visualizar, Exponer y otros similares. En este caso, el verbo siempre tiene un sustantivo a continuación u objeto directo. Ejemplos de este tipo de oraciones son: Consultar los artículos, Mostrar los saldos, Visualizar los datos, Exponer los resultados, Buscar los artículos, Localizar los repuestos, Seleccionar los clientes.

Paso FX (Paso que contiene acciones de cálculo explícitas)

Son pasos donde el verbo determina acciones de cálculo de manera explícita, tales como: Calcular, Agrupar, Promediar, Sumar, y otros similares. En este caso, el verbo siempre está acompañado de una oración que representa para el sistema una lógica matemática. Ejemplos de este tipo de oraciones son: Calcular el total de ventas, Agrupar los artículos por color, Promediar el costo del último mes.

Paso R (Paso que contiene acciones de relación)

Son pasos donde el verbo determina acciones de relación, tales como: Relacionar, Vincular, Asociar, Componer, y otros similares. En este caso, el verbo siempre tiene al menos dos sustantivos a continuación. Ejemplos de este tipo de oraciones son: Relacionar los artículos con los precios, Vincular las maquinarias con los repuestos, Asociar los impuestos con las alícuotas, Componer el producto con los insumos.

Paso F (Paso que no pertenece a ninguno de los tipos anteriores)

Son pasos donde el verbo no pertenece a ninguno de los tipos anteriores porque se refiere a una acción específica propia de “el caso”. El verbo puede presentar cualquiera de los modelos siguientes: Modelo 1: Verbo seguido de uno o más sustantivos. Por ejemplo: Comprar un artículo, Vender frutas, Arreglar las herramientas, Solucionar problemas. Modelo 2: Verbo que no tiene ningún sustantivo a continuación. Por ejemplo: Entrar, Salir.

Generalmente los pasos se describen en la siguiente secuencia: 1) Paso F, 2) Paso VC, 3) Paso CB, 4) Paso FX, 5) Paso R.

30 Etapa D. Identificar y almacenar las oraciones simples de "el caso"

En esta etapa se listan las oraciones simples de “el caso”, tomando cada párrafo de la descripción de “el caso” y extrayendo las oraciones con un solo verbo, para cargarlas en

el dispositivo de Entrada/Salida **120** Interface de Usuario Caso **122**. Por ejemplo, si el párrafo descriptivo es: “El artículo es un bien de cambio. La persona mide el artículo en una unidad de medida y registra el Pedido, sobre el cual es responsable, relacionándolo con un Tipo de Pedido.”, las oraciones simples que le corresponden son: “La persona
5 mide el artículo en una unidad de medida.”, “La persona registra el Pedido”, “La organización responsabiliza a la persona sobre el Pedido”, “El artículo es un bien de cambio”, “La persona relaciona el Pedido con un Tipo de Pedido.”

En los textos libres, existen oraciones con sujeto tácito separadas generalmente por conjunciones, dentro de un mismo párrafo. En tal caso, se expone el sujeto para
10 completar la oración, con el objetivo de convertir las oraciones con sujeto tácito en oraciones simples con sujeto expreso. Por ejemplo, en la siguiente oración separada por la conjunción “y” existe sujeto tácito: “La persona mide el artículo en una unidad de medida y registra el Pedido”, y resulta en dos oraciones simples: “La persona mide el artículo en una unidad de medida.”, “La persona registra el Pedido”.

15 En esta etapa se debe considerar que una descripción es un conjunto de oraciones simples que describen “el caso”. A cada paso de “el caso” le corresponde una oración simple que se deriva de la respuesta a las preguntas: Quién, Qué, Dónde, Cómo y Cuándo.

Una vez identificadas las oraciones simples y completadas aquellas que tienen sujeto
20 tácito, se cargan en el dispositivo de Entrada/Salida **120** Interface de Usuario Caso **122**.

En esta etapa, haciendo referencia a la Figura 2D se identifican las oraciones simples de “el caso” mediante las siguientes subetapas:

Subetapa D1. Responder a las preguntas por cada paso de “el caso”

25 En esta etapa se toma cada oración simple de “el caso” y se completan las preguntas Cuándo, Quién, Qué, Cómo y Dónde.

Para cada oración simple almacenada en la Memoria de Base de Datos Dinámica Oraciones simples y Preguntas **171**, El Procesador **150** expone cada oración en el
30 dispositivo de Entrada/Salida **120** Interface de Usuario Caso **122**, presentando las preguntas Quién, Qué, Dónde, Cómo, Cuándo. Para cada oración presentada el usuario anota la respuesta, respetando que cada respuesta sea una parte constitutiva del

significado de la oración. En algunos casos, es posible que una o varias de las preguntas no se puedan responder.

A través de la respuesta a las preguntas anteriormente mencionadas, se hacen explícitas todas las partes de la oración simple, la cual, en una modalidad preferida de la presente invención debe ser escrita en tiempo verbal presente.

Las pautas para clasificar las oraciones por pregunta, se denomina Lógica de Preguntas, como parte de las Reglas Morfo-Sintácticas definidas en la Subetapa A2.

Subetapa D2. Concatenar las respuestas de cada paso

Una vez cargadas las respuestas a las preguntas, se concatena cada respuesta en el siguiente orden: Cuándo, Quién, Qué, Cómo, Dónde. De este modo, los textos de las respuestas quedan unidos en una oración simple y se almacenan en la Memoria de Base de Datos Dinámica **170**, en la estructura lógica Oraciones simples y Preguntas **171**.

A través de la respuesta a las preguntas anteriormente mencionadas, se hacen explícitas todas las partes de la oración simple, la cual debe ser escrita en tiempo verbal presente.

15

Por ejemplo: para la oración incompleta: Solicitar su documento de identidad, si se aplica el método previamente descrito, se obtiene una oración simple: Al ingresar la persona, el operador del sistema registra el número de documento.

20 Etapa E. Identificar y almacenar los componentes funcionales.

En la presente invención, se debe entender que un componente funcional corresponde a cada palabra de la oración extraída y caracterizada automáticamente por el Procesador **150**, según la estructura gramatical y sintáctica del idioma en el cual se encuentra expresada.

25

En esta etapa, haciendo referencia a la **Figura 2E**, se describen las subetapas para identificar los componentes funcionales de la oración.

30

Subetapa E1. Descomponer la oración simple en palabras y caracterizarlas

Los componentes funcionales que genera el Procesador **150** se dividen en los siguientes tipos:

- Componentes funcionales sustantivos
 - 5 ○ Identificados en el Qué -> Sustantivo - Objeto Directo (OD)
 - Identificados en el Quién -> Sustantivo - Persona (PER)
 - Identificados en el Cuándo -> Sustantivo (SUST)
 - Identificados en el Cómo -> Sustantivo (SUST)
- Componentes funcionales adjetivos
 - 10 ○ Identificados en cualquiera de las preguntas -> Adjetivo (ADJ)
- Componentes funcionales verbos
 - Identificados en el Qué -> Verbo (VERB)
 - Identificados en el Cómo -> Verbo (VERB)
- Componentes funcionales adverbiales
 - 15 ○ Identificados en el Cómo -> Adverbio (ADV)

Las pautas para estructurar las palabras, se denomina Lógica de Palabras, como parte de las Reglas Morfo-Sintácticas definidas en la Subetapa A2 y se describen a continuación:

En esta subetapa, el Procesador **150**, configurado como Analizador Automático de
20 Oraciones **151**, ejecuta las siguientes acciones para cada oración simple de “el caso”:

(a) *Listar palabras.*

Listar por cada oración, cada una de sus palabras.

25 (b) *Identificar palabras a excluir.*

Ignorar en la selección de componentes funcionales las palabras a excluir definidas en la Subetapa B2; excepto en los casos que forman parte de sustantivos agrupados, por ejemplo: lista de precios, caja de galletas.

(c) *Identificar verbos.*

30 Identificar verbos comparando la terminación de cada palabra con la lista de terminaciones verbales definidas en la Subetapa B4 o como la primera palabra de la respuesta a la pregunta Qué de la oración simple.

(d) *Identificar sustantivos.*

Identificar sustantivos, como las palabras que no son verbos y no fueron excluidas.

5 (e) *Etiquetar sustantivos como atributo&*

Algunos sustantivos se comportan como atributos&. Se entiende por atributo& el nombre de una característica de otro sustantivo, que no es un adjetivo. En una de las modalidades de la invención, sobre la lista de sustantivos identificados, se seleccionan los atributos& de manera manual y en otra modalidad de manera automática, cuando en
10 el texto los sustantivos se enumeran entre paréntesis, el procesador los reconoce en forma automática. Ejemplo: en la oración, carga el número de documento del cliente, número de documento es un atributo& de cliente.

(f) *Identificar sustantivos agrupados.*

15 Identificar como sustantivos, los sustantivos agrupados cuando aparecen dos sustantivos consecutivos con alguna palabra especial entre ellos, que actúa como nexos.

(g) *Identificar los componentes funcionales a partir de las palabras.*

De la lista de palabras caracterizadas se identifican como componentes funcionales los
20 sustantivos y verbos, caracterizados según corresponda en VERBO, OD, SUSTANTIVO, PERSONA.

(h) *Validar uno a uno la existencia previa de cada componente funcional.*

Cuando se identifica un nuevo componente funcional de cualquier tipo, se chequea si existe previamente. Si no existe se crea, si existe, continúa.
25

(i) *Asociar componentes funcionales con la pregunta que responden.*

Tipificar a partir de la pregunta a la que pertenece la palabra el tipo de componente funcional:

- 30 i. Si pertenece al cuándo: las palabras agrupadas con un verbo son componentes funcionales adverbiales (ADV).
- ii. Si pertenecen al Qué: los sustantivos que siguen al verbo son componentes funcionales de tipo objeto directo (OD).

- iii. Si pertenecen al Quién: el primer sustantivo es un componente funcional persona (PERS).

El Procesador **150**, configurado como Analizador Automático de Oraciones **151**, realiza las acciones enumeradas previamente, como se muestra en la Figura 2E1, para obtener de cada oración simple sus componentes funcionales asociados a partir de las palabras:

- Para el idioma de "el caso" toma de la Memoria de Base de Datos estática **160**, de la estructura lógica de Idiomas, Palabras y Reglas Morfo Sintáticas **162**, las terminaciones verbales (TV), la lista de palabras especiales (PE) y dentro de ellas, la lista de palabras agrupadoras (PAG), la lista de palabras a excluir (PEX).
- Toma de la Memoria de Base de Datos dinámica **170**, de la estructura lógica Oraciones Simples, Preguntas **171**, las oraciones simples de "el caso" estructuradas en las correspondientes preguntas.
- A cada oración simple de "el caso" le asigna un número secuencial.
- Por cada oración simple de "el caso", por cada pregunta, divide una a una las palabras y las caracteriza según su tipo de palabra, así:
 - . si la pregunta es Cómo, comparar las terminaciones de cada palabra que la componen con las terminaciones verbales TV para identificar los verbos. Una vez identificados los verbos, comparar el resto de las palabras con la lista de palabras especiales PE para identificar preposiciones, artículos, conjunciones y adverbios. Las palabras consecutivas a los verbos, que no son PE, son sustantivos. Las palabras que son consecutivas a los sustantivos, pueden ser PE que forman parte de la lista de palabras agrupadoras PAG, en tal caso, la palabra PAG más su anterior y su consecutiva conforman un sustantivo agrupado. Las palabras restantes que no son PE, son adjetivos.
 - . si la pregunta es Qué, la primera palabra es un verbo. Una vez identificados los verbos, comparar el resto de las palabras con la lista de palabras especiales para identificar preposiciones, artículos, conjunciones y adverbios. Si el verbo es transitivo, la palabra consecutiva al verbo que no es PE, es un sustantivo que se comporta como objeto directo (OD). Las palabras que son consecutivas a los sustantivos, pueden ser PE que forman parte de la lista de palabras PAG, en tal caso, la palabra PAG más su anterior y su consecutiva conforman un sustantivo agrupado. Las palabras restantes que no son PE, son adjetivos.

. si la pregunta es Quién, comparar el resto de las palabras con la lista de palabras especiales para identificar preposiciones, artículos y conjunciones. Las palabras que son consecutivas a los sustantivos, pueden ser PE que forman parte de la lista de palabras PAG, en tal caso, la palabra PAG más su anterior y su consecutiva conforman un sustantivo agrupado. Las palabras restantes que no son PE, son adjetivos.

Subetapa E2. Agregar componentes funcionales no incluidos en la descripción.

En una modalidad de la invención un usuario agrega componentes funcionales que no se encuentran en la lista de los componentes funcionales identificados de forma automática. Sobre la base del ejemplo que estamos usando:

- Número de documento es un atributo detectado automáticamente a partir de la oración.
- Edad de persona es un atributo agregado por el usuario.

En una modalidad de la invención, por cada componente funcional que se agrega, se crea automáticamente el paso y la oración simple correspondiente.

Subetapa E3. Enriquecer los componentes funcionales.

En una modalidad de la invención, el usuario agrega las expresiones de fórmula asociadas a los pasos tipo FX y VC, y los mensajes asociados a dichos pasos.

(a) Agregar expresiones de fórmulas.

En una modalidad de la invención un usuario agrega las expresiones de las fórmulas que describen el comportamiento matemático de “el caso” para todos los pasos de tipo VC y FX. Por ejemplo:

- {expresión de fx}="(param1, param 2, param3) expresión= param1*param2/param3"

(b) Agregar mensajes.

En una modalidad de la invención un usuario agrega los mensajes de error, de aprobación y advertencia. Por ejemplo:

- {mensaje advertencia} ="El valor no debe ser mayor que cero"
- {mensaje ok} ="El dato se guardó correctamente"

- {mensaje de error}="Error de lectura del dato"

Subetapa E4. Aplicar reglas Morfo-Sintácticas en componentes funcionales

- 5 En una modalidad de la invención, el usuario puede habilitar el uso de reglas Morfo-Sintácticas, para que el Procesador **150** respete la lógica definida en la Subetapa A2.
- En esta modalidad, cada palabra de una oración simple, es caracterizada como Palabra **161-1** según el Meta-Modelo MAS de la Figura 2A1, y se le asigna un tipo de Palabra VERB, SUST, ART, CONJ, ADJ. También se crean las Palabras Factor **161-2** y
- 10 Factible **161-3**, asignando a los verbos transitivos la clase Factibilidad **161-4**. Todas las Palabra **161-1**, caracterizadas por tipo, son los componentes funcionales. Como se muestra en la Figura 2D, una vez creado cada componente funcional se le asocia la oración de la que proviene, la pregunta a la que pertenece, el tipo de palabra que le corresponde y el identificador de Regla Morfo-Sintáctica que representa a la secuencia
- 15 de tipos de palabras que conforman la pregunta, como se indica en la Subetapa A2.

En la selección de componentes funcionales las palabras especiales definidas en la Subetapa B2, excepto en los casos que la Regla Morfo-Sintáctica correspondiente la señale como palabra específica.

- 20 En esta modalidad, por cada verbo transitivo el Procesador **150** crea dos componentes funcionales nuevos. Un componente funcional llamado como el verbo, con el sufijo "OR" (Factor **161-2**) y un componente funcional llamado como el verbo, con el sufijo "BLE" (Factible **161-3**). El verbo transitivo es la Palabra Factibilidad **161-4**. Para el sustantivo que ejerce el verbo transitivo, Procesador **150** crea un verbo "es" asociado al
- 25 OR del verbo. Para el sustantivo destino (objeto directo de la oración), el procesador crea un verbo "es" asociado al BLE de verbo.

Etapa F. Generar los documentos de Arquitectura Funcional

- En esta etapa, haciendo referencia a la Figura 2F el Procesador **150** configurado como
- 30 Generador de Documentos **153**, genera los documentos de Arquitectura Funcional, utilizando los formatos y notaciones almacenados en la Memoria de Base de Datos estática **160**, en la estructura lógica Formatos, Notación Gráfica y Oraciones STD **163** y

los presenta en el dispositivo de Entrada/Salida **120**, Interface de Usuario de Exposición de Documentos **124**.

5 En esta etapa, haciendo referencia a la Figura 2F se generan los documentos de Arquitectura Funcional mediante las siguientes subetapas:

Subetapa F1. Establecer Reglas Morfo-Sintácticas para Arquitectura funcional

10 Según se establece en la Etapa A, por cada verbo transitivo seguido de una preposición “a” o “al”, el procesador crea para el sustantivo que ejerce el verbo, un componente sintáctico llamado como el verbo, con el sufijo “or” (en adelante llamado OR del verbo) y para el sustantivo que recibe la acción del verbo, un componente sintáctico llamado como el verbo, con el sufijo “ble” (en adelante llamado BLE del verbo). Entre el sustantivo origen y el OR del verbo, el procesador crea un nuevo sustantivo llamado
15 “Factibilidad del verbo”. Para el sustantivo origen, el procesador crea un verbo “es” asociado al BLE del verbo. Para cada sustantivo destino, el procesador crea un verbo “es” asociado al OR de verbo.

Subetapa F2. Definir la notación gráfica a utilizar en el documento de
20 **Arquitectura funcional**

En esta etapa, mediante el dispositivo de Entrada/Salida **120**, Interface de Usuario Formatos, Notación Gráfica y Oraciones STD **123**, se define la notación gráfica a utilizar en los diagramas de Arquitectura Funcional, indicando para cada componente funcional, según el tipo de palabra que le corresponde, el componente gráfico a utilizar
25 en el diagrama. Almacenar en la Memoria de Base de Datos **130** las notaciones gráficas definidas. De este modo, el usuario ingresa las notaciones gráficas en la grilla visual que se expone en Interface de Usuario Formatos, Notación Gráfica y Oraciones STD **123**, así:

30

Tipo de Palabra	Elemento gráfico	ID	Descripción
SUST		GSUST	Representa las palabras de tipo sustantivo
VERB OR-BLE, creado por el procesador		GORBLE	Representa a las palabras generadas a partir de un verbo transitivo, como OR y BLE
VERB "ser"		GSER	Representa el verbo ser
PREP		GPREP	Representa las preposiciones
VERB de cualquier otro tipo		GVERB	Representa al resto de los verbos
VERB factibilidad creado por el procesador		GFACT	Representa la relación entre el OR y el BLE

Subetapa F3. Generar documentos de Arquitectura Funcional

El Procesador 150, configurado como Generador de Documentos 153, toma los componentes funcionales almacenados en la Memoria de Base de Datos Dinámica 170, en la configuración lógica Componentes Funcionales 172, y construye un documento de Arquitectura Funcional, utilizando la notación definida en la Subetapa F2, aplicando el Meta-Modelo MAS residente en la Memoria de Base de Datos estática 160, estructura lógica de Meta-modelo MAS 161. Para lograr el diagrama de Arquitectura Funcional, el procesador recorre la lista de componentes funcionales y aplica los siguientes criterios hasta lograr un diagrama como el que se observa en la Figura 2F1:

Criterio 1. Por cada componente funcional de tipo SUST se dibuja un gráfico GSUST con el nombre del CF en su interior.

Criterio 2. Por cada componente funcional de tipo VERB se dibuja una línea entre el SUST previo y el SUST siguiente al VERB, considerando: i) si el VERB es "ser o alguna de sus conjugaciones" se dibuja un gráfico GSER y tiene dirección desde el SUST anterior al VERB hacia el SUST inmediato posterior al VERB; ii) si el VERB es

cualquier otro verbo transitivo, se dibuja un gráfico GORBLE y tiene dirección desde el SUST anterior al VERB hacia el SUST inmediato posterior al VERB; iii) si el VERB no es transitivo, se dibuja un gráfico GVERB y tiene dirección desde el SUST anterior al VERB hacia el mismo sustantivo.

- 5 Criterio 3. Por cada componente funcional de tipo PREP, sólo para los casos de las preposiciones “de” y “del”, se dibuja un gráfico GPREP entre el SUST previo a la preposición y el SUST siguiente a la preposición.

Criterio 4. Por cada verbo transitivo se dibujan los componentes gráficos asociados a las palabras creadas en base al Meta-Modelo MAS definido en la Subetapa A1: un gráfico
10 GORBLE para la palabra Factor, un gráfico GORBLE para la palabra Factible y un gráfico GFACT para la palabra Factibilidad.

