Análisis de Adecuación de Bases de Datos al Estándar ANSI/ISA-95 Utilizando un Agente Inteligente

Melina C. Vidoni[1], Aldo R. Vecchietti[1]

[1]Ingar UTN-CONICET, Santa Fe, Argentina.
{melinavidoni, aldovec}@santafe-conicet.gov.ar

Resúmen. La marcada tendencia a la globalización y comunicación entre organizaciones, conlleva la necesidad de establecer una base sobre la información a compartir, y estandarizar las estructuras de datos que se utilicen, con el objetivo de mejorar la eficiencia de dicho flujo de información. Desde hace tiempo, el estándar ANSI/ISA-95 ha ido cobrando más relevancia como uno de los principales medios para la estandarización y automatización de sistemas empresariales, MES y CPM, entre otros. Este trabajo propone la utilización de un agente inteligente basado en conocimiento que procesa lenguaje natural, para analizar y clasificar el contenido de la base de datos de un ERP, en las estructuras de información propuestas por el estándar ANSI/ISA-95, con el objetivo de favorecer un estudio de los sistemas actuales de las organizaciones en orden de poder lograr una futura integración.

Palabras Clave: ANSI/ISA-95, bases de datos, agente inteligente, bag of words, análisis.

1. Introducción

En los últimos años los principales desafíos de las empresas han sido el rápido cambio de sus entornos, lo que implica una alta necesidad de flexibilidad, agilidad, eficiencia y calidad en sus procesos. Por esto mismo, la Comisión Europea [1] recomendó la mejora de los procesos de integración a través de su estandarización y posterior automatización. La toma de decisiones integradas y la optimización colaborativa dentro de las empresas, pasó a tener un rol crucial en la interrelación de organizaciones. Con este objetivo en mente, muchas empresas han desarrollado sistemas tipo MES (Manufacturing Execution Systems) o CPM (Collaborative Production Management) con un único objetivo en mente: anular la brecha entre los procesos, las comunicaciones y los sistemas ERP (Enterprise Resource Planning) [2].

Un requerimiento fundamental para alcanzar esta integración es definir estructuras de información y herramientas sofisticadas que permitan explotar dichas configuraciones, con el objetivo de mejorar la disponibilidad y comunicación de los datos. Siguiendo esta línea, se han propuesto muchos estándares para mejorar la eficiencia y el flujo de la información, entre ellos el ANSI/ISA-95 [3].

ANSI/ISA-95 es un estándar internacional para desarrollar interfaces automatizadas entre empresas y sistemas de control, proponiendo un conjunto de modelos y

definiciones que proveen una terminología consistente para las tareas e información de manufactura y producción que deben ser intercambiadas en sistemas que se interrelacionan [4]. En los últimos años, este estándar ha sido ampliamente aceptado, debido a que especifica un modelo funcional completo [5].

Se han realizado varios trabajos académicos para favorecer el intercambio de información estandarizada según los modelos del ANSI/ISA-95, así como diferentes formas de implementación. Ya en 2009 varios autores propusieron una plataforma para el intercambio de información utilizando diagramas BPMN (Business Process Model and Notation) basados en los modelos del ANSI/ISA-95 [2]. También se han efectuado avances en el área de simulación, con el objetivo de generar especificaciones para desarrollar sistemas uniformes [6]. Otros autores han empleado ontologías para generar un framework que integra la toma de decisiones, utilizando las estructuras del ANSI/ISA-88 [3]. Finalmente, He, et al. (2012) realizaron una herramienta para el modelado de empresas, con fundamentos en el ISA-95 y en el IEC 62246.

Sin embargo, si bien muchos estudios se enfocan en diseñar nuevos sistemas y herramientas basados en el ANSI/ISA-95 [7], pocos intentan analizar los sistemas ya existentes y proveer un informe sobre su adecuación al estándar, o estudiar qué tipo de información de manufactura contienen los sistemas, a la luz de la clasificación propuesta en el ANSI/ISA-95. La realización de esto favorecería la integración entre sistemas sin obligar a las empresas a cambiar radicalmente su forma de trabajar, como así también proveer un marco para el análisis de los sistemas existentes y posibles formas de modificarlos -con el objetivo de adecuarse al ANSI/ISA-95.

A su vez, en un desarrollo previo [8], los autores encontraron la necesidad de presentar al usuario el tipo de información de manufactura contenida en cada tabla de la base de datos de un ERP. Utilizando esta idea como disparador inicial, la propuesta detallada en este artículo es utilizar un agente inteligente, cuya base de conocimiento esté dada por el ANSI/ISA-95, y que pueda clasificar el contenido de una base de datos de un ERP o sistema empresarial, en cada una de las categorías que el estándar propone. Una de las fortalezas de utilizar un agente inteligente, es la capacidad inherente del mismo de procesar el lenguaje natural.

En esta rama, también se han realizado proyectos sobre categorización de textos o estructuras, utilizando agentes inteligentes. En uno de ellos, se ha propuesto una clasificación sobre fuentes de generación de gas, utilizando algoritmos genéticos y redes neuronales para posteriormente compararlos [9]. Una investigación relevante generó un método de clasificar documentos de texto de forma automatizada, usando un conglomerado de múltiples agentes que procesaban lenguaje natural, y donde cada uno sólo podía catalogar en una sola categoría [10]. Finalmente, la idea de bolsas de palabras también ha sido empleada, a través de un modelo bayesiano, para la generación de documentos de texto usando agentes inteligentes [11].

Cabe destacar que hasta el momento no se han encontrado trabajos que empleen agentes inteligentes para analizar sistemas existentes a la luz de los conceptos propuestos por el estándar ANSI/ISA-95.

