



Universidad de Holguín
Facultad de Informática y Matemática

**Procedimiento para la obtención de un modelo ontológico para representar
como conocimiento la información contenida en bases de datos**

**Trabajo de Diploma en opción al título de
Ingeniero Informático**

Autor: Alfredo Vega Ramírez
Tutor: MsC. Irlán Grangel González
DrC. Inty Sáez Mosquera

Junio 2013

Dedicada a las personas que me vieron nacer y crecer
Mis Padres

*“...si supiese qué es lo que estoy haciendo, no lo llamaría
INVESTIGACIÓN...”*

Albert Einstein

Agradecimientos

Gracias a mi familia por el apoyo y el cariño en todos estos años.

Gracias a mi novia por su paciencia y apoyo en la realización de la tesis.

*Gracias mi tutor, amigo y ejemplo por haber confiado en mi para realizar este
trabajo.*

*Gracias a mis compañeros de trabajo por, las respuesta a cuantas preguntas
realicé y ayuda les pedí.*

Gracias a Dios por los talentos otorgados.

Resumen

El presente trabajo, propone un procedimiento a través del cual un razonador evalúa la información de una base de datos y la clasifica de forma automática en conceptos, entidades, relaciones, roles y atributos. Para realizar esta clasificación, se transforma la información de la base de datos a una ontología, a través de un algoritmo de migración.

Esta investigación surge por la necesidad de determinar el significado de la información tratada en los modelos de bases de datos, para lo cual se debe tener en cuenta el contexto donde se enmarca la misma. Lo que permite que se reutilice el modelo primario como un nuevo modelo dotado de mayor semántica, gracias a una información mejor definida.

El procedimiento propuesto se divide en dos métodos: primero, un *Algoritmo de migración*, el cual una vez obtenido un conjunto de elementos que componen el esquema relacional de una base de datos, genera una ontología con dichos elementos. El segundo método es la *Estrategia de clasificación de la información*, este consiste en una serie de consultas *SPARQL*, mediante las que se clasifica la información de la base de datos según las necesidades del estudio.

Finalmente se realizaron diez (10) casos experimentales con bases de datos relacionales en *PostgreSQL*, a través de los que se determinan la eficacia del algoritmo de migración y la validez de la clasificación obtenida mediante la estrategia de razonamiento.

Abstract

This work proposes a process to evaluate the information in databases and automatically classify concepts, entities, relationships, roles and attributes. To perform this classification a database is converted to an ontology through a migration algorithm.

This research arises from the need to determine the meaning of the information covered in the database models, for which must be taken into account the context in which it is framed. This allows to reuse the primary model as a new model to have a greater semantic, thanks to better defined information.

The proposed procedure is divided into two methods: first, a migration algorithm, in which after obtaining a set of elements comprising the relational schema of a database, generates an ontology with such elements. The second method is the strategy classification of information, this is a series of SPARQL queries. Those queries are used to classify the information contained in the database.

Finally there were ten (10) cases experimental relational database PostgreSQL, through determining the effectiveness of migration algorithm and the validity of the classification obtained by reasoning strategy.

Índice

Lista de Figuras	xi
Lista de Tablas	xiii
Introducción	1
1 Marco teórico-referencial de la investigación	7
1.1 Análisis de los Modelos de Datos	7
1.1.1 Modelo Entidad-Relación	8
1.1.1.1 Conceptos básicos del modelo E-R	8
1.1.2 Modelo Relacional	9
1.1.2.1 Componentes del modelo relacional	9
1.1.2.2 Esquema de la base de datos	11
1.1.3 Diferencias entre los modelos Entidad-Relación y Relacional	12
1.1.4 Lenguajes de Consultas de Bases de Datos	13
1.1.4.1 Estructura de las consultas en <i>SQL</i>	14
1.1.5 Gestores de Bases de Datos	14

1.1.5.1	Características fundamentales de los SGBD	15
1.1.5.2	<i>MySQL</i>	16
1.1.5.3	<i>PostgreSQL</i>	16
1.1.5.4	<i>ORACLE</i>	17
1.2	Ontologías	18
1.2.1	Definición de ontologías	18
1.2.2	Lenguajes para el desarrollo de ontologías	20
1.2.2.1	<i>RDF</i>	20
1.2.2.2	<i>RDF Schema</i>	21
1.2.2.3	<i>OWL</i>	21
1.2.2.4	<i>OWL 2</i>	21
1.2.2.5	<i>SPARQL</i>	22
1.2.3	Herramientas para el trabajo con las ontologías	22
1.2.4	Editores de ontologías	23
1.2.4.1	<i>KAON</i>	23
1.2.4.2	<i>Protégé</i>	23
1.2.4.3	<i>Topbraid</i>	24
1.2.5	Razonadores	25
1.2.5.1	<i>Pellet</i>	25
1.2.5.2	<i>Racer</i>	26
1.2.5.3	<i>Hermit</i>	26
1.2.5.4	<i>TRIPLE</i>	26

1.2.6	Bibliotecas	27
1.2.7	<i>Framawork</i> para el trabajo con ontologías	27
1.2.7.1	<i>Jena</i>	27
1.2.7.2	<i>OWL API</i>	28
1.2.8	Construcción y desarrollo de ontologías	29
1.2.8.1	Terminología básica	29
1.3	Comparación entre los modelos de BD y ontologías	33
1.3.1	Niveles de representación	33
1.3.2	La semántica en ontologías y en el modelo relacional	34
1.4	Similitudes de las BD y las ontologías	35
1.5	Generación de modelos ontológicos a partir de bases de datos	36
1.5.1	Deficiencias de los modelos de BD	38
1.5.2	Principales beneficios del empleo de las ontologías	39
1.6	Sistemas de acceso a BD mediante ontologías	40
1.6.1	Descripción del <i>plugin</i> DIG-MASTRO para Protégé	40
1.6.2	Descripción del algoritmo <i>SemanticMapper</i>	41
1.7	Determinación de eficacia	43
1.8	Conclusiones parciales	45
2	Procedimiento para la construcción del modelo ontológico	47
2.1	Alcance del modelo	47
2.2	Estrategia de solución al problema planteado	48
2.2.1	Análisis del esquema de las Bases de Datos	49

2.2.1.1	Extracción de los elementos del esquema de la BD	50
2.2.2	Descripción del algoritmo de migración	52
2.2.3	Descripción del procedimiento para la clasificación de la información . . .	62
2.2.4	Resultados obtenidos del procedimiento	65
2.2.4.1	Resultados obtenidos del algoritmo de migración: Caso experi- mental 1	65
2.2.4.2	Resultados obtenidos con el procedimiento de clasificación de la información	70
2.3	Conclusiones parciales	73
2.4	Conclusiones generales	74
2.5	Recomendaciones	74
Referencias Bibliográficas		75
3 Anexos		79
3.1	Anexo 1. Diagrama de secuencia de procedimiento propuesto	80
3.2	Anexo 2. Tipos de datos soportados en bases de datos y sus equivalentes en ontologías	82
3.3	Anexo 3. Resultados del algoritmo de migración para cada caso experimental .	82
3.4	Anexo 4. Clasificación de la información para cada caso experimental	91

Lista de Figuras

1.1	Esquema de la base de datos. Fuente: elaboración propia.	11
1.2	Conjuntos de clases a representar en Protégé. Fuente: elaboración propia. . . .	30
1.3	Conjunto de propiedades que enlazan clases (individuos) mediante relaciones. Fuente: elaboración propia	30
1.4	Taxonomía que muestra una jerarquía de clases. Fuente: elaboración propia . . .	31
1.5	Propiedades de una clase, reflejadas en una ontología en Protégé. Fuente: elaboración propia.	32
1.6	Axioma representado de una ontología. Fuente: elaboración propia.	32
1.7	Tabla <i>cliente</i> . Fuente: elaboración propia.	38
1.8	Arquitectura de los sistemas ODBA. Fuente: Tomada de (29).	41
1.9	Proceso del algoritmo <i>SemanticMapper</i> . Fuente: tomado de (45)	42
2.1	Arquitectura del modelo ontológico propuesto en la presente investigación. Fuente: elaboración propia.	48
2.2	Esquema de la base de datos Control del proceso 1 utilizada para ejemplificar el proceso. Fuente: elaboración propia.	50
2.3	Resultado de la consulta <i>SQL</i> realizada a la tabla <i>constraint-column-usage</i> del esquema de la base de datos. Fuente: elaboración propia.	53

2.4	Resultado de la consulta <i>SQL</i> realizada a la tabla <i>information-schema.columns</i> del esquema de la base de datos. Fuente: elaboración propia.	55
2.5	Resultado de la consulta <i>SQL</i> realizada a la tabla <i>pg-catalog.pg-constraint</i> del esquema de la base de datos. Fuente: elaboración propia.	58
2.6	Identificación de una relación (1:1) tras consultar la tabla <i>pg-catalog.pg-constraint</i> . Fuente: elaboración propia.	59
2.7	Declaración de instancias de clases en el modelo ontológico, mediante las tuplas de la tabla <i>pieza</i> . Fuente: elaboración propia.	60
2.8	Asociación de instancias y propiedades de datos, correspondientes a la clase <i>pieza</i> . Fuente: elaboración propia.	61
2.9	Ontología resultante una vez aplicado el algoritmo de migración a la BD Control del Proceso 1. Fuente: elaboración propia.	67
2.10	Valores de eficacia por cada caso experimental. Fuente: elaboración propia.	70
2.11	Porcientos que representan cada elemento clasificado en la composición de los elementos de una base de datos. Fuente: elaboración propia.	73
3.1	Diagrama de secuencia del Modelo ontológico propuesto. Fuente: elaboración propia.	81

Lista de Tablas

1.1	Equivalencia entre los elementos de la ontología y el esquema de la BD. Fuente: elaboración propia.	12
2.1	Muestra de tipos de datos soportados en BD mapeados a tipos de datos soportados por las ontologías. Fuente: elaboración propia.	56
2.2	Bases de datos relacionales utilizadas en los casos experimentales. Fuente: elaboración propia.	66
2.3	Elementos que componen la ontología resultante de aplicar el algoritmo de migración a la BD Control del Proceso 1. Fuente: elaboración propia.	66
2.4	Correspondencias entre tablas en la BD y conceptos en la ontología resultante . Fuente: elaboración propia.	68
2.5	Selección de correspondencias entre atributos en la BD y propiedades de datos en la ontología resultante. Fuente: elaboración propia.	68
2.6	Correspondencias entre fk en la BD y propiedades de objetos en la ontología resultante. Fuente: elaboración propia.	69
2.7	Clasificación de la información contenida en la base datos utilizada en el caso experimental 1. Fuente: elaboración propia.	71
2.8	Clasificación de conceptos, roles, atributos y relaciones para cada caso experimental. Fuente: elaboración propia.	72

3.1	Tipos de datos soportados en BD mapeados a tipos de datos soportados por las ontologías. Fuente: elaboración propia.	82
3.2	Elementos que componen la ontología resultante de aplicar el algoritmo de migración a la BD <i>Air Reservation</i> . Fuente: elaboración propia.	82
3.3	Elementos que componen la ontología resultante de aplicar el algoritmo de migración a la BD BMEWS. Fuente: elaboración propia.	84
3.4	Elementos que componen la ontología resultante de aplicar el algoritmo de migración a la BD <i>Children Institution</i> . Fuente: elaboración propia.	85
3.5	Elementos que componen la ontología resultante de aplicar el algoritmo de migración a la BD <i>Gaming Web Sites</i> . Fuente: elaboración propia.	86
3.6	Elementos que componen la ontología resultante de aplicar el algoritmo de migración a la BD <i>Open University</i> . Fuente: elaboración propia.	88
3.7	Elementos que componen la ontología resultante de aplicar el algoritmo de migración a la BD <i>Theater Booking</i> . Fuente: elaboración propia.	89
3.8	Muestra de la clasificación de la información contenida en la base dato utilizada en el caso experimental 2 <i>Air Reservation</i> . Fuente: elaboración propia.	91
3.9	Muestra de la clasificación de la información contenida en la base dato utilizada en el caso experimental 3 BMEWS. Fuente: elaboración propia.	92
3.10	Muestra de la clasificación de la información contenida en la base dato utilizada en el caso experimental 4 <i>Children Institution</i> . Fuente: elaboración propia.	92
3.11	Muestra de la clasificación de la información contenida en la base dato utilizada en el caso experimental 5 <i>Children of Family Support</i> . Fuente: elaboración propia.	93
3.12	Muestra de la clasificación de la información contenida en la base dato utilizada en el caso experimental 6 <i>Gaming Web Sites</i> . Fuente: elaboración propia.	93
3.13	Muestra de la clasificación de la información contenida en la base dato utilizada en el caso experimental 7 ISWC. Fuente: elaboración propia.	94
3.14	Muestra de la clasificación de la información contenida en la base dato utilizada en el caso experimental 8 <i>National Healt</i> . Fuente: elaboración propia.	94

3.15 Muestra de la clasificación de la información contenida en la base dato utilizada
en el caso experimental 9 *Open University*. Fuente: elaboración propia. 95

Introducción

Actualmente el éxito de la “Sociedad de la Información” produce un crecimiento exponencial de la información digital, que a diario se publica en la *Web*. Las tareas de búsqueda, recuperación, reutilización e integración de la información son cada vez más arduas, debido al carácter heterogéneo que presenta el contenido de las fuentes de datos.

La mayor parte de dicha información se encuentra almacenada en grandes bases de datos, las que son diseñadas para gestionar grandes cantidades de información(1). Las bases de datos contienen información relevante para una empresa, la que es manejada mediante un Sistema Gestor de Bases de Datos, los que tienen como objetivo principal proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente(1).

Estos sistemas de bases de datos son una colección de archivos interrelacionados y un conjunto de programas que permiten a los usuarios acceder y modificar estos, proporcionando una visión abstracta de los datos(15). Además las bases de datos se han instaurado como herramientas imprescindibles para el almacenamiento y consulta de grandes cantidades de información. Sin embargo, los métodos de consulta requieren cada vez una mayor expresividad y potencia de lenguaje(23).

Por otra parte, una de las deficiencias del actual modelo de gestión de información en la *Web* es el significado de la información contenida en estos modelo de datos, puesto que es un acuerdo informal entre los desarrolladores y los usuarios de dicho modelo(45), lo que implica que dicha información no esté explícita de manera alguna¹, sino que se encuentran implícitas en el “entorno operacional” de la fuente de datos, es decir, en las relaciones contenidas de las fuentes de datos.

¹Esta ausencia se encuentra en el hecho de que históricamente se ha dado prioridad al rendimiento, reclusando a las fases de diseño toda especificación semántica del contenido, excluyéndola completamente de la fase de ejecución.

En adición a lo anterior la semántica, la cual tiene como noción: la descripción precisa de la relación existente entre una representación particular (un modelo) y los conceptos del mundo real que dicha representación trata de modelizar. En cuanto a estos modelos según la literatura se define como una **semántica formal**(45).

Esta describe una base de datos como un modelo lógico de su esquema relacional, es decir, sus relaciones, atributos y restricciones de integridad(45), por ejemplo, un modelo relacional consiste en eliminar redundancias en los datos de una relación o dependencias incompatibles con las formas normales descritas en las normalizaciones, con la finalidad de optimizar los tiempos de respuesta a preguntas a la base de datos.

De lo antes expuesto se origina la principal deficiencia de estos modelos, el **significado de la información** tratada, para lo cual se debe tomar en cuenta el contexto donde se enmarca dicha información (45).

Esta deficiencia tiene su origen en la semántica de estos modelos, por el hecho de que la mayoría de los sistemas de bases de datos se basan en el modelo relacional que consta de una menor semántica, dado su simplicidad conceptual al representar los datos a través de tablas(15).

Respecto a lo anterior, el lenguaje de representación para estos modelos de datos Lenguaje de Consulta Estructurado (*SQL*, por sus siglas en inglés), en su definición carece de expresividad, pues carece de una estructura para la descripción de recursos. Este solo esta enfocado en el planteamiento de preguntas en bases de datos relacionales y permitir la definición de esquemas relacionales(45).

Otra deficiencia a la que se hace referencia es el hecho que, las bases de datos se diseñan para una aplicación específica a la que dará soporte. En la mayor parte de los casos sin ninguna pretensión de generalidad ni de reutilización, *a priori* se trata de **modelos locales**(26).

En correspondencia con las deficiencias antes mencionades se añade, los modelos de bases de datos funcionan según el Supuesto del Mundo Cerrado(*CWA*, por sus siglas en inglés), en los cuales la información que se omite es falsa. No así los modelos que trabajan con el Supuesto del Mundo Abierto (*OWA*, por sus siglas en inglés) en los cuales la información que se omite es desconocida (podría ser verdadera o falsa) y en los que podemos añadir nuevo conocimiento “fácilmente”. No así en las bases de datos, en las que habría que desarrollar nuevamente el proceso de diseño(25).

Referente a estas problemáticas Stefano Ceri planteó(4): “los tres problemas de investigación

más importantes en Bases de Datos solían ser 'rendimiento', 'rendimiento' y 'rendimiento'; en los próximos años los tres problemas más importantes y que verdaderamente supondrán un desafío para la investigación serán 'la semántica', 'la semántica' y 'la semántica' ”.

Como solución a la anterior deficiencia Tim Bernes Lee propone la *Web semántica*, la cual fue propuesta con el objetivo de: “exponer en la *Web* la gran cantidad de bases de datos relacionales existentes de modo que puedan ser procesadas automáticamente”(27).

La *Web semántica*¹, es una *Web* extendida, dotada de mayor significado, en la que cualquier usuario en *Internet* puede encontrar respuestas a sus preguntas de forma más rápida y sencilla gracias a una información mejor definida(28), gracias al empleo de las ontologías dentro de su arquitectura.

Las ontologías, se definen como “Una especificación explícita de una conceptualización”(16) las cuales permiten clasificar la información y extender la funcionalidad; ya que en su representación jerárquica se describe un dominio específico de conocimientos, mediante relaciones arbitrarias entre conceptos y axiomas(19). Permitiendo un entendimiento común de la estructura del conocimiento, entre personas o agentes de *software*.

Estas formas de representación de conocimiento permiten el manejo de información, la integración de datos, la interoperabilidad entre sistemas de información heterogéneos y la *Web semántica*, constituyendo un proceso evolutivo de construcción permanente(22).

En términos de los sistemas de información, las ontologías constituyen un artefacto que, codifica una descripción del mundo (real, posible, hipotético, imposible, etc.), para algún propósito(41).

Esta representación consiste en jerarquías de conceptos con atributos y relaciones, que establecen una terminología para definir redes semánticas de conceptos e instancias, interrelacionados entre sí, para describir conocimiento de un dominio, que se almacena en una base de conocimiento(8).

Por otra parte una base de conocimiento es un componente del *software* que representa una colección de información ontológicamente descrita y procesada la que provee una comprensión compartida acerca de ciertos conceptos claves en el dominio(25). Las bases de conocimiento ontológicas tienden a estar orientadas (aunque no siempre) a almacenar una gran cantidad de conocimiento, con mucho mayor nivel de detalle de lo que suele contemplarse en bases de datos, dado que esta consta de una capacidad expresiva considerablemente mayor(8).

¹<http://web3.0websemantica.comuf.com/>

Las tecnologías semánticas pueden registrar directamente conceptos que se extienden a lo largo de un número vasto de fuentes, esto posibilita resultados rápidos, relativos y comprensivos. Actualmente estas ventajas son aprovechadas para el mejoramiento integral del proceso, a partir de un mejor entendimiento del dominio de aplicación en campos tales como: Estadística, Bases de Datos, Aprendizaje Automático, Teoría de la Información, Computación Paralela, entre otros(21).

Actualmente, los sistemas de bases de datos más empleados están basados en el modelo relacional, que consta de una menor semántica dado su simplicidad conceptual al representar los datos a través de tablas. Además estas bases de datos se diseñan para una aplicación específica a la que dará soporte. En la mayor parte de los casos sin ninguna pretensión de generalidad ni de reutilización *a priori* se trata de **modelos locales**. Pues no existe un procedimiento capaz de determinar el significado de la información contenida en dichas base de datos, pues no existe un modelo con la suficiente expresividad para clasificar en conceptos, entidades (sus materializaciones en los dominios específicos) relaciones, roles y atributos dicha información, lo cual permitirá la generalización y reutilización de dichos modelos.

De lo antes expuesto se determina el siguiente **problema científico**:

¿Cómo evaluar en conceptos, entidades (sus materializaciones en los dominios específicos) relaciones, roles y atributos la información almacenada en una base de datos?

Objeto

Clasificar la información almacenada en una base de datos en *PostgreSQL* como conceptos, relaciones, atributos y roles.

Objetivo

Desarrollar un procedimiento para la obtención de un modelo ontológico para clasificar la información contenida en una base de datos en *PostgreSQL* como conceptos, entidades (sus materializaciones en los dominios específicos), relaciones, roles y atributos.

Campo de acción de la investigación

Clasificación de la información contenida en una base de datos en *PostgreSQL* empleando un modelo ontológico.

Preguntas científicas:

- ¿Qué modelo de base de datos se debe escoger para realizar el estudio?
- ¿Qué aspectos se deben tener en cuenta para determinar conceptos, relaciones, roles y atributos?
- ¿Cómo modelar la información de una base de datos como conceptos, relaciones, roles y atributos?
- ¿Qué herramientas permiten interactuar con base de datos mediante ontologías?
- ¿Qué elementos se deben tener en cuenta para el desarrollo de la estrategia de razonamiento?

Tareas científicas:

1. Analizar las características de las base de datos, con el fin de identificar los elementos a tener en cuenta para la investigación.
2. Realizar un estudio de dominio para la definición de un modelo capaz de clasificar la información de la base de datos, según las necesidades del estudio.
3. Estudiar las herramientas de acceso a base de datos mediante ontología. Se analizarán las diversas variantes de acceso de cada herramienta, para determinar la más óptima.
4. Desarrollar una estrategia de razonamiento, mediante la cual un razonador clasifique la información contenida en una base de datos.
5. Proponer un procedimiento para evaluar la información de la base de datos de acuerdo con el modelo ontológico.

Métodos Teóricos

Análisis y síntesis: para el marco teórico del análisis de la ingeniería ontológica, el estudio de los aspectos fundamentales de las base de datos y elaboración de la investigación.

Histórico -Lógico: comprensión del objeto y del campo de acción.

Inductivo - Deductivo: para analizar los resultados se conseguirán soluciones óptimas para la fundamentación.