El Procesador **150** visualiza los documentos de Arquitectura Funcional en el Dispositivo de Entrada/Salida **120**.

15

Etapa G. Identificar y almacenar los componentes de diseño.

En la presente invención, un documento de diseño de “el caso” está compuesto por los siguientes diagramas: Diagramas de diseño conceptual, Diagramas de casos de uso,
20 Diagramas de clases, Diagramas de entidad relación, Diagramas de diseño de Pantallas y Diagramas de diseño de Reportes.

Cada documento de diseño de “el caso” presenta gráficos. En particular, se utilizan en el presente invento el Diagrama de clases. A cada elemento gráfico que forma el diagrama
25 de clases según el paradigma de Orientación a Objetos (OO) le llamaremos componente de diseño (CD). Ejemplo: clase, relación, atributo, método.

Para la comprensión de la presente invención se entenderá que clase, atributo, relación, método, son componentes de diseño según se define en el paradigma de Orientación a
30 Objetos (OO).

En esta etapa, haciendo referencia a la Figura 2G, el Procesador **150** configurado como Diseñador Automático de Software **152**, diseña automáticamente el software mediante las siguientes subetapas:

5 Subetapa G1. Agrupar componentes funcionales, crear sus clases y relaciones de herencia.

En esta subetapa el Procesador **150** configurado como Diseñador Automático de Software **152**, crea las clases y relaciones de herencia, a partir de la similitud entre sus atributos, realizando las siguientes acciones:

10 (a) *Selecciona componentes funcionales*

Toma la lista de los componentes funcionales creados en la Etapa D, que se encuentran almacenados en la Memoria de Base de Datos Dinámica **170**, en la estructura lógica Componentes Funcionales **172**, exceptuando aquellos componentes funcionales etiquetados como “Es atributo” y aquellos que son de Tipo Palabra igual a VERBO.

15 Suprime los componentes funcionales que tengan una gran similitud en su nombre (por ejemplo, el 90% utilizando el algoritmo de Distancia de Levenshtein), tal es el caso de las palabras Persona y Personas, donde se considera un mismo componente funcional Persona.

20 (b) *Agrupar componentes funcionales similares*

Agrupar los componentes funcionales incluidos en el paso anterior por {enumeración de atributos&}, colocando en el mismo grupo a aquellos componentes funcionales que tienen igual {enumeración de atributos&}. Toma aquellos grupos que tienen más de un componente funcional y les asigna un nombre de palabra, por ejemplo, Palabra01,

25 Palabra02.

(c) *Crea clases por grupo*

Crea una clase por cada Grupo con más de un componente funcional, indicando como atributos cada elemento de la {enumeración de atributos&} y asignándole a la clase el

30 nombre de la palabra correspondiente al grupo.

(d) Crea relaciones de herencia para grupos

Para aquellos componentes funcionales que pertenecen a los grupos con más de un elemento, crea una relación de herencia entre cada clase correspondiente al componente funcional y su respectiva clase de grupo, dependiendo del grupo de {enumeración de atributos&} que le corresponde. Es decir, la relación de herencia existe entre clases que

5 tienen la misma {enumeración de atributos&} generalizadas en la nueva clase de grupo del paso previo.

Subetapa G2. Crear clases fórmula a partir de Pasos tipo FX y VC

10

En esta subetapa el Procesador **150** configurado como Diseñador Automático de Software **152**, realiza las siguientes acciones sobre las oraciones simples, fraccionadas por pregunta:

(a) Listar componentes funcionales VERBO y OD pertenecientes a pasos FX y VC

15 Lista todos los componentes funcionales de tipo VERBO y OD e identifica el tipo de paso que le corresponde, según lo definido en la Subetapa C3.

(b) Crea clases para tipos FX y VC

20 A partir de los componentes funcionales pertenecientes a pasos de tipo FX y VC, crea clases con el comportamiento definido por el VERBO. Por cada VERBO crear una clase, y le agrega un método llamado fórmula responsable de ejecutar la expresión indicada por {Expresion de fx}.

Por cada clase creada en el paso anterior, realizar las siguientes acciones:

- 25 a. Si el OD “Es Atributo”, crear un atributo en la clase con el nombre de OD.
- b. Si el OD no “Es Atributo”, crear una relación entre la clase del paso anterior y aquella clase que contiene a OD como su atributo (la llamaremos ClaseDestino). Llamar a esta relación Movimiento_de_ClaseDestino.

30

Subetapa G3. Crear clases dominio a partir de Pasos tipo CB

En esta subetapa el Procesador **150** configurado como Diseñador Automático de Software **152**, realiza las siguientes acciones:

- 5 (a) *Lista componentes funcionales VERBO y OD pertenecientes a pasos CB*

Lista todos los componentes funcionales de tipo VERBO y OD e identifica el tipo de paso que le corresponde, según lo definido en la Subetapa C3.

- (b) *Crea clases para pasos tipo CB*

10

A partir de los componentes funcionales de tipo CB, crea clases con el comportamiento definido por el VERBO. Por cada VERBO crea una clase denominada VERBO+OD, agrega un método llamado dominiofx, responsable de consultar datos según lo define {Expresión de fx}. Crea una clase denominada “Dominio “ + VERBO+OD con los atributos& indicados en {enumeración de atributos&}. Si la enumeración de atributos& coincide con los atributos& de alguna de las clases creadas previamente, crea una relación de herencia entre estas últimas en el sentido correspondiente.

15

El método dominiofx invoca al “Dominio “ + VERBO+OD.

20 Subetapa G4. Crear clases relación a partir de Pasos tipo R

En esta subetapa el Procesador **150** configurado como Diseñador Automático de Software **152**, realiza las siguientes acciones:

- (a) *Lista componentes funcionales VERBO y OD pertenecientes a pasos tipo R*

25 Lista todos los componentes funcionales de tipo VERBO y OD e identifica el tipo de paso que le corresponde, según lo definido en la Subetapa C3.

- (b) *Crea clases para tipos R*

A partir de los componentes funcionales de tipo R, crear una relación entre OD y SUSTANTIVO denominada OD + “para” + SUSTANTIVO.

30

Subetapa G5. Crear clases operación a partir de Pasos tipo F

En esta subetapa el Procesador **150** configurado como Diseñador Automático de Software **152**, realiza las siguientes acciones:

5 (a) *Lista componentes funcionales VERBO y OD pertenecientes a pasos F*

Lista todos los componentes funcionales de tipo VERBO y OD e identifica el tipo de paso que le corresponde, según lo definido en la Subetapa C3.

10 (b) *Crea clases para tipos F*

A partir de los componentes funcionales de tipo F, crear las siguientes clases dependiendo de si el OD “Es Atributo”:

- a. Si OD “Es Atributo”: no crear ninguna clase.
- b. Si OD no “Es Atributo”:
 - 15 i. crear una clase nombrada: VERBO + OD, la llamaremos Clase_Cab
 - ii. A la clase OD del paso tipo F, la llamaremos Clase_Rec
 - iii. Crear una relación de 1 a n entre la Clase_cab y la Clase_rec. Llamar a la relación “Movimiento”+OD.

20

Subetapa G6. Aplicar reglas Morfo-Sintácticas en componentes de diseño

En una modalidad de la invención, el usuario puede habilitar el uso de reglas Morfo-Sintácticas, para que el Procesador **150** respete la lógica definida en la Subetapa A2.

- A partir de las Reglas Morfo-Sintácticas definidas en la Subetapa A2, residentes en la
- 25 Memoria de Base de Datos estática **160**, en la estructura lógica de Idiomas, Palabras y Reglas Morfo Sintácticas **162**, el Procesador **150**, determina qué componentes de diseño de software (CD) se derivan de cada componente funcional (CF) almacenado Memoria de Base de Datos Dinámica **170**, en la estructura lógica Componentes Funcionales **172**.

Lista las preguntas, a partir de los tipos de palabras que componen la respuesta crea el identificador de regla para dicha pregunta. Luego busca ese identificador de regla en la Memoria de Base de Datos Estática **160**, en la estructura lógica de Idiomas, Palabras y Reglas Morfo Sintácticas **162**, y localiza la Regla Base por coincidencia de identificador. Puede suceder que exista más de una regla Morfo-Sintáctica con el mismo identificador de Regla Base, en cuyo caso, se selecciona la regla que coincide con la Pregunta, el tipo de paso F y la o las palabras específicas contenidas en el texto e indicadas en la regla Morfo-Sintáctica. En caso de no existir coincidencia de identificador con ninguna regla base, el Procesador **150** ignora la pregunta.

En esta modalidad, por cada VERBO transitivo identificado en las subetapas previas y según se define en la Subetapa A1, el Procesador **150** crea una Palabra **161-4** asociada al verbo y las palabras Factor **161-2** y Factible **161-3**.

Para el caso de un tipo de Paso R, el usuario puede crear una regla Morfo-Sintáctica dependiente de la Regla Base VERB-SUST-PREP-SUST, donde se trate como palabra específica el verbo “relacionar” por ejemplo, en cuyo caso el tipo de paso sería R y el Procesador 150 crearía una relación entre OD y SUSTANTIVO denominada OD + “para” + SUSTANTIVO.

Para el caso de pasos tipo FX y VC, el Procesador 150 crea una Palabra **161-4** asociada al verbo y las palabras Factor **161-2** y Factible **161-3**. El Procesador **150** le agrega a la palabra Factor **161-2**, el método llamado dominiofx.

La Figura 2A3 muestra un modelo de clases de diseño de software con un ejemplo de aplicación que describe las Reglas Morfo-Sintácticas para diseño de Software con base en el Meta-Modelo MAS **161**. Estas reglas Morfo-Sintácticas dan origen al modelado de las oraciones de “el caso”. Para diseñar software a partir del lenguaje natural, se definen Reglas Morfo-Sintácticas para diseño de Software por cada una de las preguntas que describen una oración simple (Cuándo, Quién, Qué, Cómo, Dónde). Estas reglas, almacenadas en la Memoria de Base de Datos estática Idiomas, Palabras y Reglas

Morfo-Sintácticas **162**, se comparan con el identificador de regla calculado para cada oración simple, como se indica en la Sebetapa 2A y a partir de la coincidencia se define la lógica que el Procesador **150**, configurado como Diseñador Automático de Software **152**, utiliza para generar componentes de diseño de software.

5

Etapa H. Definir formatos, notación gráfica y oraciones STD

En esta etapa, haciendo referencia a la Figura 2H se definen los formatos de los documentos de Negocio, Análisis y Diseño, y los parámetros necesarios para generarlos, mediante las siguientes subetapas:

10

Subetapa H1. Definir los formatos de salida para los documentos

En esta subetapa, mediante el dispositivo de Entrada/Salida **120**, Interface de Usuario Formatos, Notación Gráfica y Oraciones STD **123**, se definen los formatos de visualización o impresión de los documentos de negocio, los documentos de análisis, y los documentos de diseño.

15

Estas definiciones implican definir márgenes, tipografías, y niveles de ordenamiento de la secuencia en que se presentará el contenido de cada documento.

20

Para los documentos de negocio, se define en qué orden y secuencia se presentarán capas, recursos, procesos, subprocesos, momentos y pasos.

Para los documentos de análisis, se define en qué orden y secuencia se presentarán los requerimientos globales y detallados.

25

En una modalidad de la invención, es factible que un usuario modifique los formatos, notaciones gráficas y oraciones estándares cargados en las Subetapas F1, F2, F3 y F4, mediante el dispositivo de Entrada/Salida **120**, Interface de Usuario Formatos, Notación Gráfica y Oraciones STD **123**.

30

Subetapa H2. Definir las oraciones estándar de los requerimientos

En esta subetapa, mediante el dispositivo de Entrada/Salida **120**, Interface de Usuario Formatos, Notación Gráfica y Oraciones STD **123**, se ingresan las oraciones estándares para describir requerimientos, se escriben en el idioma o los idiomas que se utilizarán para generar los requerimientos y se almacenan la Memoria de Base de Datos etática **160**, en la estructura lógica Idiomas, Palabras y Reglas Morfo-Sintáticas **162**. A continuación, se describen las oraciones estándares necesarias para generar los requerimientos. Estas oraciones deben ser traducidas y almacenadas en cada uno de los idiomas elegibles que se desee generar documentos de requerimientos.

10

(a) *Oraciones estándares para requerimientos globales:*

Para generar los requerimientos Globales se definen las siguientes oraciones estándares, que aplican al idioma castellano:

15

OracionSTD_abm_Sust: “Realizar Alta, Baja, Modificación y Consulta de”

OracionSTD_abm_Pers: “Agregar, modificar, dar de baja y consultar Entidades con rol”

OracionSTD_verbo_F: “Registrar”

OracionSTD_verbo_R: “Crear regla que “

20

OracionSTD_verbo_conector_R: “con”

OracionSTD_verbo: “Crear regla que “

(b) *Oraciones estándar para requerimientos detallados para componentes SUSTANTIVO y OD*

25

Se definen las siguientes oraciones estándares para generar requerimientos detallados para componentes funcionales SUSTANTIVO y OD, que aplican al idioma castellano:

OracionSTD_Crear: “Crear un nuevo elemento “

30

OracionSTD_Agregar_Atributos: OracionSTD_Agregar_Atributos

OracionSTD_Agregar_Controlos: “Aplicar los siguientes controles al crear un elemento”

OracionSTD_Baja: “Dar de Baja un “

OracionSTD_Edicion: “Modificar un “

OracionSTD_Consulta: “Consultar los registros de “

OracionSTD_complemento_control: “aplicando los siguientes controles”

5 OracionSTD_complemento_búsqueda: “aplicando las siguientes búsquedas “

OracionSTD_crear_atributo: “Crear el atributo “

OracionSTD_validacion_atributo: “Aplicar los siguientes controles al completar el dato”

10 (c) *Oraciones estándar para requerimientos Detallados para componentes PERSONA*

Se definen las siguientes oraciones estándares para generar requerimientos detallados para componentes funcionales PERSONA, que aplican al idioma castellano:

15 OracionSTD_permitirque: “Permitir que “

OracionSTD_acciones_Persona: “realice las siguientes acciones “

OracionSTD_afectacion_Persona: “se vea afectado por las siguientes acciones “

OracionSTD_responsabilidad_Persona: “se realicen bajo la responsabilidad de “

20 (d) *Oraciones estándar para requerimientos Detallados para componentes VERBO*

Se definen las siguientes oraciones estándares para generar requerimientos detallados para componentes funcionales VERBO asociados a pasos de tipo VC o pasos de tipo FX, que aplican al idioma castellano:

25 OracionSTD_crear_Fx: “Crear una fórmula para “

OracionSTD_argumentos: “ utilizando los siguientes datos como argumento ”

OracionSTD_expresion: “ en la siguiente expresión “

OracionSTD_msj_error_Fx: “Si la fórmula arroja un error presentar el siguiente mensaje “

30 OracionSTD_msj_ok_Fx : “Si la fórmula arroja un resultado válido presentar el siguiente mensaje ”

OracionSTD_msj_advertencia_Fx: “Si la fórmula arroja una advertencia presentar el siguiente mensaje ”

Se definen las siguientes oraciones estándares para generar requerimientos detallados para componentes funcionales VERBO asociados a pasos de tipo CB, que aplican al idioma castellano:

OracionSTD_crear_búsqueda: “Crear una regla para”
 OracionSTD_exponer_atributos: “exponiendo los siguientes datos”
 OracionSTD_definir_búsqueda: “Permitir buscar los datos de”
 10 OracionSTD_conector_orden: “por”
 OracionSTD_definir_filtro: “Permitir filtrar los datos de”
 OracionSTD_definir_orden: “Permitir ordenar los datos de”
 OracionSTD_definir_agrupamiento: “Permitir agrupar los datos de”
 OracionSTD_definir_total: “Presentar los siguientes datos sumariados”

15 Se definen las siguientes oraciones estándares para generar requerimientos detallados para componentes funcionales VERBO asociados a pasos de tipo R, que aplican al idioma castellano:

20 OracionSTD_crear_regla: “Crear una regla que”
 OracionSTD_condicion: “siempre que se cumpla la condición de”
 OracionSTD_vincular: “Vincular”
 OracionSTD_conector_vincular: “con”
 OracionSTD_complemento_control: “aplicando los siguientes controles”
 25 OracionSTD_des_vincular: “Desvincular”
 OracionSTD_consultar: “Consultar”
 OracionSTD_complemento_relacionar: “relacionado con”
 OracionSTD_complemento_criteriobúsqueda: “mediante los siguientes requisitos de búsqueda”

30

Se definen las siguientes oraciones estándares para generar requerimientos detallados para componentes funcionales VERBO asociados a pasos de tipo F, que aplican al idioma castellano:

- 5 OracionSTD_permitir: “Permitir”
 OracionSTD_habilitar_persona: “Habilitar a”
 OracionSTD_complemento_accionpersona: “a decidir en la acción”
 OracionSTD_movimientos: “Permitir que los movimientos de”
 OracionSTD_complemento_acargode: “afecten a”
- 10 OracionSTD_control_nuevo: “Aplicar los siguientes controles al crear nuevo
registro de”
 OracionSTD_control_eliminar: “Aplicar los siguientes controles al eliminar
registro de”
 OracionSTD_control_modificar: “Aplicar los siguientes controles al modificar
15 registro de”
 OracionSTD_precedencia: “a partir de los siguientes registros previos:”
 OracionSTD_nuevo_movimiento: “Registrar n movimientos de”
 OracionSTD_control_nuevo_movimiento: “Aplicar los siguientes controles al
crear un nuevo movimiento de”
- 20 OracionSTD_control_eliminar_movimiento: “Aplicar los siguientes controles al
eliminar un movimiento de”
 OracionSTD_control_modificar_movimiento: “Aplicar los siguientes controles
al modificar un movimiento de”
 OracionSTD_buscar_elemento: “Buscar los elementos de”
- 25 OracionSTD_complemento_buscarelemento: “para crear un movimiento,
aplicando las siguientes búsquedas”

Subetapa H3. Definir la notación gráfica a utilizar en el diseño de Software

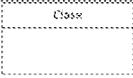
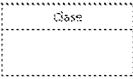
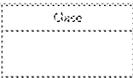
En esta etapa, mediante el dispositivo de Entrada/Salida **120**, Interface de Usuario Formatos, Notación Gráfica y Oraciones STD **123**, se define la notación gráfica a utilizar en los diagramas de diseño de software.

Los diagramas de diseño se pueden presentar en diferentes notaciones gráficas, donde una de ellas es la notación UML (ver **¡Error! No se encuentra el origen de la referencia.**), pero es posible definir elementos gráficos para representar los documentos de diseño.

Subetapa H4. Definir la notación gráfica asociada al Meta-Modelo MAS

indicando para cada elemento del Meta-Modelo MAS definido en la Subetapa A1, el componente gráfico a utilizar en el diagrama de diseño de Software. De este modo, el usuario ingresa las notaciones gráficas en la grilla visual que se expone en Interface de Usuario Formatos, Notación Gráfica y Oraciones STD **123**.

Los diagramas de diseño de software se pueden presentar en diferentes notaciones gráficas, donde una de ellas es la notación UML, pero es posible definir otros elementos gráficos para representar los documentos de diseño.