2. Descripción del Estándar ANSI/ISA-95

El estándar ANSI/ISA-95 en la Parte 3 [12] propone clasificar la información de manufactura en cuatro categorías que definen la información de productos y de producción. La Figura 1 forma parte del estándar y muestra dicha clasificación[1]:

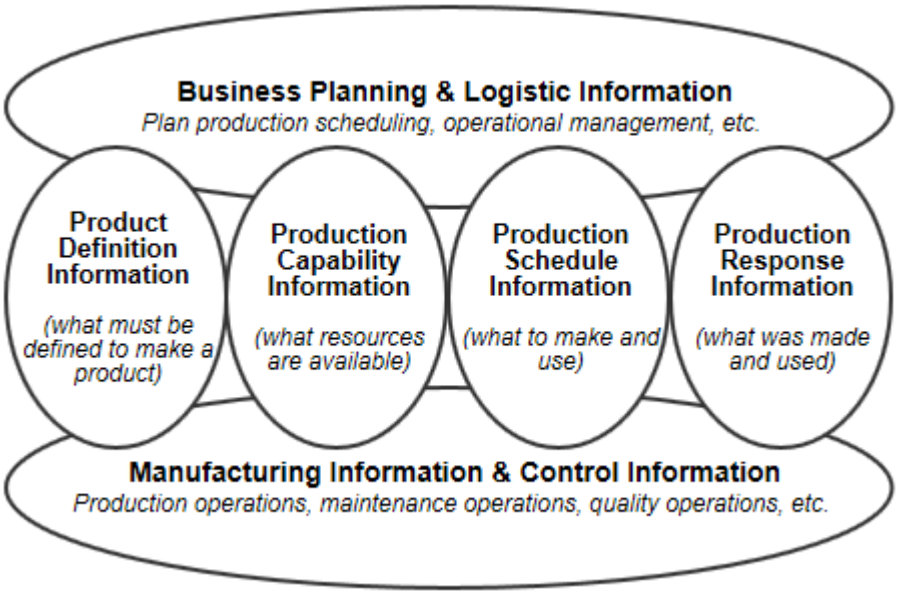

Fig. 1. Categorías de información, según ANSI/ISA-95.

Como punto de partida para el proyecto, se seleccionaron las categorías *Product Definition*, *Production Capability* y *Production Schedule*, dado que contienen el mayor porcentaje de información sobre la manufactura, y sus estructuras de datos son las que más impacto poseen en las bases de datos de los sistemas.

La Parte I del estándar [4] analiza estas categorías, dividiéndolos en subcategorías, las cuales son presentadas con definiciones y descripciones del tipo de información que contiene. Estas subcategorías pueden a su vez superponerse entre ellas, lo que a menudo es acompañado con un gráfico como el de la Figura 2; en ella, puede observarse la superposición de información dentro de la categoría *Product Definition*, lo que puede interpretarse de la siguiente forma: la definición del producto está compuesta por reglas de producción y una lista de recursos (*Bill of Resources*), la cual a su vez contiene información sobre la lista de materiales (*Bill of Materials*) para cada producto en particular. Es decir, que declara las subcategorías para la clase *Product Definition Information*, así como las relaciones de jerarquía e interrelación entre ellas.

[1] Los diagramas extraídos del ANSI/ISA-95 serán trabajados en inglés, para preservar los nombres originales.

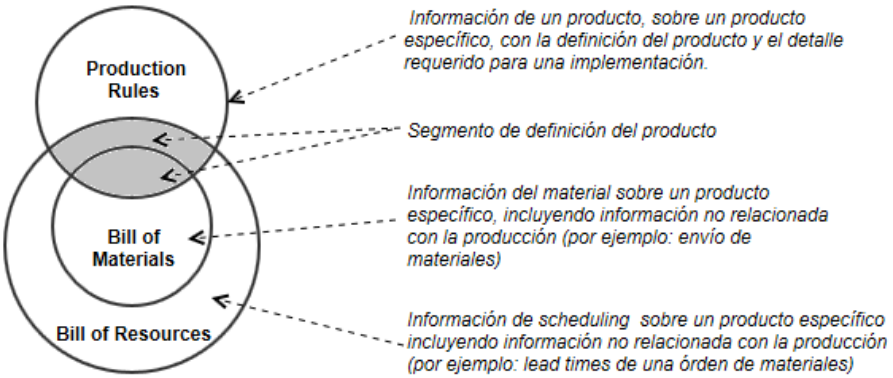

Fig. 2. Superposición de información en la Definición del Producto, para ISA-95 Parte I.

Este tipo de esquemas de superposición son detallados en ANSI/ISA-95 Parte I para cada una de las cuatro categorías principales, pero no serán descriptas en su totalidad en el presente escrito, por cuestiones de espacio.

Utilizando las clases y subclases de información de manufactura del estándar y mencionadas hasta ahora, se ha generado un grafo que abarca las categorías que serán utilizadas para analizar la información en las bases de datos de los ERP, con lo que se pretende alcanzar los objetivos de categorización y estudio de la adecuación al estándar. Este grafo puede observarse en la Figura 3:

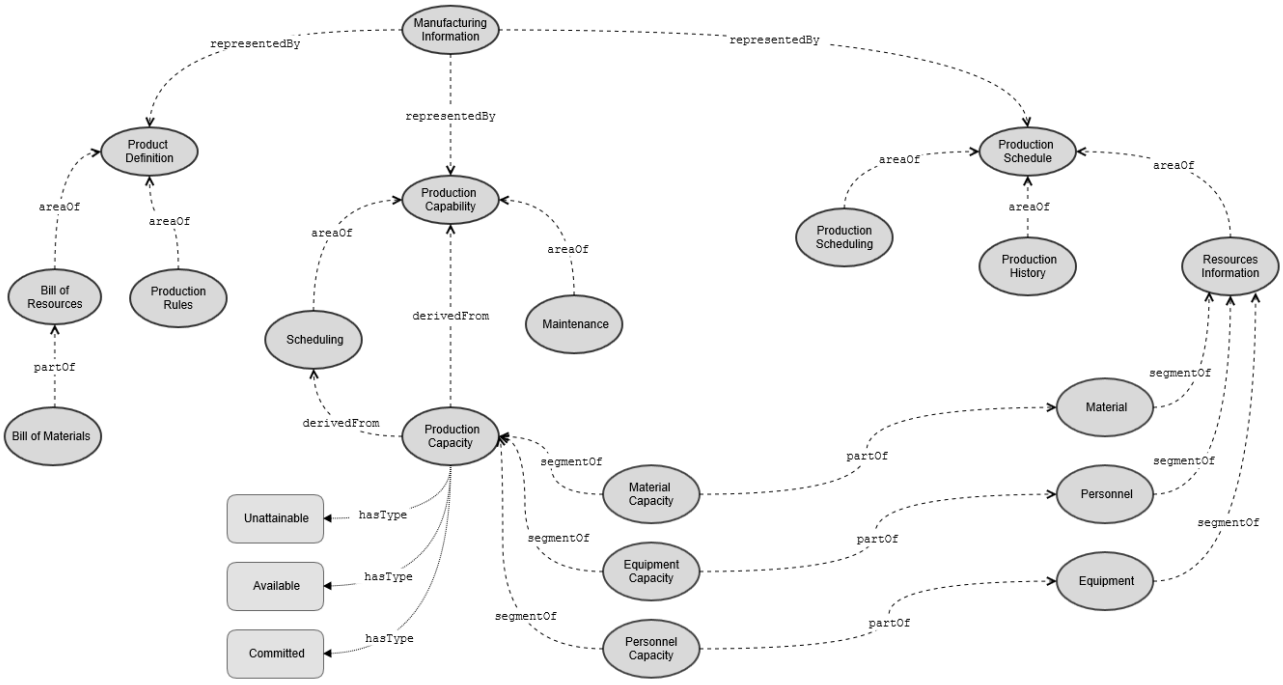

Fig. 3. Grafo utilizado como base de conocimiento por GrACED, basado en ANSI/ISA-95.