Métodos Empíricos

Entrevista: al personal especializado para conocer los aspectos principales de las bases de datos a tener en cuenta en la investigación.

Consulta de documentos: para determinar la información que se utilizará para la confección de este trabajo.

Estructura de la tesis

Este trabajo está compuesto por, Introducción, Capítulo 1, Capítulo 2 y Anexos. En el Capítulo 1 se abordan los aspectos fundamentales de las Bases de Datos y las Ontologías, así como las principales deficiencias de las bases de datos y los beneficios de las ontologías ya que la presente investigación propone la solución a los problemas de un modelo mediante los beneficios del otro. Igualmente se analizarán los principales sistemas que permiten el acceso a bases de datos mediante ontologías.

En el Capítulo 2, se describe el modelo ontológico planteado como solución en la presente investigación. Se realiza un análisis de los dos puntos fundamentales del modelo propuesto, el **algoritmo de migración** y la **estrategia de clasificación de la información**, de estos se recogen sus principales características y resultados obtenidos.

Capítulo 1

Marco teórico-referencial de la investigación

A continuación se analizan varias cuestiones relacionadas con los modelos de Bases de Datos(BD). Se analizarán los diversos modelos existentes como, los modelos relacionales, el entidad-relación, también se tendrán en cuenta, los lenguajes de consultas de BD y los gestores de bases de datos.

Por otro lado, se abordarán temas relacionados con el desarrollo de ontologías, estos temas se enfocarán principalmente en, definiciones de ontologías y los diversos lenguajes para el desarrollo de estas formas de representación de conocimientos. De igual forma, se tratarán las herramientas para el trabajo con ontologías, las cuales se dividen en: editores, razonadores y bibliotecas. De estas herramientas se expondrán, las características fundamentales de estas.

Por último se presentan diferentes aspectos relacionados con los sistemas de acceso a base de datos desde ontologías. De estos se tratarán las principales características de cada uno y las ventajas que ofrecen para lograr con efectividad el objetivo de esta investigación.

1.1 Análisis de los Modelos de Datos

Los modelos son una construcción mental a partir de la realidad en la que se reproducen los principales componentes y relaciones del segmento de la realidad analizada(47). Por tanto, un modelo de datos es una colección de herramientas conceptuales para la descripción de datos, relaciones entre datos, semántica de los datos y restricciones de consistencia(1).

En esta parte se analizarán dos modelos de datos: el modelo entidad-relación, este modelo se

basa en una percepción de un mundo real que consiste en una colección de objetos básicos, denominados entidades, y de relaciones entre estos objetos(1). El otro modelo que se analiza es el relacional, este utiliza una colección de tablas para representar tanto los datos como las relaciones entre los datos(15).

1.1.1 Modelo Entidad-Relación

El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real consistente en objetos básicos llamados *entidades* y *relaciones* entre estos objetos. Este modelo se desarrolló para facilitar el diseño de bases de datos permitiendo la especificación de un esquema de la empresa que representa la estructura lógica completa de una base de datos(1).

Por otra parte, el modelo de datos E-R es uno de los diferentes modelos de datos semánticos; el aspecto semántico del modelo yace en la representación del significado de los datos. El modelo E-R es extremadamente útil para hacer corresponder los significados e interacciones de las empresas del mundo real con un esquema conceptual(1).

1.1.1.1 Conceptos básicos del modelo E-R

Hay tres nociones básicas que emplea el modelo de datos E-R, propuestas por Chen mediante las que tratan de modelizar el mundo como un conjunto de entidades asociadas entre sí a través de: conjuntos de entidades, conjuntos de relaciones y atributos(9).

Una **entidad** se define como cualquier cosa que pueda ser identificada claramente. Equivale a un concepto, una categoría concreta o abstracta, un conjunto. Simplificando la definición anterior es una cosa u objeto en el mundo real que es distinguible de todos los demás objetos, por ejemplo, cada persona en desarrollo es una entidad.

Una **relación** por su parte representa cualquier asociación entre entidades y se caracteriza por su grado¹ y su cardinalidad².

Tanto las entidades como las relaciones se describen mediante un conjunto de **atributos** (clasificables en simples, compuestos, monovaluados, multivaluados, almacenados o derivados).

¹El número de entidades que asocia (binaria, ternaria, etc.)

²Número de ocurrencias (instancias) de cada una de las entidades asociadas mediante la relación

Estos toman valores de lo que muchos autores denominan conjuntos de valores y equivalen a los dominios definidos en el modelo relacional.

En adición a lo anteriormente expuesto, las instancias de entidades son a su vez entidades por eso Chen distingue **tipos de entidades**: para referirse a la descripción del esquema o intensión y **conjunto de entidades**: para referirse a la extensión o colección de las instancias de dicho esquema de entidad. De manera análoga se describen **tipos de relaciones** y **conjuntos de relaciones**.

Posteriormente en (7), (37) y (48), se propone una serie de extensiones (fundamentalmente la inclusión de relaciones de generalización y de agregación para permitir la definición de jerarquía de las entidades). Al modelo propuesto por Chen dando lugar a lo que se conoce como modelo Entidad Interrelación Extendido (EER).

En el presente trabajo no se hará énfasis en este modelo, debido a que el objetivo del trabajo es la evaluación de la información contenida en la BD. Los sistemas de bases de datos actuales están desarrollados mediante el modelo relacional, no en el E-R a pesar de que este sea considerado su antecesor conceptual.

1.1.2 Modelo Relacional

El modelo relacional es un modelo de menor nivel(1), con el objetivo de independizar la estructura lógica del modo de almacenamiento y cualquier detalle de tipo físico. Dicho modelo estructura los datos de manera lógica en lo que denomina relaciones, representadas en forma de tablas(10).

1.1.2.1 Componentes del modelo relacional

En el modelo relacional se distingue cuatro tipos de elementos: **relaciones, atributos, claves y tuplas**(15).

Para describir cada uno de estos elementos primero se tratarán dos conceptos fundamentales: **dominio y producto cartesiano**(10).

- **Dominio**: es un conjunto de valores del mismo tipo, por ejemplo todos los posibles ISBN de un inventario de libros o todas las posibles fechas de nacimiento de un conjunto de

personas. Un dominio se considera simple si sus valores no pueden descomponerse

- Dados n dominios (no necesariamente distintos) $D_1, D_2, D_3, \dots, D_n$ el **producto cartesiano** $D_i : i=1, 2, \dots, n$ es el conjunto de todas las n -tuplas $t_1, t_2, t_3, \dots, t_n$ tales que t_i pertenece D_i para todo i

a) Relaciones: los matemáticos la definen como subconjuntos del producto cartesiano de la lista de dominios, además se denomina de grado n , representa una simple tabla de dos dimensiones, consistente en filas y columnas de datos.

b) Atributos: es un identificador para cada uno de los dominios sobre los que se define la relación (recuérdese que el número de dominios diferentes podría ser menor que n).

c) Claves: permite identificar un conjunto de atributos suficiente para distinguir las entidades entre sí, también ayudan a identificar unívocamente a las relaciones y así a distinguir las relaciones entre sí.

Las Claves se clasifican en dos formas **Claves primarias** y **Claves candidatas o foráneas**.

Las **Claves primarias**, es una clave candidata que es elegida por el diseñador de la base de datos como elemento principal para identificar las entidades dentro de un conjunto de entidades.

Las **Claves candidatas o foráneas**, es cualquier conjunto de atributos que pueden ser elegidos como una claves de una relación. Si K es una clave candidata para la relación R , es un subconjunto de atributos de R con las siguientes propiedades (no contingentes, sino esenciales, permanentes):

1- No hay dos filas de R que tengan el mismo valor para K .

2- Si se elimina cualquiera de los atributos de K la anterior propiedad de unicidad se pierde.

d) Tuplas: es un conjunto de pares $(A:v)$, donde A es un atributo y v es un valor perteneciente al dominio de A : Atributo1 :valor1 , Atributo2 :valor2 , Atributo3 :valor3 , ..., Atributon :valorn.

De los componentes antes mencionados y a consideración del autor la presente investigación se centrará en las **Claves**, ya que mediante estas se logra:

- Las claves primarias para identificar las entidades en la base de datos

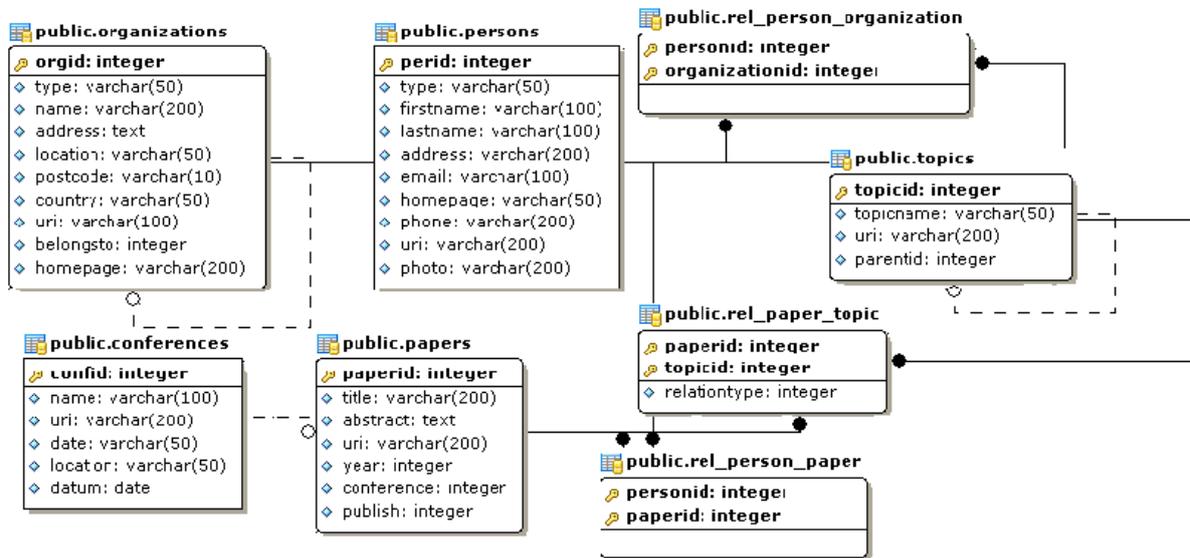


Figura 1.1: Esquema de la base de datos. Fuente: elaboración propia.

- Las claves foráneas permiten inferir relaciones entre entidades
- Las claves foráneas para determinar clases y subclases

1.1.2.2 Esquema de la base de datos

Cuando se habla de bases de datos se debe diferenciar entre el **esquema de la base de datos**, o diseño lógico de la misma, y el **ejemplar de la base de datos**, que es una instantánea de los datos de la misma en un momento dado(15).

Un esquema de BD describe la estructura de una Base de datos, en un lenguaje formal soportado por un Sistema Gestor de Base de datos. En una base de datos relacional, el esquema define sus tablas, sus campos en cada tabla y las relaciones entre cada campo y cada tabla. En la figura 1.1 se muestra un esquema relacional de base de datos.

El esquema es generalmente almacenado en un Diccionario de Datos. Aunque generalmente el esquema es definido en un lenguaje de base de datos, el término se usa a menudo para referirse a una representación gráfica de la estructura de base de datos (Diseño lógico de la base de datos).

En el siguiente trabajo, de acuerdo con lo anterior analizado se empleará directamente el esquema de la BD para la generación del modelo ontológico, dado que este contiene toda la información concerniente a la BD, así como las relaciones entre cada uno de los elementos de conforman el mismo. Además el modelo ontológico resultante será estructurado empleando los elementos del esquema según lo mostrado en la tabla 1.1.

Elementos de la ontología	Elementos del esquema
Clases de la ontología	Tablas de la base de datos
Axioma <i>hasKey</i>	Llaves de las tablas
Propiedades de datos	Atributos de las tablas
Propiedades de objetos	Tablas que intervienen en las relaciones de la BD
Instancias	Tuplas de la tablas

Tabla 1.1: Equivalencia entre los elementos de la ontología y el esquema de la BD. Fuente: elaboración propia.

1.1.3 Diferencias entre los modelos Entidad-Relación y Relacional

La siguiente sección está dirigida a resaltar las principales diferencias entre los modelos de datos Entidad-Relación y el modelo Relacional.

Como principal diferencia se resalta que el modelo E-R a pesar de considerarse el “antecesor conceptual” del modelo Relacional, es un modelo de datos de alto nivel. Está basado en una percepción de un mundo real que consiste en una colección de objetos básicos, denominados **entidades**, y de **relaciones** entre estos objetos. Mientras que el modelo Relacional es un modelo de menor nivel. Usa una colección de tablas para representar tanto los datos como las relaciones entre los datos(1).

En cuanto a la **semántica** de los modelos, el aspecto semántico del modelo E-R yace en la representación del significado de los datos. El modelo E-R es extremadamente útil para hacer corresponder los significados e interacciones de las empresas del mundo real con un esquema conceptual. A diferencias del modelo relacional que consta de una menor semántica dado su simplicidad conceptual al representar los datos a través de tablas(15).

A pesar de su simplicidad conceptual, actualmente una vasta mayoría de productos de bases de datos se basan en el modelo relacional. Los diseñadores formulan generalmente el diseño del esquema de la base de datos modelando primero los datos en alto nivel, usando el modelo E-R, y después traduciéndolo al modelo relacional, por esto en la presente investigación se centrará la atención en el modelo **Relacional**.

1.1.4 Lenguajes de Consultas de Bases de Datos

El lenguaje de consultas de BD se basa en el álgebra relacional, cuyas operaciones básicas permitirán extraer la información deseada de las relaciones sobre las que se aplican(10).

Unión, Intersección, Diferencia y Producto Cartesiano: puesto que las relaciones son conjuntos, los operadores usuales para conjuntos serán aplicables (los tres primeros restringidos al caso de relaciones compatibles, es decir, que tengan el mismo conjunto de atributos de modo que el resultado de la operación sea a su vez una relación).

Selección o restricción: Sobre una relación R , escrito $R[f(at_1, at_2, at_3, \dots, at_n)]$ es el conjunto de tuplas de R que satisfacen la expresión *booleana* f .

Proyección: sobre un conjunto de atributos, escrito $R[at_1, at_2, at_3, \dots, at_n]$ es la relación resultante de eliminar todas las columnas de R excepto las correspondientes a los atributos especificados en $at_1, at_2, at_3, \dots, at_n$.

Reunión: sobre dos relaciones R y S , escrito $R[f(at_{R1}, \dots, at_{Rn}, \dots, at_{S1}, \dots, at_{Sm})]S$, es el subconjunto del producto cartesiano de las dos relaciones R y S que satisfacen la expresión *booleana* f .

El Lenguaje de Consulta Estructurado (*SQL*, por sus siglas en inglés) es el lenguaje más común para plantear preguntas en bases de datos relacionales. Implementa las operaciones del álgebra relacional mencionadas anteriormente además de permitir la definición de esquemas relacionales.

SQL contiene otras capacidades además de la consulta en bases de datos. Incluye características para definir la estructura de los datos, para la modificación de los datos en la base de datos y para la especificación de restricciones de seguridad(1).

1.1.4.1 Estructura de las consultas en SQL

Una consulta en *SQL* consta básicamente de tres(3) cláusulas obligatorias **SELECT**, **FROM** y **WHERE** a las que se le pueden añadir opcionalmente otras tres(3) cláusulas **GROUP BY**, **HAVING** y **ORDER BY**.

La cláusula **SELECT** corresponde a la operación proyección del álgebra relacional. Se usa para listar los atributos deseados del resultado de una consulta.

La cláusula **FROM** corresponde a la operación producto cartesiano del álgebra relacional. Lista las relaciones que deben ser analizadas en la evaluación de la expresión.

La cláusula **WHERE** corresponde al predicado selección del álgebra relacional. Este engloba a los atributos de las relaciones que aparecen en la cláusula **FROM**. A continuación se muestra una consulta *SQL* mediante la cual se ejemplificará lo expuesto anteriormente.

```
SELECT número-préstamo
FROM préstamo
WHERE nombre-sucursal = 'Navecerrada';
```

En la consulta mostrada se determina el *número-préstamo* de la sucursal “*Navecerrada*”, esta información se obtiene desde la tabla *préstamo*, cada elemento que compone la consulta está ubicado como se explicó anteriormente.

De acuerdo con las características antes mencionadas, en la presente investigación se empleará el lenguaje de consulta *SQL*, mediante el cual se realizará la extracción de los elementos de interés para el modelo ontológico, los que permitirán la clasificación de la información.

1.1.5 Gestores de Bases de Datos

Un Sistema Gestor de Bases de Datos(SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente, protegiendo los datos a los que tienen acceso los usuarios, de las actualizaciones simultáneas por otros usuarios (15).

En esta sección se tratarán los aspectos fundamentales relacionados con los SGBD.

1.1.5.1 Características fundamentales de los SGBD

Los sistemas gestores de bases de datos presentan un grupo de características que le permiten realizar las tareas para las que son desarrollados estos sistemas. A continuación se muestran algunas de las fundamentales(15).

Mecanismos de Seguridad: estos sistemas garantizan la seguridad de la bases de datos limitando el acceso a la misma, del personal autorizado, generalmente estarán restringidos en cuanto al acceso a ciertos datos en particular y aquellos que puedan actualizar.

Integridad de los datos: esta se garantiza por medio de restricciones sobre los valores que pueden tomar los elementos de los datos y por las capacidades de recuperación y respaldo que ofrece el sistema.

Acceso concurrente a los datos por varios usuarios: proporciona mecanismos físicos que permiten a varios usuarios tener acceso de forma rápida y eficiente a diferentes datos relacionados. Se utiliza un mecanismo de bloqueo para proteger los datos que son actualizados por un usuarios, mientras al mismo tiempo garantiza el acceso concurrente a la base de datos y un tiempo de respuesta del sistema a otros usuarios aceptable.

Consultas e informes dirigidos al usuario: estos sistemas brindan herramientas de manipulación de datos dirigidos al usuario. Los lenguajes de consultas fáciles de usar permiten a los usuarios formular consultas y pedir informes únicos directamente a la bases de datos, mediante dichas herramientas.

Facilidades para el desarrollo de aplicaciones: garantizan una asistencia importante a los programadores de aplicaciones, mediante herramientas como generadores de pantallas, de menús y de reportes, los generadosres de aplicaciones, compiladores y las facilidades para la definición de los datos y las vistas de los datos, contribuyen a que el proceso de programación sea apreciablemente más eficiente.

Las siguientes secciones están dedicada al análisis de estos sistemas, haciendo énfasis en los que son soportados por las herramientas de acceso a la información de BD mediante ontologías, por resultar de interés en la presente investigación.

1.1.5.2 **MySQL**

*MySQL*¹, el sistema de gestión de bases de datos *SQL Open Source* más popular. Se desarrolló originalmente para tratar grandes bases de datos mucho más rápido que soluciones existentes y ha sido usado con éxito en entornos de producción de alto rendimiento durante varios años.

Otras características de este sistema son:

- Uso completo de *multi-threaded*, mediante *threads* del *kernel*. Pueden usarse fácilmente multiple *CPUs* si están disponibles
- Proporciona sistemas de almacenamiento transaccionales y no transaccionales
- Relativamente sencillo de añadir otro sistema de almacenamiento. Esto es útil si desea añadir una interfaz *SQL* para una base de datos propia
- Las funciones *SQL* están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible. Normalmente no hay reserva de memoria tras toda la inicialización para consultas
- El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor. También está disponible como biblioteca y puede ser incrustado (*linkado*) en aplicaciones autónomas. Dichas aplicaciones pueden usarse por sí mismas o en entornos donde no hay red disponible

Este SGBD es soportado por la gran mayoría de las herramientas de las herramientas que permiten el acceso de información de BD mediante ontologías.

1.1.5.3 **PostgreSQL**

El gestor de bases de datos *PostgreSQL*, es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales. *PostgreSQL* utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando(36).

¹<http://www.mysql.com/>

Como características de este gestor se señalan:

- Provee nativamente soporte para: números de precisión arbitraria, texto de largo ilimitado, Figuras geométricas (con una variedad de funciones asociadas)
- Los usuarios pueden crear sus propios tipos de datos. Algunos ejemplos son los tipos de datos del Sistema de Información Geográfica GIS creados por el proyecto *PostGIS*¹
- Soporta *triggers* y funciones
- Se pueden usar en una gran cantidad de lenguajes de programación como: *PL/PgSQL*, *C*, *C++*, *Java* *PL/Java web*, *PL/Perl*, *PIPHP*, *PL/Python*, *PL/Ruby*

El SGBD *PostgreSQL* es uno de los sistemas soportados por las herramientas que permiten el acceso de información de BD mediante ontologías.

1.1.5.4 ORACLE

El gestor de bases de datos *ORACLE*², uno de los SGBD más utilizados en el mundo de las grandes empresas, con licencia privativa. Se considera como uno de los sistemas de bases de datos más completos y es el proveedor mundial líder de software para administración de información. Consta de una estructura de memoria, llamada Área Global del Sistema (SGA), y de unos procesos *background* utilizados por el servidor *Oracle* para manejar una base de datos. Cada instancia puede abrir y utilizar sólo una base datos en cualquier punto y momento.

Oracle es un Sistema Gestor de Bases de Datos con características objeto-relacionales, que pertenece al modelo evolutivo de SGBD. Sus características principales son las siguientes:

- Entorno cliente/servidor
- Usuarios concurrentes
- Alto rendimiento en transacciones
- Sistemas de alta disponibilidad

¹Extensión que añade soporte de objetos geográficos a *PostgreSQL* y permite realizar análisis mediante consultas *SQL* espaciales o mediante conexión a aplicaciones GIS

²<http://www.oracle.com>

- Disponibilidad controlada de los datos de las aplicaciones
- Adaptación a estándares de la industria, como *SQL-92*.
- Autogestión de la integridad de los datos
- Replicación de entornos

Por otra parte la característica fundamental que resulta de interés a la presente investigación es que este SGBD es soportado por las herramientas que permiten el acceso de información de BD mediante ontologías.

En la presente investigación se trabajará sobre **PostgreSQL**, como caso particular de su objeto de estudio. Se escoge este SGBD por los beneficios que presenta encuaneto a: la extracción de datos de interés para el trabajo y la definición de información mediante los catálogos del esquema de la base de datos.