Notación gráfica para documentos de diseño de software		
Identificador MAS	Nombre MAS	Gráfico
161-0	H Palabra-Palabra	
161-1	Palabra	
161-2	Factor	
161-3	Factible	
161-4	Factibilidad	

161-5	RA Factor-Factibilidad	<input checked="" type="checkbox"/>
161-6	RA Factible-Factibilidad	<input checked="" type="checkbox"/>
161-7	RA Palabra-Palabra	<input checked="" type="checkbox"/>
161-8	RH Palabra-Factible	<input type="checkbox"/>
161-9	RH Palabra-Factor	<input type="checkbox"/>

Etapa I. Generar automáticamente documentos de negocio, análisis y diseño

5 En esta etapa, haciendo referencia a la Figura 2I el Procesador **150** configurado como Generador de Documentos **153**, genera los documentos de negocio, análisis y diseño, utilizando los formatos y notaciones almacenados en la Memoria de Base de Datos estática **160**, en la estructura lógica Formatos, Notación Gráfica y Oraciones STD **163** y los presenta en el dispositivo de Entrada/Salida 120, Interface de Usuario de Exposición
 10 de Documentos **124** mediante las siguientes subetapas:

Subetapa II. Generar documentos de Negocio

En esta etapa, el Procesador **150** configurado como Generador de Documentos **153**,
 15 compone el texto en un documento de negocio. En la presente invención, un documento de negocio es un documento de análisis, que expone las oraciones simples almacenadas en la Memoria de Base de Datos dinámica **170**, en la estructura lógica Oraciones simples y Preguntas **171**, realizando las siguientes actividades:

20 *a). Ordena dimensiones, actividades temporales y oraciones simples*

Ordena jerárquicamente los textos, donde la mayor jerarquía la tienen las capas y la menor jerarquía los pasos, como se puede observar en la **Figura 3**, así:

- Las capas **310** tienen recursos **320**
- Los recursos **320** tienen procesos **330**
- 25 - Los procesos **330** tienen subprocesos **340**
- Los subprocesos **340** tienen momentos **350**

- Los momentos **350** tienen pasos **360** que pueden ser hasta de cinco tipos según se explicó anteriormente.
- Los pasos **360** tienen oraciones simples **370** con sus respectivas preguntas completas.

5

b). Une en forma consecutiva todos los componentes previos ordenados

En una modalidad preferida y haciendo referencia a la Figura 3, la oración simple se obtiene uniendo las respuestas para cada una de las preguntas el orden preferido sería primero Cuándo, luego Quién, luego Qué, luego Cómo y finalmente Dónde, lo cual no
10 significa que no se pueda cambiar el orden.

Cuando se describen todos los procesos mediante este método, se logra una redacción completa y detallada de “el caso” expresada en lenguaje natural.

Subetapa I2. Generar documentos de Análisis

15 En esta etapa, el Procesador **150** configurado como Generador de Documentos **153**, estructura requerimientos funcionales y genera automáticamente documentos de Análisis. En la presente invención, un documento de Análisis expone los Requerimientos obtenidos a partir de los componentes funcionales residentes en la Memoria de Base de Datos dinámica **170**, en la estructura lógica Componentes
20 Funcionales **172**, realizando las siguientes actividades:

a) Arma requerimientos funcionales globales

Los requerimientos funcionales globales de “el caso” son aquellas actividades que describen el proceso en el orden en el que deben ejecutarse con el fin de que se puedan
25 interpretar y determinar la acción a llevar a cabo, desde el punto de vista de los requerimientos funcionales necesarios para construir una aplicación de software.

El Procesador 150 toma los componentes funcionales almacenados en la Memoria de Base de Datos dinámica **170**, en la estructura lógica Componentes Funcionales **172** y
30 selecciona solo aquellos que son SUSTANTIVO, PERSONA, OD y VERBO. Para cada uno de los componentes funcionales mencionados, genera requerimiento global que forma parte del alcance de “el caso” realizando definiendo los requerimientos globales

para componentes SUSTANTIVO (SUST, OD, PER) y definiendo los requerimientos globales para componentes VERBO.

Requerimientos globales para componentes SUSTANTIVO

- 5 El requerimiento Global se construye creando por cada componente funcional sustantivo (SUSTANTIVO, PERSONA y OD), siempre que no sea atributo, una oración etiquetada y enumerada consecutivamente como la que se indica en la **¡Error! No se encuentra el origen de la referencia.**, utilizando las oraciones estándares definidas en la Subetapa F2.

10

Tabla 1

Componente funcional	Cómo se arma el requerimiento Global
PERSONA	<p>“Agregar, modificar, dar de baja y consultar Entidades con rol ” + PERSONA.</p> <p>Ejemplo: Agregar, modificar, dar de baja y consultar Entidades con rol ‘operador del sistema’</p>
SUSTANTIVO	<p>“Realizar Alta, Baja, Modificación y Consulta de” + SUSTANTIVO.</p> <p>Ejemplo: Realizar Alta, Baja, Modificación y Consulta de ‘registro de personas’</p>
OD	<p>“Realizar Alta, Baja, Modificación y Consulta de” + OD.</p> <p>Ejemplo: Realizar Alta, Baja, Modificación y Consulta de ‘número de documento’</p>

Requerimientos globales para componentes VERBO

- 15 El requerimiento Global se construye creando por cada componente funcional VERBO, una oración etiquetada y enumerada consecutivamente.

En este caso, la oración se genera por cada tipo de paso, a partir del VERBO y utiliza en su constitución el OD y SUSTANTIVO que lo acompañan en la pregunta Qué como se muestra en la **¡Error! No se encuentra el origen de la referencia.:**

Tabla 2

Componente funcional	Tipo paso	Cómo se arma el requerimiento Global
VERBO	VC	“Crear regla que” + VERBO + OD
VERBO	CB	“Crear regla que” + VERBO + OD
VERBO	FX	“Crear regla que” + VERBO + OD
VERBO	R	“Crear regla que” + VERBO + OD + “con” + SUSTANTIVO
VERBO	F	“Registrar” + OD

5

De diferentes pasos y a través de un mismo componente funcional se escribe un mismo requerimiento Global. En este caso, se escribe una única vez el requerimiento Global para evitar repetición y se relaciona tantas veces como corresponda a los pasos correspondientes.

10

b) Arma requerimientos funcionales detallados

En esta etapa, se escriben los requerimientos detallados de “el caso” para cada requerimiento Global, tanto los explícitos, entendiendo por explícitos aquellos que están claramente enunciados en las oraciones, como los implícitos, entendiendo por implícitos aquellos que se saben necesarios, pero no están descriptos puntualmente.

15

Para el entendimiento de la presente invención, las oraciones presentadas entre comillas corresponden a oraciones preferidas, tomadas de la Memoria de Base de Datos estática **160**, residentes en la estructura lógica Idiomas, Palabras y Reglas Morfo Sintáticas **162**. Estas oraciones, sin embargo, pueden ser cambiadas por oraciones de equivalente significado aplicables al idioma elegido para el caso, en la modalidad presentada en la Subetapa F2.

20

Por cada tipo de componente funcional, a partir de la definición de su Requerimiento Global se definen los requerimientos detallados de “el caso” utilizando las oraciones estándares definidas en la Subetapa F2.

5 Para mayor comprensión del presente invento, se debe considerar, que para generar los requerimientos detallados, se realiza la sustitución de las palabras con el criterio expuesto a continuación:

- 10 - SUSTANTIVO, hace referencia a los componentes funcionales de este tipo definidos en la Subetapa D2.
- PERSONA, hace referencia a los componentes funcionales de este tipo definidos en la Subetapa D2.
- OD, hace referencia a los componentes funcionales de este tipo definidos en la Subetapa D2.
- 15 - VERBO, hace referencia a los componentes funcionales de este tipo definidos en la Subetapa D2.
- {enumeración de atributos&}: es la lista de atributos& que describen a SUSTANTIVO, formada por aquellos SUSTANTIVOS que fueron marcados como atributos& en la Subetapa D2, más los adicionados en la Subetapa D3.
- 20 - {enumeración de atributos*}: es parte de la lista de atributos& definidos en la Subetapa D2, que se utiliza para buscar, filtrar, ordenar, agrupar o sumarizar la búsqueda o consulta.
- {enumeración de atributos& OD}: es el conjunto de atributos& de la lista de atributos& definidos en la Subetapa D2, formada por los elementos que se encuentran asociados al componente OD a través de los requerimientos detallados creados en la Subetapa G3.
- 25 - {enumeración de atributos& SUSTANTIVO}: es el conjunto de atributos& de la lista de atributos& definidos en la Subetapa D2, formada por los elementos que se encuentran asociados al componente SUSTANTIVO a través de los requerimientos detallados creados en Subetapa G3.
- 30 - {enumeración de atributos& PERSONA}: es el conjunto de atributos& de la lista de atributos& definidos en en la Subetapa D2, formada por los elementos

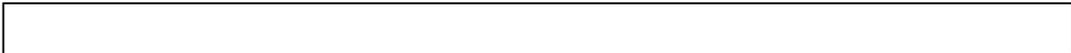
que se encuentran asociados al componente PERSONA a través de los requerimientos detallados creados en Subetapa G3.

- 5 - {enumeración de requerimientos globales de tipo VC}: es el conjunto de requerimientos globales, formada por los requerimientos globales que fueron definidos a partir del componente SUSTANTIVO o el componente OD pertenecientes a pasos de tipo VC identificados en la Subetapa E2. Esta lista contiene de 0 a n requerimientos globales. Si la lista tiene 0 requerimientos globales, no se generan los requerimientos detallados que hacen referencia a requerimientos globales de tipo VC.
- 10 - {enumeración de requerimientos globales de tipo CB}: es una lista con requerimientos globales generados a partir de pasos de tipo CB identificados en la Subetapa E3. Esta lista contiene de 0 a n requerimientos globales. Si la lista tiene 0 requerimientos globales, no se generan los requerimientos detallados que hacen referencia a requerimientos globales de tipo CB.
- 15 - {enumeración de requerimientos globales de tipo F}: es una lista con requerimientos globales generados a partir de pasos de tipo F identificados en la Subetapa E5. Esta lista contiene de 0 a n requerimientos globales. Si la lista tiene 0 requerimientos globales, no se generan los requerimientos detallados que hacen referencia a requerimientos globales de tipo F.
- 20 - {expresión de fx}: es la expresión de la fórmula que utiliza como argumentos los datos enumerados y fue declarada en la Subetapa D3.
- {mensaje de error}: es el texto del mensaje que se desea presentar en el sistema si ocurriera un error en la ejecución de la fórmula, el cual fue definido en la Subetapa D3.
- 25 - {mensaje ok}: es el texto del mensaje que se desea presentar en el sistema si el resultado de la validación o control fuese el correcto, el cual fue definido en la Subetapa D3.
- {mensaje advertencia}: es el texto del mensaje que se desea presentar en el sistema si el resultado de la validación o control NO fuese el correcto, el cual fue
30 definido en la Subetapa D3.

Define requerimientos detallados para componentes SUSTANTIVO

A partir de los componentes funcionales de tipo SUSTANTIVO, residentes en la Memoria de Base de Datos Dinámica 170, de la estructura lógica Componentes Funcionales 172, y su respectivo requerimiento Global, se generan los siguientes requerimientos Detallados:

Componente Funcional	Requerimientos Global
SUSTANTIVO	“Realizar Alta, Baja, Modificación y Consulta de” + {SUSTANTIVO}
<p><u>Requerimientos detallados:</u></p> <p>Para cada componente funcional de tipo SUSTANTIVO se escriben las siguientes oraciones que constituyen los siguientes requerimientos detallados, si el SUSTANTIVO no es marcado como atributo:</p> <ol style="list-style-type: none"> 1. OracionSTD_Crear+ SUSTANTIVO + OracionSTD_Agregar_Atributos + {enumeración de atributos&} 2. OracionSTD_Agregar_Controles+ SUSTANTIVO + {enumeración de requerimientos globales de tipo VC} 3. OracionSTD_Baja+ SUSTANTIVO + OracionSTD_complemento_control + {enumeración de requerimientos globales de tipo VC} 4. OracionSTD_Edicion + SUSTANTIVO+ “ aplicando los siguientes controles ” + {enumeración de requerimientos globales de tipo VC} 5. OracionSTD_Consulta + SUSTANTIVO+ OracionSTD_complemento_búsqueda+{enumeración de requerimientos globales de tipo CB} <p>Si el SUSTANTIVO es marcado como atributo, se definen los siguientes requerimientos:</p> <ol style="list-style-type: none"> 1. OracionSTD_crear_atributo + SUSTANTIVO 2. OracionSTD_validacion_atributo + SUSTANTIVO+ {enumeración de requerimientos globales de tipo VC} 	



Define requerimientos detallados para componentes PERSONA

A partir de los componentes funcionales de tipo PERSONA , residentes en la Memoria de Base de Datos Dinámica **170**, de la estructura lógica Componentes Funcionales **172**,
 5 y su respectivo requerimiento Global, se generan los siguientes requerimientos Detallados:

Componente Funcional	Requerimientos Global
PERSONA	"Agregar, modificar, dar de baja y consultar Entidades con rol " + PERSONA
<p><u>Requerimientos detallados:</u></p> <p>Para cada componente funcional de tipo PERSONA se escriben los siguientes requerimientos detallados:</p> <ol style="list-style-type: none"> 1. OracionSTD_Crear+ PERSONA + OracionSTD_Agregar_Atributos + {enumeración de atributos&} 2. OracionSTD_Agregar_Atributos + PERSONA+ {enumeración de requerimientos globales de tipo VC} 3. OracionSTD_Baja+ PERSONA + OracionSTD_complemento_control + {enumeración de requerimientos globales de tipo VC} 4. OracionSTD_Edicion + PERSONA + OracionSTD_complemento_control + {enumeración de requerimientos globales de tipo VC} 5. OracionSTD_permitirque +PERSONA + OracionSTD_acciones_Persona+{enumeración de requerimientos globales de tipo F} 6. OracionSTD_permitirque +PERSONA + OracionSTD_afectacion_Persona +{enumeración de requerimientos globales de tipo F} 7. OracionSTD_permitirque +{enumeración de requerimientos globales de tipo F} + OracionSTD_responsabilidad_Persona + PERSONA 8. OracionSTD_Consulta + PERSONA + OracionSTD_complemento_criteriobusqueda:+{enumeración de requerimientos globales de tipo CB} 	



Define requerimientos detallados para componentes VERBO

A partir de los componentes funcionales de tipo VERBO , residentes en la Memoria de Base de Datos Dinámica **170**, de la estructura lógica Componentes Funcionales **172**, y su respectivo requerimiento Global, se generan los siguientes requerimientos Detallados:

Componente Funcional	Requerimientos Global
VERBO (tipo de paso VC)	"Crear regla que " + VERBO + OD
<p><u>Requerimientos detallados:</u></p> <p>Para cada componente funcional de tipo VERBO, que participa en un paso de tipo VC se definen los siguientes requerimientos detallados:</p> <ol style="list-style-type: none"> 1. OracionSTD_crear_Fx+ VERBO + OD + OracionSTD_argumentos + {enumeración de atributos&} + OracionSTD_expresion + {expresión de fx} 2. OracionSTD_msj_error_Fx+ {mensaje de error} 3. OracionSTD_msj_ok_Fx + {mensaje ok} 4. OracionSTD_msj_advertencia_Fx+ {mensaje advertencia} 	
VERBO (tipo de paso CB)	"Crear regla que " + VERBO + OD
<p><u>Requerimientos detallados:</u></p> <p>Para cada componente funcional de tipo VERBO (tipo de paso CB) se definen los siguientes requerimientos detallados:</p> <ol style="list-style-type: none"> 1. OracionSTD_crear_busqueda + VERBO + OD + OracionSTD_exponer_atributos + {enumeración de atributos&} 2. OracionSTD_definir_búsqueda + OD + OracionSTD_conector_orden+ {enumeración de atributos*} 3. OracionSTD_definir_filtro+ OD + OracionSTD_conector_orden+ {enumeración de atributos*} 4. OracionSTD_definir_orden+ OD + OracionSTD_conector_orden+ 	

{enumeración de atributos*} 5. OracionSTD_definir_agrupamiento+ OD + OracionSTD_conector_orden+ {enumeración de atributos*} 6. OracionSTD_definir_total+ {enumeración de atributos*}	
VERBO (tipo de paso FX)	“Crear regla que “ + VERBO + OD
<p><u>Requerimientos detallados:</u></p> <p>Para cada componente funcional de tipo VERBO, que participa en un paso de tipo VC se definen los siguientes requerimientos detallados:</p> <ol style="list-style-type: none"> 1. OracionSTD_crear_Fx+ VERBO + OD + OracionSTD_argumentos + {enumeración de atributos&} + OracionSTD_expresion + {expresión de fx} 2. OracionSTD_msj_error_Fx+ {mensaje de error} 3. OracionSTD_msj_ok_Fx + {mensaje ok} 4. OracionSTD_msj_advertencia_Fx+ {mensaje advertencia} 	
VERBO (tipo de paso R)	“Crear regla que “ + VERBO + OD + “con” + SUSTANTIVO
<p><u>Requerimientos detallados:</u></p> <p>Para cada componente funcional de tipo VERBO (tipo de paso R) se definen los siguientes requerimientos detallados:</p> <ol style="list-style-type: none"> 1. OracionSTD_crear_regla + VERBO + OD + “con” + SUSTANTIVO + OracionSTD_condicion + {enumeración de atributos& OD} + {expresión de fx} + {enumeración de atributos& SUSTANTIVO} 2. OracionSTD_vincular +OD + OracionSTD_conector_vincular + SUSTANTIVO+ OracionSTD_complemento_control + {enumeración de requerimientos globales de tipo VC} 3. OracionSTD_desvincular+OD + “con” + SUSTANTIVO+ OracionSTD_complemento_control + {enumeración de requerimientos globales de tipo VC} 	

<p>4. OracionSTD_ consultar+OD + OracionSTD_ complemento_relacionar+ SUSTANTIVO + OracionSTD_ complemento_criteriobusqueda+{enumeración de requerimientos globales de tipo CB}</p>	
<p>VERBO (tipo de paso F)</p>	<p>“Registrar” + OD</p>
<p><u>Requerimientos detallados:</u></p> <p>Para cada componente funcional de tipo VERBO (tipo de paso F) se definen los siguientes requerimientos detallados, si OD no es marcado como atributo:</p> <p>Para cada componente funcional de tipo VERBO (tipo de paso F) se definen los siguientes requerimientos detallados:</p> <ol style="list-style-type: none"> 1. OracionSTD_permitir + VERBO + OD + OracionSTD_Agregar_Atributos + {enumeración de atributos&} 2. OracionSTD_habilitar_persona+{enumeración de PERSONA} + OracionSTD_complemento_accionpersona+ VERBO + OD 3. OracionSTD_movimientos+OD + OracionSTD_complemento_acargode+ {enumeración de PERSONA} 4. OracionSTD_control_nuevo+ OD +”: ”+ {enumeración de requerimientos globales de tipo VC} 5. OracionSTD_control_eliminar+ OD +”: ”+ {enumeración de requerimientos globales de tipo VC} 6. OracionSTD_control_modificar + OD + OracionSTD_complemento_control + {enumeración de requerimientos globales de tipo VC} 7. OracionSTD_permitir + VERBO + OD + OracionSTD_precedencia+{enumeración de requerimientos globales de tipo F} 8. OracionSTD_nuevo_movimiento+OD 9. OracionSTD_control_nuevo_movimiento+ OD + {enumeración de requerimientos globales de tipo VC} 	

10. OracionSTD_control_eliminar_movimiento+	OD	+	OracionSTD_complemento_control + {enumeración de requerimientos globales de tipo VC}
11. OracionSTD_control_modificar_movimiento		+ OD	+ OracionSTD_complemento_control + {enumeración de requerimientos globales de tipo VC}
12. OracionSTD_buscar_elemento		+ OD	+ OracionSTD_complemento_buscarelemento +{enumeración de requerimientos globales de tipo CB}

Define requerimientos detallados para componentes OD

5 A partir de los componentes funcionales de tipo OD, residentes en la Memoria de Base de Datos Dinámica **170**, de la estructura lógica Componentes Funcionales **172**, y su respectivo requerimiento Global, se generan los siguientes requerimientos Detallados:

Componente Funcional	Requerimientos Global
OD	“Realizar Alta, Baja, Modificación y Consulta de” + OD
<p><u>Requerimientos detallados:</u></p> <p>Para cada componente funcional de tipo SUSTANTIVO se escriben las siguientes oraciones que constituyen los siguientes requerimientos detallados, si el SUSTANTIVO no es marcado como atributo:</p> <ol style="list-style-type: none"> 1. OracionSTD_Crear+ SUSTANTIVO + OracionSTD_Agregar_Atributos + {enumeración de atributos&} 2. OracionSTD_Agregar_Atributos + SUSTANTIVO + {enumeración de requerimientos globales de tipo VC} 3. OracionSTD_Baja+ SUSTANTIVO + OracionSTD_complemento_control + {enumeración de requerimientos globales de tipo VC} 4. OracionSTD_Edicion + SUSTANTIVO+ OracionSTD_complemento_control + 	

{enumeración de requerimientos globales de tipo VC}

5. OracionSTD_Consulta + SUSTANTIVO+ “ aplicando las siguientes búsquedas
“+{enumeración de requerimientos globales de tipo CB}

Si el OD es marcado como atributo, se definen los siguientes requerimientos:

1. OracionSTD_crear_atributo + OD
2. OracionSTD_validacion_atributo + OD+ {enumeración de requerimientos globales de tipo VC}

Subetapa I3. Generar documentos de Diseño de Software

Según la notación gráfica definida en la Subetapa F4, el Procesador **150**, configurado como Generador de Documentos **153**, genera un archivo XML donde se almacena entre tags el nombre de cada componente de diseño y el código de la notación gráfica que le corresponde.