De esta forma, en el grafo de la Fig. 3 el nodo raíz representa a la totalidad de información de manufactura, mientras que los nodos de nivel 1 son las grandes categorías de la Fig. 1 (sin usar a *Production Response Information*). Por otro lado, los nodos de niveles sucesivos e inferiores fueron obtenidos de los gráficos de superposición (por ejemplo, las categorías visualizadas en la Fig. 2, son representadas como los hijos del nodo *Product Definition* en el grafo de la Fig. 3) y de las descripciones de cada categoría. Los nodos de forma rectangular representan categorías que, si bien están denotadas en el estándar, fueron graficadas pero no serán empleadas para el posterior análisis de las bases de datos.

Por otro lado, los arcos y sus etiquetas representan los tipos de relaciones entre los nodos:

- Los arcos etiquetados como `representedBy` unen a la raíz con las principales categorías denotando que los nodos hermanos no se superponen entre sí, pero que los hijos de dichos nodos sí pueden superponerse.
- Un arco `partOf` indica que el nodo hijo es una especialización del padre; por ejemplo, en la Fig. 2 puede observarse que *Bill of Materials* está incluido completamente dentro de *Bill of Resources*.
- Arcos tipo `segmentOf` representan relaciones de tipo "compone a" y a su vez indican que los nodos hermanos pueden superponerse y tener información que se clasifique en más de uno de estos nodos.
- Los arcos `areaOf` son similares a los `segmentOf`, sólo que la pertenencia hacia alguno de los nodos es mayor (podemos decir que un nodo pesa más que sus hermanos) y en el caso de obtener información que clasifique en más de uno, se emplearán dichos pesos para evaluar la pertenencia más fuerte.
- Finalmente, los arcos `derivedFrom` indican que todos los padres de dicho nodo tienen la misma relevancia.

Por esto mismo, hay algunos nodos que son más sencillos que otros en su estructura. Por ejemplo, los nodos *Material Capacity*, *Equipment Capacity* y *Personal Capacity*, participan en dos tipos de relaciones: una relación `partOf` y una del tipo `segmentOf`. Por un lado, el estándar define a estos tres nodos como parte de la capacidad de producción, en particular referidos a los materiales, equipo y personal, junto con sus posibles superposiciones (por ejemplo: equipos automatizados). Sin embargo, en una sección posterior, se define a *Resource Information* al componerlo de información de materiales, equipo y personal, incluyendo datos que van más allá de la capacidad de los mismo; es por esto que los nodos *Capacity* fueron incluidos con relaciones `partOf`, mientras que siguen especificando la capacidad de producción.

Finalmente, cabe destacar que este grafo fue implementado en un archivo XML [13], el cual almacena los nodos, sus nombres y las relaciones entre ellos; el agente presentado en la siguiente sección trabaja con esta representación XML del grafo para clasificar en las categorías representadas como nodos.

3. Estructura del Agente Inteligente

Un agente inteligente es una entidad autónoma inserta en un ambiente, que perci-

be lo que sucede en él a través de percepciones (realizadas mediante sensores), y responde a ellas actuando de forma racional, a través de acciones ejecutadas por actuadores [14]. Por otro lado, los agentes basados en conocimiento especializan la definición anterior, y poseen una representación del conocimiento y un proceso de razonamiento que lo ejecuta y puede combinar el conocimiento general con las percepciones actuales para poder inferir aspectos ocultos del estado actual, antes de seleccionar acciones [14]. Estos agentes son muy utilizados en el procesamiento de lenguaje natural, dado que su comprensión radica en inferir los estados ocultos, es decir, la semántica detrás de las palabras.

El agente inteligente propuesto fue denominado GrACED, por las siglas en inglés de *Grammar Agent for Classifying ERP Databases*, y en la Figura 4 puede observarse su estructura básica distribuida en los componentes mencionados en la definición de agente inteligente.

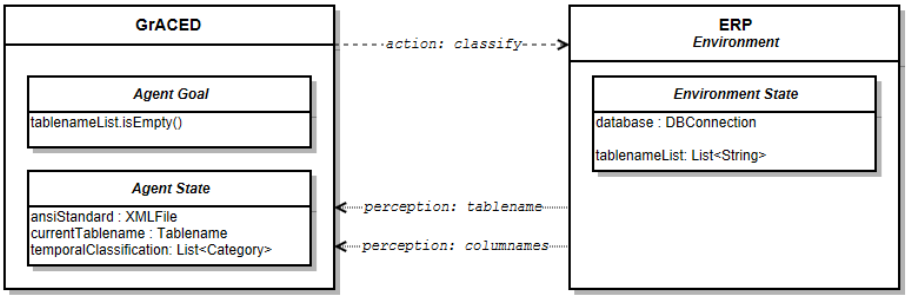

Fig. 4. Componentes de GrACED.

De esta forma, GrACED representa a la entidad autónoma (el agente inteligente en sí) que se encuentra inserto en un ambiente, representado por el ERP que se desea analizar. Este ambiente posee un estado compuesto por los datos de conexión a la base de datos del ERP y la lista de los nombres de las tablas existentes en dicha base.

Por otro lado, GrACED tiene dos percepciones, las cuales están relacionadas: percibir el siguiente nombre de tabla a analizar, y percibir los nombres de las columnas de dicha tabla. Estas percepciones son almacenadas en el estado del agente, más en particular en la variable `currentTablename: Tablename`; los otros son componentes del estado del agente son una referencia a la base de conocimiento (en la variable, el cual será desarrollado en la siguiente sección) y una lista temporal para las preclasificaciones obtenidas para la tabla que se está analizando. Esta última parte del estado del agente es la que almacena los resultados intermedios de la única acción que ejecuta GrACED: clasificar.

Finalmente, el agente también tiene una prueba de meta, la cual le permite evaluar si ha llegado a su objetivo, o si aún necesita continuar trabajando.