1.2 Ontologías

Una ontología ofrece conocimiento compartido y formalizado acerca de un dominio concreto(50). Esta se basan en teorías lógicas con una semántica bien definida (a diferencia de las descripciones de documentos *XML* o la estructura *HTML*), reutilizables entre aplicaciones. Que se encargan de proporcionar la columna vertebral de la *Web Semántica* ya que en ellas reside la semántia explícita que servirá de base a los procesos de integración, composición, búsqueda o inferencia que harán de la *Web*.

1.2.1 Definición de ontologías

La definición de ontología ha recibido múltiples definiciones a lo largo de la historia. Por lo que existen diferentes propuestas de diferentes áreas. El presente trabajo se centrará en las definiciones desde el punto de vista de la Inteligencia Artificial y desde el punto de vista de los Sistemas Informáticos.

Desde el punto de vista de la Inteligencia Artificial (IA)

Se encuentran varias definiciones como la que brinda Quine(42), la que apareció en Neches(43) y la que aporta Tom Gruber(16). Esta última es una de las definiciones más extendidas.

“Una ontología es una especificación explícita de una conceptualización (...) En sistemas de Inteligencia Artificial, lo que existe es lo que puede ser representado (...) Así, en el contexto de IA, la ontología de un programa es un conjunto de términos, las definiciones asocian nombres de entidades del universo del discurso con textos comprensibles (...) y axiomas formales que limitan la interpretación y buen uso de dichos términos. Formalmente, una ontología es una teoría lógica”.

Para comprender la definición anterior, es necesario entender **¿Qué es una conceptualización?**

- Una conceptualización es una interpretación estructurada de una parte del mundo que usan los seres humanos para pensar y comunicar en ella
- Para un informático, una conceptualización puede ser la clasificación de sistemas informáticos atendiendo a su naturaleza física en sistemas *hardware*, sistemas *software* y sistemas *firmware*
- Por explícita se dice que los conceptos y las restricciones se definen de forma evidente

Por su parte Nicola Guarino(18) hizo varias críticas tras examinar siete (7) posibles interpretaciones de ontologías, y propuso una alternativa a la definición brindada por Gruber.

“Un punto de inicio en este esfuerzo clarificador será el cuidadoso análisis de la interpretación dada por Gruber. El problema principal de dicha interpretación es que se basa en la noción de conceptualización. Una conceptualización es un conjunto de relaciones extensionales que describen un estado particular, mientras que la noción que se tiene en mente es intensional, esto es, algo como una rejilla conceptual al que se le impone varios estados posibles”.

En adición a lo anterior y según este autor, las ontologías que se usan en aplicaciones reales son realmente adaptaciones ingenieriles de ontologías, no ontologías propiamente dichas, por ejemplo, siguiendo esta perspectiva, “*el peso de una persona es una masa*” es una afirmación que podría aparecer en una ontología, mientras que “*el peso de una persona es un número*” puede aparecer en una adaptación de la ontología debido a limitaciones de tiempo y recursos. Sin embargo, esta idea estricta de ontología no es usada ni por los más puristas en IA.

2- Desde el punto de vista de los Sistemas Informáticos (SI)

Aparecen diferentes criterios de Nicola Guarino en el 2000(33). Además resulta interesante conocer la formulación que realiza el W3C¹.

“Una ontología define los términos a utilizar para describir y representar un área de conocimiento. Las ontologías son utilizadas por las personas, las bases de datos, y las aplicaciones que necesitan compartir un dominio de información (...) Las ontologías incluyen definiciones de conceptos básicos del dominio, y las relaciones entre ellos, que son útiles para los ordenadores. Codifican el conocimiento de un dominio y también el conocimiento que extienden los dominios. En este sentido, hacen el conocimiento reutilizable”.

A juicio del autor, en la presente investigación se tratarán las ontologías desde el punto de vista de los SI. Pues está orientada a la ilustración en las computadoras de vocabularios comunes que permiten entender, especificar, explotar, implementar y reutilizar la información que ha sido situada en los mismos.

1.2.2 Lenguajes para el desarrollo de ontologías

Una vez que los componentes de la ontología están definidos, puede ser representada en varios lenguajes. Los mismos surgieron a comienzos de 1990 y se basan principalmente en lógicas de primer orden, en marcos (*frames*) combinados por las lógicas descriptivas (*description logic*), por sus siglas en idioma inglés.

A continuación se listan algunos de ellos en orden cronológico:

1.2.2.1 RDF

*RDF (Resource Description Framework)*² como lo indican sus siglas en inglés *RDF*. Desarrollado con el objetivo de especificar contenido semántico, estandarizado, interoperable en base a lenguajes sobre redes semánticas; con el fin de describir recursos de la *Web*. La combinación

¹[http:// www.w3.org/](http://www.w3.org/)

²<http://www.w3.org/RDF/>

de *RDF* con otras herramientas como *RDF Schema* y *OWL* permite añadir significado a las páginas, y es una de las tecnologías esenciales de la *Web semántica*.

1.2.2.2 *RDF Schema*

Se considera una extensión semántica de *RDF* con primitivas basadas en marcos. La combinación de ambas suele denotarse como *RDF(S)*¹ Este lenguaje no es muy expresivo, sólo permite la representación de conceptos, taxonomías de conceptos y relaciones binarias. Y posee algunos motores de inferencia, que fueron creados específicamente para este lenguaje, en especial para verificación de restricciones.

1.2.2.3 *OWL*

OWL (Ontology Web Lenguaje) como lo indican sus siglas en inglés *OWL*: se basa en *XML* y *RDF*, es el estándar actual del *W3C*. Permite representar los elementos de lógica descriptiva, además tiene mayor capacidad expresiva. Es el más usado en *Internet*, estando sus elementos definidos con las fuentes de *RDF*.

Tiene tres (3) variantes o sub-lenguajes, según la complejidad que se necesite especificar, estas se muestran en orden ascendente de complejidad: (1) *OWL Lite*, (2) *OWL DL* y (3) *OWL Full*.

Debido a la generalidad de la utilización de *OWL* en unión con *RDF* y *RDF(S)*, se determinan estos lenguajes para ser utilizados en la investigación. A pesar de que existen varias ontologías disponibles en formato *DAML*, en la investigación se enfocará en los lenguajes antes mencionados.

1.2.2.4 *OWL 2*

Es un lenguaje de representación del conocimiento que permite desarrollar ontologías, clases, propiedades, valores de datos que son automatizados como documentos de la *Web semántica*. Las ontologías en *OWL 2*² tienen como objetivo, facilitar un modelo de marcado construido

¹[http:// www.gsi.dit.upm.es /](http://www.gsi.dit.upm.es/)

²<http://www.w3.org/TR/owl>

sobre *RDF* y codificado en *XML*. *OWL 2* se conoce también como *OWL 1.1*. Esta nueva versión añade potencia y velocidad, estandariza las características más demandadas por los usuarios de *OWL* y presenta perfiles tales como: *OWL2 EL*, *OWL2 QL* y *OWL2 RL*; los cuales permiten mejorar la escalabilidad en las aplicaciones típicas.

1.2.2.5 **SPARQL**

*SPARQL (Simple Protocol and RDF Query Language)*¹, por sus siglas en inglés. Es un lenguaje de consulta del ámbito de la *Web Semántica* del *W3C*. Define la sintaxis y la semántica necesaria para una expresión de consulta sobre un grafo *RDF* y las diferentes formas de resultados obtenidos. Es un lenguaje consultivo para *RDF*, paralelo al estándar *SQL* de bases de datos relacionales, que ayuda a obtener los recursos de diferentes fuentes identificables.

Su misión es devolver todas las triplas o componentes solicitados basándose en la comparación de una tripleta pasada como parámetro de la consulta (grafo básico) con todas las triplas que componen el grafo *RDF*(22).

Objetivos de las consultas SPARQL

1. Extraer información en forma de *URIs* y literales
2. Extraer sub-estructuras *RDF*
3. Construir nuevas estructuras *RDF* partiendo de resultados de consultas

De esta forma el *SPARQL* posibilita el acceso a un único conjunto de datos y mediante un mensaje de salida se observa los resultados obtenidos a partir de la consulta. A consideración del autor en la presente investigación se utilizará este lenguaje mediante el cual realizará la evaluación de la información resultante de la estrategia de razonamiento.

1.2.3 **Herramientas para el trabajo con las ontologías**

En los últimos años, el número de entornos de desarrollo para ontologías y de herramientas relacionadas han crecido considerablemente(14). Las herramientas aportan soporte para el

¹<http://xsparql.deri.org/>

proceso de desarrollo y uso de las ontologías. Estas herramientas se basan en **Editores**, **Razonadores** y **Bibliotecas** destinadas para el trabajo con las ontologías.

1.2.4 Editores de ontologías

Los editores surgen en entornos de desarrollo un tanto más ambiciosos, con el objetivo de integrar ontologías a sistemas de información existentes. A continuación se presentan las características de los editores más relevantes de ontologías.

1.2.4.1 KAON

KAON como lo indican sus siglas en inglés (*Karlsruhe Ontology*)¹ o *KAON 1* como también se conoce, es un entorno flexible y extensible. Es un proyecto escrito en *Java* de fuente libre con infraestructura para editar y mantener ontologías. También es el sucesor de *OntoEdit* y fue desarrollado en la universidad de *Karlsruhe*. Brinda una interfase de programación *Java* para acceder a ontologías e instancias, independientemente del formato de implementación de la ontología. Provee una simple herramienta para generar portales *Web*. Además de *KAON 1*, se desarrolló el proyecto *KAON 2*, sucesor del proyecto de *KAON 1*, por lo que también se conoce como una extensión del *KAON 1* el cual usó inicialmente *RDF(S)*. El *KAON 2*² se basa en *OWL-DL* y *F-Logic*.

1.2.4.2 Protégé

*Protégé*³ es un editor de ontologías y a la vez un *framework*, que permite trabajar con bases de conocimiento *Open Source*. Este editor permite en conjunto la modelación de ontologías, mediante el *Protégé Frames* y el *Protégé OWL*. El *Protégé Frames* posibilita modelar ontologías de acuerdo al *OKBC*, como lo indican sus siglas en inglés (*Open Knowledge Base Connectivity protocol*). En este modelo una ontología consiste, en una jerarquía de clases que representan los conceptos más importantes del dominio. Todo esto se realiza mediante una interfaz gráfica muy intuitiva(24).

¹<http://kaon1.semanticweb.org>

²<http://kaon2.semanticweb.org>

³<http://protege.stanford.edu/>

El *Protégé OWL* permite construir ontologías en base a *OWL*. Sigue la misma filosofía que *Protégé-Frames*, pues brinda flexibilidad para usar las funcionalidades de varios razonadores. El *Protégé* fue desarrollado por el *Stanford Medical Informatics*.

Para la realización de este trabajo se decidió utilizar *Protégé* porque es, una herramienta muy versátil y en la construcción de ontologías, es la que más usuarios tiene actualmente. Con esta herramienta se puede fácilmente crear clases y jerarquías, declarar propiedades para las clases, crear instancias e introducir valores; todo ello en un entorno de menús, botones, cuadros de diálogo y representaciones gráficas fáciles de usar. Se modela combinando *frames* y su propio lenguaje interno para definir ontologías *OWL*, pero permite también trabajar con *RDF* de modo transparente. Soporta directamente los tres principales paradigmas de ingeniería del conocimiento (mapas conceptuales, marcos y lógicas descriptivas).

1.2.4.3 *Topbraid*

*TopBraid*¹ es una herramienta creada por *Top-Cuadrant* que se basa en *plug-ins* de *Eclipse* para trabajar con modelos semánticos usando: *RDF*, *OWL*, *SWRL*.

***TopBraid* consta principalmente de las siguientes vistas**

- **Navegador:** despliega los archivos contenidos en el *workspace* de *Eclipse*
- **Clases, Propiedades y Asociaciones:** presentan los recursos y las jerarquías del modelo actual
- **Instancias:** muestra las instancias de la clase seleccionada en la vista de clases
- **Dominio:** presenta las propiedades que tienen en su dominio la clase seleccionada
- **Herencia:** despliega las superclases y restricciones de herencia de la clase actual
- **Editor de Recursos:** muestra la información del recurso actual incluyendo mecanismos de edición, código fuente y gráficas
- **SPARQL:** ofrece una interfaz para realizar consultas en el modelo
- **Reglas:** presenta las reglas definidas para el modelo en formato *SWRL*

¹<http://www.topbraidcomposer.com/>

A consideración del autor además del *Protégé OWL*, se empleará *TopBraid*, ya que el mismo ofrece una serie de vista que favorecen la investigación, principalmente las de *SPARQL*, donde se desarrollarán las consultas para la extracción de información del modelo resultante y las de edición de gráficos y diagramas que contribuyen a los procesos de desarrollo del modelo ontológico.

1.2.5 Razonadores

Inferir es deducir información adicional. El código que realiza esta tarea se le llama **razonador**. Para hacer inferencias se debe de crear un modelo inferido a partir de un razonador y después de eso, todas las consultas que se hagan a este modelo devolverán información deducida.

Los razonadores ¹, chequean la consistencia de la ontología, según el conjunto de reglas definidas en las diferentes clases. Re-clasifican los conceptos en la jerarquía de clases inferida y clasifican las instancias en la clase correspondiente(46). El proceso de razonamiento en la investigación que aquí se plantea, resulta de gran importancia ya que éste proceso permite evaluar la veracidad o no de algún hecho presente en el modelo que se propone.

Los razonadores se clasifican en dos (2) tipos.

1- Razonadores lógica descriptiva

1.2.5.1 *Pellet*

Pellet ², es un razonador open source para *OWL-DL* construido en *Java*. Originalmente se desarrolló en el *Laboratorio Mindswap de Maryland*. Este razonador soporta toda la expresividad de la lógica descriptiva detrás de *OWL-DL*, incluye el razonamiento con nominales (clases definidas por extensión). Se basa en los algoritmos *Tableau* desarrollados para lógicas expresivas potentes. Recientemente, *Pellet* ha sido extendido para soportar las nuevas características de la propuesta *OWL 1.1*.

¹<http://forge.morfeoproject.org/>

²<http://clarkparsia.com/>

1.2.5.2 *Racer*

*Racer*¹, es un razonador para la lógica descriptiva. Su nombre comercial es *RacerPro*. Al contrario que el resto de los razonadores de lógica descriptiva, *Racer* no tiene licencias libres que permitan su utilización.

1.2.5.3 *Hermit*

*Hermit*², es uno de los actuales proyectos de investigación del Laboratorio de Computación de la Universidad de *Oxford*. Aunque puede ser empleado con cualquier ontología, los investigadores toman como punto de partida los requerimientos de ontologías médicas a fin de construir un razonador potente.

El objetivo del proyecto *Hermit* es, desarrollar algoritmos de razonamiento escalables e implementar un prototipo que pueda eficientemente manejar ontologías enormes o complejas y volúmenes importantes de datos.

Hermit es un razonador para ontologías que emplean *OWL* y en su construcción se emplea cálculo *hypertableau* a fin de proveer razonamiento más eficiente. Utiliza *Direct Semantic* y supera las pruebas de conformidad de *OWL 2 DL*, *OWL 2 QL* y *OWL 2 RL*. Teniendo en cuenta estas características en la presente investigación se utilizará este razonador para evaluar la información mediante la estrategia de razonamiento que se propone.

2- Razonadores de programación lógica

A continuación se presenta uno de los razonadores más habituales para la programación lógica.

1.2.5.4 *TRIPLE*

Es un lenguaje y su correspondiente motor de inferencia para la *Web semántica*.

*TRIPLE*³ proporciona soporte para *RDF* y para un subconjunto de *OWL Lite*. Su sintaxis se

¹[http:// www.racer.com/](http://www.racer.com/)

²<http://materias.fi.uba.ar/>

³<http://forge.morfeoproject.org/>

basa en *F-Logic* y proporciona soporte para símbolos de función, el tratamiento de la igualdad o negación por defecto.

Una tripleta *RDF* (s, p, o) se representa utilizando expresiones de la forma $S[P - > O]$. El motor de inferencia de *TRIPLE* se basa en el sistema *Prolog XSB*.

1.2.6 Bibliotecas

Las Bibliotecas¹, son los sistemas de almacenamiento de ontologías, que permiten que las mismas estén en las bases de datos y se les añadan nueva información. El empleo de los razonadores, permite comprobar la estabilidad de las ontologías, en el desarrollo de aplicaciones basadas en *RDF*, *OWL* u otros lenguajes similares. En este aspecto las bibliotecas son necesarias a la hora de leer y procesar las ontologías definidas por estos lenguajes.

1.2.7 Framework para el trabajo con ontologías

A continuación se describen dos de los principales framework par el trabajo con ontologías. Estos son los más empleados actualmente, de estos se tratarán sus principales características.

1.2.7.1 Jena

*Jena*², es una biblioteca para el desarrollo de aplicaciones, en el lenguaje de programación *Java* para la *Web semántica*. En la actualidad existen (2) dos versiones.

Jena 1: principalmente soporte para *RDF*. Capacidades de razonamiento limitadas.

Jena 2: incluye además una *API* para el manejo de ontologías. Soporta el lenguaje *OWL*.

Soporte de inferencia del framework *Jena*

Jena proporciona a los desarrolladores un *API* para el tratamiento de los grafos *RDF*. Estos grafos son representados internamente como un modelo abstracto que se puede consultar mediante el lenguaje de consultas para *RDF*, *SPARQL*. Además *Jena* permite el tratamiento

¹[http:// eprints.ucm.es/](http://eprints.ucm.es/)

²[http://jena.sourceforge.net//](http://jena.sourceforge.net/)

del lenguaje *OWL*, al utilizar el modelo semántico de *RDF*, y permite que los razonadores de lógica descriptiva, como *Pellet* o *FaCT++* se puedan conectar de forma externa.

Razonadores internos del *framework Jena*

Independientemente de la capacidad de *Jena* para conectarse con motores de inferencia externos. *Jena* tiene su propio subsistema de inferencia que consiste en un motor híbrido con encadenamiento hacia-delante y hacia-atrás. Y proporciona varios razonadores.

- Razonador transitivo
- Razonador para *RDF(S)*
- Razonador para *OWL*

1.2.7.2 *OWL API*

*OWL API*¹ como lo indican sus siglas en inglés es la (*application programming interface for ontology web language*). En una interfase de programación en *Java* que permite la implementación de referencia para crear y manipular las ontologías en el lenguaje *OWL*. La última versión es, *API 3.x* desarrollada principalmente en la Universidad de *Manchester*, esta versión se enfoca hacia *OWL 2*.

***OWL API* incluye los siguientes componentes:**

Una *API* para *OWL 2* y una eficiente en memoria de implementación de referencia

RDF/XML parser y escritor

OWL/XML parser y escritor

OWL(Manchester) analizador de sintaxis funcional y escritor

OBO analizador de archivos planos en formato

Interfases para trabajar con razonadores, tales como, *Pellet*, *Hermit* y *Racer*

A juicio del autor se utilizará *OWL API* para desarrollar el proceso de creación del modelo ontológico de los datos obtenidos de la BD, ya que este *framework* permite la implementación

¹<http://techwiki.openstructs.org/>

de referencias para crear y manipular ontologías de una manera mejor definida que las forma propuesta en el *framawork Jena*.

1.2.8 Construcción y desarrollo de ontologías

En este contexto se aporta un punto de inicio el cual permite, conocer cómo se realiza el desarrollo de ontologías desde varios aspectos fundamentales.

1.2.8.1 Terminología básica

Las ontologías proporcionan un vocabulario común de un área, definen diferentes niveles de formalismo, el significado de los términos y relaciones entre ellos. El conocimiento en ontologías se formaliza principalmente usando los siguientes aspectos, desde el punto de vista tecnológico(16).

*- Componentes

Una ontología consta de un conjunto no vacío de conceptos identificados como entidades relevantes en el dominio a modelar, un conjunto de relaciones y atributos que describen los conceptos que pueden ser propios o heredados en una especialización.

a) Los conceptos: son las ideas básicas que se intentan formalizar. Los conceptos pueden ser clases, las cuáles a su vez pueden ser conjuntos de individuos que tienen características en común, estas clases pueden ser subclases de otras clases: todos los individuos de la subclase son también individuos de la superclase (Pero no viceversa)(3). En la figura 1.2 un conjunto de clases a representar en el *Protégé*.

b) Relaciones: representan la interacción y enlace entre los conceptos. Las relaciones básicas son sub-clase-de, parte-de, conectada-a. Otros ejemplos de relaciones ontológicas son las relaciones temporales (que implican precedencia en el tiempo) y las topológicas (que implican conexión espacial entre objetos)(5).

A continuación se muestra en la figura 1.3 un conjunto de relaciones.

En general, las relaciones pueden ser de los tipos:

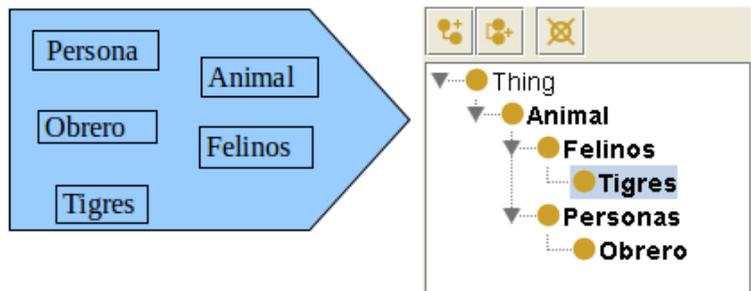


Figura 1.2: Conjuntos de clases a representar en Protégé. Fuente: elaboración propia.