Para la modalidad de la invención donde se elige la notación gráfica UML, se utiliza el estándar xmi y se genera el XML que representa el diagrama de clases.

El el Procesador **150**, configurado como Generador de Documentos **153**, genera los documentos de diseño, tomando los datos almacenados en la memoria de base de datos correspondientes al diseño y exportando ese contenido en un formato XML particular, según lo define la notación gráfica elegida en en la Subetapa F4.

7. Ejemplo de una aplicación del método.

A partir de una descripción de un proceso en lenguaje natural, el sistema produce de manera automática documentos de Negocio, Documentos de análisis y Documentos de diseño de "el caso".

Los componentes del sistema son aquellos que permiten la obtención de los documentos mencionados:

1. Dispositivo de Entrada/Salida **120**: es el dispositivo mediante el cual se ingresan los textos en lenguaje natural, utilizando las siguientes configuraciones:
 - a. Interface de Usuario Idioma **121**.
 - b. Interface de Usuario Caso **122**.
 - 5 c. Interface de Usuario Formatos, Notación Gráfica y Oraciones STD **123**
 - d. Interface de Usuario de Exposición de Documentos **124**
2. CPU **110**: es el dispositivo de procesamiento del sistema 100 formado por:
 - a. Memoria General **140**: Realiza las siguientes funciones según su configuración:
 - 10 i. Diagramas **141**.
 - ii. Matriz de Oraciones **142**
 - iii. Matriz de CF **143**
 - iv. Matriz de CD **144**
 - b. Procesador **150**: Realiza las siguientes funciones según su configuración:
 - 15 i. Analizador automático de oraciones **151**
 - ii. Diseñador automático de Software **152**
 - iii. Generador de Documentos **153**
3. Memoria de Base de Datos **130**:
 - a. Memoria de Base de Datos Estática **160**:
 - 20 i. Meta-Modelo MAS **161**.
 - ii. Idiomas, Palabras y Reglas Morfo-Sintácticas **162**.
 - iii. Formatos, Notación Gráfica y Oraciones STD **163**.
 - b. Memoria de Base de Datos Dinámica **170**
 - i. Oraciones simples Preguntas **171**.
 - 25 ii. Componentes Funcionales **172**
 - iii. Componentes de Diseño **173**

Dispositivo de Entrada/Salida

La herramienta está compuesta por memorias permanentes que son capaces de producir pantallas en un monitor con zonas en las cuales el usuario puede ingresar datos o ejecutar acciones.

Interface de Usuario Caso

La interface de Usuario Caso permite al usuario ingresar los datos solicitados y ejecutar acciones para grabarlos en la memoria de base de datos.

La herramienta para funcionar necesita que el usuario ingrese datos en lenguaje natural (capas, recursos, procesos, subprocesos, momentos, pasos) según lo establece el método.

Para tal objetivo, presenta esta interface de carga que permite ingresar estos datos y relacionarlos entre sí con la secuencia establecida en el método, donde cada uno de los componentes (dimensiones, actividades temporales y oraciones simples) requieren de los siguientes datos para ser cargadas:

- 10 - Nombre
- Descripción
- Elementos dependientes relacionados

Estos datos se presentan en la interface de la **¡Error! No se encuentra el origen de la referencia.**, donde el {texto1} y {texto2} es reemplazado por el nombre del componente según corresponda, a medida que el usuario va completando los datos en la secuencia establecida:

- En primer lugar {texto1}="Capa" y {texto2}="Recurso"
- Al ejecutar la acción Editar (8.2), se presenta nuevamente la interface de la **¡Error! No se encuentra el origen de la referencia.**, solo que en esta oportunidad {texto1}="Recurso" y {texto2}="Proceso"
- 20 ○ Al ejecutar la acción Editar (8.2), se presenta nuevamente la interface de la **¡Error! No se encuentra el origen de la referencia.**, solo que en esta oportunidad {texto1}="Proceso" y {texto2}="Subproceso"
- 25 ■ Al ejecutar la acción Editar (8.2), se presenta nuevamente la interface de la **¡Error! No se encuentra el origen de la referencia.**, solo que en esta oportunidad {texto1}="Subproceso" y {texto2}="Momento"
- Al ejecutar la acción Editar (8.2), se presenta nuevamente la interface de la **¡Error! No se encuentra el origen de la referencia.**, solo que en esta oportunidad {texto1}="Momento" y {texto2}="Paso"
- 30

Una vez completado el paso, el editar uno de ellos, se presenta la interface de oración de la **¡Error! No se encuentra el origen de la referencia.**, donde el usuario deberá responder a las preguntas para lograr la oración simple completa estructurada del modo que define el método. También deberá elegir en la lista Tipo de Paso (9.2.) entre las

5 siguientes opciones:

- F: Funcional
- VC: Validación/Control
- CB: Consulta/Búsqueda
- FX: Cálculo

10 - R: Relación

En la medida que el usuario va completando las preguntas, la herramienta va completando la Oración (9.8.) concatenando el texto ingresado en las preguntas, en el orden que se exponen en la interface: 9.3; 9.4; 9.5; 9.6; 9.7.

15 Una vez concluida la tarea de carga de los datos, el usuario puede accionar de la siguiente manera:

- Grabar: graba en la memoria Base de Datos (7.3.1.) los datos cargados.
- Cancelar: descartar los datos cargados sin grabarlos en la memoria Base de Datos (7.3.1.).

20 *Visualización de Análisis*

La visualización de análisis (7.1.2.) que se presenta en el monitor consta de pantallas que presentan los datos cargados organizados de tal manera que le posibilita al usuario de la herramienta la lectura y comprensión de “el caso” orientándolo a la realización de un análisis preciso para la correcta generación de requerimientos funcionales Globales y

25 Detallados. Estos requerimientos son generados de manera automática por la herramienta solicitando confirmaciones y selecciones de algunos datos por parte del usuario.

Visualización de análisis de Palabras

Esta interface presenta al usuario las oraciones simples cargadas en forma estructurada

30 (**¡Error! No se encuentra el origen de la referencia.**). A partir de esta interface la herramienta permite ejecutar el Procesador de Análisis y el Procesador de Documentos de Análisis.

Se presenta al usuario una tabla con la totalidad de las oraciones simples cargadas (10.1.) y para cada una de ellas es posible ejecutar la acción Analizar. Esta acción llama al Procesador de Análisis quien devuelve como resultado la lista de palabras contenidas en la oración (10.2.).

5 Cada palabra está caracterizada por:

- Palabra (10.2.1): es la palabra detectada por el Procesador de Análisis.
- Pregunta (10.2.2): es la pregunta a la que pertenece la palabra dentro de la oración: Cuándo, Quién, Qué, Cómo, Dónde.
- Tipo Componente (10.3.3): es un tipo que asigna el Procesador de Análisis a cada palabra de forma automática: SUS, ADV, VERB, OD, ADJ
- 10 - En Alcance (10.3.4): es la columna que le permite al usuario de la herramienta indicar si desea que la palabra en cuestión forme parte del alcance de “el caso” a diseñar, puede elegir entre las opciones: SI / NO
- Es atributo& (10.3.5): es la columna que le permite al usuario de la herramienta
- 15 indicar si desea que la palabra en cuestión se comporte como un atributo& de una clase, puede elegir entre las opciones: SI / NO

Una vez concluida la tarea de análisis de palabras, el usuario puede accionar de la siguiente manera:

- Grabar: graba en la memoria Base de Datos (7.3.1.) los datos cargados.
- 20 - Cancelar: descartar los datos cargados sin grabarlos en la memoria Base de Datos (7.3.1.).

Desde esta interface es posible ejecutar la acción Analizar para cada una de las palabras identificadas por el Procesador de Análisis (7.5). En tal caso se ejecuta el Procesador de Documentos de Análisis (7.2.2) y luego se presenta la Visualización de análisis de

25 Requerimientos (**¡Error! No se encuentra el origen de la referencia.**).

Visualización de análisis de Requerimientos

Esta interface (**¡Error! No se encuentra el origen de la referencia.**) presenta al usuario los requerimientos Globales y Detallados generados por el Procesador de Documentos de Análisis.

30 Se presenta al usuario una tabla con la totalidad de los requerimientos Globales (11.1.1) generados y por cada uno de ellos una tabla anidada con los Requerimientos Detallados (11.1.2) que corresponden a cada Requerimiento Global.

Cada requerimiento global está caracterizado por:

- Palabra (10.2.1): es la palabra detectada por el Procesador de Análisis.
- Tipo Componente (10.3.3): es un tipo que asigna el Procesador de Análisis a cada palabra de forma automática: SUS, ADV, VERB, OD, ADJ
- 5 - Tipo Paso (9.2.): es el valor elegido por el usuario en la interface de carga.
- NroReqGlobal (11.1.3): es un número correlativo generado por la herramienta.
- Requerimiento Global (11.1.4): es la oración correspondiente al requerimiento global producida por el Procesador de Documentos de Análisis.

Cada requerimiento detallado está caracterizado por:

- 10 - NroReqDet (11.1.5): es un número correlativo generado por la herramienta.
- Req Detallado (11.1.6): es la oración correspondiente al requerimiento detallado producida por el Procesador de Documentos de Análisis.
- Atributos& (11.1.7): es la lista de atributos& que describen la Palabra, deben ser elegidos por el usuario de una lista de atributos& que se genera mediante:
 - 15 ○ La inserción manual del usuario
 - La generación automática cuando se trata de una palabra marcada por el usuario como “Es Atributo”
- Mensaje (11.1.8): es la lista de mensajes que el usuario quiere especificar para responder a las situaciones de error, acción correcta, acción incorrecta como
 - 20 ○ Inserción manual del usuario.
- Expresión (11.1.9): es la expresión a calcular que el usuario debe especificar cuando se trata de Tipo Paso = VC o FX
- ReqGlobal Asociado (10.1.10): es la lista de requerimientos globales que
 - 25 permite al usuario referenciar un ReqGlobal de esta lista y asociarlo a un requerimiento detallado. Esto sucede cuando en la descripción del alcance no se encuentra alguna particularidad de este tipo descripta (con lo cual el sistema no puede generar la relación en forma automática) y en ese caso es el usuario la agrega.
- 30 Una vez concluida la tarea de análisis de palabras, el usuario puede accionar de la siguiente manera:
 - Grabar: graba en la memoria Base de Datos (7.3.1.) los datos cargados.

- Cancelar: descartar los datos cargados sin grabarlos en la memoria Base de Datos (7.3.1.).

Visualización de Diseño

Esta interface (7.1.3.) (**¡Error! No se encuentra el origen de la referencia.**) presenta al usuario las palabras detectadas por el Procesador de Análisis, para poder diseñar “el caso” a partir de los resultados del análisis.

Se presenta al usuario una tabla con la totalidad de las palabras analizadas, y por cada una de ellas una tabla anidada con las palabras relacionadas a partir de los Requerimientos Asociados (12.1.1).

10 Cada palabra está caracterizada por los siguientes datos provenientes de la fase de análisis:

- Palabra (10.2.1): palabra que forma parte del alcance a diseñar.
- Pregunta (10.2.2): es la pregunta a la que pertenece la palabra dentro de la oración: Cuándo, Quién, Qué, Cómo, Dónde.
- 15 - Tipo Componente (10.3.3): es el tipo asignado por el Procesador de Análisis a cada palabra: SUS, ADV, VERB, OD, ADJ
- Tipo Paso (9.2.): es el valor elegido por el usuario en la interface de carga.
- Es atributo& (10.3.5): característica que indicó el usuario en fase de análisis, puede ser SI / NO.

20 Desde esta interface es posible ejecutar la acción Diseñar para cada una de las palabras listadas. En tal caso se ejecuta el Procesador de Diseño (7.6) y luego se presenta la visualización de diseño (12.2) con la lista de clases generadas para la palabra y sus relacionadas.

Cada una de las clases diseñadas está caracterizada por:

- 25 - Clase (12.2.1): Nombre de la clase diseñada por el Procesador de Diseño.
- TipoClase (12.2.2): Puede ser R o NoR.
- Atributos& (12.2.3): Es la lista de atributos& asignados a la clase. En todos los casos pertenecen a la lista de atributos& declarada en fase de análisis.
- Métodos (12.2.4): Es la lista de métodos asignados a la clase. Estos métodos son
- 30 generados por el Procesador de Diseño.

Una vez diseñadas las clases es posible ejecutar la acción ver diagrama (12.1.3) que presentará en pantalla el diagrama de clases, similar al del ejemplo de la **¡Error! No se encuentra el origen de la referencia.**

Una vez concluida la tarea de análisis de palabras, el usuario puede accionar de la siguiente manera:

- Grabar: graba en la memoria Base de Datos (7.3.1.) los datos cargados.
- Cancelar: descartar los datos cargados sin grabarlos en la memoria Base de Datos (7.3.1.).

Memoria General

10 Procesador

Para producir en forma automática documentos de análisis y diseño de “el caso”, la herramienta provee un procesador con tres funcionalidades: procesamiento de análisis que ejecuta la acción de análisis, procesamiento de diseño que ejecuta la acción de diseño y procesamiento de documentos para producir los documentos resultantes.

15 *Analizador automático de Oraciones*

El Procesador de Análisis toma el conjunto de oraciones simples y completas cargadas por el usuario en la memoria Base de Datos. A partir de dichas oraciones, ejecuta tres algoritmos:

- 20 - El algoritmo (**¡Error! No se encuentra el origen de la referencia.**) que identifica las palabras relevantes para la construcción de “el caso”, método constitutivo del presente invento descrito en 0.
- El algoritmo (**¡Error! No se encuentra el origen de la referencia.**) que genera los requerimientos globales de “el caso” en forma automática, método constitutivo del presente invento descrito en **¡Error! No se encuentra el origen de la referencia.**
- 25 - El algoritmo (**¡Error! No se encuentra el origen de la referencia.**) que genera los requerimientos detallados de “el caso” en forma automática, método constitutivo del presente invento descrito en **¡Error! No se encuentra el origen de la referencia.**

Algoritmo de palabras

Este algoritmo (**¡Error! No se encuentra el origen de la referencia.**) ejecuta la rutina descripta para cada una de las oraciones existentes y requiere que en la base de datos se encuentren cargadas dos listas:

- 5 - PAG={palabras agrupadoras} Es el conjunto de palabras que determinan una agrupación, por ejemplo: la palabra “de” es una palabra agrupadora cuando actúa en “lista de precio”
- PEX={palabras a excluir} Es el conjunto de palabras que se desea excluir del análisis. Normalmente está formado por las preposiciones, conjunciones y
- 10 artículos.

Una vez ejecutado el algoritmo contamos con todas las palabras convertidas en componentes funcionales de “el caso”, calificadas por tipo de componente y relacionadas entre sí a través de la pregunta.

El tipo de componente describe:

- 15 - SUS: SUSTANTIVO
- PER: SUSTANTIVO PERSONA
- ADV: CONSTRUCCIÓN ADVERBIAL
- VERB: VERBO
- OD: SUSTANTIVO OBJETO DIRECTO
- 20 - ADJ: ADJETIVO

Algoritmo de requerimientos globales

Este algoritmo (**¡Error! No se encuentra el origen de la referencia.**) ejecuta la rutina descripta para cada una de las palabras identificadas en el alcance y requiere que en la

25 base de datos se encuentre cargada la siguiente lista:

- reqgl_abm_Sust: es la oración que se utilizará para componer el requerimiento global cada vez que se identifique un componente SUS en el alcance. En el método se sugiere algo similar a: "Realizar Alta, Baja, Modificación y Consulta de " + {PalabraSustantivo}
- 30 - reqgl_abm_Pers: es la oración que se utilizará para componer el requerimiento global cada vez que se identifique un componente PER en el alcance. En el método se sugiere algo similar a: "Agregar, modificar, dar de baja y consultar Entidades con rol " + {PalabraPersona}

- reqgl_verboF: es la oración que se utilizará para componer el requerimiento global cada vez que se identifique un componente VERB vinculado a un tipo de paso F en el alcance. En el método se sugiere algo similar a: “Registrar” + {palabraVerbo} + {palabrOD}
- 5 - reqgl_verboR: es la oración que se utilizará para componer el requerimiento global cada vez que se identifique un componente VERB vinculado a un tipo de paso R en el alcance. En el método se sugiere algo similar a: “Crear regla que “ +{palabraVerbo} + {palabraOD} +” con “+ {palabraSustantivo}
- reqgl_verbo: es la oración que se utilizará para componer el requerimiento global cada vez que se identifique un componente VERB en el alcance, vinculado a un tipo de paso que no sea F o R . En el método se sugiere algo similar a: “Crear regla que “ +{palabraVerbo} + {palabraOD}

10 Una vez ejecutado el algoritmo contamos con todos los requerimientos globales de ”el caso” almacenados en la base de datos.

15 *Algoritmo de requerimientos detallados*

Este algoritmo (**¡Error! No se encuentra el origen de la referencia.**) ejecuta la rutina descrita para cada una de las palabras identificadas en el alcance y requiere que en la base de datos se encuentren cargadas las oraciones estándares en el idioma elegido para la descripción de “el caso”.

20 *Procesamiento de Diseño*

El Procesador de Diseño toma el conjunto de componentes funcionales y los complementos que fueron agregados en la fase de análisis:

- Atributos
- Mensajes
- 25 - Expresiones
- Reqs Asociados

Todos ellos relacionados con los componentes funcionales que forman parte del alcance de “el caso”.

Se ejecuta el algoritmo de la **¡Error! No se encuentra el origen de la referencia.** que
30 implementa la 0 y genera automáticamente las clases y relaciones que constituyen el diagrama de diseño de ”el caso” resultante.

Procesamiento de Documentos

La herramienta está compuesta por memorias permanentes que son capaces de producir de manera automática documentos que se presentan en un monitor con zonas en las cuales el usuario puede observar datos provenientes de las memorias de carga.

5 *Procesamiento de Documentos de Negocio*

El Generador de Documentos de Negocio toma los datos cargados en la interface de Carga, que residen en la Base de Datos y aplica el siguiente algoritmo para generar el documento de Negocio:

- 10 a) Toma la totalidad de los datos cargados almacenados en el orden indicado por el método accediendo a la base de datos con las siguientes columnas y tantas filas como pasos haya cargado el usuario: Capa, Recurso, Proceso, Subproceso, Momento, Paso, Cuándo, Quién, Qué, Cómo, Dónde, Oración.
- b) Concatena Capa&Recurso&Proceso en la variable Título (17.1)
- c) Concatena Subproceso&Momento en la variable Subtitulo (17.2)
- 15 d) Presenta cada paso con un nivel de sangría mayor (17.3)
- e) Presenta cada oración con un nivel de sangría mayor (17.4)

Al documento resultante se lo denomina **Documento de Negocio** (**¡Error! No se encuentra el origen de la referencia.**) y la herramienta permite almacenarlo en Archivo de Documentos (7.3.1), imprimirlo en dispositivos de impresión (impresoras, 20 ploters, ...) y presentarlo en dispositivos de visualización (monitores, pantallas, proyectores, ..)