3.1 Generación de la Base de Conocimiento

Como se mencionó previamente, GrACED trabaja directamente con lenguaje natural, por lo cual se ha decidido trabajar en idioma inglés debido a su amplia acepta-

ción y alcance para la comunicación y programación.

Sin embargo, cuando el agente percibe la información de una tabla -el nombre de la misma, y los nombres de las columnas- obtiene dos conjuntos de palabras que no tienen secuencia ni relación entre ellas. Por esto, se decidió utilizar el acercamiento de *bags of words* (BoW) o bolsas de palabras. Las bolsas de palabras son una representación simplificada utilizada en el procesamiento de lenguaje natural, donde cada clase o documento se representa en un multiset (o bolsas) de palabras, sin considerar la gramática (formación de sentencias) ni el orden de las palabras [11].

Dado que al analizar las tablas se obtienen tanto el nombre de la misma, como los nombres de las columnas, surge una situación: muy a menudo, las palabras que se encuentran formando parte del nombre de la tabla, no son las mismas que pueden estar en los nombres de las columnas, aún dentro de la misma categoría (por ejemplo, *Production Rules*). Debido a esto, se ha decidido mantener dos bolsas de palabras por categoría: una para las palabras en los nombres de las tablas, y otra para las columnas.

Como se explicó anteriormente, el grafo de la Fig. 3 es el que contiene las categorías en las que clasifica GrACED, de forma que cada nodo del mismo tiene ahora dos bolsas de palabras asociadas. Debido a que el grafo se guarda como un archivo XML, cada nodo es un elemento dentro del mismo el cual tiene atributos para el nombre, y para el nombre de archivo de cada bolsa de palabra, diferenciando el uso de cada una. En la Fig. 5 puede observarse un extracto del código XML que representa al nodo *Bill Of Materials*, donde los atributos `columnameBow` y `tablenameBow` son los que guardan el nombre de archivo de las respectivas BoW:

```
<!--BILL OF MATERIALS NODE-->
<tns:node tns:nodeName="Bill Of Materials" tns:columnameBow="bom_col.xml"
tns:tablenameBow="bom_tab.xml" tns:usable="true">
    <tns:relation tns:relationName="partOf"/>
</tns:node>
```

Fig. 5. Extracto del código XML que representa al nodo Bill of Materials.

Finalmente, para completar la base de conocimiento (BC) era necesario considerar algo más: muchas veces se usan sinónimos (palabras escritas de diferente forma pero que significan lo mismo) o abreviaturas (convenciones ortográficas que acortan la escritura de cierto término o expresión) al momento de nombrar las tablas o columnas. Agregar cada combinación para cada palabra a las BoW no es conveniente, ya que no sólo introduce redundancia y aumenta el tiempo de procesamiento, sino que también reduce los porcentajes de pertenencia al final de la clasificación al agregar palabras innecesarias en las bolsas.

Sin embargo, las palabras pueden tener distintos significados, sinónimos y abreviaturas, dependiendo de la categoría en la que estén siendo clasificadas; por ejemplo, tanto `product` como `production` puede ser abreviado como `prod`, y esa diferencia sólo puede reconocerse por la categoría en la que está la bolsa original. De esta forma, también se agregaron archivos con sinónimos y abreviaturas, los cuales fueron relacionados directamente a cada palabra en cada BoW. Estos archivos serán directamente nombrados como *Synonyms Files*, incluso si contienen abreviaturas.

Las bolsas de palabras y los archivos de sinónimos fueron generadas a partir de las bases de datos de cuatro ERPs de código abierto: Adempiere [15], OpenERP [16],

ERPNext [17] y JFire [18]. El procedimiento empleado fue manual, y se describe a continuación:

1. Se listaron los nombres de tablas y las columnas de cada ERP.
2. Para cada ERP:
    a. Se clasificó manualmente cada tabla, considerando las descripciones de contenido del estándar ANSI/ISA-95.
    b. Se separaron las palabras que conformaban cada nombre de tabla y las de los nombres de columnas. Por ejemplo, el nombre de tabla `stock_inventory_move`, se transformó en tres palabras: `stock`, `inventory` y `move`.
3. Manteniendo la distinción del origen de las palabras (es decir, si eran de los nombres de tablas o de los nombres de columnas) se agruparon todas las palabras de cada categoría (todas las pertenientes a *Bill of Materials*, las pertenecientes a *Production Rules*, etc.).
4. Para cada grupo de palabras:
    a. Se contó la cantidad de veces que aparecía cada palabra, para obtener la "relevancia" o "nivel de descripción" que aporta la misma para una categoría.
    b. Separadamente , se anotaron cada palabra y los sinónimos de la misma.
    c. Se sumó la cantidad de apariciones de la palabra y sus sinónimos.
    d. Dándole un peso total de 100 a cada BoW, se otorgó un peso a cada palabra, considerando la cantidad de apariciones encontrada en el punto anterior.

De esta forma, en la Figura 6 puede observarse la estructura completa del agente, con el detalle de la BC. El principal componente es el grafo basado en la clasificación del estándar ANSI/ISA-95; por cada nodo de dicho grafo, hay un par de BoW (una para las tablas, y una para las columnas, como se explicó anteriormente) y, a su vez, muchas de las palabras en cada bolsa contienen referencias a archivos de sinónimos.

Por otro lado, también se denota que la acción del agente escribe dos archivos como resultados: uno para las tablas que no entraron dentro de ninguna, y para las que entraron en una o más. Esto sucede así, dado que no todas las tablas de la base de datos de un ERP contienen información de manufactura, que es lo que aquí se está buscando categorizar.

3.1 Algoritmo de Razonamiento

La acción del agente posee dos grandes partes: una primera parte de clasificación usando los nodos "habilitados" del grafo, y que sirve para encontrar qué tipo de información almacena cada tabla; la segunda parte trabaja sólo con las tablas que pertenecen a una o más categorías, y propaga sus porcentajes de pertenencias hacia arriba en el grafo, con el objetivo de encontrar la adecuación de la BD al estándar.