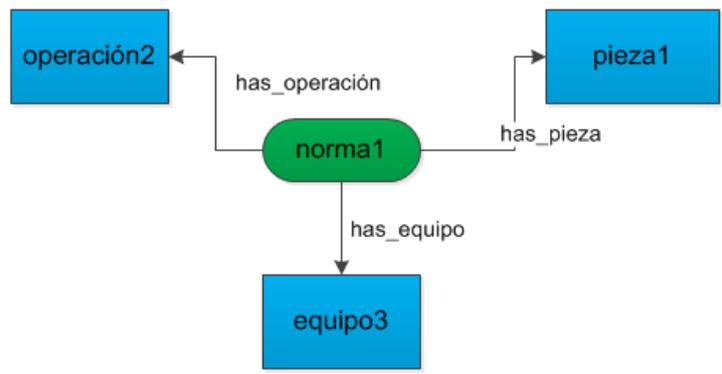


Figura 1.3: Conjunto de propiedades que enlazan clases (individuos) mediante relaciones. Fuente: elaboración propia

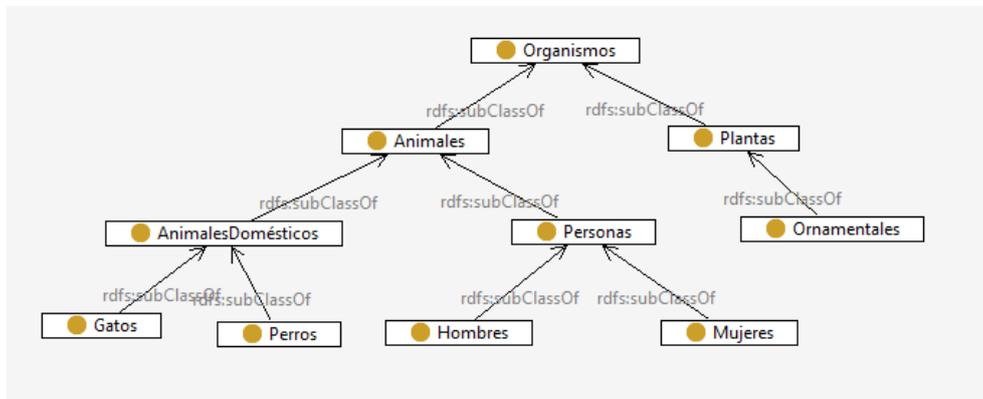


Figura 1.4: Taxonomía que muestra una jerarquía de clases. Fuente: elaboración propia

- **Taxonomía:** es una organización ontológica basada en una relación de orden parcial llamada *is-a*, a través de la cual se agrupan las entidades y son subsumidas por clases de más alto nivel(33).

Con las relaciones clase-subclase se obtiene una **taxonomía**, lo cual representa y clasifica una jerarquía de clases, donde una clase puede tener varias superclases.

A continuación se muestra en la siguiente figura 1.4 un ejemplo de una taxonomía.

A continuación se presentan las propiedades satisfechas por las relaciones taxonómicas(30).

- **Asimetría:** esta propiedad significa que la inclusión de una clase de individuos X, en una clase Y ,implica la no inclusión de Y en X. Formalmente, esta propiedad garantiza que:(X *is-a* Y) si y únicamente no (Y *is-a* X)
- **Transitividad:** sea X incluido en una clase Y, que a su vez está incluido en una clase Z, ambas inclusiones a través de relaciones *is-a*, entonces, existe una relación *is-a* entre las clases X y Z. Formalmente, (X *is-a* Y) y (Y *is-a* Z) \longrightarrow (X *is-a* Z)
- **Irreflexividad:** admitir la reflexividad en relaciones taxonómicas solo tiene sentido para modelar tautologías. Una tautología es la expresión de un mismo hecho de distintas maneras. La relación taxonómica se considera no reflexiva. Formalmente, no (X *is-a* X)

En la figura 1.5 se muestran algunas de éstas propiedades, a través del Protégé.

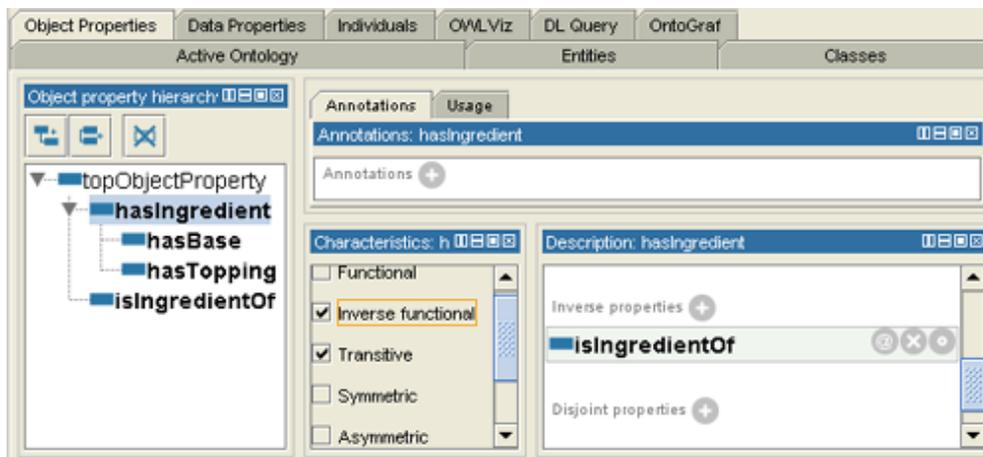


Figura 1.5: Propiedades de una clase, reflejadas en una ontología en Protégé. Fuente: elaboración propia.

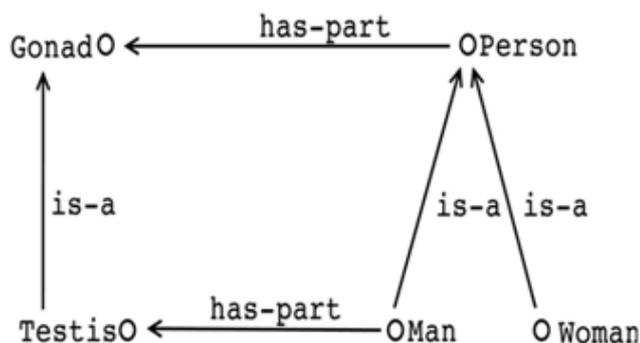


Figura 1.6: Axioma representado de una ontología. Fuente: elaboración propia.

c) Los axiomas: se usan para modelar verdades que se cumplen siempre en la realidad modelada(6). Los axiomas definidos en una ontología pueden ser **estructurales** o **no estructurales**.

Los axiomas **estructurales** establecen condiciones relacionadas a las jerarquías de la ontología, conceptos y atributos definidos; y los axiomas **no estructurales** establecen relaciones entre atributos de un concepto y son específicos de cada dominio. Los axiomas permiten la formalización del conocimiento en una ontología, como se muestra en la figura 1.6.

A juicio del autor, los componentes antes analizados son de vital importancia, ya que la información extraída de la base de datos desempeñará el papel de dichos componentes de la siguiente manera:

Conceptos → Tablas de la BD.
Relaciones → Llaves foráneas de la BD.
Axiomas → Declaración del axioma *hasKey* en el modelo resultante.

1.3 Comparación entre los modelos de BD y ontologías

En esta sección se analizan las diferencias entre los modelos de BD y ontologías. Esta comparación estará centrada los aspectos relacionados con el **nivel de representación** y la **semántica** ambos modelos.

1.3.1 Niveles de representación

En la bibliografía referente a Bases de Datos e Inteligencia Artificial se expone la diferencia en el nivel de representación entre los modelos conceptuales y los modelos lógicos. En cuanto a las BD se establecen diferencias entre los modelos de **alto nivel** o **conceptuales** y los modelos de **bajo nivel** o **físicos**. Los primeros manejan conceptos próximos al modo en que los humanos perciben la realidad y los segundos manejan conceptos próximos al modo en que las máquinas almacenan los datos. Entre estos niveles se ubica una clase intermedia: los de **representación** o **implementación** cuyos conceptos son comprensibles fácilmente por humanos pero que no se alejan demasiado del modo en que dichos datos están almacenados en las máquinas(13).

En adición a lo anterior los autores proponen dentro de la clasificación de modelo conceptual al modelo E-R y como modelo físico al modelo relacional(13)(45).

En la Inteligencia Artificial también se han establecido diferencias dentro de las cual se establece, los modelos **conceptuales** o **semánticos** y modelos **lógicos**(32). Los primeros basados en nociones más próximas a estructuras cognitivas humanas tales como conceptos o relaciones entre ellos, y los segundos basados en formalismos matemático-lógicos, que implementan de forma operativa los anteriores. En consideración con lo anterior las ontologías están incluidas dentro los modelos conceptuales o semánticos.

A continuación se muestra un ejemplo que permite representar el modo en que los humanos perciben la realidad y su representación a través de un modelo conceptual:

La afirmación del lenguaje natural: “*Todo vehículo pertenece a su propietario*”, tendría su equivalente en un lenguaje conceptual similar a: “*Todo individuo del concepto vehículo estará relacionado mediante la relación perteneceA con otro individuo del concepto persona*”.

De lo anteriormente expuesto se concluye que la primera y principal diferencia entre los modelos de BD y ontologías radica en el nivel de representación en el cual se encuentra uno y otro. Se puede afirmar que el modelo relacional trata de organizar la estructura e integridad de los datos, mientras que las ontologías se ocupan de especificar el significado de la conceptualización subyacente a dichos datos(45).

Por otra parte los componentes del modelo relacional son estructuras lógicas que se refieren directamente a los datos (relaciones, tuplas), mientras que los componentes utilizados por las ontologías tratan de capturar entidades e interrelaciones entre ellas y para ello manejan estructuras conceptuales independientes del formalismo lógico en que se especifiquen, tales como conceptos, interrelaciones o atributos(45).

En adición a la comparación, desde el punto de vista de la perspectiva o el alcance, puede afirmarse de manera general que un modelo de datos como el relacional está orientado a una aplicación específica a la que dará soporte. En la mayor parte de los casos sin ninguna pretensión de generalidad ni de reutilización *a priori* se trata de un modelo **local**(26). No así las ontologías que tratan de ser conceptualizaciones consensuadas y en mayor o menor medida independientes del uso que se haga de ellas, que facilitará al menos en teoría su reutilización(35)(39).

1.3.2 La semántica en ontologías y en el modelo relacional

Tanto un modelo relacional como una ontología o un modelo ER (en tanto que modelos de nivel conceptual) abstraen entidades de un dominio de discurso concreto¹. Por lo que existiría una correspondencia entre sus elementos (y extensiones instancias en una ontología y tuplas en el modelo relacional) y las entidades del dominio de discurso que modelizan(45).

Por otra parte, si el modelo está descrito mediante algún lenguaje formal y la correspondencia entre dicho lenguaje y una conceptualización sobre el dominio de discurso es explícita y formalizable, se dirá que el lenguaje tiene una **semántica formal**. La anterior idea permite definir la noción de la semántica de un modelo como la descripción precisa de la relación existente

¹ con diferentes finalidades, explotación mediante consultas en el caso del modelo relacional, conceptualización en el caso de la ontología y el modelo Entidad-Relación

entre una representación particular (un modelo) y los conceptos del mundo real que dicha representación trata de modelizar(45).

De acuerdo la definición anterior, existe otra acepción del término semántica algo menos rigurosa pero por otra parte muy frecuentemente utilizada en la literatura sobre Bases de Datos y Sistemas de Información para referirse a la capacidad expresiva de un lenguaje de representación. No resulta extraño encontrar en la literatura afirmaciones como “*RM/T¹ o el modelo E-R tienen más semántica que el modelo relacional*” para dar a entender que dispone de un conjunto de elementos (primitivas) de representación más extenso o que al menos permite capturar más conocimiento(31)

Según lo planteado con anterioridad, se establece que para las ontología como para el modelo relacional puede definirse una **semántica formal**. En el caso del modelo relacional propone una semántica formal describiendo una base de datos como un modelo lógico de su esquema relacional, es decir, sus relaciones, atributos y restricciones de integridad(44).

De la misma forma, cualquier lenguaje de implementación de ontología se acompaña de una teoría semántica que especifica formalmente su significado, por ejemplo la teoría semántica propuesta por Hayes para *RDFS*(20) y la de Patel-Schneider y otros para *OWL*(38).

Por otra parte, desde la perspectiva de expresividad, las primitivas y restricciones, las de las ontología son abstractas como por ejemplo concepto, interrelación, identidad, rigidéz, obligatoriedad, entre otras. Características de un nivel superior e independiente del tipo de implementación, mientras que el el modelo relacional están orientadas a la implementación, como las claves primarias, los tipos de datos o los valores nulos(45).

1.4 Similitudes de las BD y las ontologías

En anteriores secciones se abordaron las principales diferencias entre los modelos de BD y las ontologías, sin embargo a pesar de las mismas estos modelos presentan similitudes llegando incluso a decirse que: el modelo E-R es también un modelo conceptual como se ha visto en cierto sentido y bajo ciertas restricciones puede ser considerado también una ontología tal y como describen Gómez-Pérez y colegas en(34).

A continuación se muestran las principales similitudes entre estos modelos(40):

¹Extensión del modelo relacional

- Los modelos ontológicos son capaces de definiciones nativas de jerarquía así como una base de datos jerárquicas lo que significa que ambas estructuras de datos de tipo de árbol puede estar eficazmente organizada
- Las ontología son una red semántica algo así como la base de datos, por tanto en ambos modelos se puede modelar cualquier cosa del mundo real
- Varios modelos de ontologías están basados lógica marcos o *frame* algo así como las base de datos orientadas a objetos, lo que significa que la orientación de objeto puede ser conservada independientemente del modelo de dato

En adición a lo anteriormente expuesto se añade que las especificaciones claves de la *Web Semántica* fueron esfuerzos conscientes para combinar los mejores atributos de la base de datos relacional: la consistencia, el modelado flexible, natural fácil para usar con datos no estructurados y las relaciones lógicas.

1.5 Generación de modelos ontológicos a partir de bases de datos

Como se mencionó anteriormente el propósito de la *Web Semántica*, es la exposición en la *Web* del gran número de bases de datos relacionales existentes. La presente sección está dirigida a una de las soluciones del propósito de la *Web Semántica* la **generación de modelos ontológicos a partir de bases de datos**. Para desarrollar este análisis se abordarán las siguientes cuestiones: **a) Deficiencias de los modelos de BD. b) Principales beneficios del empleo de las ontologías.**

Antes de referirnos a las cuestiones antes mencionados se hace necesario analizar tres conceptos fundamentales **datos, información y conocimiento**, ya que los mismos permiten determinar los fronteras de los modelos analizados en la presente investigación.

De acuerdo con la bibliografía consultada, los **datos** son, hechos instancias o valores específicos que describen sucesos y entidades, entre los tipos de datos que existen se puede encontrar: atómicos, únicos, ocurrencias singulares, por ejemplo un número, una letra o cualquier símbolo que represente cantidad, medida, palabra o descripción. Los datos de por si, no contienen información, su importancia radica en asociarse a un contexto donde se convierten en información(12).

Por otra parte un concepto relacionado estrechamente con el término datos es el de **metadatos**, estos son datos que describen otros datos o información acerca de información, estos son una manera de enriquecer los datos a tal punto que las aplicaciones puedan interactuar con la información. Según Bernes Lee los metadatos son, “información entendible para el ordenador sobre recursos *Web* u otras cosas”. En el contexto de la *Web* son datos que se pueden guardar, intercambiar y procesar por medio del ordenado de tal forma que permiten ayudar a la identificación, descripción, clasificación y localización del contenido de documentos o recursos(40).

De acuerdo a lo planteado anteriormente antes de referirnos al término información resulta importante definir contexto, ya que es un conjunto de relaciones alrededor de un elemento que ayuda a determinar su significado, por ejemplo el banco puede ser una institución financiera o el un lugar donde sentarse. Teniendo en cuenta lo antes planteado, **información** se refiere a datos que se encuentran en el contexto de otros datos. Según Pollock en (40), el concepto de información esta relacionado con el significado que brindan los datos.

Muy relacionado con la información se encuentra el concepto **conocimiento**, ya que la información conduce a obtener un conocimiento. El conocimiento es información en el contexto de otra información, por ejemplo en el contexto de las ontologías, el contenido del conocimiento esta expresado por los conceptos referentes al dominio que la misma modela(49). Por otra parte Pollock define conocimiento como una red de conceptos que provee un contexto para los datos y la información(40).

De acuerdo con lo anterior explicado se concluye:

Con relación a los términos mencionados anteriormente los modelos de bases de datos están destinados para el tabajo a nivel de **información**, ya que los diversos sistemas de bases de datos gestionan la información en tablas y en dichas tablas representan ningún tipo de conocimiento, solo un conjunto de datos resultantes del proceso de diseño de la BD. Por ejemplo:

En la figura 1.7 se presenta la tabla *cliente*, esta tabla puede conformar una BD cualquiera.

Los campos de la tabla *cliente* contienen datos refentes a dicha entidad, suficientes para la aplicación la cual se diseñó la BD, pero insuficientes para representar algún tipo de conocimiento sobre esta entidad. Pues los atributos que conforman la tabla varían en dependencia del contexto de dicha base de datos, por ejemplo la tabla *cliente* en un BD (x) puede contener más atributos que una entidad *cliente* en una BD y, debido a que, aunque se trabaje sobre el mismo concepto, este se modela en contexto diferentes.

<i>nombre-cliente</i>	<i>calle-cliente</i>	<i>ciudad-cliente</i>
Santos	Mayor	Peguerinos
Gómez	Carretas	Cerceda
López	Mayor	Peguerinos
Pérez	Carretas	Cerceda
Rupérez	Ramblas	León
Abril	Preciados	Valsaín
Valdivieso	Goya	Vigo
Fernández	Jazmín	León
González	Arenal	La Granja
Rodríguez	Yeserías	Cádiz
Amo	Embajadores	Arganzuela
Badorrey	Delicias	Valsaín

Figura 1.7: Tabla *cliente*. Fuente: elaboración propia.

Mientras que las ontologías son desarrolladas para representar un determinado **conocimiento** de un dominio, incluyen definiciones de conceptos básicos del dominio, y las relaciones entre ellos, codificando el conocimiento de un dominio y también el conocimiento que extienden los dominios. En este sentido, hacen el conocimiento reutilizable.

1.5.1 Deficiencias de los modelos de BD

Según la bibliografía referente a BD la principal deficiencia de estos modelos son la semántica, lo cual se produce por el hecho de que la mayoría de los sistemas de bases de datos se basan en el modelo relacional que consta de una menor semántica, dado su simplicidad conceptual al representar los datos a través de tablas(15).

En relación a lo anterior, el lenguaje de representación para estos modelos de datos *SQL*, en su definición carece de expresividad, dado que este solo está enfocado en el planteamiento de preguntas en bases de datos relacionales y permitir la definición de esquemas relacionales. Mientras los lenguajes utilizados para el desarrollo de ontologías, tienen mayor capacidad expresiva , dotando a estos modelos de mayor semántica que los modelos de BD(45).

Otras de las deficiencias es que las bases de datos se diseñan para una aplicación específica a la que dará soporte. En la mayor parte de los casos sin ninguna pretensión de generalidad

ni de reutilización *a priori* se trata de **modelos locales**(26)

Por otra parte, otra de las deficiencias de estos modelos es que funcionan con el Supuesto del Mundo Cerrado (*CWA*, por sus siglas en inglés), en los cuales la información que se omite es falsa. No así los modelos que trabajan con el Supuesto del Mundo Abierto (*OWA*, por sus siglas en inglés) como las ontologías en las cuales la información que se omite es desconocida (podría ser verdadera o falsa) y en los que podemos añadir nuevo conocimiento “fácilmente”. No así en las BD, en las que habría que desarrollar nuevamente el proceso de diseño(25).

1.5.2 Principales beneficios del empleo de las ontologías

Entre los beneficios de las ontologías se destacan(25):

- Proveen una comprensión /definición común y compartida acerca de ciertos conceptos claves en el dominio.
- Proveen una forma de reutilizar conocimiento de dominio.
- Conjuntamente con los lenguajes de descripción de ontología proveen una forma de codificar conocimiento y semántica algo semejante que las máquinas pueden comprender.

Por otro lado, hoy día la principal ventaja que aportan las ontologías, desde el punto de vista de las bases de datos y de la recuperación de información, tiene que ver con el empleo simultáneo de bases de datos de muy distinta naturaleza, características y formato, las que son consultadas simultáneamente para recuperar la información buscada(11).

El contenido de dichas bases de datos puede ser de contenido tan dispar como la información textual poco estructurada(la clásica que forma parte de los buscadores de *Internet*), documentación fotográfica, documentación geográfica, museográfica, urbanística, espacios naturales, etc.

Otra de las grandes ventajas de las ontologías es la posibilidad de utilizar la información pre-existente en dichas bases de datos a través de la propia ontología, sin tener que renunciar a la base de datos original de partida. De manera que pueden convivir y seguir empleándose simultáneamente bases de datos iniciales y ontología(11).

Tan solo debemos tener presente la necesidad de actualizar la ontología con los nuevos datos que vayan añadiéndose a las bases de datos. Además tampoco es necesario introducir de

nuevo manualmente -aunque podría hacerse- las bases de datos en la nueva estructura que hemos desarrollado con la ontología.

1.6 Sistemas de acceso a BD mediante ontologías

Los sistemas de acceso a bases de datos a través de ontologías(ODBA, por sus siglas en inglés), tienen como propósito de utilizar una ontología, o sea una conceptualización formal del dominio, para mediar el acceso a una fuente de datos(29). En estos sistemas el usuario especifica una base de datos relacional como una fuente de datos y una ontología que describe el dominio de conocimiento, mediante la cual se manipularán los datos de la BD, a través de un mapeo semántico definido previamente.

Estos sistemas en su arquitectura están compuestos por:

1. Un razonador para responder las consultas de los usuarios, teniendo en cuenta la ontología, los mapeos semánticos con las fuente de los datos
2. Una ontología que describe el dominio específico de la información tratada
3. Un mapeo semántico entre la ontología y la fuente de los datos
4. Una base de datos relacional como fuente de los datos

La interacción de estos componentes se muestran en la figura 1.8, donde se describe la arquitectura de estos sistemas.

1.6.1 Descripción del *plugin* DIG-MASTRO para Protégé

El *plugin* DIG-MASTRO para Protégé descrito en (2), es un sistema mediante el cual se logra acceder datos almacenados en una base de datos relacional a través de una ontología del dominio, además permite:

- Describir mapeos semánticos conectando la fuente de los datos con las entidades de la ontología

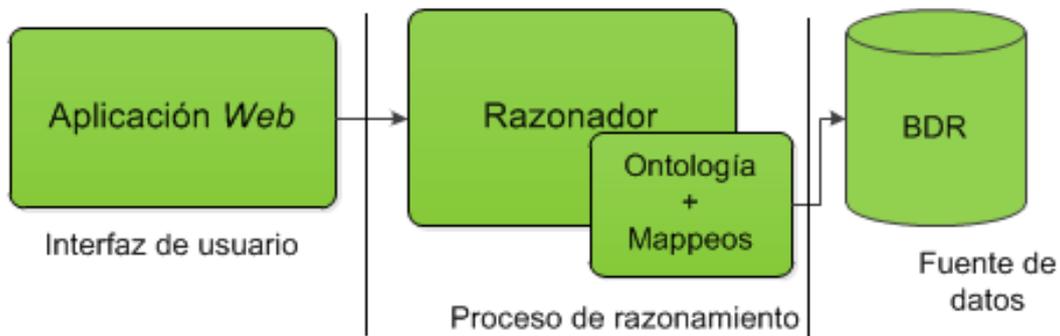


Figura 1.8: Arquitectura de los sistemas ODBA. Fuente: Tomada de (29).