Procesamiento de Documentos de Análisis

El Procesador de Documentos de Análisis toma la totalidad de los Datos Cargados y Requerimientos generados, tanto globales como detallados y aplica el siguiente 25 algoritmo para generar el documento de Análisis:

- a) Utiliza el mismo algoritmo que en el documento de Negocio para generar Título (17.1), Subtitulo (17.2) y paso (17.3)
- b) Presenta cada Requerimiento Global asociado al paso con un nivel de sangría mayor que la sangría del paso anterior (18.1)
- 30 c) Presenta cada Requerimiento Detallado asociado al Requerimiento Global con un nivel de sangría mayor que la sangría del Requerimiento Global anterior (18.2)

- Al documento resultante se lo denomina **Documento de Análisis** (**¡Error! No se encuentra el origen de la referencia.**) y la herramienta permite y la herramienta permite almacenarlo en Archivo de Documentos (7.3.1), imprimirlo en dispositivos de impresión (impresoras, ploters, ...) y presentarlo en dispositivos de visualización
- 5 (monitores, pantallas, proyectores, ..) (7.3.1)

Procesamiento de Documentos de Diseño

El Procesador de Documentos de Diseño toma los datos de las clases identificadas por el Procesador de Diseño y genera un archivo XML según el estándar xmi para lectura de diagramas UML.

- 10 Al documento resultante se lo denomina **Documento de Diseño de Clases** y se puede observar un ejemplo en la **¡Error! No se encuentra el origen de la referencia..**
Este documento resultante se almacena en Archivo de Documentos (7.3.1)

Memoria de Base de Datos

- La herramienta está compuesta por memorias permanentes que son capaces de
- 15 almacenar los resultados producidos por la herramienta:

Base de Datos

La base de datos en la cual se almacenan los datos de carga, análisis y diseño producidos por la herramienta, administrada por un motor de base de datos.

Archivo de Documentos

- 20 La base de datos en la cual se almacenan los documentos generados es una estructura establecida en un disco rígido administrada por el file server del sistema operativo.

REIVINDICACIONES

1. Un método implementado por computador para generar documentos de
5 Arquitectura Funcional a partir de frases expresadas en lenguaje natural para describir un caso, que son almacenadas en una memoria de base de datos **130**, caracterizado por las etapas:
- A. cargar a través de un Dispositivo de Entrada/Salida **120** el Meta-Modelo MAS en la Memoria de Base de Datos **130**.
- 10 B. ingresar a través de la Interface de Usuario Idioma **121**, del Dispositivo de Entrada/Salida **120** los idiomas elegibles, las características sintácticas y gramaticales de cada uno de los idiomas elegibles y almacenar estos datos en la Memoria de Base de Datos **130**;
- C. ingresar a través de la Interface de Usuario Caso **122**, del dispositivo de
15 visualización **120** la descripción de “el caso” como texto en lenguaje natural, en uno de los idiomas escogidos de la lista de idiomas elegibles de la Etapa B y luego almacenarla en la Memoria de Base de Datos **130**;
- D. identificar las oraciones simples del caso, correspondientes a los pasos de la Etapa C y cargarlas a través de la Interface de Usuario Caso **122**, del Dispositivo de
20 Entrada/Salida **120** para luego almacenarlas en la Memoria de Base de Datos **130**;
- E. identificar automáticamente los componentes funcionales de las oraciones simples de la Etapa D, con un Procesador **150** configurado como Analizador automático de Oraciones **151** y almacenarlos en la Memoria de Base de Datos **130**;
- 25 F. generar automáticamente los documentos de arquitectura funcional a partir de los componentes funcionales de la Etapa E con un Procesador **150** configurado como Generador de Documentos **153**, aplicando la notación gráfica definida específicamente para este tipo de Documentos.
- 30
2. El método de acuerdo a la reivindicación 1, caracterizado porque la etapa A comprende las subetapas:

5 A1. Almacenar Meta-Modelo MAS en la Memoria de Base de Datos **130**, el cual define la lógica que el Procesador 150 utilizará para generar los documentos de análisis y diseño. Esta lógica implica la generación automática de Palabras Factor, Palabras Factible y Palabras Factibilidad a partir de los verbos transitivos.

10 A2. Almacenar lógica MAS para reglas Morfo-Sintácticas en la Memoria de Base de Datos **130**, la cual establece que toda oración en lenguaje natural se puede estructurar a partir de a) la lógica de preguntas: define a qué pregunta (Cuándo, Quién, Qué, Cómo, Dónde) pertenece la palabra dentro de una oración simple, b) la lógica de tipo de paso: define qué tipo de verbo compone la pregunta Qué (FX, VC, CB, R, F) y c) la lógica de palabras: define de qué tipo de palabra se trata.

15

3. El método de acuerdo a la reivindicación 1, caracterizado porque la etapa B comprende las subetapas:

20 B1. ingresar desde el dispositivo de Entrada/Salida **120** la secuencia de adjetivación que utiliza el idioma elegible declarado y almacenarla en la Memoria de base de datos **130**;

25 B2. ingresar desde el dispositivo de Entrada/Salida **120** las palabras especiales del idioma elegible que pueden hallarse en el texto de las oraciones simples de la etapa D y se desean excluir de la lista de componentes funcionales, y almacenarlas en la Memoria de base de Datos **130**;

30 B3. ingresar desde el dispositivo de Entrada/Salida **120** las palabras agrupadoras del idioma elegible que se utilizan como nexo de componentes funcionales que constan de más de una palabra y pueden hallarse en el texto de las oraciones simples de la etapa D, y almacenarlas en la Memoria de base de Datos **130**;

B4. ingresar desde un dispositivo de Entrada/Salida **120** las terminaciones verbales de los verbos en infinitivo del idioma elegible que pueden hallarse en el texto de las oraciones completas de la etapa D, y almacenarlas en la Memoria de base de Datos **130**.

5

4. El método de acuerdo a la reivindicación 1, caracterizado porque la etapa C comprende las subetapas:

C1. elegir el idioma en el cual se describe el caso, de los idiomas elegibles definidos en la Etapa B, en el dispositivo de entrada/salida **120**, configurado como Interface de Usuario del Caso **122**;

10

C3. identificar los Pasos del caso aplicando la lógica de Tipo de paso definida en la Subetapa A2 y almacenarlos en la Memoria de base de Datos **130**.

15

5. El método de acuerdo a la reivindicación 1, caracterizado porque la etapa C comprende identificar los componentes que estructuran el contexto de "el caso", capas, recursos, procesos, subprocesos y momentos, y almacenarlos en la Memoria de Base de Datos **130**;

20

6. El método de acuerdo a la reivindicación 1, caracterizado porque la etapa D comprende las subetapas:

D1. responder a las preguntas Quién, Qué, Dónde, Cómo y Cuándo por cada uno de los pasos de "el caso" identificados en la etapa B y almacenar como texto las respuestas en una memoria de base de datos;

25

D2. concatenar el texto de las respuestas de cada paso en una oración simple y almacenar como texto de dicha oración en una memoria de base de datos.

30

7. El método de acuerdo a la reivindicación 1, caracterizado porque la etapa E comprende descomponer la oración simple en palabras, caracterizarlas según el idioma elegido e identificar los componentes funcionales de la oración simple y almacenarlos en la Memoria de Base de Datos **130**;
8. El método de acuerdo a la reivindicación 1, caracterizado porque en la etapa E un usuario agrega componentes funcionales que no se encuentran en la lista de componentes funcionales identificados de manera automática a partir de oraciones simples.
9. El método de acuerdo a la reivindicación 1, caracterizado porque en la etapa E un usuario agrega las expresiones de las fórmulas que describen el comportamiento matemático de “el caso” para todos los pasos de tipo VC y FX.
10. El método de acuerdo a la reivindicación 1, caracterizado porque en la etapa E se aplican reglas Morfo-Sintácticas definidas en la Subetapa A2, a partir de las cuales el Procesador **150** excluye las palabras específicas declaradas en la Regla y crea los componentes funcionales nuevos para los casos de verbos transitivos: un componente funcional llamado como el verbo, con el sufijo “OR” y un componente funcional llamado como el verbo, con el sufijo “BLE”.
11. El método de acuerdo a la reivindicación 1, caracterizado porque la etapa F comprende las subetapas:
- F1. aplicar reglas Morfo-Sintácticas definidas en la Subetapa A2, a partir de las cuales el Procesador **150** excluye de los componentes funcionales las palabras específicas declaradas en la Regla y crea los componentes funcionales nuevos para los casos de verbos transitivos: un componente funcional llamado como el verbo, con el sufijo “OR” y un componente funcional llamado como el verbo, con el sufijo “BLE”.

5 F2. definir la notación gráfica a utilizar en los diagramas de Arquitectura Funcional, mediante el dispositivo de Entrada/Salida **120**, Interface de Usuario Formatos, Notación Gráfica y Oraciones STD **123**, indicando para cada componente funcional, según el tipo de palabra que le corresponde, el componente gráfico a utilizar en el diagrama y almacenarlos en la Memoria de Base de Datos **130**.

10 F3. Generar documentos de Arquitectura funcional con el Procesador **150**, configurado como Generador de Documentos **153**, a partir de los componentes funcionales almacenados en la Memoria de Base de Datos **130**, utilizando la notación definida en la Subetapa F2 y visualizarlos en el Dipositivo de Entrada/Salida **120**;

15 12. Un método implementado por computador para generar documentos de especificación de diseño de software a partir de frases expresadas en lenguaje natural para describir un caso, que son almacenadas en una memoria de base de datos **130**, caracterizado por las etapas:

A. cargar a través de un Dispositivo de Entrada/Salida **120** el Meta-Modelo MAS en la Memoria de Base de Datos **130**.

20 B. ingresar a través de la Interface de Usuario Idioma **121**, del Dispositivo de Entrada/Salida **120** los idiomas elegibles, las características sintácticas y gramaticales de cada uno de los idiomas elegibles y almacenar estos datos en la Memoria de Base de Datos **130**;

25 C. ingresar a través de la Interface de Usuario Caso **122**, del dispositivo de visualización **120** la descripción de “el caso” como texto en lenguaje natural, en uno de los idiomas escogidos de la lista de idiomas elegibles de la Etapa B y luego almacenarla en la Memoria de Base de Datos **130**;

30 D. identificar las oraciones simples del caso, correspondientes a los pasos de la Etapa C y cargarlas a través de la Interface de Usuario Caso **122**, del Dispositivo de Entrada/Salida **120** para luego almacenarlas en la Memoria de Base de Datos **130**;

- E. identificar automáticamente los componentes funcionales de las oraciones simples de la Etapa D, con un Procesador **150** configurado como Analizador automático de Oraciones **151** y almacenarlos en la Memoria de Base de Datos **130**;
- 5 F. generar automáticamente los documentos de arquitectura funcional a partir de los componentes funcionales de la Etapa E con un Procesador **150** configurado como Generador de Documentos **153**, aplicando la notación gráfica definida específicamente para este tipo de Documentos.
- G. identificar automáticamente los componentes de diseño a partir de los componentes funcionales de la Etapa E, con un Procesador **150** configurado como Diseñador automático de Software **152** y almacenarlos en la Memoria de Base de Datos **130**;
- 10 H. ingresar a través de la Interface de Usuario Formatos, Notación Gráfica y Oraciones STD **123**, del dispositivo de visualización **120** los formatos de salida para documentos de negocio, análisis y diseño, los parámetros de oraciones estándares para requerimientos y la notación gráfica para diagramas de diseño y almacenarlos en la Memoria de Base de Datos **130**;
- I. generar automáticamente los documentos de negocio a partir de las oraciones simples de la Etapa D, los documentos de análisis a partir de los componentes funcionales de la Etapa E y los documentos de diseño a partir de los componentes de diseño de la Etapa F, con un Procesador **150** configurado como Generador de Documentos **153**, aplicando los Formatos, Notaciones gráficas y Oraciones STD definidas en la Etapa G;
- 20
13. El método de acuerdo a la reivindicación 12, caracterizado porque la etapa G
- 25 comprende las subetapas:
- G1. seleccionar y crear grupos de componentes funcionales, sus clases y relaciones de herencia y almacenarlos en la Memoria de Base de Datos **130**;
- 30 G2. crear clases fórmula a partir de pasos tipo FX y VC y almacenarlas en la Memoria de Base de Datos **130**;

- G3. crear clases dominio a partir de pasos tipo CB y almacenarlas en la Memoria de Base de Datos **130**;
- 5 G4. crear clases relación a partir de pasos tipo R y almacenarlas en la Memoria de Base de Datos **130**;
- G5. crear clases operación a partir de pasos tipo F y almacenarlas en la Memoria de Base de Datos **130**;
- 10 14. El método de acuerdo a la reivindicación 12, caracterizado porque en la etapa E un usuario un usuario agrega mensajes de error, de aprobación y advertencia.
- 15 15. El método de acuerdo a la reivindicación 12, caracterizado porque la etapa I comprende las subetapas:
- 15 I1. generar documentos de Negocio
- I2. generar documentos de Análisis
- I3. generar documentos de Diseño de Software
- 20 16. El método de acuerdo a la reivindicación 15, caracterizado porque la subetapa I1 genera los documentos de negocio mediante las actividades:
- a) ordenar dimensiones, actividades temporales y oraciones simples;
- b) unir en forma consecutiva todos los componentes previos ordenados; y
- c) visualiza los documentos de negocio en el dispositivo de Entrada/Salida
- 120.**
- 25 17. El método de acuerdo a la reivindicación 15, caracterizado porque la subetapa I2 genera los documentos de análisis mediante las actividades:
- a) armar requerimientos funcionales globales definiendo requerimientos globales para componentes SUSTANTIVO y definir requerimientos globales para componentes VERBO
- 30 b) armar requerimientos funcionales detallados definiendo requerimientos detallados para componentes SUSTANTIVO, PERSONA, VERBO, y OD; y

c) visualiza los documentos de análisis en el dispositivo de Entrada/Salida **120**.

18. El método de acuerdo a la reivindicación 15, caracterizado porque la subetapa I3 genera los documentos de diseño y los visualiza en el dispositivo de Entrada/Salida **120**.

19. El método de acuerdo a la reivindicación 12, caracterizado porque en la etapa H un usuario puede componer en plantillas la estructura del texto de los documentos de análisis funcional de “el caso”, utilizando el dispositivo de Entrada/Salida **120**, en la Interface de Usuario Formatos , Notación Gráfica y Oraciones STD **123**, para luego almacenarlas en la Memoria de Base de Datos **130**.

20. El método de acuerdo a la reivindicación 12, caracterizado porque en la etapa I un usuario almacenan los componentes de diseño de “el caso” en formato de lenguaje de marcado extensible (xmi).

21. Un sistema para generar documentos de arquitectura funcional y documentos de negocio, análisis y especificación de diseño de software a partir de descripciones expresadas en frases en lenguaje natural para describir un caso, que son almacenadas en una base de datos, que comprende:

un dispositivo de entrada/salida **120** configurado como Interface de Usuario Idioma **121**, Interface de Usuario Caso **122**, Interface de Usuario Formatos , Notación Gráfica y Oraciones STD **123** e Interface de Usuario de Exposición de Documentos **124** para ingresar los idiomas elegibles y su estructura, para ingresar la descripción de “el caso” en lenguaje natural, para ingresar los formatos y notaciones gráficas para generación de documentos y para visualizar los documentos de arquitectura funcional y documentos de negocio, análisis y especificación de diseño de software;

una memoria general **140** en comunicación con el dispositivo de entrada/salida **120**, que interactúa con el procesador **150**, configurada para almacenar de manera volátil la descripción del caso, los componentes funcionales, los componentes de diseño

y los documentos de arquitectura funcional y documentos de negocio, análisis y especificación de diseño de software.

un procesador **150** configurado para recibir de un usuario al menos una descripción en lenguaje natural, en el que dicha descripción incluye el caso e identifica
5 los pasos del caso;

dicho procesador configurado como analizador automático de oraciones **151** para crear oraciones simples correspondientes a los pasos del caso y analizarlas para generar automáticamente componentes funcionales a partir de las oraciones simples;

dicho procesador configurado como diseñador automático de software **152** para
10 identificar componentes de diseño del caso a partir de los componentes funcionales; y,

dicho procesador configurado como generador de documentos **153** para generar documentos de arquitectura funcional y documentos de negocio, análisis y especificación de diseño de software, a partir de los componentes funcionales y componentes de diseño;

15 una memoria de base de datos **130** en comunicación con el procesador **150**, configurada para almacenar en forma estática **160** un Meta-modelo MAS **161**, Idiomas, Palabras y Reglas Morfo Sintáticas **162** y Formatos, Notación Gráfica y Oraciones STD **163** y también configurada para almacenar en forma dinámica **170**, Oraciones simples y Preguntas **171**, Componentes Funcionales **172** y Componentes de Diseño **173**.