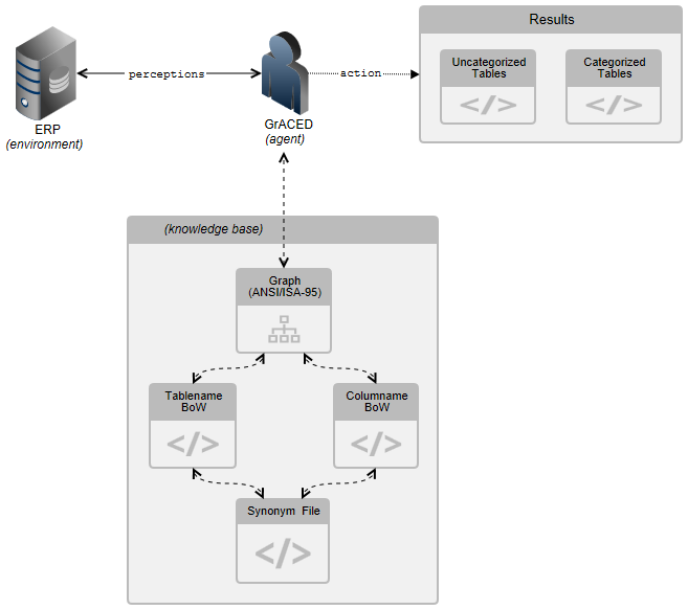

Fig. 6. Estructura completa del agente, la base de conocimiento y los resultados.

En este trabajo se desarrollará sólo la primera parte de clasificación, la cual se ejecuta a través de un algoritmo de razonamiento. Este algoritmo también tiene dos pasos: el primero, es tomar el nombre de la tabla y clasificarlo utilizando las bolsas de palabras para los nombres de tablas. Este proceso puede verse en la Figura 7:

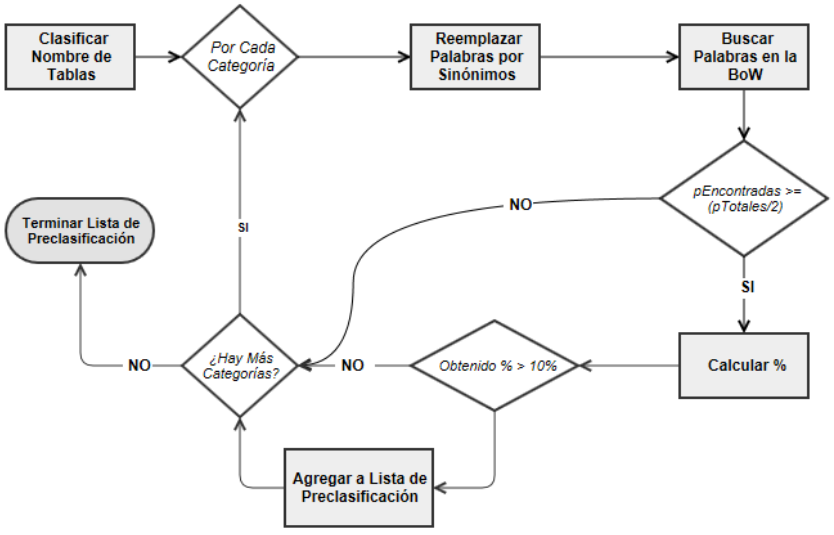

Fig. 7. Paso 1 del algoritmo de razonamiento: preclasificación de nombre de tablas.

En la Fig. 6 puede notarse que hay dos "filtros" para determinar si una clasificación es adecuada o no. El primero de ellos se realiza comparando la cantidad total de palabras del nombre de una tabla, contra la cantidad de ellas que fueron encontradas en la BoW; para poder pasar a la siguiente etapa, al menos la mitad de las palabras del nombre debe estar en la bolsa. El segundo filtro implica calcular el porcentaje de pertenencia con la fórmula que será detallada más adelante, y si dicho valor es menor a un 10%, la clasificación es descartada.

El porcentaje del segundo filtro fue elegido debido a que las bolsas de palabras para las tablas son menores en tamaño que las de las columnas y, además, la cantidad de palabras que hay por nombre de tabla es también mucho menor, por lo que al tener más de un 10%, ya se encuentran palabras de cierta relevancia. Sin embargo, posteriormente se discuten otros análisis que se hacen sobre los resultados del agente.

El segundo paso para la clasificación, es evaluar las palabras en los nombres de las columnas, sólo en las categorías que sobrepasaron la clasificación con el nombre de la tabla. Los pasos seguidos son muy similares a los de la Figura 6, pero con un solo filtro: evaluar que el porcentaje de pertenencia obtenido sea mayor que un 15%. Si el filtro no es superado, la categoría es removida de la preclasificación de la tabla. Dado que todas las bolsas pesan 100, y las BoW para las columnas tienen mayor cantidad de palabras (por ende, menos pesadas), tras evaluar distintas posibilidades, se llegaron a mejores resultados con este valor de filtro.

Por último, la fórmula de este cálculo es la misma que se emplea para obtener la pertenencia de los nombres de las tablas, y está dada por la ecuación (1), donde el numerador representa a la suma de los pesos de las palabras encontradas en una BoW, mientras que el denominador es el peso total de dicha bolsa:

$$Pertenencia = \frac{pesosEncontrados}{100} \quad (1)$$

3.3 Resultados de la Clasificación

GrACED genera varios tipos de resultados. Por un lado, para favorecer el análisis general de la base de datos, se obtiene un gráfico de torta con la proporción de tablas que contienen información de manufactura, y las que no. Esto es especialmente útil para analizar la distribución de los datos y la relevancia que cada empresa le da a los mismos.

Por otro lado, y de forma más específica, GrACED permite observar un gráfico de barras para cada tabla, mostrando las categorías en las que fue clasificada, y el porcentaje que obtuvo al ser analizada por nombre de tabla y nombre de columna. Es decir, que cada tabla puede tener pertenencia a más de una categoría, debido a la ambigüedad del lenguaje natural.

Las pertenencias que surgen del análisis de los nombres de tablas y de columnas no se combinan, dado que al mantenerse separados se obtiene una mayor riqueza al momento de analizarla; sin embargo, sí se muestra el promedio de ambos porcentajes de pertenencia. Esta combinación siempre pertenece a uno de los tres tipos de clasificación detallados a continuación:

• **Falsos Positivos (Tricky):** son clasificaciones en las que la pertenencia del nombre de la tabla es mucho mayor que la obtenida con los nombres de las columnas. Esto sucede en casos donde el nombre de la tabla tiene palabras muy específicas para una categoría, mientras que las columnas tienen palabras genéricas cuyos pesos son medios o bajos. Sin embargo, no se descartan porque más allá de la combinación de pertenencias obtenidas, la tabla puede contener información relevante.
• **Positivos Totales (True):** representan a aquellas clasificaciones donde ambas pertenencias tienen porcentajes altos, ya que fueron encontradas muchas palabras clave de peso elevado. Por lo general, muy pocas tablas por categoría pertenecen a este tipo.
• **Neutrales (Neutral):** son clasificaciones no englobadas en las anteriores, donde generalmente ambas pertenencias son de nivel medio, y sólo contienen palabras de relevancia intermedia, no muy importantes pero tampoco genéricas. Suelen ser tablas que almacenan información complementaria a las Positivas Totales.