- Enviar las descripciones de los componentes de la fuente de los datos al razonador
- Publicar las búsquedas realizadas por el razonador como la vista de los resultados

El principal inconveniente de estos sistemas, es la imposibilidad de crear una ontología de la fuente de los datos, requiriendo de una nueva ontología por cada fuente de datos que presente información diferente del dominio de la ontología. Como caso particular en el *plugin* descrito utilizan la ontología LUBM, esta describe un proceso de *benchmark* académico.

1.6.2 Descripción del algoritmo *SemanticMapper*

El algoritmo *SemanticMapper*, descrito en (45), es un procedimiento para el establecimiento de correspondencias semánticas, dados una ontología **O** y un modelo relacional **M**. Este algoritmo permite la generación automática de un recubrimiento conceptual del modelo relacional **M** basado en la ontología **O** y expresarlo utilizando el lenguaje R2O. Posteriormente permitiendo la recuperación de información a través del procesador *ODEMapster*. La figura 1.9 ilustra el proceso de funcionamiento del algoritmo.

El lenguaje **R2O** no es un lenguaje de representación de conocimiento, este hace referencia a dos modelo preexistentes, un modelo relacional y una ontología cada uno descrito en su correspondientes lenguajes (*SQL* y *OWL*) respectivamente. Otras de las características de este lenguaje es su adecuación para escenarios en los que se lleva a cabo procesamiento de consultas (*query processing*) indicando al sistema procesador como utilizar las fuentes para recuperar la información que contienen. R2O define un conjunto extensible de primitivas de

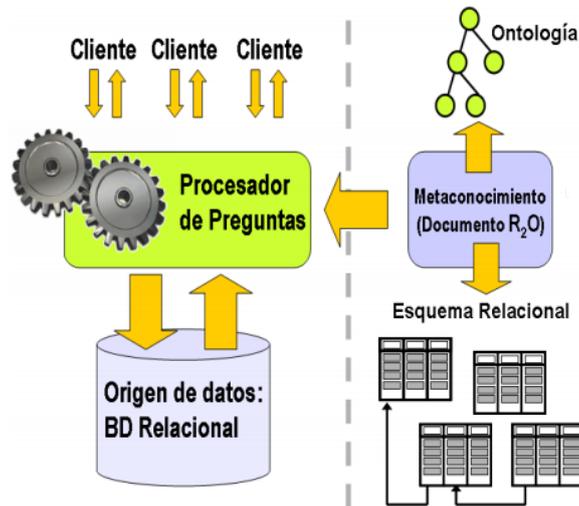


Figura 1.9: Proceso del algoritmo *SemanticMapper*. Fuente: tomado de (45)

condición y transformación para conseguir la independencia de SGBD y permitir la extensión de la capacidad expresiva del lenguaje de consultas de un SGBD particular. Además Los elementos de una ontología que tiene en cuenta R2O son los son los conceptos, atributos e interrelaciones. En lo que respecta al modelo relacional se reduce a relaciones y atributos.

Por otra parte, el procesador **ODEMapster** permite la recuperación de información, que se realiza a través de preguntas al modelo definido en R2O, esto se realiza de dos variantes:

1. **Recuperación de información dirigida por las consultas:** información semántica sólo se genera bajo demanda para responder a una pregunta concreta, realizadas sobre el documento en R2O.
2. **Recuperación de información masiva** o de tipo de repositorio (*Warehouse*): La información semántica se genera mediante un proceso de migración masivo que crea un repositorio persistente que contendrá un volcado completo de la fuente de datos. No define (no declara, no crea) conceptos, atributos e interrelaciones que no existiesen ya, sino que se limita a poner en relación los elementos definidos en un modelo relacional con los definidos en una ontología (todos ellos preexistentes).

Este algoritmo se desarrolló sobre la base de los sistemas ODBA: **la ontología y el modelo relacional entre los que se establece el alineamiento son completamente indepen-**

dientes, solo se tiene que cumplir que: Deberá existir una intersección no vacía entre los dominios de discurso de ambos modelos (el relacional y la ontología), o sea:

Cualquier concepto **O1** modelado en la ontología **O**, deberá coincidir con algún elemento **M1** en un modelo relacional **M**.

A *grosso modo*, este algoritmo no es un modelo de representación del conocimiento sino de descripción de correspondencias entre modelos ya descritos en sus correspondientes formalismos (relacional y ontologías) y lenguajes de representación (*OWL* en el caso de las ontologías y *SQL-DDL* en el caso del modelo relacional).

1.7 Determinación de eficacia

Para la evaluación de la **eficacia** entendida como, la calidad de un sistema en función de la cantidad de recursos que requiere, se analizan los indicadores precisión(**precision**) y exhaustividad (**recall**) descritos en (45) y las combinaciones de las anteriores conocidas como **Media Armónica o Medida E**, estos indicadores son analizados a continuación.

- Precisión(P): Indicador que permite poner de relieve la proporción de correspondencias correctas(CCD) del total de las descubiertas(CD), la expresión matemática de la precisión entre modelos es la siguiente:

$$P = \frac{nCCD}{nCD}$$

- La exhaustividad o índice de recuperación(R): Indicador que captura la proporción de correspondencias correctas descubiertas(CCD) con respecto al total de existentes(CC). Se trata de una medida inversamente proporcional a la precisión. La expresión matemática de la exhaustividad entre modelos es la siguiente:

$$R = \frac{nCCD}{nCC}$$

- Medida E (E): Como se observa, la precisión y la exhaustividad están compensadas, por lo que un sistema con una exhaustividad muy alta pero con baja precisión y viceversa no será adecuado. La medida E es una función que combina las dos anteriores y toma valores en el intervalo [0, 1]. Permite además ponderar la prioridad de la precisión sobre

la exhaustividad o viceversa. La importancia relativa de una con respecto a otra se indica mediante un parámetro constante **b**. Su expresión matemática es la siguiente:

$$F = \frac{1 + b^2}{\frac{b^2}{R} + \frac{1}{P}}$$

Donde valores de $b > 1$ dan prioridad a P sobre R y valores de $b < 1$ dan prioridad a R sobre P . En el caso en que $b = 1$, ambas métricas se ponderan por igual y la función toma el nombre de **medida F** o media armónica.

A consideración del autor, en el presente trabajo se utilizarán los indicadores analizados para determinar la eficacia del procedimiento que se propone como resultado de la presente investigación.

1.8 Conclusiones parciales

Del lo analizado a lo largo de este capítulo se concluye:

- El modelo relacional, en el cual se basan los modelos de bases de datos actuales, carece de semántica debido a su simplicidad conceptual al representar los datos a través de relaciones, las que se manifiestan en tablas.
- El lenguaje *SQL*, en su definición, no presenta alguna estructura para la descripción de los datos, pues este solo se destina al planteamiento de preguntas a la BD y la representación de esquemas relacionales.
- Las ontologías permiten resultados rápidos y comprensibles, gracias a una información mejor definida y detallada de lo que suele almacenarse en BD, ya que presentan una capacidad expresiva considerablemente mayor.
- Mediante el empleo de las ontologías se puede, utilizar la información preexistente en dichas bases de datos a través de la propia ontología sin tener que renunciar a la base de datos original de partida, de manera que pueden convivir y seguir empleándose simultáneamente bases de datos iniciales y ontología.
- Los sistemas OBDA son incapaces de crear una ontología de la fuente de datos, requiriendo una nueva ontología para cada vez que se trabaje sobre un dominio de conocimiento diferente. Además estos sistemas no presentan mecanismo alguno para la clasificación de la información contenida en las BD.
- Actualmente no existe procedimiento alguno capaz de evaluar la información almacenada en bases de datos.

Capítulo 2

Procedimiento para la construcción del modelo ontológico

En este capítulo se describe la solución propuesta al problema científico de la presente investigación. Esta se divide en dos métodos, el **algoritmo de migración**, este realiza la migración de una base de datos a una ontología, de acuerdo con las correspondencias existentes entre los elementos de ambos modelos. El otro método es una estrategia para la **clasificación de la información**, compuestas por un grupo de consultas *SPARQL* que clasifican como conceptos, relaciones, roles y atributos la información de la base datos.

2.1 Alcance del modelo

En este punto se tratan una serie de cuestiones para delimitar el alcance del modelo propuesto en la presente investigación.

1. En primer lugar, en el presente trabajo se propone un procedimiento para la migración de la información de la base de datos a una ontología. Este procedimiento cumple con el requerimiento fundamental para los sistemas de migración: **la ontología que es construida a partir de una BD debe tener el más alto grado de similitud.**
2. En segundo lugar, una estrategia de clasificación de la información almacenada en la base de datos. Este consiste en una serie de consultas *SPARQL* aplicadas a la ontología resultante del algoritmo de migración. Dichas consultas clasifican la información en **conceptos, relaciones, roles y atributos.**

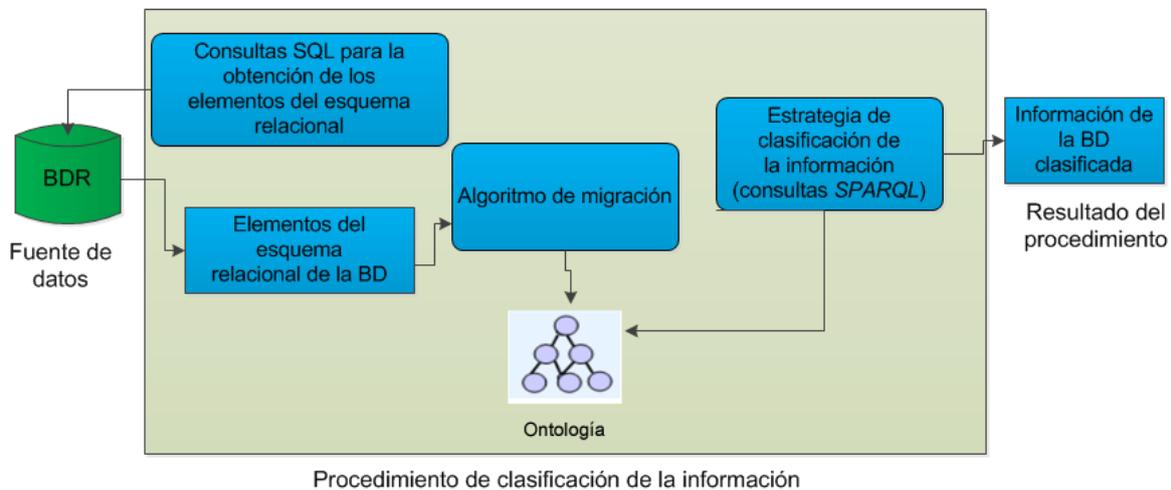


Figura 2.1: Arquitectura del modelo ontológico propuesto en la presente investigación. Fuente: elaboración propia.

Para realizar los dos procedimientos en los que se divide el modelo propuesto en la presente investigación, se diseñó la arquitectura mostrada en la figura 2.1.

La arquitectura mostrada en la figura 2.1, consta de tres (3) partes: la **Fuente de datos**, una base de datos relacional, el **Procedimiento para la clasificación de la información**, compuesto por dos (2) métodos, el primero el **algoritmo de migración**. Este genera una ontología del esquema relacional de la base de datos, una vez obtenidos un conjunto de elementos del esquema relacional de la BD a través de una serie de consultas *SQL*.

Una vez construida la ontología se aplica el segundo método, la **estrategia para la clasificación de la información**, compuesto por una serie de consultas *SPARQL*. A través del cual se obtiene como **Resultado del proceso**, la información clasificada en **conceptos, relaciones, roles y atributos**.

2.2 Estrategia de solución al problema planteado

De acuerdo con lo planteado en el capítulo 1, se diseñó un procedimiento mediante el cual se obtiene un **modelo ontológico** de un esquema relacional de bases de datos. Este procedimiento constituye la solución al problema científico a solucionar en la presente investigación.

En el Anexo 1 se muestra el diagrama de secuencia el cual ilustra como se realiza el procedimiento propuesto. A continuación se describen las actividades del procedimiento.

2.2.1 Análisis del esquema de las Bases de Datos

De acuerdo con lo analizado en la subsección 1.1.2.2, el procedimiento resultante en esta investigación se centra en la obtención de una serie de elementos que conforman el esquema relacional de la BD para la creación de un ontología. Se utilizó el esquema relacional ya que este describe la estructura de una Base de datos, definiendo sus tablas, sus campos en cada tabla y las relaciones entre cada campo y cada tabla. Para la descripción del procedimiento se utiliza el esquema mostrado en la figura 2.2.

La base de datos **Control del proceso 1** mostrada en la figura 2.2, se diseñó para almacenar la información concerniente al proceso de producción de piezas en una industria mecánica. La misma se compone de siete (7) entidades: **pieza, equipo, operación, obrero, auxiliar, directo y norma**, las cuales almacenan toda la información del proceso. Cada uno de estas entidades está compuesta por sus llaves, sus atributos y las relaciones entre cada una de estas entidades. Esta BD permite a los usuarios tener una visión abstracta de la información del proceso para el cual se diseñó dicha BD.

Dentro de los elementos contenidos en el esquema, en este proceso se emplean: **tablas** que conforman la base de datos, los **atributos llaves**¹ de estas, los **atributos** de cada tabla, los **tipos de datos** de cada atributo, las restricciones de cada tabla² y las **tuplas** de cada tabla. Mediante la relación establecida entre estos elementos se conforma el modelo ontológico resultante del presente proceso.

¹Se denomina atributo llave al atributo escogido como llave de la entidad en la BD.

²En este caso las restricciones se refieren a las llaves primaria y las llaves foráneas

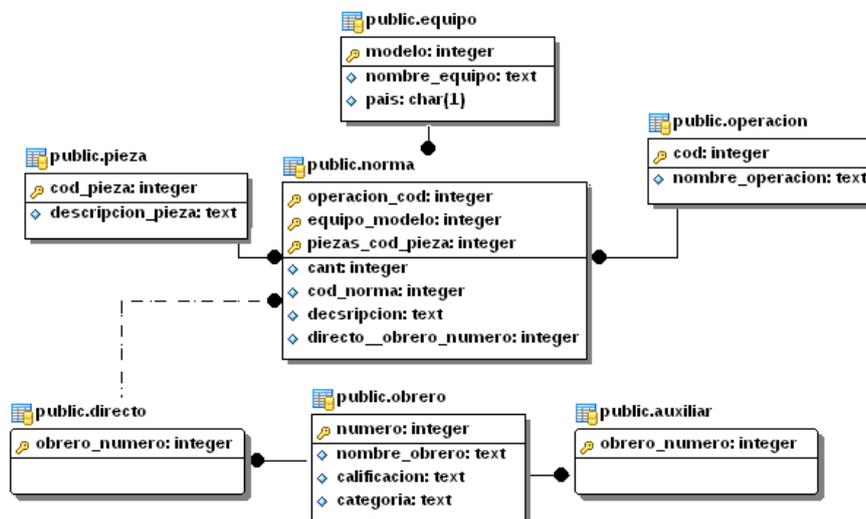


Figura 2.2: Esquema de la base de datos Control del proceso 1 utilizada para ejemplificar el proceso. Fuente: elaboración propia.

2.2.1.1 Extracción de los elementos del esquema de la BD

La obtención de los elementos del esquema se realiza mediante una serie de consultas a las tablas *constraint-column-usage*, *information-schema.columns* y *pg-catalog.pg-constraint*, las que almacenan información correspondiente al esquema relacional de la BD. Estas tablas están contenidas en los catálogos del gestor *PostgreSQL*, escogido como caso particular en la presente investigación debido a las beneficios de este analizados en la sección 1.1.5.3.

En la primera tabla se almacena información correspondiente a las tablas de la BD sus atributos llaves y las restricciones de cada atributo llave. A través del resultado de dicha consulta se generó la jerarquía de clases y el axioma *haskey* en la ontología, los que conforman el modelo de la siguiente manera:

Tablas del esquema de la BD → Clases de la ontología.

Llaves de las tablas → Axioma *haskey*.

La consulta diseñada para capturar los elementos del esquema de la BD de la tabla *constraint-column-usage* se muestra a continuación:

```

SELECT
constraint-column-usage.table-name,
constraint-column-usage.column-name,
constraint-column-usage.constraint-name
FROM
information-schema.constraint-column-usage;

```

En la consulta que se muestra en el cuadro anterior, el campo **table-name** se obtienen todas las tablas del esquema de la base de datos. En el campo **column-name**, se capturan los atributos llaves de cada tabla y en el campo **constraint-name** se determinan el tipo de restricción de cada tabla, ya sea llave primaria o llave foránea. Esta consulta permite relacionar a cada tabla del esquema con su respectivo identificador y el tipo de restricción presente. Dicha relación facilita la generación de la jerarquía de clases del modelo y la declaración del axioma *hasKey*.

Por otra parte para generar las propiedades de datos, así como el dominio y el rango de las mismas, se consultó la segunda tabla. En esta se obtiene información del esquema relacionada con las tablas de la BD, sus atributos y los tipos de datos de cada atributo. Con el resultado de esta consulta se logró establecer las propiedades de datos en el modelo ontológico de la siguiente forma:

Atributos de las tablas → Propiedades de datos.

Tablas a las que pertenecen cada atributo → Dominio de las propiedades de datos.

Tipo de dato de cada atributo → Rango de las propiedades de datos.

La consulta mediante la cual se obtienen los elementos del esquema de la BD de la tabla **information-schema.columns** se diseñó de la siguiente forma:

```

SELECT
columns.table-name,
columns.column-name,
columns.data-type
FROM
information-schema.columns
WHERE columns.table-schema = 'public';

```

En el campo **table-name** de la consulta mostrada en el cuadro anterior, se obtienen todas las tablas que componen el esquema de la base de datos. En el campo **column-name**, se capturan todos los atributos de cada una de las tablas de la BD y en el campo **data-type** los tipos de datos de cada uno de los atributos. La estructura descrita permite asociar cada tabla con sus atributos y los tipos de datos de dichos atributos. Esta asociación es aprovechada en el modelo para la generación de las propiedades de datos.

Otra de las tablas del esquema de la base de datos consultada fue **pg-catalog.pg-constraint**. Esta tabla contiene información sobre las restricciones de las BD, a través de las cuales se establecen las relaciones en el modelo ontológico. Dichas relaciones serán establecidas de acuerdo con las siguientes relaciones:

- Relación (M:M) → Restricción de Mínima cardinalidad.
- Relación (1:1) → Restricción de Máxima cardinalidad.
- Relación (1:M) → Restricción de Mínima cardinalidad.

La consulta diseñada para obtener la información de las restricciones del esquema de la BD se estructuró de la siguiente manera:

```
SELECT
pg-constraint.conname,
pg-constraint.conkey,
pg-constraint.confkey
FROM
pg-catalog.pg-constraint;
```

En la consulta que se muestra en el cuadro anterior, en el campo **conname** se determinan las restricciones del esquema de la BD. Mientras que los campos **confkey**, permiten determinar la conformación de dichas restricciones, si las llaves primarias están compuestas por una o más llaves, la posición de las llaves foráneas en las tablas, etc.

2.2.2 Descripción del algoritmo de migración

Una vez obtenidos los elementos del esquema relacional de la BD mediante las consultas descritas anteriormente se realiza el **algoritmo de migración**. Este permite crear una on-

	table_name character varying	column_name character varying	constraint_name character varying
1	pieza	cod_pieza	piezas_pkey
2	operacion	cod	operacion_pkey
3	equipo	modelo	equipo_pkey
4	obrero	numero	obrero_pkey
5	directo	obrero_numero	directo_pkey
6	auxiliar	obrero_numero	auxiliar_pkey
7	norma	operacion_cod	norma_pkey

Figura 2.3: Resultado de la consulta SQL realizada a la tabla **constraint-column-usage** del esquema de la base de datos. Fuente: elaboración propia.

tología con los elementos de su esquema relacional, mediante los pasos que se describen a continuación.

1- Crear jerarquía de clases

Como se mencionó anteriormente al consultar la tabla **constraint-column-usage**, se obtienen las tablas de la BD, sus atributos llaves y las restricciones de cada atributos llave en el esquema de la base de datos. Con esta información se procede a generar la jerarquía de clases del modelo ontológico, la que se desarrolló de la siguiente manera:

- Superclase de clases y subclases en la ontología

Para mostrar este paso, se parte del resultado de la consulta realizada a la tabla del esquema de la BD, el cual se muestra en la figura 2.3.

Se realiza un análisis para comprobar si existe repetición secuencial de algún atributo llave con la siguiente estructura:(**nombre de la tabla**)-(**atributo llave**), en el campo de los atributos llaves de la consulta. Dado que si existe esta repetición estamos en presencia de una superclase de clases y de sus subclases. Lo anteriormente expuesto se define a continuación:

Definición 1. Superclases y subclases.

Sea A una tabla cualquiera con un atributo llave Apk y dos tablas B y C cuyos atributos llaves están expresados de la forma A-Apk. Se determina que, como las llaves primarias de B y C describen a otra tabla y su llaves, estas tablas serán subclases de dicha tabla en el modelo ontológico resultante.

Al aplicar la anterior definición al modelo en el resultado mostrado en la figura 2.3 se obtiene: la tabla **obrero** que presenta como atributo llave *número* y las tablas **auxiliar** y **directo** que

presentan como atributo llave *obrero-número*. Se determina que **auxiliar** y **directo** son tipos de obreros. En el modelo ontológico, obrero será una superclase, auxiliar y directo sus subclases.

A continuación se muestra la declaración de la clase **obrero** y sus subclases **auxiliar** y **directo** en el modelo ontológico, resultante de aplicar el presente procedimiento:

```
<owl:Class>
<owl:Class rdf:about="#obrero">
<owl:Class/>
<owl:Class>
<owl:Class rdf:about="#auxiliar">
<rdfs:subClassOf rdf:resource="#obrero"/>
<owl:Class/>
<owl:Class>
<owl:Class rdf:about="#directo">
<rdfs:subClassOf rdf:resource="#obrero"/>
<owl:Class/>
```

*- **Agregar un conjunto de clases**

En el caso de no presentar repetición en el campo de los atributos llaves del resultado de la consulta, con la característica antes descrita. Se toman las tablas que conforman el esquema de la BD y se crean un conjunto de clases *sibling*¹.