20

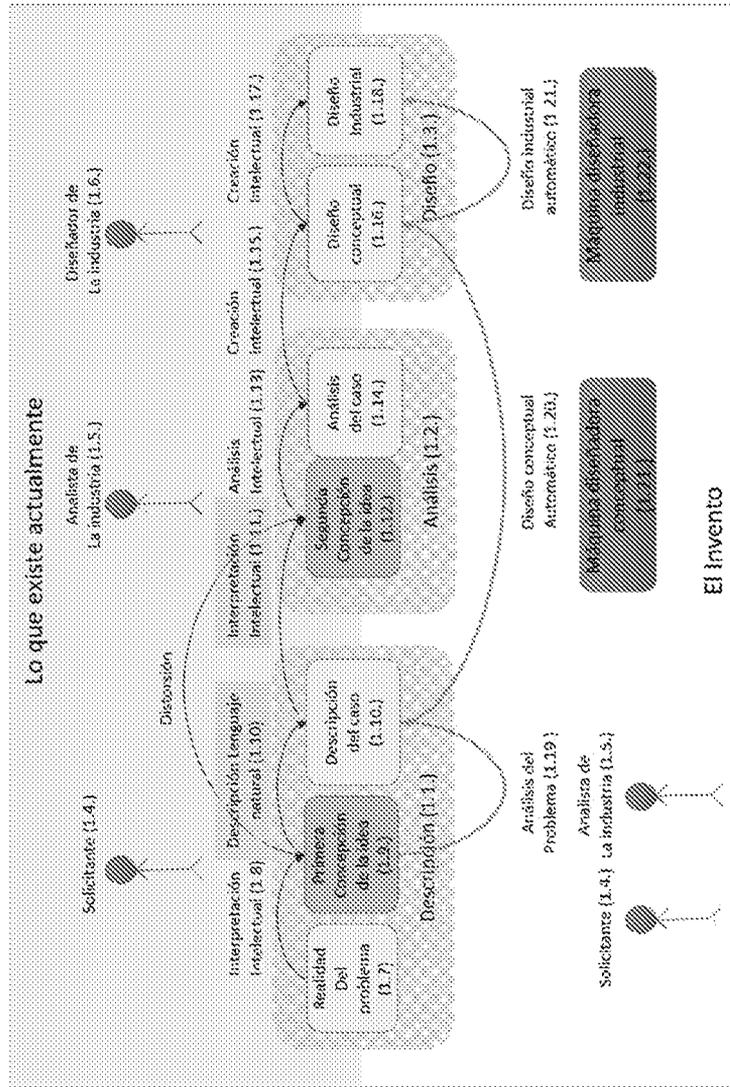


FIG. I

FIG. II

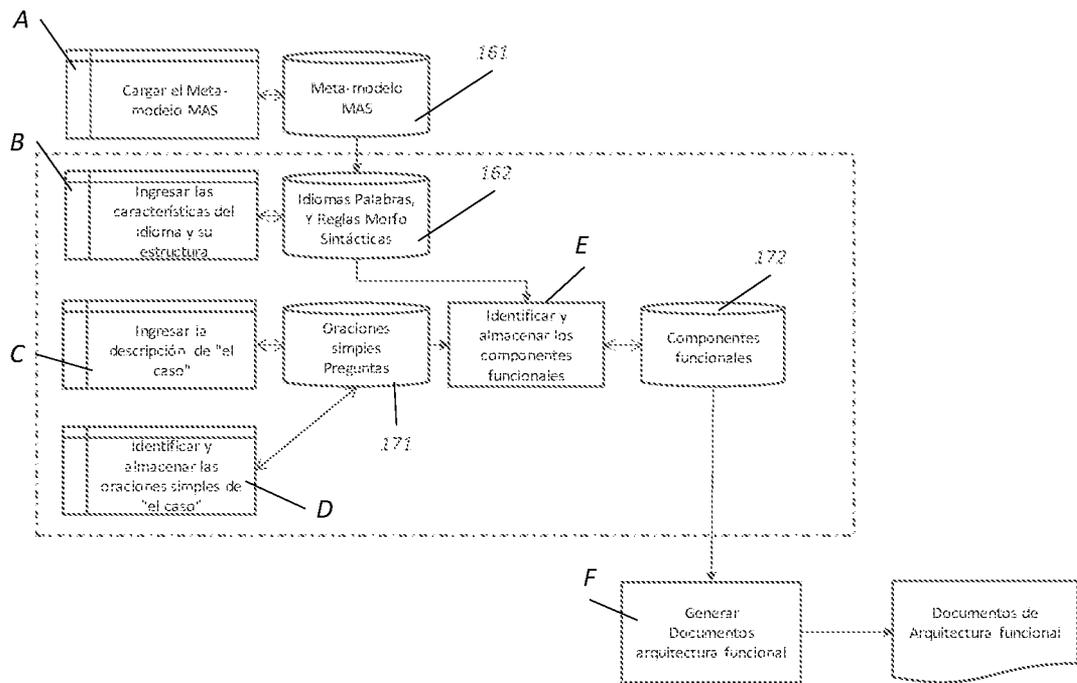


FIG. 1

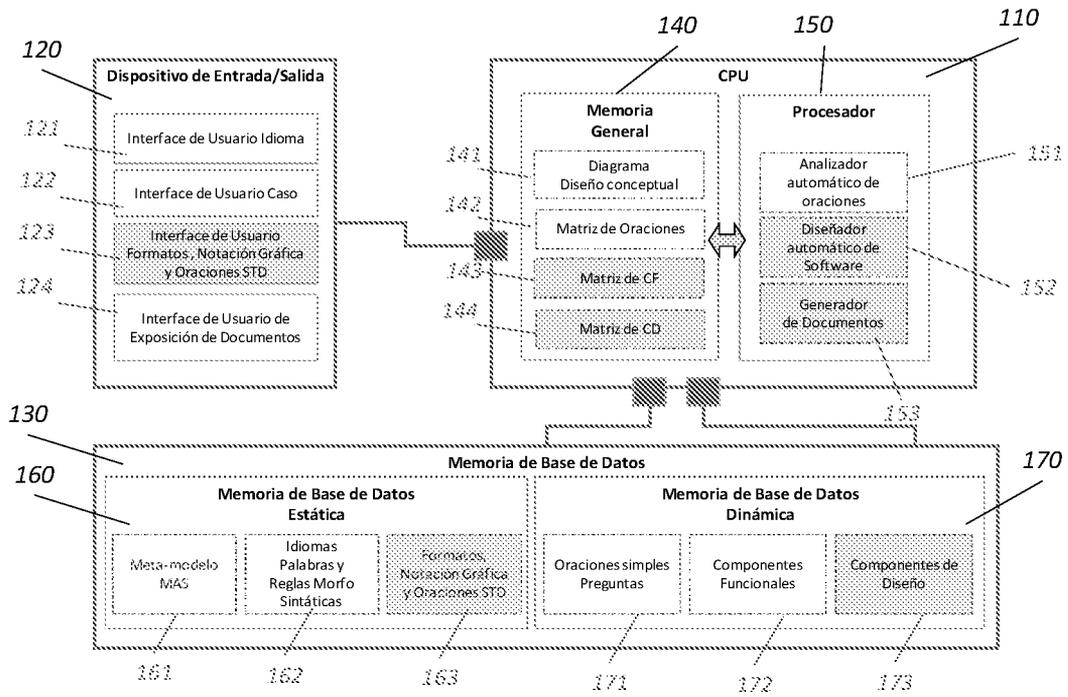


FIG. 2

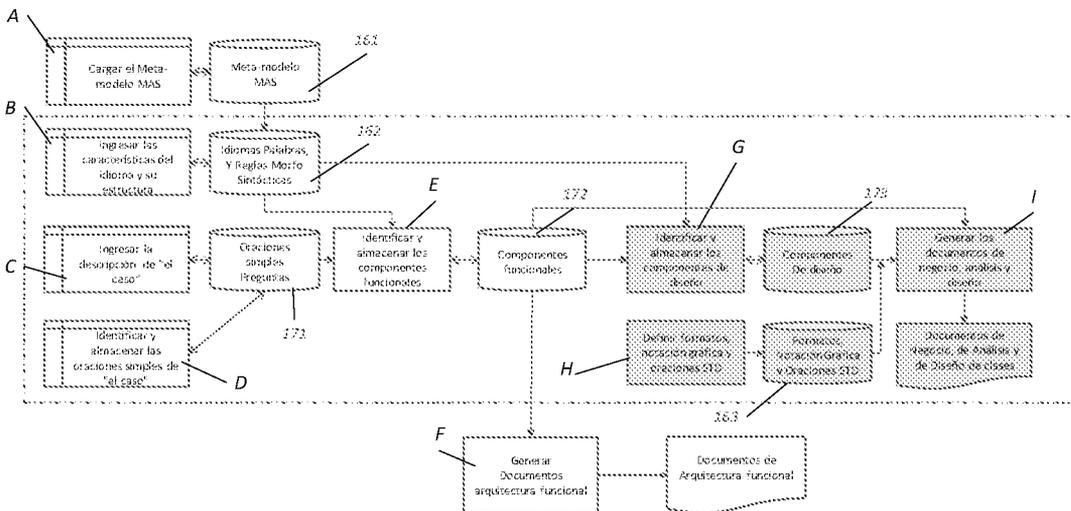


FIG. 2A

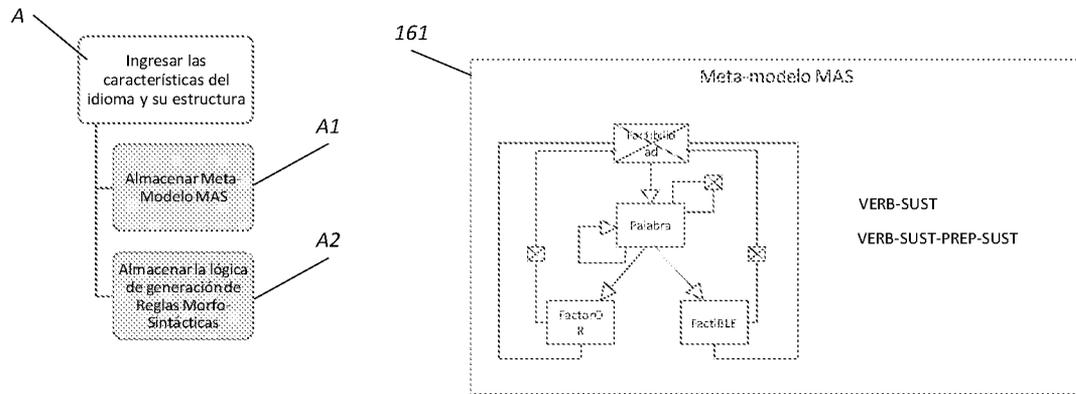


FIG. 2A1

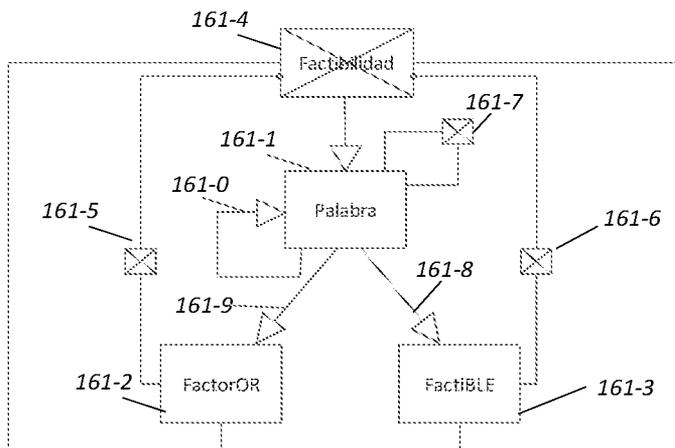


FIG. 2A2

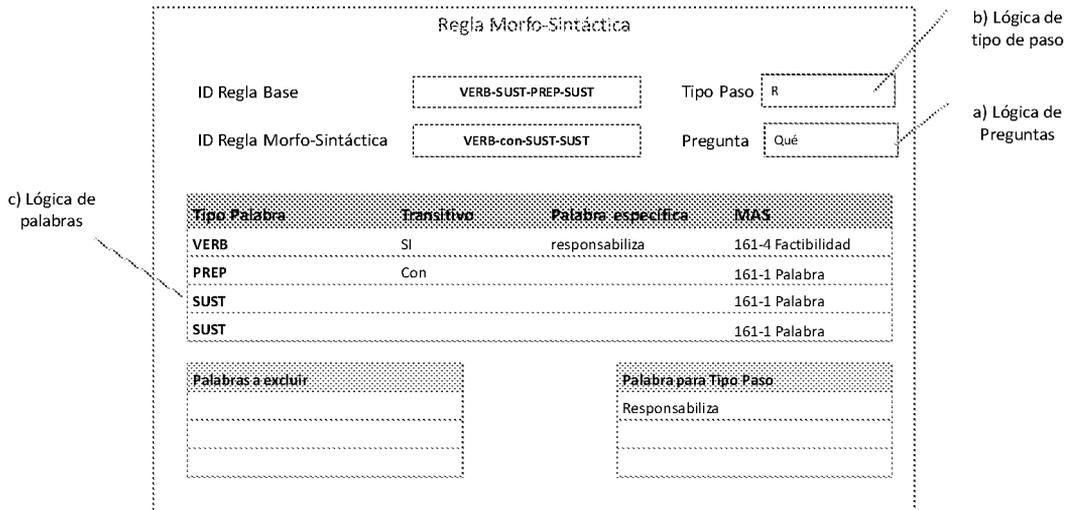
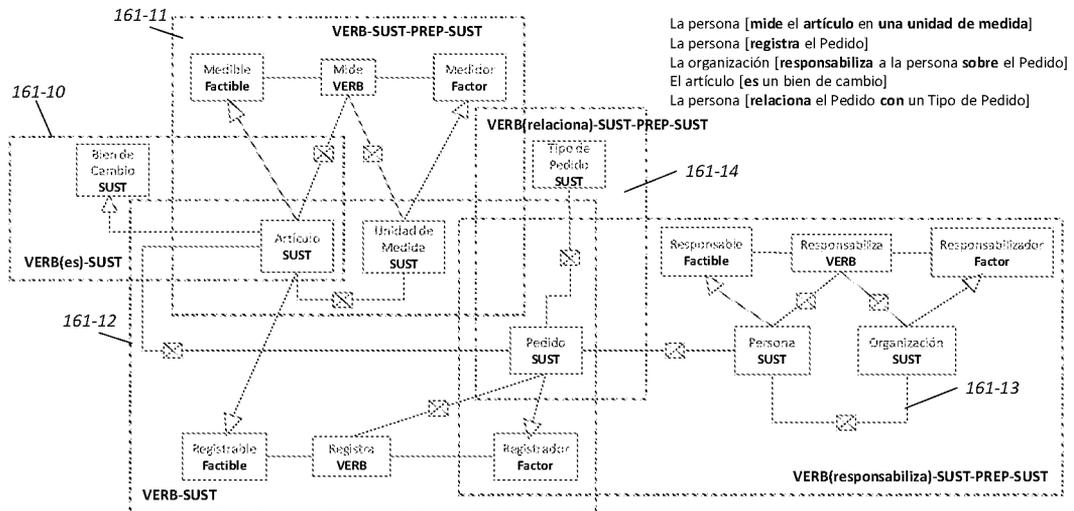