4. Implementación

La implementación de un agente inteligente es una tarea compleja por lo que se decidió utilizar FAIA [19]: un framework generado en Java, que ofrece una estructura de clases abstractas que generan varios tipos de agentes inteligentes (reactivos, basados en metas, basados en conocimiento, etc.), y que sirven de marco para implementar la funcionalidad básica de todo agente (la entidad, el ambiente, estado del ambiente, estado del agente, percepciones y acciones).

A su vez, la base de conocimiento ha sido implementada en XML, como se mencionó previamente, debido a la portabilidad, flexibilidad y universalidad que este lenguaje ofrece, además de permitir una fácil modificación y agregado de palabras. Otra ventaja es que a partir de la versión Java 8, las librerías para la lectura/escritura de este tipo de archivos ya se encuentran incorporadas en el lenguaje, quitando la necesidad de utilizar archivos JAR externos.

4.1 Consideraciones de Sintaxis y Semántica

Como puede notarse, la implementación de un agente que procese lenguaje natural, siempre va a depender de dos situaciones, ajenas al mismo: la sintaxis y la semántica de las palabras. Si las palabras son escritas en idiomas que el agente no comprende, o con errores ortográficos, éste no podrá procesarlas. Lo mismo sucede si la semántica de las palabras no es utilizada adecuadamente; por ejemplo, si una columna se llama `cellphone_number` pero en realidad contiene un nombre de persona física, el agente generará una clasificación basada en el nombre de la columna, y no en el contenido de la misma, ya que la semántica de la etiqueta ha sido usada erróneamente.

Un punto importante relacionado con las palabras, es la separación de las mismas. Generalmente, en los lenguajes de programación se utilizan convenciones de nombres (o *naming conventions* por el nombre en inglés), que establecen métodos para separar las palabras. Las bases de datos actuales no tienen ninguna convención preestablecida -y aunque la tuvieran, no hay forma de asegurar que los desarrolladores las utilizarían- por lo que el problema de la separación no resulta trivial. Sin embargo, está fuera del alcance actual de GrACED.

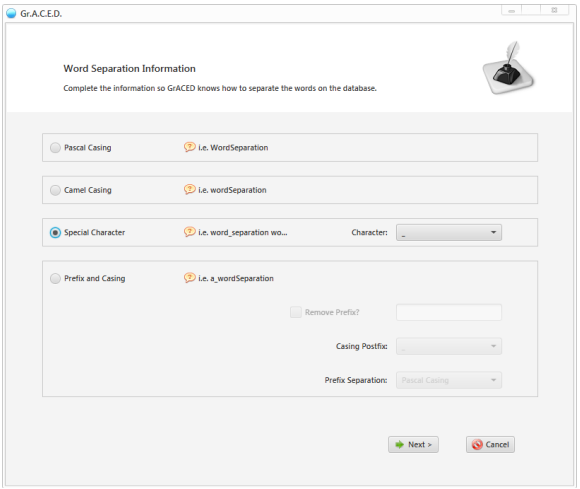

Fig. 6. Interface gráfica que solicita información sobre la separación de palabras.

Para solucionar esto, antes de comenzar la clasificación, el agente solicita que se le instruya qué método de separación va a emplearse, lo cual puede observarse en la Figura 8. Los tipos de separación comprendidos, por el momento, son:

- **Pascal Casing**: las palabras se escriben juntas, y cada una empieza con letra capitalizada. Ejemplo: `UnEjemploDePalabras`.
- **Camel Casing**: similar al anterior, la primera palabra lleva letra minúscula. Ejemplo: `unEjemploDePalabras`.
- **Separación por Caracteres**: las palabras son escritas en minúsculas, y cada una se separa usando un carácter especial (punto, espacio, guion medio, guion bajo). Ejemplo: `un_ejemplo_de_palabras`.
- **Separación Mixta**: es una separación más compleja y personalizable, y permite seleccionar un prefijo que será eliminado y no analizado, una separación para el prefijo del resto del nombre, y una para el nombre restante.

Cabe mencionar que si una base de datos no mantiene una semántica adecuada, ni consistencia en el método de separación de palabras, GrACED no clasificará a su máxima capacidad.

4.2 Casos de Estudio

Con el objetivo de realizar un primer test de la estructura del agente, sus acciones y procedimientos de clasificación, se realizaron dos casos de estudio, en donde sólo se ha utilizado una parte del grafo de clasificación generado a partir del estándar (Fig. 3): la rama de *Product Definition* y sus subnodos. El principal motivo de esta selección fue para, en la etapa de inicio, dejar de lado la herencia múltiple de algunas categorías, y concentrarse en el algoritmo de razonamiento, y el proceso de GrACED.

Para esto, se han estudiado las bases de datos de dos sistemas diferentes: Adempiere y OpenERP, los cuales serán detallados a continuación, y se dejaron como trabajo futuro evaluar Dolibarr [20] y Libertya [21].

4.2.1 Caso de Estudio: OpenERP

OpenERP [16] es una suite ERP de código abierto, publicado con una licencia AGPL2 [22] e implementado como una aplicación web. Su funcionamiento se centra en la lógica de negocios y en el módulo MRP. Esta suite también fue utilizada por los autores en el caso de estudio de una investigación previa [8].

La base de datos de OpenERP sí mantiene consistencia en la convención de nombres, usando siempre las letras en minúsculas, separadas con guiones bajos. En la Tabla 1 puede observarse un ejemplo de una tabla SQL de dicha suite:

Tabla 1. Tabla SQL de la base de datos de OpenERP.

Nombre de Tabla: `mrp_bom`			
Nombres de Columnas			
id	create_uid	create_date	write_date
write_uid	product_uos_qty	date_stop	code
sequence	product_qty	product_uos	product_efficiency
product_rounding	date_start	company_id	product_id
bom_id	routing_id	position	type

Como puede observarse, la separación es consistente tanto para el nombre de tabla como para las columnas, y las palabras empleadas no son sólo genéricas, sino que también posee varias palabras relevantes en la clase BOM. Tras realizar un estudio manual que implicó separar las palabras de la BD (provenientes tanto de los nombres de tablas como de los nombres de columnas) demostró que la correctitud en la separación de palabras para este ERP es de 97.4%.