2- Agregar axioma *hasKey*

El axioma *hasKey* es introducido en *OWL 2*, a través de este axioma identifican únicamente a una clase dentro de la ontología.

Para agregar este axioma al modelo ontológico, lo primero es controlar que todas las tablas de la BD están contenidas en la ontología como clases. Luego cada atributo llave de cada tabla se declara en la ontología como una propiedad de objeto inversa y funcional. Posteriormente con cada clase y su propiedad de objeto inversa y funcional. La estructura del resultado de la consulta facilita la estructuración del axioma.

A continuación se muestra la declaración del axioma *haskey* de la clase *pieza*, en el modelo ontológico:

¹Estas clases se definen como clases independientes

table_name character vary	column_name character varying	data_type character varying
operacion	cod	integer
operacion	nombre_operacion	text
obrero	numero	integer
obrero	nombre_obrero	text
obrero	calificacion	text
obrero	categoria	text

Figura 2.4: Resultado de la consulta *SQL* realizada a la tabla *information-schema.columns* del esquema de la base de datos. Fuente: elaboración propia.

```

<owl:Class>
  <owl:Class rdf:about="#pieza">
    <owl:hasKey rdf:parseType="Collection">
      <rdf:Description rdf:about="#cod-pieza"/>
    </owl:hasKey>
  </owl:Class>

```

3- Crear propiedades de datos

Las propiedades de datos en las ontologías describen información del dominio de conocimiento descrito. El procedimiento propuesto en la presente investigación, transforma los atributos de las tablas de la BD como propiedades de datos en el modelo ontológico. Para realizar este paso se consultó la tabla *information-schema.columns* del esquema de la BD. En el resultado de la consulta se obtienen: el nombre de la tablas, los atributos de las mismas y los tipos de datos de cada atributo, como se muestra en la figura 2.4.

Para la declaración de las propiedades de datos se utilizaron los tres tipos de elementos obtenidos en la consulta:

- Atributos de las tablas de la BD, estos son, las propiedades de datos en la ontología.
- Tablas de la BD, estas se emplean para declara el dominio de cada propiedad de datos.
- Los tipos de datos de cada atributo, estos se utilizan para declarar el rango de las propiedades de datos.

La declaración del **dominio** de las propiedades de datos, se realiza de una manera sencilla. Pues la estructura de los datos del resultado de la consulta permite asociar a cada tabla con sus atributos correspondientes. Por tanto cada tabla -clases en la ontología es agregada como dominio de la propiedad de dato.

Por otra parte, para la declaración del **rango** de estas propiedades de datos requirió realizar un proceso de conversión o *mapping*. Esto permitió transformar los tipos de datos que son contenidos en bases de datos¹ a tipos de datos soportados en las ontologías. Algunos de los tipos de datos a las que se le realizó esta transformación se muestran en la tabla 2.1. Los otros tipos de datos transformados son mostrados en el Anexo 2.

Tabla 2.1: Muestra de tipos de datos soportados en BD mapeados a tipos de datos soportados por las ontologías. Fuente: elaboración propia.

Tipos de datos en la BD	Tipos de datos en la ontología
character	xsd:string
character varying	xsd:string
money	xsd:double
double precision	xsd:double
time with time zone	xsd:data-time-tamp

Luego de realizar esta conversión cada tipo de dato, es asignado como rango de su propiedad de dato correspondiente.

La declaración en el modelo ontológico de la propiedad de datos **descripción**, en la cual se describe información de la clase **pieza** se muestra a continuación:

```
<owl:DatatypeProperty rdf:about="#descripcion">
  <rdf:type rdf:resource="#FunctionalProperty"/>
  <rdfs:domain rdf:resource="#pieza"/>
  <rdfs:range rdf:resource="#string"/>
</owl:DatatypeProperty>
```

¹Aunque particularmente este proceso se realiza para PostgreSQL los tipos de datos que se transforman son soportados en cualquier gestor de base de datos.

3- Crear relaciones en la ontología

Para realizar este paso se tuvieron en cuenta los tipos de relaciones presentes en el diseño de bases de datos. A través de estas se estructuró las relaciones entre clases la ontología.

Los tipos de relaciones que se tuvieron en cuenta son:

- Relación (1:M): la tabla que representa la entidad mucho adquiere la llave primaria de la entidad 1 como atributo y los atributos de la relación.
- Relación (1:1): si las dos tablas tienen sentido obligatorio, la llave primaria es adquirida por una de ellas como atributo de la otra. Si el sentido de la relación es opcional, la llave es adquirida por la entidad obligatoria. Si es opcional en ambos sentidos se crea una nueva clase con las llaves de ambas clases y los atributos de la relación.
- Relación (M:M): surge una nueva tabla para representar la relación con la llave primaria formada por la unión de las llaves de las entidades involucradas en la relación. Como atributos; los atributos de la relación.

De acuerdo con lo planteado anteriormente, el procedimiento para establecer las relaciones en el modelo ontológico, se centró en el control de las llaves foráneas. Pues las tablas que presenten este tipo de llaves serán las clases que contendrán las relaciones. Posteriormente las tablas a las cuales hacen referencias las llaves foráneas se crearán en el modelo como propiedades de objetos para describir la relación.

*- Agregar relaciones (M:M) en el modelo ontológico

Cómo resultado de este tipo de relación, se crea una nueva tabla cuya llave primaria se conforma con las llaves de las entidades que intervienen en la relación. Lo primero es identificar la tabla cuya llave primaria esté compuesta por otras llaves. Esto se obtiene en campo **confkey** perteneciente a la tabla **pg-catalog.pg-constraint** del esquema de la base de datos, como es mostrado en la figura 2.5.

Como se observa en la figura 2.5, la llave primaria de la tabla **norma** está compuesta por varias llaves, según la estructura de esta mostrada en el campo **confkey**. Por tanto en la clase **norma** se contendrá la relación en el modelo ontológico.

Luego se determinan las entidades cuyas llaves primarias conforman la llave primaria de la relación **norma**. Esto se realiza con las entidades cuyas llaves se contienen en la relación

	conname name	conkey smallint[]	confkey smallint[]
5	equipo_pkey	{1}	
6	obrero_pkey	{1}	
7	directo_pkey	{1}	
8	auxiliar_pkey	{1}	
9	norma_pkey	{1,2,3}	
10	fk_directo_obrero	{1}	{1}
11	fk_auxiliar_obrero	{1}	{1}
12	fk_norma_operacion	{1}	{1}
13	fk_norma_equipo	{2}	{1}
14	fk_norma_piezas	{3}	{1}

Figura 2.5: Resultado de la consulta *SQL* realizada a la tabla ***pg-catalog.pg-constraint*** del esquema de la base de datos. Fuente: elaboración propia.

como llaves foráneas. En la figura 2.5 la estructura de las llaves foráneas se representan **fk-(tabla relación)-(tabla dueña de la llave foránea)**.

De acuerdo con esta estructura las tablas cuyas llaves foráneas que intervienen en la relación **norma** son: **pieza, equipo, operación**. Estas entidades son tomadas como propiedades de objetos para la declaración de la relación en el modelo ontológico.

Obtenidas las tablas que intervienen en la relación, se declara en la ontología a través de restricciones de **Mínima** cardinalidad, dado que las relaciones (M:M) se parafrasean:

“En una norma intervienen como mínimo una pieza, una operación y un equipo”.

***- Crear relaciones (1:1) en el modelo ontológico**

En este tipo de relación la llave foránea puede estar en cualquiera de las dos entidades dependiendo de el sentido en que se plantee la relación. Entonces nada más lógico que identificar la entidad que absorbe la llave primaria de la otra entidad como llave foránea. Para explicar este proceso se utiliza la figura 2.6.

En la figura 2.6, en el campo **confkey** la entidad que presenta la llave foránea se describe mediante un número que indica la existencia de la llave foránea en la entidad. Cabe señalar que en este caso se trabaja directamente sobre este tipo de llave, no sobre la llave primaria pues la entidad relación mantiene su llave primaria.

Luego de identificar dicha entidad se procede a estructurar la relación igual al proceso descrito

	conname name	conkey smallint[]	confkey smallint[]
7	cliente_pkey	{1}	
8	vehiculo_pkey	{1}	
9	parqueo_pkey	{1}	
10	norma_pkey	{1,2,3}	
11	fk_cliente_cuenta	{2}	{1}
12	fk_parqueo_vehiculo	{2}	{1}
13	fk_norma_pieza	{1}	{1}

Figura 2.6: Identificación de una relación (1:1) tras consultar la tabla *pg-catalog.pg-constraint*.
Fuente: elaboración propia.

para las relaciones (M:M). Aclarar que en este caso el tipo de cardinalidad es **Máxima**, dado que el parafraseo de este tipo de relación es el siguiente:

“A un cliente le corresponde como máximo una cuenta”.

*- Crear relaciones (1:M) en el modelo ontológico

Este tipo de relación es una mezcla de las dos formas de relaciones anteriormente descritas. En este caso la entidad relación que corresponde al valor máximo absorbe la llave de la otra entidad, estructurándose la relación de acuerdo con el procedimiento anteriormente descrito.

Resaltar que en este tipo de relación la cardinalidad será **Mínimo**, dado que este tipo de relación se parafrasea:

“A un parqueo le corresponde como mínimo un vehículo”.

A continuación se presenta la modelación ontológica de las relaciones (1:M):

```

<owl:Class>
<owl:Class rdf:about="#parqueo">
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty rdf:resource="#vehículo"/>
<owl:minCardinality rdf:datatype="#nonNegativeInteger">1
</owl:minCardinality>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

```

	cod_pieza integer	descripcion_pieza text	
1	12	pieza1	– Individual 1
2	13	pieza2	– Individual 2
3	14	pieza3	– Individual 3

Figura 2.7: Declaración de instancias de clases en el modelo ontológico, mediante las tuplas de la tabla **pieza**. Fuente: elaboración propia.

*- Crear instancias de clases

Los procesos descritos con anterioridad, permiten relacionar los elementos del esquema relacional de la BD, para desarrollar una forma de representación de conocimiento. El procedimiento que se describe a continuación, permite la representación de la información de la BD a través de **instancias** de clases en la ontología.

Una vez obtenidas todas las tuplas de cada tabla en la base de datos, se declaran las instancias de clases (tablas de la BD) en el modelo ontológico. Esta declaración se realiza sobre la siguiente definición:

Definición 2. Creación de instancias.

En una tabla T de un modelo relacional M, las instancias de T representada como clase en una ontología O, corresponderán al conjunto de tuplas que componen la tabla T en el modelo M.

La figura 2.7 refleja lo planteado por la idea anterior.

Al declarar instancias de la clase **pieza** según la idea representada en la figura 2.7. Se nombra cada instancia según la clase (tabla) seguido del número correspondiente a dicha tupla. Las instancias de la clase **piezas** son declaradas: **pieza1, pieza2..pieza-n**, de acuerdo con la cantidad de tuplas en la tabla correspondiente.

Luego de la declaración de las respectivas instancias de cada clase, se declaran las **data property assertion**. Mediante las cuales se representan la información de cada instancia. La estructura de este axioma se realiza de la siguiente manera:

Lo primero es obtener en la ontología las propiedades de datos y sus respectivos tipos de datos por cada clase. Luego las instancias correspondientes a dicha clase, como se muestra

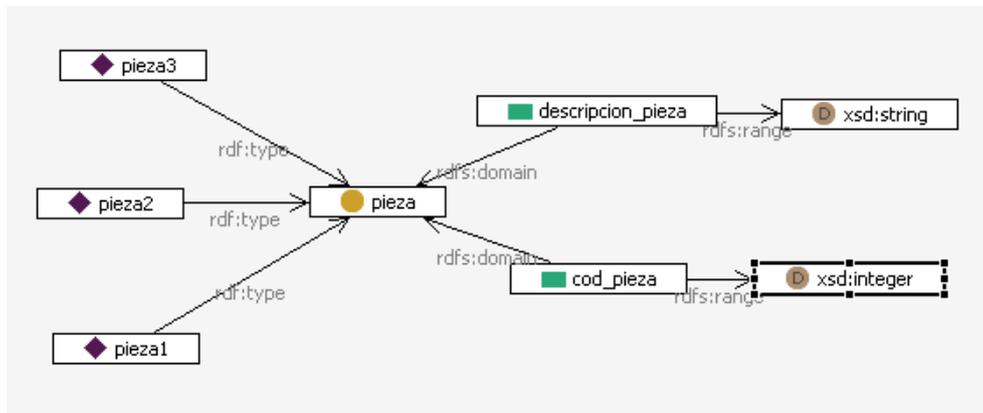


Figura 2.8: Asociación de instancias y propiedades de datos, correspondientes a la clase **pieza**.
Fuente: elaboración propia.

en la figura 2.8.

La asociación mostrada en la figura 2.8, permite declarar las triplas correspondientes a este axioma con la siguiente característica:

instancia *propiedad* **valor de la tupla correspondiente**

Al estructurar las tripla para la primera intancias de la clase **pieza**, de acuerdo con los que se muestran en la figura 2.7, se obtiene:

pieza1 *cod* – *pieza* **12**

pieza1 *descripcion* – *pieza* **pieza1**

Señalar que para declarar estas triplas es necesario:

Realizar el control de los tipos de datos de cada propiedad de datos y del valor que corresponde en la tripla. Una vez realizado este control, las **data property assertion** en el modelo ontológico se modelan:

```
<owl:NamedIndividual rdf:about="#pieza1">
<rdf:type rdf:resource="#pieza"/>
<cod-pieza rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">12 </cod-pieza>
<descripcion-pieza rdf:datatype="http://www.w3.org/2001/XMLSchema#string">pieza1
</descripcion-pieza>
</owl:NamedIndividual>
```

2.2.3 Descripción del procedimiento para la clasificación de la información

Uno de los requerimientos para resolver el problema científico de la presente investigación es, el diseño de una **estrategia de razonamiento**. A través de la cual un razonador evalúe la información de la base de datos en, **conceptos, relaciones, roles y atributos**.

El procedimiento que se propone se centra en la modelación de las definiciones de **conceptos, relaciones, roles y atributos** dadas por Nicola Guarino en (17) como consultas *SPARQL*. Mediante estas consultas se clasificará la información de la BD en las categorías antes señaladas. A continuación se describe el procedimiento propuesto.

1- Clasificación de conceptos

Guarino plantea, "los conceptos no tienen nombre al menos en el plano conceptual, de tenerlo, entonces no necesitaríamos ontologías"(17). Para materializar (por así llamarlo) un concepto se necesita una entidad. Esta entidad necesitaría un nombre y un identificador. Esto para identificarla de forma universal (al menos, en el dominio discursivo que se trate).

Como caso particular de la presente investigación, se transforma una BD a una ontología. En las bases de datos, las entidades se identifican a través de una llave primaria y en las ontologías esto es modelado con el axioma *hasKey*.

A continuación se muestra un modelo de consulta en el cual se logra definir el axioma mencionado. Este permite definir las entidades y su identificador. Estas entidades los conceptos presentes en la base de datos.

```
SELECT ?x ?y
WHERE{
  ?x owl:haskey ?y
}
```

Al estructural la tripla **?x owl:haskey ?y**, el valor de la variable **x** corresponderá al nombre de la entidad. El valor de la variable **y** corresponderá al identificador de dicha entidad en el universo de discurso modelado.

2- Clasificación de las relaciones

Estas surgen al “relacionar dos conceptos o más conceptos cualesquiera en un universo discursivo y como resulta de esta surge un nuevo concepto(17)”. En adición a la definición anterior, se considera como relación a toda entidad resultante de una determinada asociación de dos o más entidades.

De acuerdo con lo anterior, el modelo de consulta que se diseñó para clasificar las relaciones debe ser capaz de cumplir con los siguientes requisitos:

- Identificar las entidades que interviene en la relación.
- Identificar la forma mediante la cual se establece dicha relación.

En el siguiente cuadro se propone un modelo de consulta que cumple con los requisitos anteriormente descritos.

```
SELECT ?x ?y ?z
WHERE{
  ?x rdfs:subClassOf ?y
  ?y owl:onProperty ?z
}
```

La tripla **?x rdfs:subClassOf ?y**, permite obtener las formas de relación¹ presente en la entidad correspondiente a la variable **x**, representadas a través del axioma **subClassOf**. En este punto cabe señalar que mediante las formas de relación corresponden a la variable **y**.

Por otra parte, la tripla **?y owl:onProperty ?z**, permite identificar las entidades que intervienen en la forma de relación representada en la variable **y** como propiedades. Pues las entidades que intervienen en una determinada relación se convertirán en propiedades de objetos para describir dicha relación. Estas entidades corresponden al valor de la variable **z**.

3- Clasificación de atributos

¹Como caso particular de la presente investigación las formas de relación son de Cardinalidad

Guarino reconoce a un **atributo** como, “un concepto que, en el dominio de interés, tiene una interpretación”(17). Con relación a lo anterior, se entiende como atributo, todo concepto que a través de su interpretación se describa una determinada entidad en el dominio de interés.

Ontológicamente esto es modelado a través de las propiedades de datos. Por tanto para clasificar un atributo se hace indispensable obtener:

- El dominio que se describe, la clase.
- El concepto mediante el cual se describe dicho dominio, la propiedad de dato.

A continuación se muestra un modelo de consulta *SPARQL*, mediante el cual se definen los atributos del modelo ontológico.

```
SELECT ?x ?y
WHERE{
  ?x rdfs:domain ?y
  ?x rdf:type owl:DatatypeProperty
}
```

En la tripla **?x rdfs:domain ?y**, los conceptos correspondientes al valor de la variable **x**. Estos describen la información correspondiente a los dominios representados por la variable **y**.

Por otro lado, la tripla **?x rdf:type owl:DatatypeProperty**, garantiza que los conceptos de la variable **x**, sean del tipo propiedad de dato.

4- Clasificación de roles

Un rol es, “un concepto que participa de una relación particular, no semánticamente rígido¹ si al participar de esta relación, es necesario una Entidad para explicarlo”(17).

De acuerdo con la anterior definición, las entidades dentro del modelo que cumplen la condición para ser clasificada como un **rol** son las subclases. Pues para la definición de dichas entidades en el dominio de conocimiento será necesario utilizar la superclase.

El modelo de consulta que se propone a continuación, permite obtener las subclases de la ontología, los roles presentes en la base de datos.

¹No se puede realizar definición alguna de la entidad a través de la relación que participa.

```
SELECT ?x ?y
WHERE {
  ?x rdfs:subClassOf ?y
  ?y rdf:type owl:Class
}
```

En la tripla **?x rdfs:subClassOf ?y**, el valor de la variable **x**, corresponden a las subclases del modelo y el valor de la variable **y** a la superclase. Esta puede ser una clase del dominio o la clase *owl:Thing* como superclase general en el modelo.

Para garantizar que la superclase sea una de las clases del dominio se ha añadido la tripla **?y rdf:type owl:Class**. La que garantiza que la variable **y** sea una clase en el modelo.

2.2.4 Resultados obtenidos del procedimiento

Para evaluar el procedimiento propuesto en la presente investigación, se realizaron diez (10) casos experimentales. Cada caso experimental cuenta como origen de datos un modelo relacional extraído de *Data Base Answers*¹. Estos representan información de diferentes dominios.

Estas bases de datos están desarrolladas sobre el gestor **PostgreSQL**, todas cumplen con los requisitos de normalización. Esto garantiza que no existe redundancia en sus datos. Cada uno de los modelos utilizados son descritos en la tabla 2.9².

A continuación se describe el **caso experimental 1**, a través de este se muestran los resultados del **algoritmo de migración** y del **procedimiento para la clasificación de la información**. El resto de los resultados de los casos experimentales se muestran en el Anexo 3.

2.2.4.1 Resultados obtenidos del algoritmo de migración: Caso experimental 1

- **Caso experimental 1**. Base de datos relacional Control del Proceso 1

¹<http://www.databseanswer.org/data-model>

²Significados de las abreviaturas en la tabla: NT= número de tablas, NA= número de atributos, NPK= número de llaves primarias y NFK= número de llaves frías

Tabla 2.2: Bases de datos relacionales utilizadas en los casos experimentales. Fuente: elaboración propia.

Nombre	Descripción del dominio	NT	NA	NPK	NFK
Control del Proceso 1	Producción de piezas	7	18	7	4
<i>Air Reservation</i>	Reservaciones de vuelos	10	39	10	7
BMEWS	Inteligencia de negocio	6	8	6	2
<i>Children Institution</i>	Casas de acogida de niños (Unicef)	9	10	9	2
<i>Children of Family Support</i>	Adopción de niños (Unicef)	10	13	10	7
<i>Gaming Web Sites</i>	Video juegos	10	32	10	11
ISWC	Publicaciones científicas	9	33	9	8
<i>National Healt</i>	Control de pacientes	7	25	7	6
<i>Open University</i>	Maticulas universitarias	10	24	10	10
<i>Theater Booking</i>	Reservación de teatros	8	29	8	6

Para el primer caso experimental se utilizó como origen de datos la BD **Control del Proceso 1**, esta contiene información acerca del proceso productivo en una industria mecánica. Los elementos que estructuran esta base de datos se muestran en la tabla 2.2. Tras aplicar el algoritmo de migración se obtiene como resultado la ontología mostrada en la figura 2.9.

En la tabla 2.3, se muestra los elementos que componen a la ontología obtenida de la base de datos utilizada en el caso experimental.

Tabla 2.3: Elementos que componen la ontología resultante de aplicar el algoritmo de migración a la BD Control del Proceso 1. Fuente: elaboración propia.

Caso experimental	Conceptos	Atributos	Relaciones	Axioma <i>hasKey</i>
1	7	18	4	7

Tras migrar la base de datos a ontología según el algoritmo propuesto, se analizan los indicadores para la evaluación de la **eficacia** tratados en la sección 1.7. Mediante estos se determina el principal requerimiento de los procedimiento de migración, el máximo grado de similitud.

Para realizar la evaluación de la eficacia se toman los parámetros **conceptos**, **relaciones** y **atributos** por ser parámetros generales y medibles en ambos modelos.