FIG. 2A3



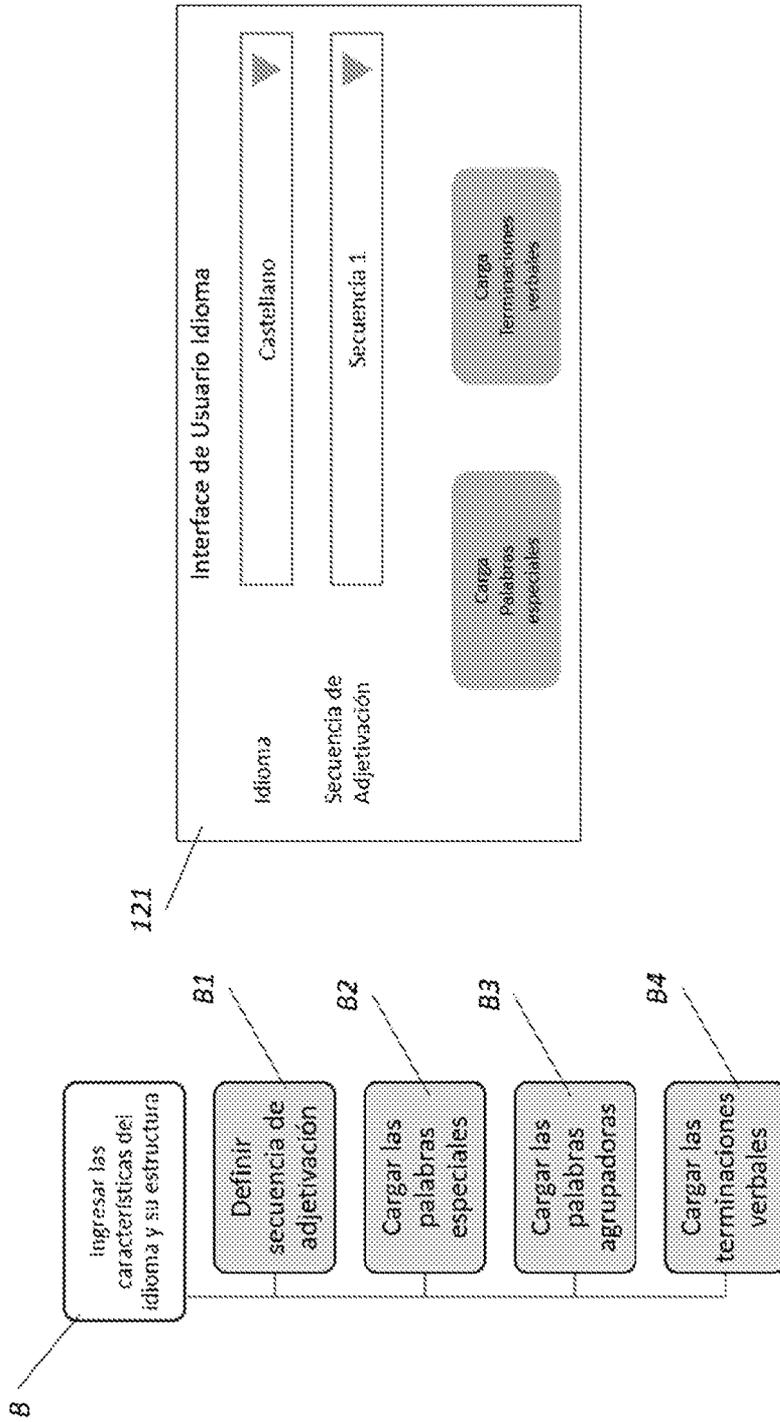


FIG. 2B

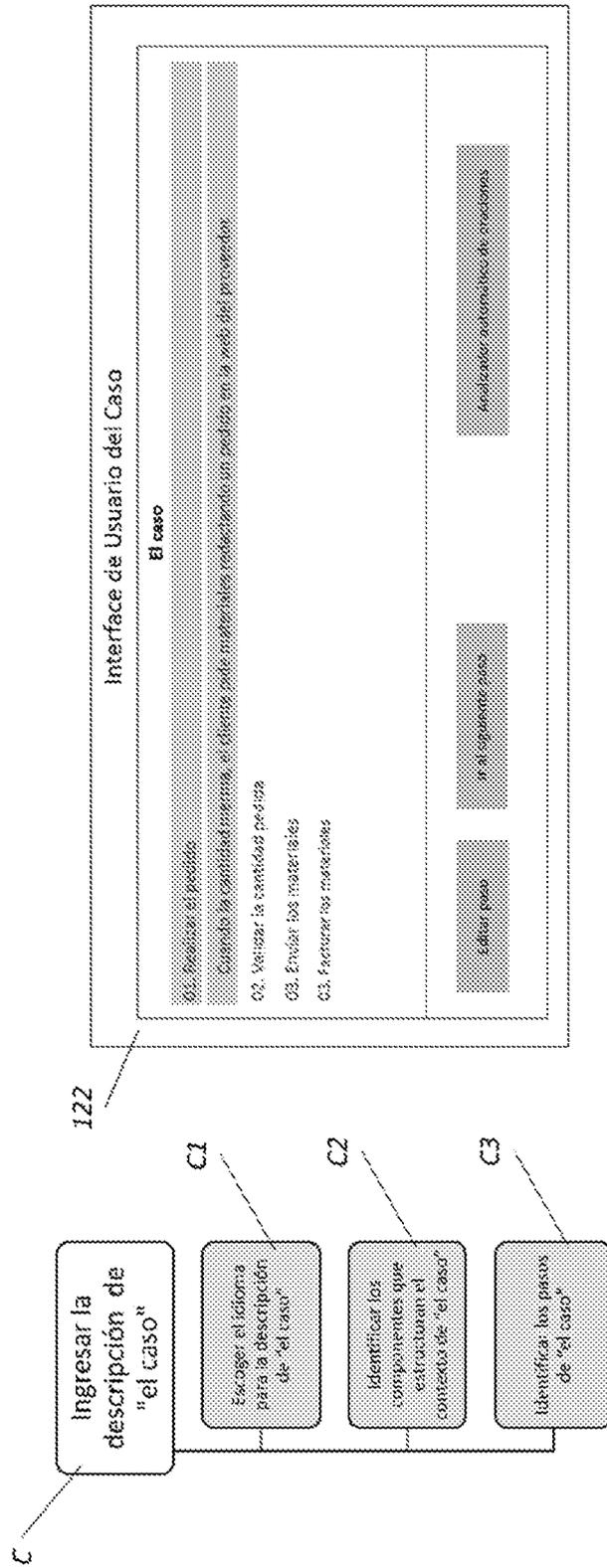


FIG. 2C

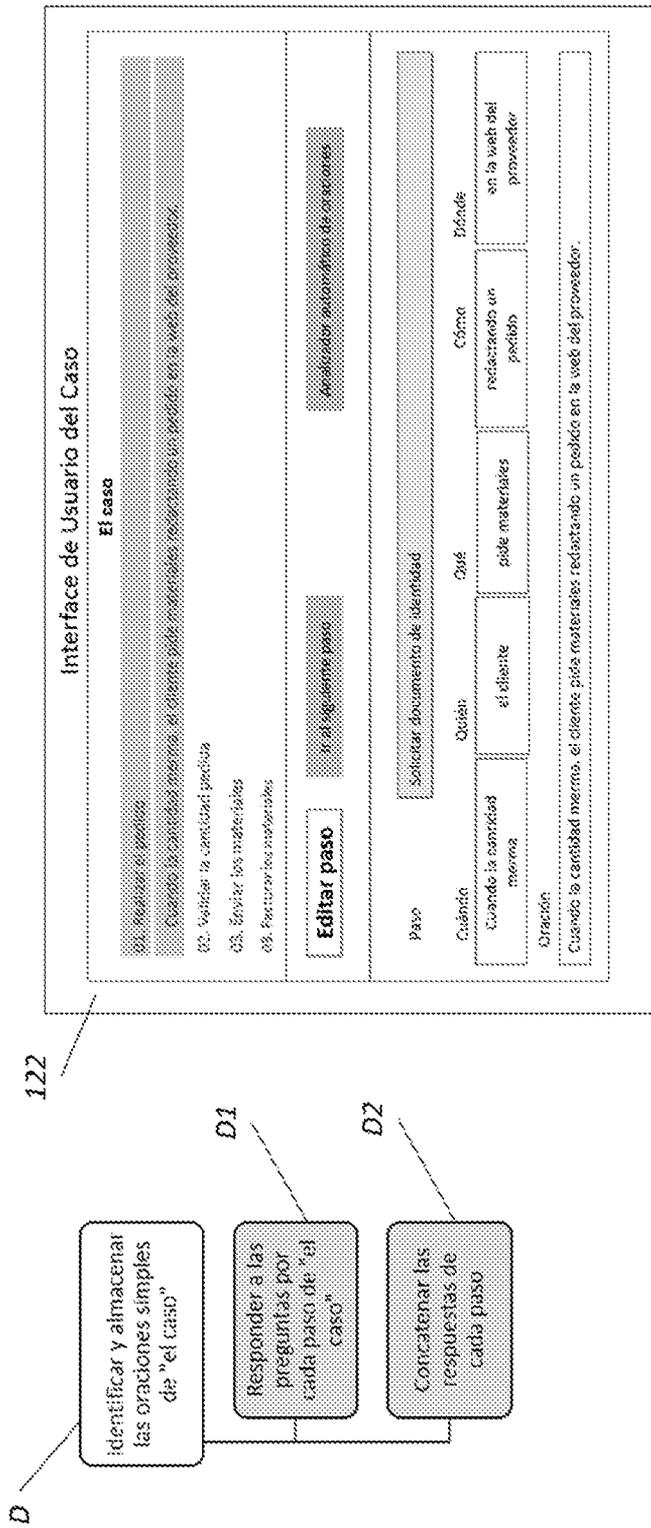


FIG. 2D

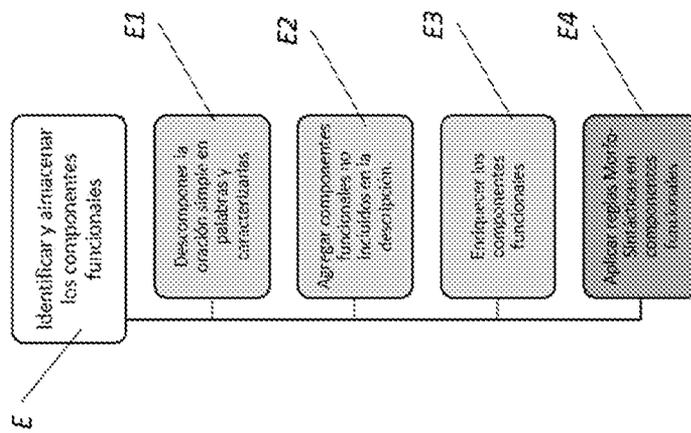
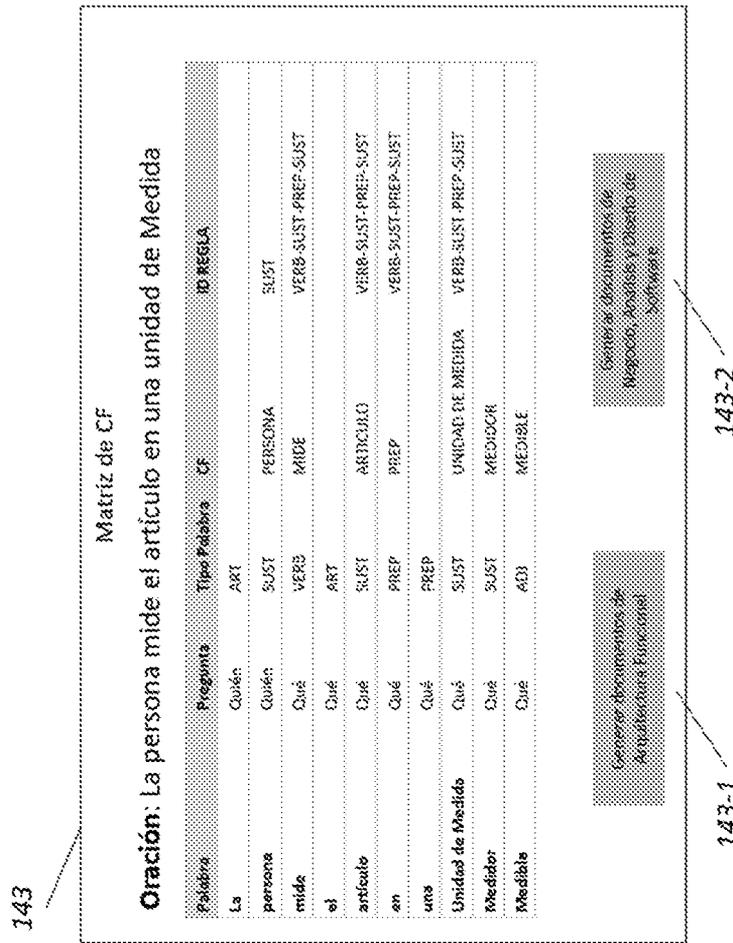


FIG. 2E

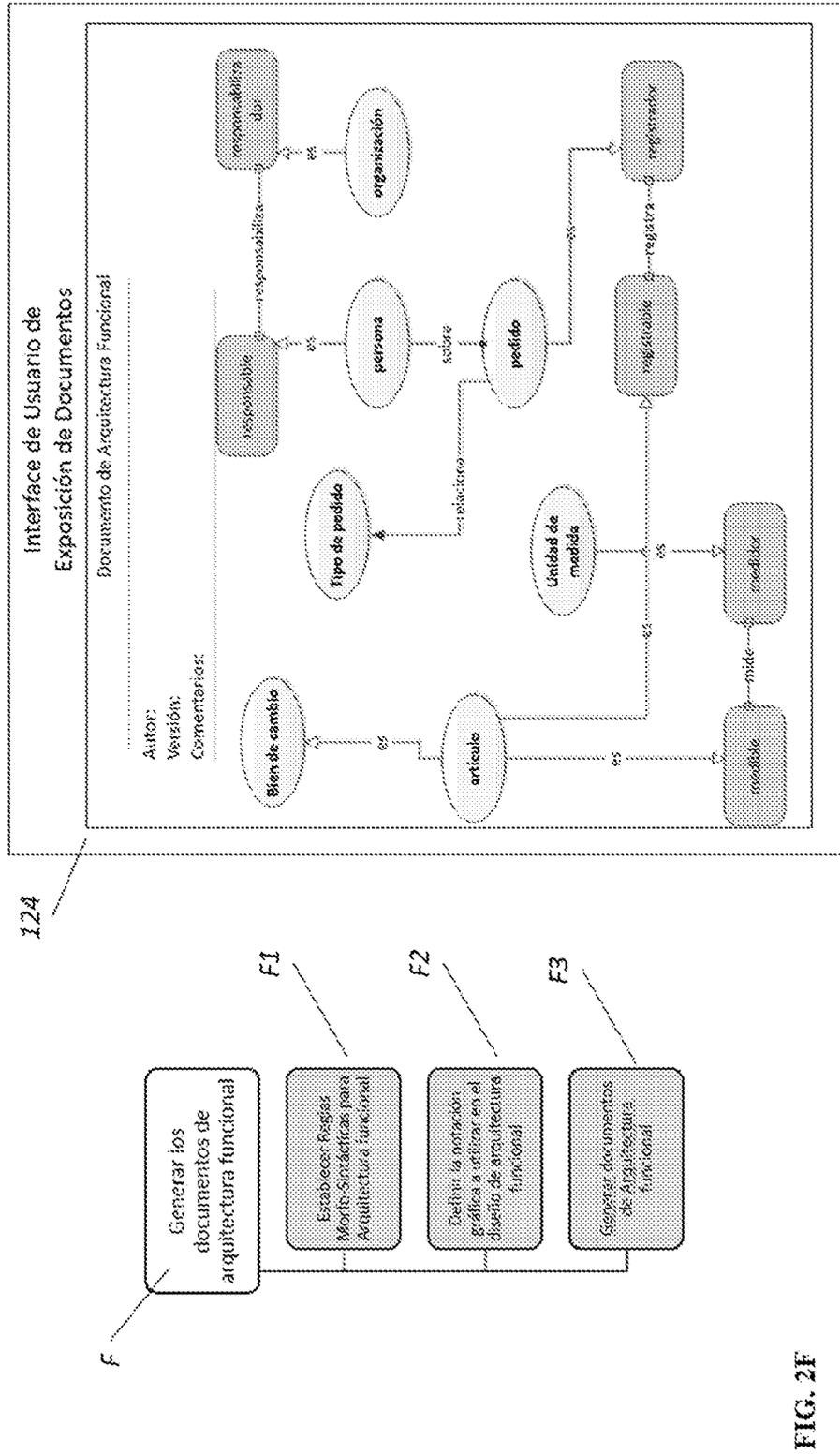
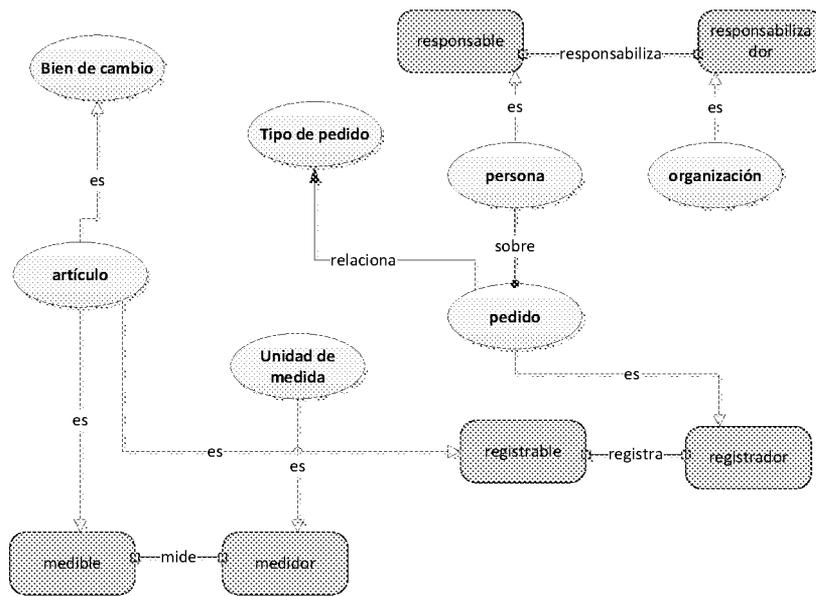


FIG. 2F1



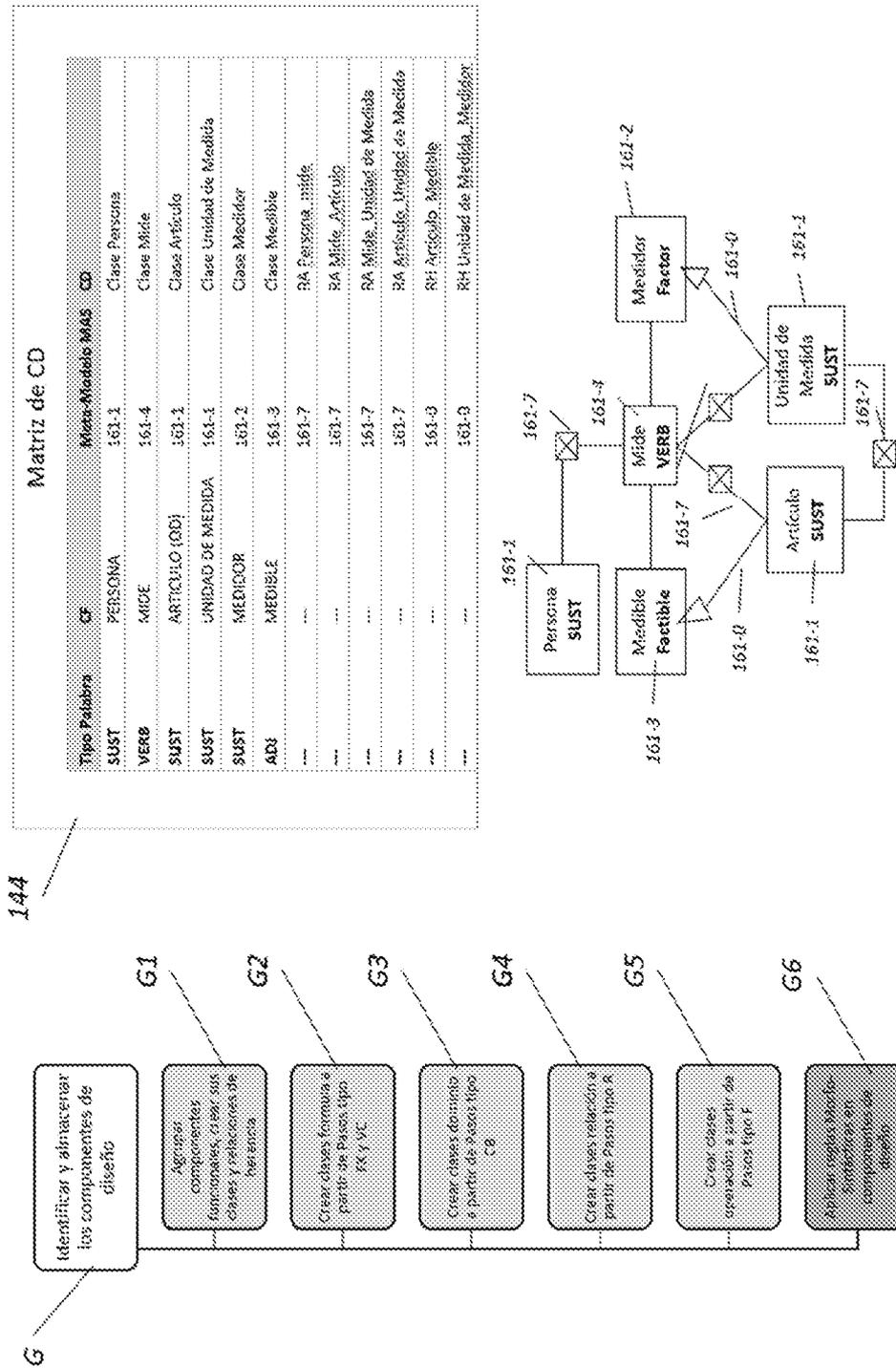


FIG. 2G

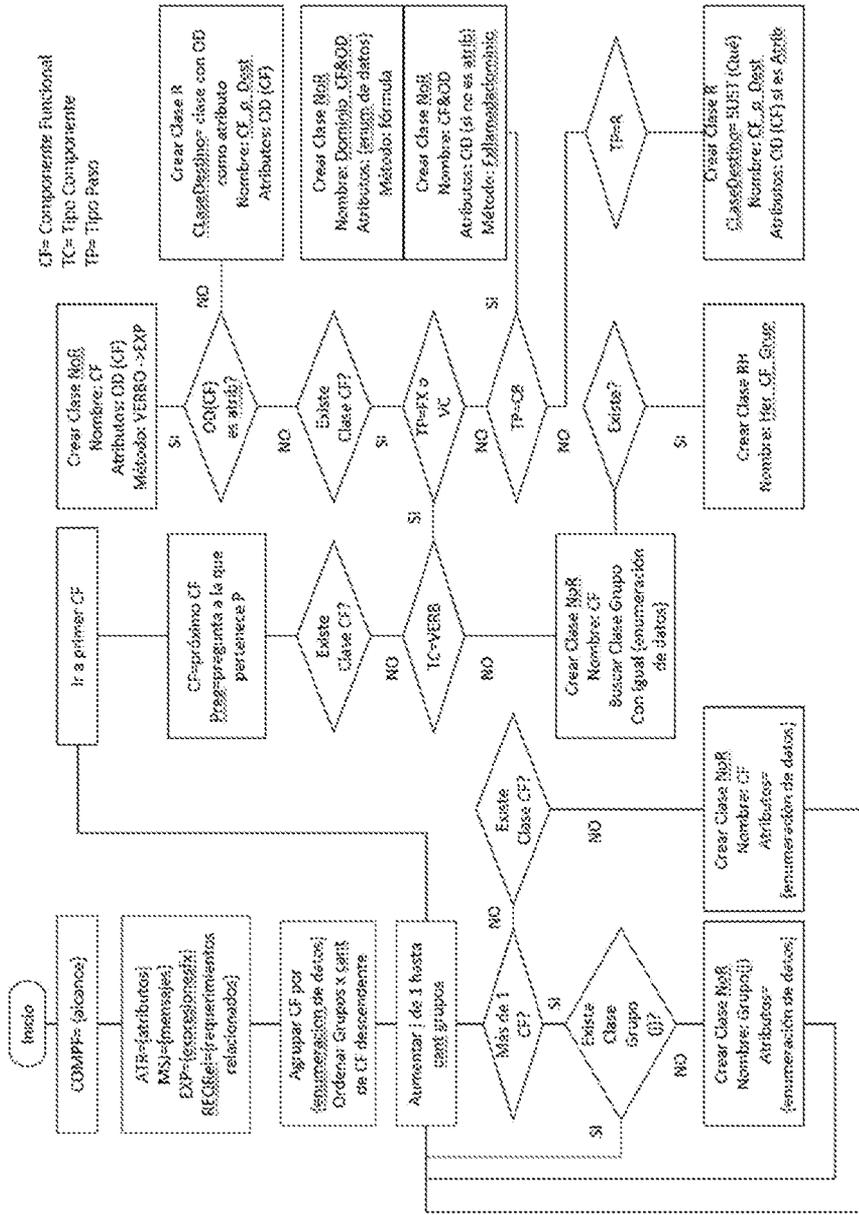


FIG. 2G1

Interface de Usuario Formatos , Notación Gráfica y Oraciones STD			
Notación gráfica para documentos de diseño de software			
Identificador MAS	Nombre MAS	Gráfico	
161-0	H Palabra-Palabra		
161-1	Palabra		
161-2	Factor		
161-3	Factible		
161-4	Factibilidad		
161-5	RA Factor-Factibilidad		
161-6	RA Factible-Factibilidad		
161-7	RA Palabra-Palabra		
161-8	RH Palabra-Factible		
161-9	RH Palabra-Factor		