Este ejemplo fue analizado con GrACED y algunos de los resultados pueden verse a continuación en la Fig. 9 (izquierda), donde se puede observar una de las pestañas con resultados generados por el agente, donde del total de tablas, el 11.12% contiene información sobre la categoría *Product Definition* y el 88.88% no.

Por otro lado, en la Fig. 9 (derecha) se observa la separación en tipos de clasificación, para todas las tablas que han sido categorizadas. Aquí se cuenta el total de clasificaciones, ya que una tabla puede pertenecer a más de una categoría; de esta forma, el 27.71% de las clasificaciones son del tipo Falsos Positivos (Tricky), el 63.85% son Neutrales, y el 8.44% son Positivos Totales (True).

Analizando estos resultados, observamos que:

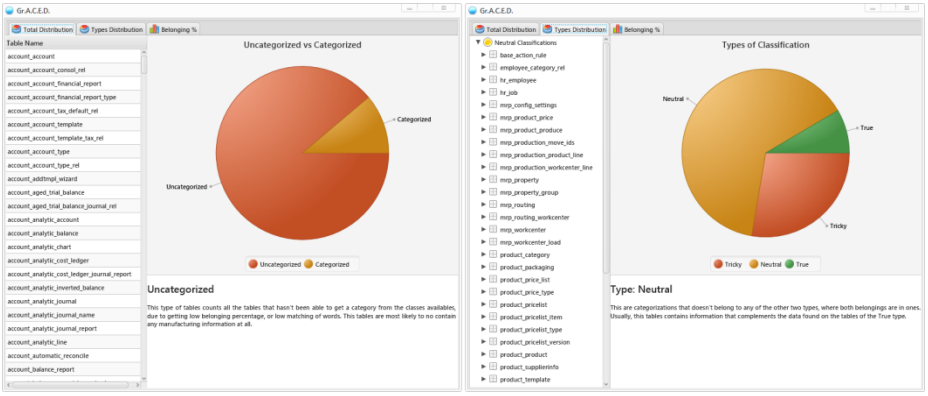

Fig. 9. Resultado comparando tablas categorizadas contra no categorizadas (izquierda). Resultado de las tablas categorizadas, separadas por tipo de categorización (derecha).

- GrACED declaró como Positivos Totales en una categoría a aquellas tablas que contienen la mayor cantidad posible de palabras relevantes (por ejemplo, la plantilla de un producto, o un nodo de la BOM), obteniendo una pertenencia mayor al 50% tanto en el nombre de la tabla, como en el de las columnas.
- Las Neutrales son tablas con información que emplea mayormente palabras genéricas, y que tiene pocas palabras claves. No porque una tabla tenga categorizaciones de tipo Neutral es "menos correcta", sino que GrACED no puede extraer la cantidad de datos semánticos necesarios como para obtener porcentajes de mayor envergadura.
- Los Falsos Positivos para OpenERP son tablas que tienen información complementaria para las otras categorizaciones, como ser las categorías de productos. Para ingresar a este tipo, se requiere que el porcentaje de pertenencia del nombre de la tabla sea menor al 25%, y que a su vez, la pertenencia obtenida por los nombres de columnas, sea menor que el obtenido con el nombre de tabla.

Para evaluar el comportamiento obtenido con GrACED se compararon los resultados obtenidos con el agente, contra una clasificación manual realizada por expertos. De esta forma, los expertos realizaron 64 categorizaciones, y GrACED coincidió con 53, lo que representa un 82.81% de coincidencias. A su vez, el agente agregó 26 categorizaciones, de las cuales 18 fueron posteriormente consideradas correctas por los expertos, tras estudiar el contenido de información y palabras de las mismas.

4.2.2 Caso de Estudio: Adempiere.

Adempiere [23] es un ERP desarrollado bajo una licencia GNU General Public License [24] resultante de un fork[2] de otro ERP de código libre, llamado Compiere.

Este sistema, desplegado con la base de datos en Oracle 11g XE, fue analizado uti-

[2] Un **fork** sucede cuando los desarrolladores copian el código fuente de un paquete de software y comienzan un desarrollo independiente sobre éste, creando un software distinto.

lizando GrACED. Sin embargo, si bien las tablas se encuentran en idioma inglés, esta suite funciona como un contra-ejemplo a la hora de mostrar los problemas que pueden surgir de un mal empleo de sintaxis y semántica. En la Tabla 2, puede encontrarse un ejemplo de una tabla SQL extraída de dicho ERP:

Tabla 2. Ejemplo de tabla SQL de la base de datos de Adempiere.

Nombre de Tabla: m_production			
Nombres de Columnas			
m_production_id	ad_client_id	ad_org_id	isactive
created	createdby	updated	updatedby
name	description	movementdate	iscreated
posted	processed	processing	ad_orgtrx_id
ad_project_id	c_campaing_id	c_activity_id	user1_id
user2_id			

El primer problema encontrado, es una separación de palabras inconsistente: en algunos casos se emplea un carácter especial (el guión bajo) pero en otros casos directamente no hay carácter separador, ni separación Pascal/Camel, dado que todos los nombres son o todos en mayúsculas, o todos en minúsculas, sin seguir ninguna convención de nombres. Por ejemplo, una de las columnas usa sólo guiones bajos para separar el nombre (como ser `m_production_id`) mientras que otras no emplean separación (como `movementdate`).

Otro problema en esta base de datos, es el uso de palabras genéricas como "`bname`" o "`description`", sin emplear otros modificadores que agreguen mayor valor semántico, lo que disminuye considerablemente el porcentaje de pertenencia obtenido al intentar clasificar las tablas de esta base de datos.

Dado que la separación de palabras no es un problema trivial, no poder desagregarlas adecuadamente -y de forma consistente- conlleva a un problema mayor para buscar dichas palabras en las BoW y en los archivos de sinónimos, resultando en una clasificación pobre.

Un despliegue completo y online de las tablas de la BD de Adempiere pueden encontrarse en Adempiere SchemaSpy [15].

Conclusiones

El presente trabajo propone la estructura básica para un agente inteligente basado en conocimiento y denominado GrACED, el cual trabaja con lenguaje natural y que utiliza una base de conocimiento generada a partir de la estructura de datos y modelos de información propuestos en el estándar ANSI/ISA-95.