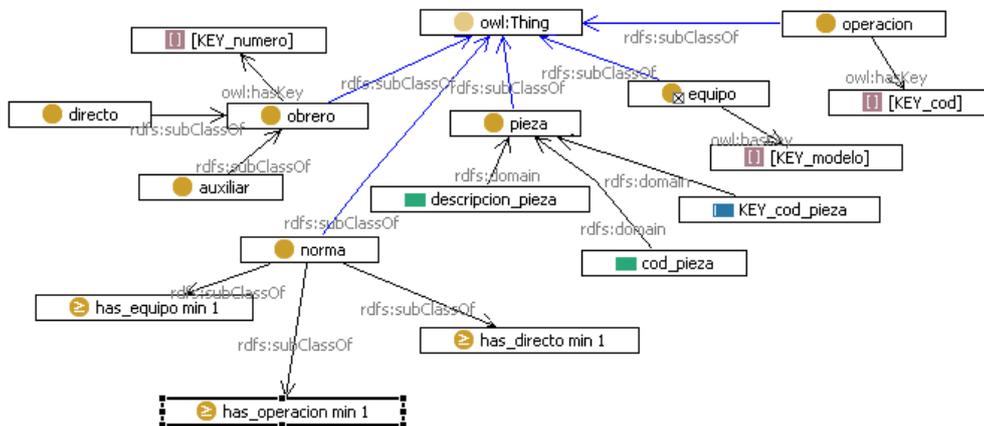


Figura 2.9: Ontología resultante una vez aplicado el algoritmo de migración a la BD Control del Proceso 1. Fuente: elaboración propia.

- Análisis de la correspondencias de conceptos

Antes de realizar la evaluación para los conceptos, cabe señalar que en el presente trabajo se desarrolla un procedimiento de migración. Por lo que los indicadores de correspondencias correctas descubiertas (CCD), correspondencias correctas (CC) y descubiertas (CD) son iguales. Pues estos procedimientos crean un modelo originado de los elementos de otro modelo existente, esta aclaración es aplicable los atributos y relaciones.

De acuerdo con las correspondencias entre las tablas del esquema relacional y los conceptos que componen la ontología, mostrados en la tabla 2.11. Los indicadores de correspondencias toman los siguientes valores, CCD = 7, CC = 7 y las CD = 7. Con estos valores se procede al cálculo de los indicadores de eficacia:

1. Precisión:

$$P = \frac{n_{CCD}}{n_{CD}} = \frac{7}{7} = 1$$

2. Exhaustividad:

$$R = \frac{n_{CCD}}{n_{CC}} = \frac{7}{7} = 1$$

Tabla 2.4: Correspondencias entre tablas en la BD y conceptos en la ontología resultante . Fuente: elaboración propia.

Tablas	Conceptos(owl:Class)
pieza	pieza
operación	operación
obrero	obrero
equipo	equipo
auxiliar	auxiliar
directo	directo
norma	norma

3. Medida F (medida E con b=1):

$$F = \frac{1 + b^2}{\frac{b^2}{R} + \frac{1}{P}} = \frac{1 + 1^2}{\frac{1^2}{1} + \frac{1}{1}} = 1$$

- Análisis de la correspondencias de atributos

De la misma forma, se analizan las correspondencias para los atributos. La tabla 2.12 muestra las correspondencias entre una selección de los atributos en la base de datos y los de la ontología.

Tabla 2.5: Selección de correspondencias entre atributos en la BD y propiedades de datos en la ontología resultante. Fuente: elaboración propia.

Atributos	Propiedades de datos(owl:DataTypeProperty)
modelo	modelo
nombre-equipo	nombre-equipo
país	país
número	número
nombre-obrero	nombre-obrero
calificación	calificación
categoría	categoría

Para el caso de los atributos, los indicadores de correspondencias toman los siguientes valores, CCD = 18, CC = 18 y CD = 18. Como se aprecia los valores de los indicadores de

correspondencias son iguales, por tanto, los valores de los indicadores de eficacia para los atributos van a ser igual a 1.

- Precisión: $P = 1$, - Exhaustividad: $R = 1$, - Medida F: $F = 1$

- Análisis de la correspondencias de relaciones

En el caso de las relaciones, las correspondencias se determinan entre las **fk** del modelo relacional y las propiedades de objetos que relacionan conceptos en la ontología. Las correspondencias para las relaciones son mostradas en la tabla 2.13

Tabla 2.6: Correspondencias entre **fk** en la BD y propiedades de objetos en la ontología resultante.
Fuente: elaboración propia.

Relaciones en la BD	Propiedades de objetos(<i>owl:ObjectProperty</i>)
fk-norma-operación	has-operación
fk-norma-equipo	has-equipo
fk-norma-pieza	has-pieza
fk-norma-directo	has-directo

Para las relaciones los indicadores de correspondencias toma los valores CCD = 4, CC = 4 y CD = 4. Los indicadores de eficiencia toman al igual que en los casos anteriores el valor 1:

- Precisión: $P = 1$, - Exhaustividad: $R = 1$, - Medida F: $F = 1$

La eficacia del algoritmo de migración puede medirse como la media ponderada de las medidas F de las correspondencias entre conceptos, atributos y relaciones.

$$Ef(Caso) = \frac{C}{C + A + R} * F_{concep.} + \frac{A}{C + A + R} * F_{atrib.} + \frac{R}{C + A + R} * F_{relac.}$$

$$C = \#conceptos + \#tablas$$

$$A = \#atrib.ont + \#atrib.BD$$

$$R = \#rel.ont + \#fk$$

Así la eficacia (Ef) del algoritmo de migración para el caso experimental 1 será:

$$Ef(Experimento1) = \frac{7}{7 + 18 + 4} * 1 + \frac{18}{7 + 18 + 4} * 1 + \frac{4}{7 + 18 + 4} * 1 = 0.99$$

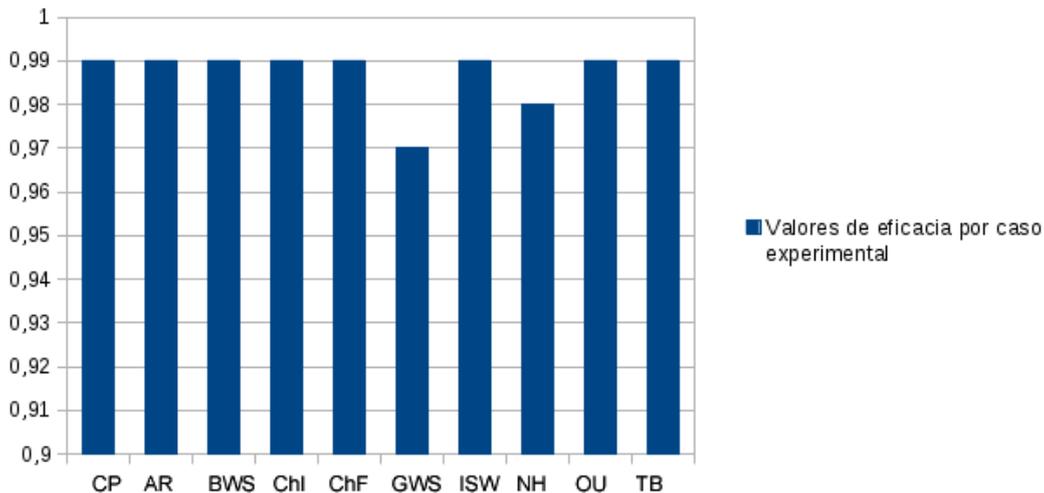


Figura 2.10: Valores de eficacia por cada caso experimental. Fuente: elaboración propia.

Como se aprecia el valor de la eficacia para el caso experimental 1 es de 0.99. Esto significa que con el algoritmo propuesto en la presente investigación para este caso experimental cumple con el principal requerimiento para los sistemas de migración. El máximo grado de similitud entre la base de datos y la ontología resultante.

La figura 2.10¹ muestra una gráfica que ilustra los valores de eficacia para cada caso experimental realizado para probar el algoritmo de migración.

Al determinar la media muestral de los valores de eficacia para cada caso experimental se obtiene una $\overline{M}(Ef) = 0.98$. Con este valor se determina, el algoritmo de migración propuesto en la presente investigación garantiza el **máximo grado de similitud** entre la base de datos y la ontología resultante.

2.2.4.2 Resultados obtenidos con el procedimiento de clasificación de la información

El procedimiento de clasificación de la información que se propuso en la sección 2.2.3, evalúa como conceptos, atributos, relaciones y roles la información contenida en la base de datos. Al

¹Significado de las abreviatutas en la figura: CP= Control de Proceso 1, AR= *Air Reservation*, BWS= BMEWS, ChI= *Children Institution*, ChF= *Children of Family Support*, GWS= *Gaming Web Sites*, ISW= ISWC, NH= *National Health*, OU= *Open University*, TB= *Theater Booking*

aplicar dicho procedimiento en el caso experimental 1 se obtiene el resultado mostrado en la tabla 4.14¹.

La evaluación para el caso de los **conceptos** coinciden con las tablas que componen la BD. Esto determina que en una base de datos los conceptos modelados son las tablas. En el caso de las **relaciones**, son las tablas en que se relacionan otras mediante la llave foránea (véase la tabla 2.6). De acuerdo con lo anterior, en el caso experimental 1 la entidad **norma** constituye una relación.

En el caso de los **atributos**, estos serán los atributos de las tablas en la BD. Por último los **roles** para esta BD son las entidades **auxiliar** y **directo**, estas son tipos de la entidad **obrero** (véase sección 2.2.2). Esto implica que en toda definición que se realice para estas entidades se deberá incluir a la entidad **obrero** en la misma, de acuerdo con la definición de rol dada por Guarino.

Tabla 2.7: Clasificación de la información contenida en la base datos utilizada en el caso experimental 1. Fuente: elaboración propia.

Conceptos	Relaciones	Atributos	Roles
pieza	norma	clasificación	auxiliar
operación		cantidad	directo
equipo		categoría	
obrero		cod-norma	
norma		cod-pieza	
directo		modelo	
auxiliar		país	
		descripción	
		número	

La tabla 2.8, recoge cuantitativamente la clasificación de conceptos, relaciones, roles y atributos. Realizadas mediante el mecanismo de clasificación propuesto en la presente investigación para cada caso experimental. La clasificación cualitativa para cada caso experimental se muestra en el Anexo 4.

¹En el caso de los atributos se muestra una selección

Tabla 2.8: Clasificación de conceptos, roles, atributos y relaciones para cada caso experimental.

Fuente: elaboración propia.

Nombre	Conceptos	Relaciones	Atributos	Roles
Control del Proceso 1	7	1	18	2
<i>Air Reservation</i>	10	7	39	-
BMEWS	6	2	8	-
<i>Children Institution</i>	9	2	10	-
<i>Children of Family Support</i>	10	7	13	-
<i>Gaming Web Sites</i>	10	11	32	-
ISWC	9	8	33	-
<i>National Healt</i>	7	6	25	-
<i>Open University</i>	10	10	24	-
<i>Theater Booking</i>	8	6	29	-

Por otra parte, en la figura 2.11, se presentan los porcentajes de los elementos que son clasificados para los casos experimentales realizados.

Como se aprecia en la figura 2.11, el menor porcentaje corresponde al caso de los **roles**. Pues no todas las bases de datos existentes requieren en su diseño de especializaciones entre entidades para representar la información. Esto depende del dominio para el cual se diseñe la BD.

No así en el caso de los **atributos** dado que ocupan el más alto porcentaje. A través de estos se representa la información contenida en las bases de datos.

Para el caso de los **conceptos** y **relaciones** tienen porcentajes semejantes. Estos son los elementos de los cuales se obtiene la información.

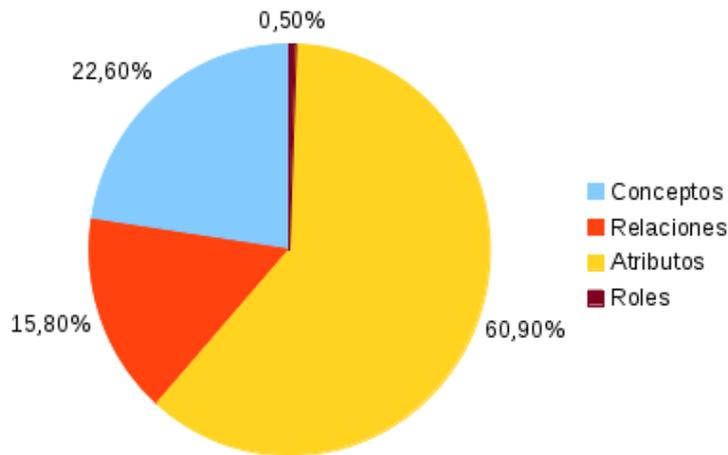


Figura 2.11: Porcientos que representan cada elemento clasificado en la composición de los elementos de una base de datos. Fuente: elaboración propia.

2.3 Conclusiones parciales

- Los elementos que componen la base de datos mediante los que se logra definir equivalente en las ontologías son, las tablas, las llaves primarias, las llaves foráneas, los atributos y las tuplas.
- Al migrar una base de datos a ontología se logra un modelo que facilita la clasificación de sus componentes, gracias a una información dotada de mayor significado.
- Los procedimientos de migración proveen un modelo mejor definido, al proporcionar una visión total de los elementos del modelo migrado. Lo anterior se dificulta en el caso de los sistemas ODBA, que proporcionan una visión parcial de los datos.
- El lenguaje de consultas *SPARQL* proporciona de forma eficiente, la definición de un mecanismo de clasificación de la información.
- El mayor porcentaje dentro de los elementos que componen una base de datos lo ocupan los atributos, dado que a través de estos se representa la información contenida en la base de datos.

2.4 Conclusiones generales

- Se desarrolló un procedimiento que clasifica la información almacenada en una base de datos en *PostgreSQL*, según las necesidades del estudio. Este procedimiento está compuesto por dos métodos, un algoritmo de migración y una estrategia de razonamiento, a través de estos se cumple con el objeto de estudio de la presente investigación.
- Los resultados obtenidos al probar el algoritmo de migración propuesto con diez (10) bases de datos relacionales sobre el gestor *PostgreSQL* demuestran que, este método garantiza el máxima de similitud entre las base de datos y la ontología resultante, con un alto grado de eficacia.
- La estrategia de clasificación propuesta, determina conceptos, relaciones, atributos y roles en la información almacenada en una base de datos en *PostgreSQL*. En total correspondencia con las definiciones de estos elementos sugeridas por Nicola Guarino.
- El mayor porcentaje de composición de las bases de datos corresponden a los atributos. Pues a través de estos se cumple el principal objetivo de los modelos de base de datos: la representación de información.

2.5 Recomendaciones

1. Someter el procedimiento a validación por expertos humanos. Sobre todo porque puede darse el caso de que el razonador clasifique como un rol algo que no es más que una relación, pero las relaciones y la información evaluada, llevó al razonador a hacer esta inferencia.
2. Extender el procedimiento propuesto a otros gestores de bases de datos existentes. A modo de generalizar este modelo.

Referencias Bibliográficas

- [1] ABRAHAM SILBERSCHATZ, H.F.K.B.L., BELL LABORATORIES (1990). *FUNDAMENTOS DE BASES DE DATOS*. Instituto Indio de Tecnología, Bombay. 1, 7, 8, 9, 12, 13
- [2] ANTONELLA POGGI, M.R., MARIANO RODRIGUEZ-MURO (2010). Ontology-based database access with dig-m astro and the obda plugin for protégé. 40
- [3] BAADER, F. & NUTT., W. (2003). Basic description logics. 29
- [4] BORGIDA, A. & MYLOPOULOS, J. (2004). *Data semantics revisited*. In *Semantic Web and Databases*.. 2
- [5] BORST, W. (1997). *Construction of Engineering Ontologies*.. Ph.D. thesis, University of Tweenty. 29
- [6] BORST, W. (1997). *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. 32
- [7] C. DOS SANTOS, E.N. & FURTADO., A. (1979). A data type approach to the entity-relationship model. 9
- [8] CASTELLS, P. (2010). Búsqueda semántica basada en conocimiento del dominio. 3
- [9] CHEN, P. (1975). The entity - relationship model. towards a unified view of data. 8
- [10] CODD., E. (1970). *A relational model of data for large shared databanks*.. 9, 13
- [11] CONTRERAS, J. (2000). Tutorial ontologías. 39
- [12] DAVIS, M. (2004). *The Business Value of Semantic Technologies. A TopQuadrant Special Report*.. 36
- [13] ELMASRI, R. & NAVATHE, S. (2000). *Fundamentals of Database Systems*.. 33
- [14] FOO, D. (1999). *Ontology research and development*. 22
- [15] GARY W. HANSEN, J.V.H. (1995). *Sistemas de Bases de Datos*. 1, 2, 8, 9, 11, 12, 14, 15, 38

- [16] GRUBER, T. (1993). A translation approach to portable ontology specifications. 3, 18, 29
- [17] GUARINO, N. (1992). Concepts, attributes, and arbitrary relations: Some linguistic and ontological criteria for structuring knowledge bases. 62, 63, 64
- [18] GUARINO, N. (1998). Formal ontology and information systems. 19
- [19] GUARINO, N. (2008). Formal ontology and information systems. 3
- [20] HAYES, P. (2004). Rdf semantics. 35
- [21] HÉCTOR OSCAR NIGRO, D.X., SANDRA GONZÁLEZ CÍSARO (2002). Ontologías en el proceso de descubrimiento de conocimiento en bases de datos. 4
- [22] HERMAN., I. (2009). Introduction to the semantic web. 3, 22
- [23] IGNACIO J. BLANCO, M.A.V., CARMEN MARTÍNEZ-CRUZ² (2006). Arquitectura para la integración de esquemas relacionales difusos basada en ontologías: una aplicación para la web. *Dpto. Ciencias de la Computación e I. A., Universidad de Granada.*. 1
- [24] JIM, E.S.V. (2006). Getting started with protege-frames. 23
- [25] JOHN HEBELER, R.B.A.P.L., MATTHEW FISHER (2000). *Semantic Web Programming*. Wiley Publishing, Inc. 2, 3, 39
- [26] KASHYAP, V. & SHETH, A.P. (1992). *Database Semantics Conference on Interoperable Database Systems*. 2, 34, 39
- [27] LEE, T.B. (1998). Relational databases on the semantic web. 3
- [28] LEE, T.B. (2001). The semantic web. 3
- [29] MARIANO RODRIGUEZ LUNO, D.C., LINA LUBYTE (2009). Realizing ontology base data access: A plug-in for proteégé. xi, 40, 41
- [30] MATTHEW HORRIDGE, A.R., HOLGER KNUBLAUCH (2004). *A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools*. 31
- [31] MEERSMAN, R. (1996.). *An essay on the role and evolution of data(base) semantics*. 35
- [32] MYLOPOULOS., J. (2004). Course on conceptual modelling. 33
- [33] N. GUARINO, C.W. (2000). Identity and subsumption. the semantics of relationships: An interdisciplinary perspective, kluwer. 20, 31
- [34] NDEZ L OPEZ, A.G.P.M.F. & CORCHO., O. (2003). Ontology engineering. with examples from the areas of knowledge management, e-commerce and the semantic web. 35
- [35] NOY, N. & KLEIN, M. (2003). *Ontology evolution: Not the same as schema evolution.*. 34

- [36] OBANDO, P.A. (2010). Administración del sgbd postgresql. 16
- [37] P. SCHEUERMANN, G.S. & WEBER, H. (1979). Abstraction capabilities and invariant properties modeling with the entity-relationship approach. 9
- [38] PETER F. PATEL-SCHNEIDER, P.H. & HORROCKS, I. (2004). Owl web ontology language semantics. 35
- [39] PETER SPYNS, R.M. & JARRAR, M. (2002). *Data modelling versus ontology engineering..* 34
- [40] POLLOCK, J.T. (2009). *Semantic Web For Dummies*. Wiley Publishing, Inc. 35, 37
- [41] PRESUTTI., V. (2009). Handbook of ontologies for business interaction. ontologies for business interaction. 3
- [42] QUINE, W. (1961). From a logical point of view, nine logico-philosophical essays. *Cambridge, Massachusetts: Harvard University Press*. 18
- [43] R. NECHES, T.G.T.S.T., E. FININ (1991). Enabling technology for knowledge sharing. 18
- [44] REITER, R. (1984). *On Conceptual Modelling*. 35
- [45] RODRIGUEZ, J.B. (2007). *Modelo para la definición automática de correspondencias semánticas entre ontologas y modelos relacionales*. Ph.D. thesis, Universidad Politécnica de Madrid. xi, 1, 2, 33, 34, 35, 38, 41, 42, 43
- [46] ROMAN, F. (2006). Notas de clases para introduccion a la lgica. 25
- [47] RUIZ, F. (2000). Modelos de datos. 7
- [48] SMITH, J. & SMITH, D. (1977). *Database abstractions aggregation and generalization*. 9
- [49] TOJANOVIC., N.E.S.T..L.J.S. (2002). Searching for the knowledge in the semantic web. *American Association for Artificial Intelligence*. 37
- [50] WILEY, J. & SONS (2003). *Towards the Semantic Web. Ontology-driven Knowledge Management..* 18

Capítulo 3

Anexos

3.1 Anexo 1. Diagrama de secuencia de procedimiento propuesto

El analista ejecuta unas consultas a la base de datos, mediante la cual obtiene una serie de elementos de su esquema relacional. Luego con los elementos del esquema, valida si existe una repetición secuencial dentro de las llaves primaria con la estructura (tabla_llave primaria), para determinar si existe una relación de clases y subclase. Las clases que presentan este forma de llave primaria serán subclase de la tabla a la que se hace referencia mediante dicha llave. Luego las tablas que no presentan este tipo de llave serán creadas como un conjunto de clases en la ontología. Luego se valida si todas las tablas de la base de datos están contenidas en el modelo como clases, seguido se toman las llaves primarias de cada tabla para crear el axioma haskey. Luego se toman tablas, los atributos de cada tabla y sus tipos de datos para crear las propiedades de datos en la ontología. En este paso se controlan los tipos de datos de cada atributo, para conformar los rangos de las propiedades de datos, ya que los que no son soportados por las ontologías son convertidos a un equivalente que se contenga. Después se controlan las llaves foráneas de la base de datos para establecer las relaciones en el modelo, las relaciones (M:M) se crean como relaciones de cardinalidad Mínima, las relaciones (1:1) con cardinalidad Máxima y las relaciones (1:M) de cardinalidad Mínima. Luego se capturan todas las tuplas de las tablas que componen la base de datos para crear las instancias de clases en la ontología. En este paso se tienen en cuenta las propiedades de datos de cada clase y los tipos de datos de estas propiedades, para conformar las datapropertyassertion. Luego de generar la ontología con los elementos del esquema relacional, se aplican un conjunto de consultas SPARQL, para clasificar la información contenida en la base de datos