123

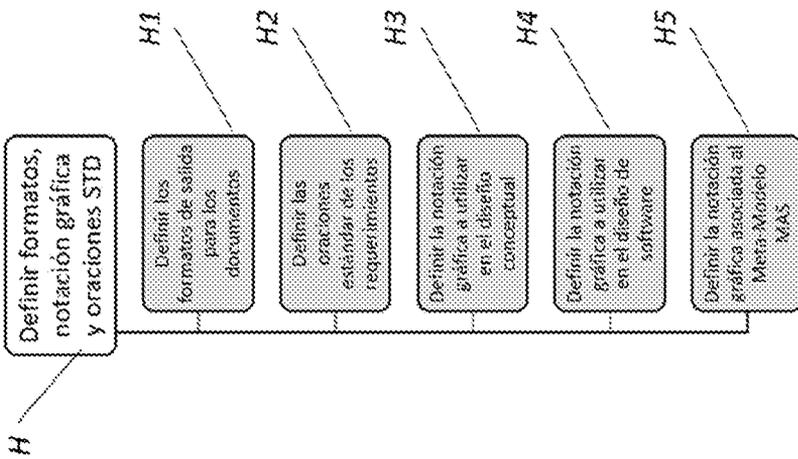


FIG. 2H

FIG. 21

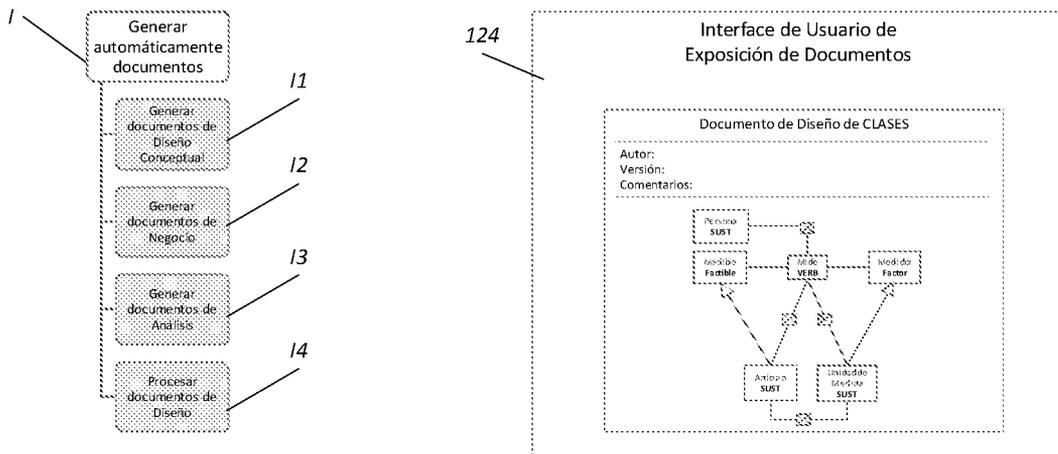


FIG. 211

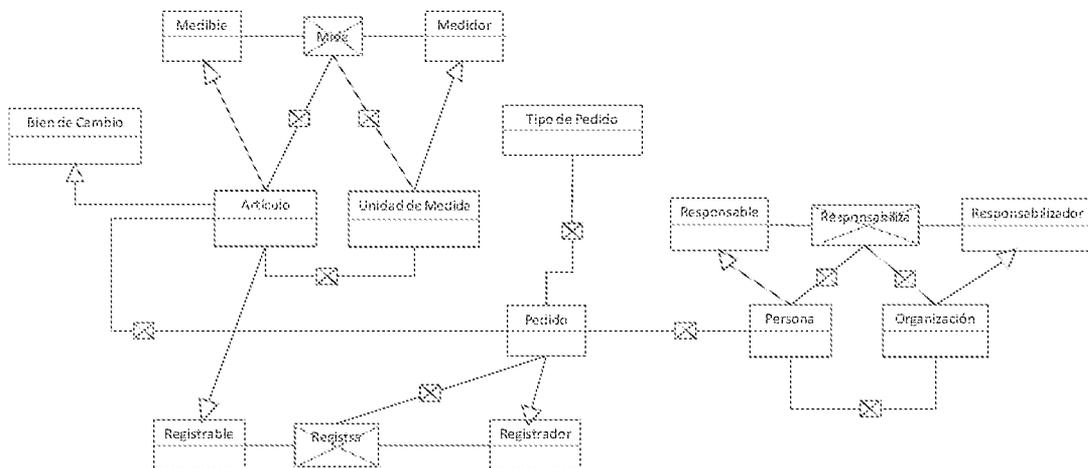


FIG. 212

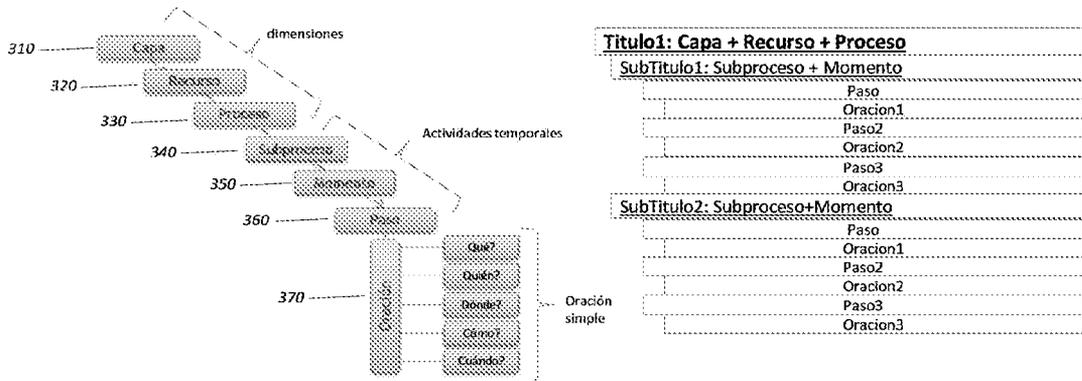


FIG. 213

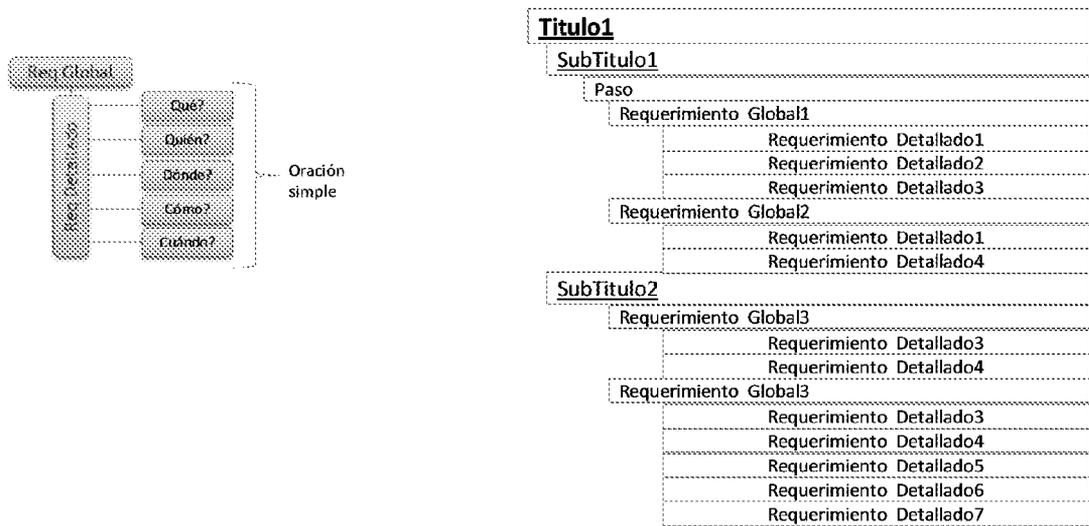
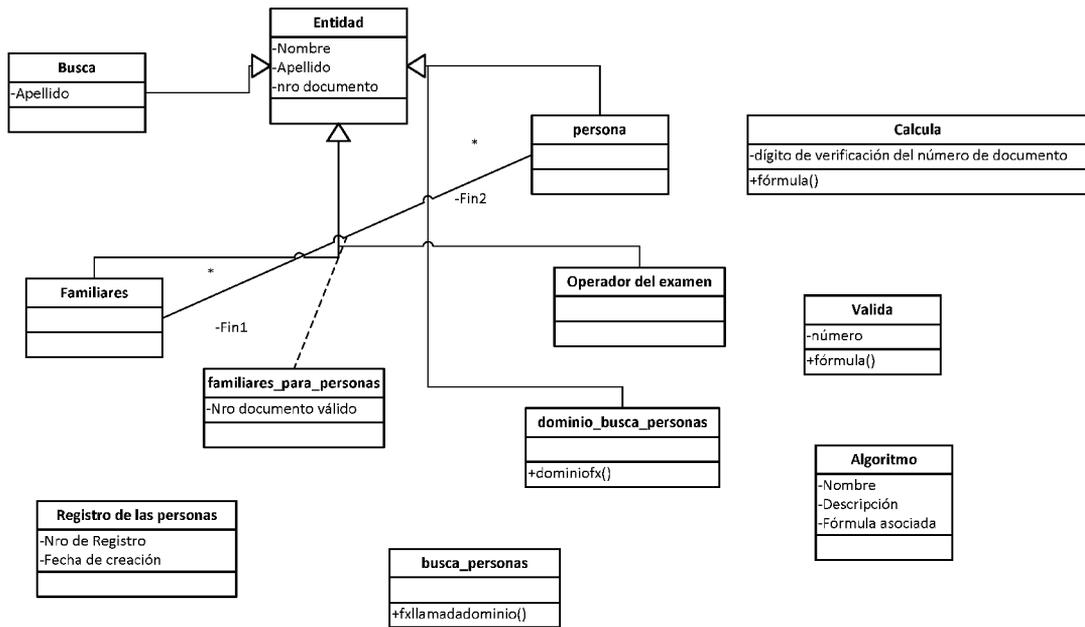


FIG. 3



INTERNATIONAL SEARCH REPORT

International application No.

PCT/IB2016/052465

A. CLASSIFICATION OF SUBJECT MATTER (CIP) G06F 9/45 (2016.01) According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) (CIP) G06F 9/45 Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) EPOQUE, THOMSON, GOOGLE, ESP@CENET, INAPI		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US7555742B2 (SOSY INC.) 30 June 2009. Col. 14, L. 1-4, L. 16-18, L. 49-53; Col. 15 L. 5-7, L. 10-15; Col. 19, L. 5-10; Col. 20, L. 38-40; Col. 22, L. 49-53, 62-67; Col. 44, L. 12-20; Reiv. 1, 3; Ref. 106, 112, 114, 116, 232, 238.	1, 7, 10-13, 21
Y	WO2008/131028A2 (THE BOEING COMPANY) 30 October 2008. Pag. 19, L. 17-20; Reiv. 11; Fig. 4, 7; Ref. 206, 212, 214, 404, 418, 604, 706.	1, 7, 10-13, 21
A	US 8561014 B2 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 15 October 2013. Col. 4, L. 66-67; Col. 5, L. 5-8; Col. 6, L. 16-22. Reiv. 1, 9, 14; Fig 3.	
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 22 July 2016 (22.07.16)		Date of mailing of the international search report 25 August 2016 (25.08.16)
Name and mailing address of the ISA/ INAPI, Av. Libertador Bernardo Facsimile No. O'Higgins 194, Piso 17, Santiago, Chile		Authorized officer ARAYA LARA, Hugo Boris Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/IB2016/052465

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. Claims Nos.: **1.A-D, 2-6, 8, 9, 12.A-D,H, 14, 19, 20**
because they relate to subject matter not required to be searched by this Authority, namely:
These claims relate to a subject matter that is considered by this International Searching Authority to be affected by PCT Rules 39.1(iii) and 67.1(iii), concerning schemes, rules or methods of doing business, performing purely mental acts or playing games. In conclusion, no expert opinion will be established in respect of novelty, inventive step and industrial applicability for these claims (PCT Article 17(2)(a)(b)).
2. Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
3. Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.
3. As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
4. No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

- The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/IB2016/052465

Patent document cited in the search copy	Publication date	Patent family member(s)	Publication date
US 7555742 (B2)	30-06-2009	US2007/168907 (A1)	19-07-2007
		US7555742 (B2)	30-06-2009
		AU4194501 (A)	15-10-2001
		AU2001/241945 (B2)	11-11-2004
		CA2403975 (A1)	11-10-2001
		CA2723005 (A1)	11-10-2001
		EP1272927 (A2)	08-01-2003
		JP2004/506962 (A)	04-03-2004
		JP2007/257654 (A)	04-10-2007
		JP2009/193592 (A)	27-08-2009
		US6681383 (B1)	20-01-2004
		US2002/062475 (A1)	23-05-2002
		US7137100 (B2)	14-11-2006
		US2002/100014 (A1)	25-07-2002
		US7278130 (B2)	02-10-2007
		US2004/153992 (A1)	05-08-2004
		US7334216 (B2)	19-02-2008
		US2007/089103 (A1)	19-04-2007
		US7584451 (B2)	01-09-2009
		US2008/275910 (A1)	06-11-2008
		US7941438 (B2)	10-05-2011
		US2008/209391 (A1)	28-08-2008
		US8037449 (B2)	11-10-2011
		US2004/233232 (A1)	25-11-2004
		US8141031 (B2)	20-03-2012
		US2009/125546 (A1)	14-05-2009
		US8245183 (B2)	18-08-2012
		US2009/132995 (A1)	21-05-2009
		US8365138 (B2)	29-01-2013
		US2003/167455 (A1)	04-09-2003
WO0175593 (A2)	11-10-2001		
-----	-----	-----	-----
WO 2008/131028 (A2)	30-10-2008	US2008/255997 (A1)	16-10-2008
-----	-----	-----	-----
US 8561014 (B2)	15-10-2013	US2010275179 (A1)	28-10-2010
		CN101872341 (A)	27-10-2010
		JP2010257452 (A)	11-11-2010
		JP5536518 (B2)	02-07-2014

INFORME DE BÚSQUEDA INTERNACIONAL

Solicitud internacional N°
PCT/IB2016/052465

A. CLASIFICACIÓN DEL OBJETO DE LA SOLICITUD

(CIP) G06F 9/45 (2016.01)

De acuerdo con la Clasificación Internacional de Patentes (CIP) o según la clasificación nacional y CIP.

B. SECTORES COMPRENDIDOS POR LA BÚSQUEDA

Documentación mínima buscada (sistema de clasificación seguido de los símbolos de clasificación)

(CIP) G06F 9/45

Otra documentación consultada, además de la documentación mínima, en la medida en que tales documentos formen parte de los sectores comprendidos por la búsqueda

Bases de datos electrónicas consultadas durante la búsqueda internacional (nombre de la base de datos y, si es posible, términos de búsqueda utilizados)

EPOQUE, THOMSON, GOOGLE, ESP@CENET, INAPI

C. DOCUMENTOS CONSIDERADOS RELEVANTES

Categoría*	Documentos citados, con indicación, si procede, de las partes relevantes	Relevante para las reivindicaciones N°
Y	US7555742B2 (SOSY INC.) 30 Junio 2009. Col. 14, L. 1-4, L. 16-18, L. 49-53; Col. 15 L. 5-7, L. 10-15; Col. 19, L. 5-10; Col. 20, L. 38-40; Col. 22, L. 49-53, 62-67; Col. 44, L. 12-20; Reiv. 1, 3; Ref. 106, 112, 114, 116, 232, 238.	1, 7, 10-13, 21
Y	WO2008/131028A2 (THE BOEING COMPANY) 30 Octubre 2008. Pag. 19, L. 17-20; Reiv. 11; Fig. 4, 7; Ref. 206, 212, 214, 404, 418, 604, 706.	1, 7, 10-13, 21
A	US 8561014 B2 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 15 Octubre 2013. Col. 4, L. 66-67; Col. 5, L. 5-8; Col. 6, L. 16-22. Reiv. 1, 9, 14; Fig 3.	

En la continuación del Recuadro C se relacionan otros documentos Los documentos de familias de patentes se indican en el Anexo

* Categorías especiales de documentos citados:	"T"	documento ulterior publicado con posterioridad a la fecha de presentación internacional o de prioridad que no pertenece al estado de la técnica pertinente pero que se cita por permitir la comprensión del principio o teoría que constituye la base de la invención.
"A" documento que define el estado general de la técnica no considerado como particularmente relevante.	"X"	documento particularmente relevante; la invención reivindicada no puede considerarse nueva o que implique una actividad inventiva por referencia al documento aisladamente considerado.
"E" solicitud de patente o patente anterior pero publicada en la fecha de presentación internacional o en fecha posterior.	"Y"	documento particularmente relevante; la invención reivindicada no puede considerarse que implique una actividad inventiva cuando el documento se asocia a otro u otros documentos de la misma naturaleza, cuya combinación resulta evidente para un experto en la materia.
"L" documento que puede plantear dudas sobre una reivindicación de prioridad o que se cita para determinar la fecha de publicación de otra cita o por una razón especial (como la indicada).	"&"	documento que forma parte de la misma familia de patentes.
"O" documento que se refiere a una divulgación oral, a una utilización, a una exposición o a cualquier otro medio.		
"P" documento publicado antes de la fecha de presentación internacional pero con posterioridad a la fecha de prioridad reivindicada.		

Fecha en que se ha concluido efectivamente la búsqueda internacional. 22/07/2016 22/julio/2016	Fecha de expedición del informe de búsqueda internacional 25/08/2016 25/agosto/2016
Nombre y dirección postal de la Administración encargada de la búsqueda internacional INAPI, Av. Libertador Bernardo O'Higgins 194, Piso 17, Santiago, Chile N° de fax	Funcionario autorizado ARAYA LARA, Hugo Boris N° de teléfono 56-2-28870550 56-2-28870551

INFORME DE BÚSQUEDA INTERNACIONAL

Solicitud internacional N°

PCT/IB2016/052465

Recuadro II Observaciones cuando se estime que algunas reivindicaciones no pueden ser objeto de búsqueda (continuación del punto 2 de la primera hoja)

Este informe de búsqueda internacional no se ha realizado en relación a ciertas reivindicaciones según el Artículo 17.2)a) por los siguientes motivos:

1. Las reivindicaciones N°s: 1.A-D, 2-6, 8, 9, 12.A-D,H, 14, 19, 20 se refieren a un objeto con respecto al cual esta Administración no está obligada a proceder a la búsqueda, a saber:
Estas cláusulas se refieren a una materia que esta Administración considera afectada por las Reglas PCT 39.1(iii) y 67.1(iii), concernientes a planes, principios o métodos para hacer negocios, para actos puramente intelectuales o en materia de juego. En conclusión, ninguna opinión experta será establecida para novedad, nivel inventivo y aplicación industrial de dichas cláusulas (Art.17(2)(a)(b)).
2. Las reivindicaciones N°s:
se refieren a elementos de la solicitud internacional que no cumplen con los requisitos establecidos, de tal modo que no pueda efectuarse una búsqueda provechosa, concretamente:
3. Las reivindicaciones N°s:
son reivindicaciones dependientes y no están redactadas de conformidad con los párrafos segundo y tercero de la Regla 6.4.a).

Recuadro III Observaciones cuando falta unidad de invención (continuación del punto 3 de la primera hoja)

La Administración encargada de la búsqueda internacional ha detectado varias invenciones en la presente solicitud internacional, a saber:

1. Dado que todas las tasas adicionales requeridas han sido satisfechas por el solicitante dentro del plazo, el presente informe de búsqueda de tipo internacional comprende todas las reivindicaciones que pueden ser objeto de búsqueda.
2. Dado que todas las reivindicaciones que pueden ser objeto de búsqueda podrían serlo sin realizar un esfuerzo que justifique tasas adicionales, esta Administración no requirió el pago de tasas adicionales.
3. Dado que tan sólo una parte de las tasas adicionales requeridas ha sido satisfecha dentro del plazo por el solicitante, el presente informe de búsqueda de tipo internacional comprende solamente aquellas reivindicaciones respecto de las cuales han sido satisfechas las tasas, concretamente las reivindicaciones N°s:
4. Ninguna de las tasas adicionales requeridas ha sido satisfecha por el solicitante dentro de plazo. En consecuencia, el presente informe de búsqueda de tipo internacional se limita a la invención mencionada en primer término en las reivindicaciones cubiertas por las reivindicaciones N°s:

Indicación en cuanto a la protesta

- Se acompañó a las tasas adicionales la protesta del solicitante y, en su caso, el pago de una tasa de protesta.
- Se acompañó a las tasas adicionales la protesta del solicitante, pero la tasa de protesta aplicable no se pagó en el plazo establecido en el requerimiento.
- El pago de las tasas adicionales no ha sido acompañado de ninguna protesta.

INFORME DE BÚSQUEDA INTERNACIONAL

Información relativa a miembros de familias de patentes

Solicitud internacional N°

PCT/IB2016/052465

Documento de patente citado en Informe de Búsqueda	Fecha de Publicación	Miembro(s) de Familia	Fecha de Publicación
US 7555742 (B2)	30-06-2009	US2007/168907 (A1)	19-07-2007
		US7555742 (B2)	30-06-2009
		AU4194501 (A)	15-10-2001
		AU2001/241945 (B2)	11-11-2004
		CA2403975 (A1)	11-10-2001
		CA2723005 (A1)	11-10-2001
		EP1272927 (A2)	08-01-2003
		JP2004/506962 (A)	04-03-2004
		JP2007/257654 (A)	04-10-2007
		JP2009/193592 (A)	27-08-2009
		US6681383 (B1)	20-01-2004
		US2002/062475 (A1)	23-05-2002
		US7137100 (B2)	14-11-2006
		US2002/100014 (A1)	25-07-2002
		US7278130 (B2)	02-10-2007
		US2004/153992 (A1)	05-08-2004
		US7334216 (B2)	19-02-2008
		US2007/089103 (A1)	19-04-2007
		US7584451 (B2)	01-09-2009
		US2008/275910 (A1)	06-11-2008
		US7941438 (B2)	10-05-2011
		US2008/209391 (A1)	28-08-2008
		US8037449 (B2)	11-10-2011
		US2004/233232 (A1)	25-11-2004
		US8141031 (B2)	20-03-2012
		US2009/125546 (A1)	14-05-2009
		US8245183 (B2)	18-08-2012
US2009/132995 (A1)	21-05-2009		
US8365138 (B2)	29-01-2013		
US2003/167455 (A1)	04-09-2003		
WO0175593 (A2)	11-10-2001		
WO 2008/131028 (A2)	30-10-2008	US2008/255997 (A1)	16-10-2008
US 8561014 (B2)	15-10-2013	US2010275179 (A1)	28-10-2010
		CN101872341 (A)	27-10-2010
		JP2010257452 (A)	11-11-2010
		JP5536518 (B2)	02-07-2014