GrACED se enlaza con un sistema ERP y permite analizar su base de datos para clasificar las tablas en cada una de las categorías propuestas por el ANSI/ISA-95 Parte I [4], y posteriormente obtener la adecuación de la base al estándar.

Su funcionalidad fue evaluada a través de dos casos de estudio, empleando siste-

mas ERP de código abierto: Adempiere y OpenERP, logrando tanto un comportamiento favorable, como así también alcanzando a demostrar la importancia de la semántica dentro de la composición de las bases de datos.

Como prototipo del proyecto, la implementación actual de GrACED ha logrado buenos resultados por lo que surgen varios trabajos futuros, entre ellos, lograr la utilización completa del grafo de clasificación, y evaluar dos casos más de estudio: Dolibarr y Libertya.

Otro punto importante, es lograr la propagación de pertenencia a las distintas categorías. Si se observa el grafo en la Figura 3, los arcos entre los nodos tienen etiquetas; estas etiquetas están asociadas a una fórmula de cálculo, la cual permite obtener el grado en que los nodos hijos componen a su/s padre/s. La propuesta es que, utilizando las pertenencias obtenidas en la clasificación básica desarrollada en este trabajo, se pueda propagar el porcentaje hacia arriba en el grafo con el objeto de encontrar el impacto que cada tabla tiene en el total de la información de manufactura contenida en la base de datos. Al lograr una pertenencia total, también puede estudiarse la adecuación de la base de datos al ANSI/ISA-95.

Bibliografía

[1] EU-Commission, MANUFACTURE - A vision for 2020. Assuring the future of manufacturing in Europe., Office for Official Publications of the European Communities, 2004.

[2] I. Harjunkoski, R. Nyström y A. Horch, «Integration of scheduling and control - Theory or practice?,» *Computers and Chemical Engineering*, vol. 33, pp. 1909-1918, 2009.

[3] E. Muñoz, E. Capón-García, A. Espuña y L. Puigjaner, «Ontological framework for entreprise-wide integrated decision-making at operational level,» *Computers and Chemical Engineering*, vol. 42, pp. 217-234, 2012.

[4] ISA, ANSI/ISA-95.00.01-2000. Enterprise-Control System Integration. Part 1: Models and terminology, ISBN: 1-55617-727-5, 2000.

[5] L. Prades, F. Romero, A. Estruch, A. García-Dominguez y J. Serrano, «Defining a Methodology to Design and Implement Business Process Models in BPMN according to the Standard ANSI/ISA-95 in a Manufacturing Enterprise,» *The Manufacturing Engineering Society International Conference, MESIC 2013*, vol. 63, pp. 115-122, 2013.

[6] C. Kardos, G. Popovics, B. Kádár y L. Monostori, «Methodology and data-structure for a uniform system's specification in simulation projects,» de *Forty Six CIRP Conference on Manufacturing Systems 2013*, 2013.

[7] D. Brandl, «Business to manufacturing (B2M) collaboration between business and manufacturing using ISA-95,» *Revue de l' electricite et de l' electronique*, nº 8, pp. 46-52, 2002.

[8] M. Vidoni y A. Vecchietti, «E2OL: Sistema de Planeamiento y Scheduling Personalizable e Integrable con ERPs,» de *1º Congreso Nacional de Ingeniería Informática y Sistemas de Información*, Córdoba, 2013.

[9] O. P. Quiñonez-Gámez y R. G. Camacho-Velázquez, «Validation of production data by using an AI-based classification methodology; a case in the Gulf of Mexico,» *Journal of Natural Gas Science and Engineering*, vol. 3, pp. 729-734, 2011.

[10] Y. Fu, W. Ke y J. Mostafa, «Automated Text Classification Using a Multi-Agent Framework,» de *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, Denver, Colorado, 2005.

[11] H. M. Wallach, «Topic Modeling: Beyond Bag-of-Words,» de *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburg, PA, 2006.

[12] ISA, ANSI/ISA-95.00.03-2005. Enterprise-Control System Integration. Part 3: nActivity models of manufacturing operations management, 1-55617-955-3 ed., ISA, 2005.

[13] W3C Recommendation, «Extensible Markup Language (XML) 1.1 (Second Edition),» 2006. [En línea]. Available: http://www.w3.org/TR/xml11/#sec-xml11. [Último acceso: 01 04 2014].

[14] P. Norvig y S. Russel, Artificial Intelligence: A Modern Approach, Prentice Hall, 2010.

[15] Adempiere, «SchemaSpy Analysis of Adempiere340,» 2012. [En línea]. Available: http://www.adempiere.com/schemaspy/. [Último acceso: 20 April 2014].

[16] OpenERP S.A., «OpenERP,» 2012. [En línea]. Available: https://www.openerp.com/. [Último acceso: 20 April 2014].

[17] Panorama Consulting Solutions, «ERPNext,» 19 Noviembre 2010. [En línea]. Available: http://panorama-consulting.com/erp-vendors/erpnext/. [Último acceso: 2014].

[18] NightLabs Consulting GmbH, «JFire,» 2011. [En línea]. Available: http://www.jfire.net/. [Último acceso: 2014].

[19] J. Roa, M. Gutierrez, M. Pividori y G. Stegmayer, «How to develop intelligent agents in an easy way with FAIA,» de *Quality and Communicability for Interactive Hypermedia Systems: Concepts and Practices for Design*, IGI global, ed. Francisco V. Cipolla Ficarra, 2010, pp. 120-140.

[20] L. Destailleur, «Dollibar ERP/CRM,» 2014. [En línea]. Available: http://www.dolibarr.org/. [Último acceso: 2014].

[21] F. Cristina, M. Mauprivez, M. Nerón Cap, J. M. Castro y F. Bonafine, «Libertya ERP,» 2011. [En línea]. Available: http://www.libertya.org/producto/preguntas-frecuentes. [Último acceso: 2014].

[22] GNU Affero, «Affero General Public Licence,» 2007. [En línea]. Available: http://www.gnu.org/licenses/agpl-3.0.html. [Último acceso: 20 April 2014].

[23] B. C. Pamungkas, «ADempiere 3.4 ERP Solutions,» Birmingham, UK, Packt Publishing, 2009.

[24] Free Software Foundation Inc., «GNU General Public Licence,» 29 June 2007. [En línea]. Available: https://gnu.org/licenses/gpl.html. [Último acceso: 2014 April 20].

[25] D. He, A. Lobov y J. L. Matinez-Lastra, «ISA-95 Tool for Enterprise Modeling,» de *ICONS 2012: The Seventh International Conference on Systems*, 2012.