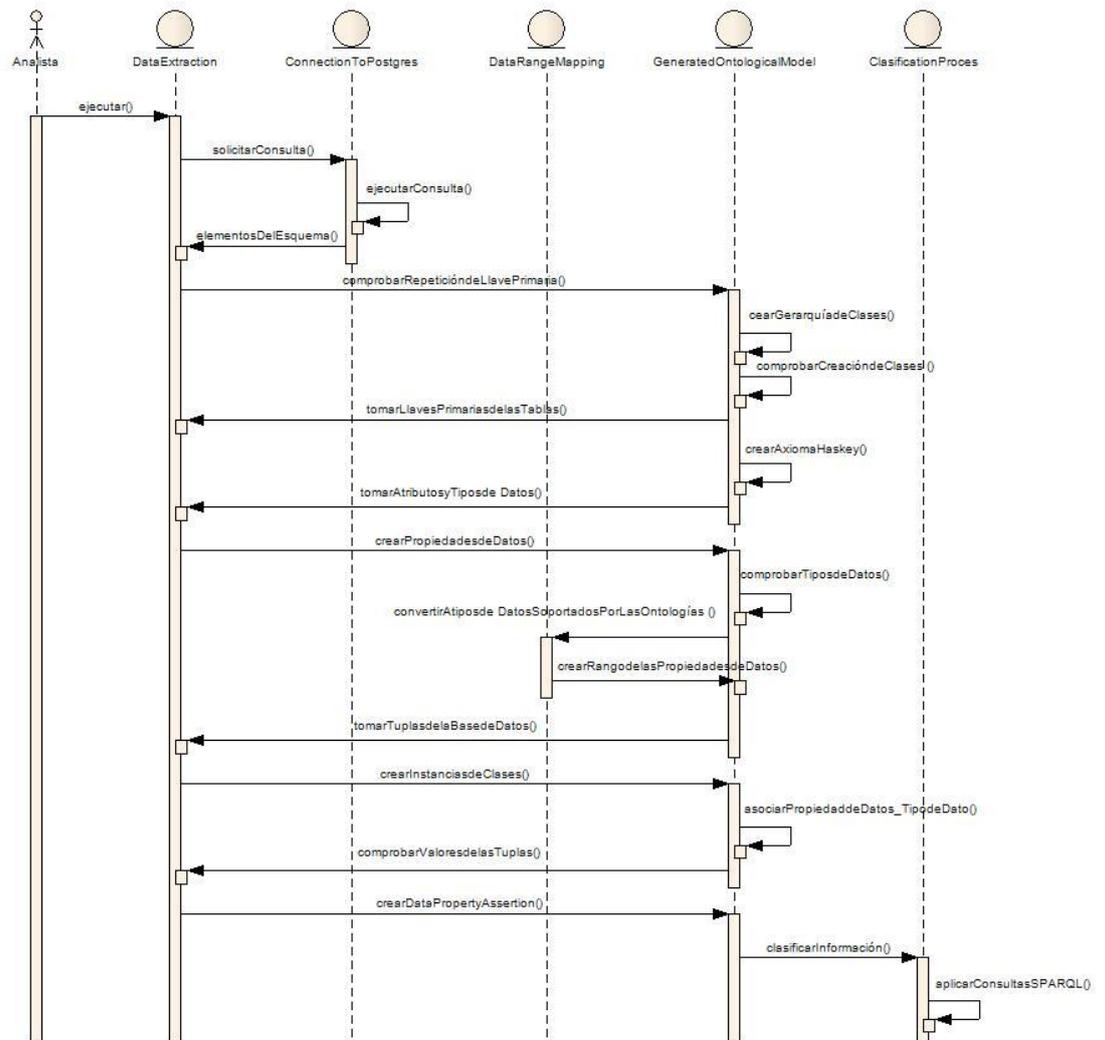


Figura 3.1: Diagrama de secuencia del Modelo ontológico propuesto. Fuente: elaboración propia.

3.2 Anexo 2. Tipos de datos soportados en bases de datos y sus equivalentes en ontologías

Tabla 3.1: Tipos de datos soportados en BD mapeados a tipos de datos soportados por las ontologías. Fuente: elaboración propia.

Tipos de datos en la BD	Tipos de datos en la ontología
character	xsd:string
character varying	xsd:string
money	xsd:double
double precision	xsd:double
time with time zone	xsd:data-time-tamp
serial	xsd;float
interval	xsd;long
year	xsd:date time
numeric	xsd;decimal
line	xsd;long
int2vector	xsd:int
double precision	xsd;double
real	xsd;float
bit	xsd;plaint literal

3.3 Anexo 3. Resultados del algoritmo de migración para cada caso experimental

- **Caso experimental 2.** Base de datos *Air Reservation*

Tabla 3.2: Elementos que componen la ontología resultante de aplicar el algoritmo de migración a la BD *Air Reservation*. Fuente: elaboración propia.

Caso experimental	Conceptos	Atributos	Relaciones	Axioma <i>hasKey</i>
2	10	39	8	10

Cálculo de los indicadores de eficacia:

- Precisión de correspondencias de conceptos:

$$P = \frac{nCCD}{nCD} = \frac{10}{10} = 1$$

- Exhaustividad de correspondencias de conceptos:

$$R = \frac{nCCD}{nCC} = \frac{10}{10} = 1$$

- Medida F (medida E con b=1):

$$F = \frac{1 + b^2}{\frac{b^2}{R} + \frac{1}{P}} = \frac{1 + 1^2}{\frac{1^2}{1} + \frac{1}{1}} = 1$$

- Precisión de correspondencias de atributos:

$$P = \frac{nCCD}{nCD} = \frac{39}{39} = 1$$

- Exhaustividad de correspondencias de atributos:

$$R = \frac{nCCD}{nCC} = \frac{39}{39} = 1$$

- Medida F (medida E con b=1):

$$F = \frac{1 + b^2}{\frac{b^2}{R} + \frac{1}{P}} = \frac{1 + 1^2}{\frac{1^2}{1} + \frac{1}{1}} = 1$$

- Precisión de correspondencias de conceptos:

$$P = \frac{nCCD}{nCD} = \frac{8}{8} = 1$$

- Exhaustividad de correspondencias de conceptos:

$$R = \frac{nCCD}{nCC} = \frac{8}{8} = 1$$

- Medida F (medida E con b=1):

$$F = \frac{1 + b^2}{\frac{b^2}{R} + \frac{1}{P}} = \frac{1 + 1^2}{\frac{1^2}{1} + \frac{1}{1}} = 1$$

- Eficacia del caso experimental

$$Ef(Caso) = \frac{C}{C + A + R} * F_{concep.} + \frac{A}{C + A + R} * F_{atrib.} + \frac{R}{C + A + R} * F_{relac.}$$

$$Ef(Experimento2) = \frac{10}{10 + 39 + 8} * 1 + \frac{38}{10 + 39 + 8} * 1 + \frac{8}{10 + 39 + 8} * 1 = 0.99$$

Tabla 3.3: Elementos que componen la ontología resultante de aplicar el algoritmo de migración a la BD BMEWS. Fuente: elaboración propia.

Caso experimental	Conceptos	Atributos	Relaciones	Axioma <i>hasKey</i>
3	7	9	2	7

- **Caso experimental 3.** Base de datos BMEWS

Cálculo de los indicadores de eficacia:

- Precisión de correspondencias de conceptos:

$$P = \frac{nCCD}{nCD} = \frac{7}{7} = 1$$

- Exhaustividad de correspondencias de conceptos:

$$R = \frac{nCCD}{nCC} = \frac{7}{7} = 1$$

- Medida F (medida E con b=1):

$$F = \frac{1 + b^2}{\frac{b^2}{R} + \frac{1}{P}} = \frac{1 + 1^2}{\frac{1^2}{1} + \frac{1}{1}} = 1$$

- Precisión de correspondencias de atributos:

$$P = \frac{nCCD}{nCD} = \frac{9}{9} = 1$$

- Exhaustividad de correspondencias de atributos:

$$R = \frac{nCCD}{nCC} = \frac{9}{9} = 1$$

- Medida F (medida E con b=1):

$$F = \frac{1 + b^2}{\frac{b^2}{R} + \frac{1}{P}} = \frac{1 + 1^2}{\frac{1^2}{1} + \frac{1}{1}} = 1$$

- Precisión de correspondencias de relaciones:

$$P = \frac{nCCD}{nCD} = \frac{2}{2} = 1$$

- Exhaustividad de correspondencias de relaciones:

$$R = \frac{nCCD}{nCC} = \frac{2}{2} = 1$$

- Medida F (medida E con b=1):

$$F = \frac{1 + b^2}{\frac{b^2}{R} + \frac{1}{P}} = \frac{1 + 1^2}{\frac{1^2}{1} + \frac{1}{1}} = 1$$

- Eficacia del caso experimental

$$Ef(Caso) = \frac{C}{C + A + R} * F_{concep.} + \frac{A}{C + A + R} * F_{atrib.} + \frac{R}{C + A + R} * F_{relac.}$$

$$Ef(Experimento3) = \frac{7}{7 + 9 + 2} * 1 + \frac{9}{7 + 9 + 2} * 1 + \frac{2}{7 + 9 + 2} * 1 = 0.99$$

• **Caso experimental 4.** Base de datos *Children Institution*

Tabla 3.4: Elementos que componen la ontología resultante de aplicar el algoritmo de migración a la BD *Children Institution*. Fuente: elaboración propia.

Caso experimental	Conceptos	Atributos	Relaciones	Axioma <i>hasKey</i>
4	9	10	2	9

Cálculo de los indicadores de eficacia:

- Precisión de correspondencias de conceptos:

$$P = \frac{n_{CCD}}{n_{CD}} = \frac{9}{9} = 1$$

- Exhaustividad de correspondencias de conceptos:

$$R = \frac{n_{CCD}}{n_{CC}} = \frac{9}{9} = 1$$

- Medida F (medida E con b=1):

$$F = \frac{1 + b^2}{\frac{b^2}{R} + \frac{1}{P}} = \frac{1 + 1^2}{\frac{1^2}{1} + \frac{1}{1}} = 1$$

- Precisión de correspondencias de atributos:

$$P = \frac{n_{CCD}}{n_{CD}} = \frac{10}{10} = 1$$

- Exhaustividad de correspondencias de atributos:

$$R = \frac{n_{CCD}}{n_{CC}} = \frac{10}{10} = 1$$

- Medida F (medida E con b=1):

$$F = \frac{1 + b^2}{\frac{b^2}{R} + \frac{1}{P}} = \frac{1 + 1^2}{\frac{1^2}{1} + \frac{1}{1}} = 1$$

- Precisión de correspondencias de conceptos:

$$P = \frac{nCCD}{nCD} = \frac{2}{2} = 1$$

- Exhaustividad de correspondencias de conceptos:

$$R = \frac{nCCD}{nCC} = \frac{2}{2} = 1$$

- Medida F (medida E con b=1):

$$F = \frac{1 + b^2}{\frac{b^2}{R} + \frac{1}{P}} = \frac{1 + 1^2}{\frac{1^2}{1} + \frac{1}{1}} = 1$$

- Eficacia del caso experimental

$$Ef(Caso) = \frac{C}{C + A + R} * F_{concep.} + \frac{A}{C + A + R} * F_{atrib.} + \frac{R}{C + A + R} * F_{relac.}$$

$$Ef(Experimento4) = \frac{9}{9 + 10 + 2} * 1 + \frac{10}{9 + 10 + 2} * 1 + \frac{2}{9 + 10 + 2} * 1 = 0.99$$

• **Caso experimental 6.** Base de datos *Gaming Web Sites*

Tabla 3.5: Elementos que componen la ontología resultante de aplicar el algoritmo de migración a la BD *Gaming Web Sites*. Fuente: elaboración propia.

Caso experimental	Conceptos	Atributos	Relaciones	Axioma <i>hasKey</i>
6	10	32	8	10

Cálculo de los indicadores de eficacia:

- Precisión de correspondencias de conceptos:

$$P = \frac{nCCD}{nCD} = \frac{10}{10} = 1$$

- Exhaustividad de correspondencias de conceptos:

$$R = \frac{nCCD}{nCC} = \frac{10}{10} = 1$$

- Medida F (medida E con b=1):

$$F = \frac{1 + b^2}{\frac{b^2}{R} + \frac{1}{P}} = \frac{1 + 1^2}{\frac{1^2}{1} + \frac{1}{1}} = 1$$

- Precisión de correspondencias de atributos:

$$P = \frac{nCCD}{nCD} = \frac{32}{32} = 1$$

- Exhaustividad de correspondencias de atributos:

$$R = \frac{nCCD}{nCC} = \frac{32}{32} = 1$$

- Medida F (medida E con b=1):

$$F = \frac{1 + b^2}{\frac{b^2}{R} + \frac{1}{P}} = \frac{1 + 1^2}{\frac{1^2}{1} + \frac{1}{1}} = 1$$

- Precisión de correspondencias de conceptos:

$$P = \frac{nCCD}{nCD} = \frac{8}{11} = 0,72$$

- Exhaustividad de correspondencias de conceptos:

$$R = \frac{nCCD}{nCC} = \frac{8}{11} = 0,72$$

- Medida F (medida E con b=1):

$$F = \frac{1 + b^2}{\frac{b^2}{R} + \frac{1}{P}} = \frac{1 + 1^2}{\frac{1^2}{1} + \frac{1}{1}} = 0.72$$

- Eficacia del caso experimental

$$Ef(Caso) = \frac{C}{C + A + R} * F_{concep.} + \frac{A}{C + A + R} * F_{atrib.} + \frac{R}{C + A + R} * F_{relac.}$$

$$Ef(Experimento6) = \frac{10}{10 + 32 + 8} * 1 + \frac{32}{10 + 32 + 8} * 1 + \frac{8}{10 + 32 + 8} * 0.72 = 0.97$$

- **Caso experimental 9.** Base de datos *Open University*

Tabla 3.6: Elementos que componen la ontología resultante de aplicar el algoritmo de migración a la BD *Open University*. Fuente: elaboración propia.

Caso experimental	Conceptos	Atributos	Relaciones	Axioma <i>hasKey</i>
9	7	25	6	7

Cálculo de los indicadores de eficacia - Precisión de correspondencias de conceptos:

$$P = \frac{nCCD}{nCD} = \frac{7}{7} = 1$$

- Exhaustividad de correspondencias de conceptos:

$$R = \frac{nCCD}{nCC} = \frac{7}{7} = 1$$

- Medida F (medida E con b=1):

$$F = \frac{1 + b^2}{\frac{b^2}{R} + \frac{1}{P}} = \frac{1 + 1^2}{\frac{1^2}{1} + \frac{1}{1}} = 1$$

- Precisión de correspondencias de atributos:

$$P = \frac{nCCD}{nCD} = \frac{25}{25} = 1$$

- Exhaustividad de correspondencias de atributos:

$$R = \frac{nCCD}{nCC} = \frac{25}{25} = 1$$

- Medida F (medida E con b=1):

$$F = \frac{1 + b^2}{\frac{b^2}{R} + \frac{1}{P}} = \frac{1 + 1^2}{\frac{1^2}{1} + \frac{1}{1}} = 1$$

- Precisión de correspondencias de relaciones:

$$P = \frac{nCCD}{nCD} = \frac{6}{6} = 1$$

- Exhaustividad de correspondencias de relaciones:

$$R = \frac{nCCD}{nCC} = \frac{6}{6} = 1$$

- Medida F (medida E con b=1):

$$F = \frac{1 + b^2}{\frac{b^2}{R} + \frac{1}{P}} = \frac{1 + 1^2}{\frac{1^2}{1} + \frac{1}{1}} = 1$$

- Eficacia del caso experimental

$$Ef(Caso) = \frac{C}{C + A + R} * F_{concep.} + \frac{A}{C + A + R} * F_{atrib.} + \frac{R}{C + A + R} * F_{relac.}$$

$$Ef(Experimento9) = \frac{7}{7 + 25 + 6} * 1 + \frac{25}{7 + 25 + 6} * 1 + \frac{6}{7 + 25 + 6} * 1 = 0.99$$

• **Caso experimental 10.** Base de datos *Theater Booking*

Tabla 3.7: Elementos que componen la ontología resultante de aplicar el algoritmo de migración a la BD *Theater Booking*. Fuente: elaboración propia.

Caso experimental	Conceptos	Atributos	Relaciones	Axioma <i>hasKey</i>
10	8	29	6	8

Cálculo de los indicadores de eficacia - Precisión de correspondencias de conceptos:

$$P = \frac{n_{CCD}}{n_{CD}} = \frac{8}{8} = 1$$

- Exhaustividad de correspondencias de conceptos:

$$R = \frac{n_{CCD}}{n_{CC}} = \frac{8}{8} = 1$$

- Medida F (medida E con b=1):

$$F = \frac{1 + b^2}{\frac{b^2}{R} + \frac{1}{P}} = \frac{1 + 1^2}{\frac{1^2}{1} + \frac{1}{1}} = 1$$

- Precisión de correspondencias de atributos:

$$P = \frac{n_{CCD}}{n_{CD}} = \frac{29}{29} = 1$$

- Exhaustividad de correspondencias de atributos:

$$R = \frac{n_{CCD}}{n_{CC}} = \frac{29}{29} = 1$$

- Medida F (medida E con b=1):

$$F = \frac{1 + b^2}{\frac{b^2}{R} + \frac{1}{P}} = \frac{1 + 1^2}{\frac{1^2}{1} + \frac{1}{1}} = 1$$

- Precisión de correspondencias de conceptos:

$$P = \frac{nCCD}{nCD} = \frac{6}{6} = 1$$

- Exhaustividad de correspondencias de conceptos:

$$R = \frac{nCCD}{nCC} = \frac{6}{6} = 1$$

- Medida F (medida E con b=1):

$$F = \frac{1 + b^2}{\frac{b^2}{R} + \frac{1}{P}} = \frac{1 + 1^2}{\frac{1^2}{1} + \frac{1}{1}} = 1$$

- Eficacia del caso experimental

$$Ef(Caso) = \frac{C}{C + A + R} * F_{concep.} + \frac{A}{C + A + R} * F_{atrib.} + \frac{R}{C + A + R} * F_{relac.}$$

$$Ef(Experimento10) = \frac{8}{8 + 29 + 6} * 1 + \frac{29}{8 + 29 + 6} * 1 + \frac{6}{8 + 29 + 6} * 1 = 0.99$$

3.4 Anexo 4. Clasificación de la información para cada caso experimental

Tabla 3.8: Muestra de la clasificación de la información contenida en la base de datos utilizada en el caso experimental 2 *Air Reservation*. Fuente: elaboración propia.

Conceptos	Relaciones	Atributos	Roles
intinerary legs	reservation payments	day number	
payments	legs	payment date	
legs	intinerary reservation	reservationid	
passengers	intinerary legs	last name	
reservation payments	flight schedules	city	
ref calendar		payment amount	
bookings agents		agentid	
air ports		passegersid	
intinerary reservation		agent details	
flight schedules		airport code	
		airport location	
		country	
		paymentid	
		phone number	
		legsid	
		airport name	

Tabla 3.9: Muestra de la clasificación de la información contenida en la base dato utilizada en el caso experimental 3 BMEWS. Fuente: elaboración propia.

Conceptos	Relaciones	Atributos	Roles
report types	report data items values	data martid	
report definitions	datamart data values	report data item	
report data items values		datatype code	
data types		reportid	
datamart definitions		data itemid	
datamart data values			

Tabla 3.10: Muestra de la clasificación de la información contenida en la base dato utilizada en el caso experimental 4 *Children Institution*. Fuente: elaboración propia.

Conceptos	Relaciones	Atributos	Roles
children relatives	children relatives	childid	
institutions	children in institution	rolesid	
special needs		offenceid	
children in institution		institutionid	
status		status code	
contacts		contactsid	
relatives		relativeid	
juvenil offences		needsid	
roles			

Tabla 3.11: Muestra de la clasificación de la información contenida en la base dato utilizada en el caso experimental 5 *Children of Family Support*. Fuente: elaboración propia.

Conceptos	Relaciones	Atributos	Roles
event	professional at event	individualid	
children	individual at event	eventid	
family members	family members	familyid	
family	event	cuortsid	
professionals		professionalid	
cuorts		casesid	
individual at event		childid	
cases			
individuals			
professional at event			

Tabla 3.12: Muestra de la clasificación de la información contenida en la base dato utilizada en el caso experimental 6 *Gaming Web Sites*. Fuente: elaboración propia.

Conceptos	Relaciones	Atributos	Roles
members	continents	middle name	alliances members
cities	members	country details	members resource
continents	alliances members	continent details	
alliances members	resources	worldid	
resources	countries	memberid	
members resource	members resource	alliance name	
resources types		resourceid	
world		date alliance formed	
alliances		city name	
		resources type code	
		country name	
		world details	
		resource name	

Tabla 3.13: Muestra de la clasificación de la información contenida en la base dato utilizada en el caso experimental 7 ISWC. Fuente: elaboración propia.

Conceptos	Relaciones	Atributos	Roles
relation1	relation4	type	
relation4	relation3	confid	
organizations	relation2	date	
conferences	relation1	name	
topics	papers	phone	
relation2		address	
relation3		title	
papers		location	
persons		uri	
		photo	
		postcode	
		email	
		home page	

Tabla 3.14: Muestra de la clasificación de la información contenida en la base dato utilizada en el caso experimental 8 *National Health*. Fuente: elaboración propia.

Conceptos	Relaciones	Atributos	Roles
staff	patients medications	medicationid	
ref medications types	medications	first name	
patients medications	appointments	medications types code	
patients		staffid	
medications		other details	
appointments		middle name	
		medication type name	
		medication description	
		commets	
		patientid	
		date time of appointment	
		dosage	
		medication unit cost	

Tabla 3.15: Muestra de la clasificación de la información contenida en la base dato utilizada en el caso experimental 9 *Open University*. Fuente: elaboración propia.

Conceptos	Relaciones	Atributos	Roles
curses scheduled	staff curse supervision	first name	
staff curse supervision	student curse registrations	phone number	
staff	staff research interest	staffid	
staff in research project	curses scheduled	middle name	
curses offered	staff in research project	date of birth	
student	research project	other details	
areas of research	student curse registrations	curses ofereringid	
staff research interest		project description	
student curse registrations		studentid	
research project		gender	
		project name	
		job title	
		curses oferering name	
		areas ofresearchid	