

**TRANSFORMACIÓN DE ESQUEMAS DE BASES DE DATOS RELACIONALES
AL LENGUAJE DE ONTOLOGÍAS WEB**



ANEXOS

**CARLOS ALBEIRO BRAVO LLANOS
CARLOS ALBERTO SUAREZ MUÑOZ**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
PROGRAMA DE INGENIERÍA DE SISTEMAS
POPAYÁN
2013**

**TRANSFORMACIÓN DE ESQUEMAS DE BASES DE DATOS RELACIONALES
AL LENGUAJE DE ONTOLOGÍAS WEB**

**CARLOS ALBEIRO BRAVO LLANOS
CARLOS ALBERTO SUAREZ MUÑOZ**

Directora
PhD. CAROLINA GONZALEZ SERRANO

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
PROGRAMA DE INGENIERÍA DE SISTEMAS
POPAYÁN
2013**

CONTENIDO

ANEXO 1. REGLAS DE TRANSFORMACIÓN	6
ANEXO 2. MANUAL DE USUARIO	17
ANEXO 3. MANUAL TÉCNICO.....	21
ANEXO 4. SCRIPTS DE LOS RDBS	25
ANEXO 5. RESULTADOS DE LAS TRANSFORMACIONES	35
ANEXO 6. RESULTADOS OWL	41
ANEXO 7. HISTORIAS DE USUARIO	76
BIBLIOGRAFÍA.....	78

LISTA DE TABLAS

Tabla 1. Resultado transformación I – RDBS Observación de aves	35
Tabla 2. Resultado transformación II – RDBS Proyectos de investigación.....	35
Tabla 3. Resultado transformación III – RDBS Alojamientos rurales.....	36
Tabla 4. Resultados transformación IV – RDBS Gestión de proyectos informáticos	37
Tabla 5. Resultados transformación V – RDBS Observación de aves	38
Tabla 6. Resultados transformación VI – RDBS Proyectos de investigación	39
Tabla 7. Resultados transformación VII – RDBS Alojamientos rurales.....	39
Tabla 8. Resultados transformación VIII – RDBS Gestión de proyectos informáticos	40

LISTA DE FIGURAS

Figura 1. Formulario para transformación de RDBS a OWL2	18
Figura 2. Ejemplo - Datos para transformación.....	18
Figura 3. Información adicional para RDBS con herencia.....	19
Figura 4. Resultado de la transformación	19
Figura 5. Documentación del prototipo	20
Figura 6. Información de los autores.....	20
Figura 7. Arquitectura del sistema	21
Figura 8. Diagrama de despliegue del sistema	21
Figura 9. Diagrama de paquetes y clases del servicio	22
Figura 10. Diagrama de componentes del servicio	23
Figura 11. Diagrama de paquetes y clases del cliente	24
Figura 12. Diagrama de componentes del cliente	24
Figura 13. Historia de usuario - Mostrar relaciones de herencia	76
Figura 14. Historia de usuario - Transformar RDBS a OWL2.....	77

ANEXO 1. REGLAS DE TRANSFORMACIÓN

En esta sección se describen los conjuntos de reglas de transformación usados en los estudios, que son utilizados en el desarrollo del presente proyecto, se presenta cada título del estudio con su respectivo conjunto de reglas.

A hybrid strategy to construct scientific instrument ontology from relational database model (Chen, Liu, & Shang, 2012).

Se consideran cinco conjuntos de reglas que son

1. **Reglas para tablas:** Para las tablas se diferencian tres tipos de relaciones.
 - a. Relaciones básicas: Son relaciones que no tienen llaves foráneas. Estas son mapeadas a *owl:class*.
 - b. Relaciones dependientes: la relación A depende de la relación B si la relación A solo tiene llaves foráneas de B, las llaves foráneas de A son la llave primaria de B y la llave primaria de A está compuesta totalmente por sus llaves foráneas. Estas relaciones transformadas de dos maneras:
 - i. Si el conjunto de las llaves primarias de A es igual al conjunto de las llaves primarias de B, la relación B es transformada a *owl:class* (C_B), y las llaves no primarias de A son transformadas en atributos de C_B . Pero A no necesita ser transformada en *owl:class*.
 - ii. Si el conjunto de las llaves primarias de A es subconjunto del conjunto de las llaves primarias de B, A es transformada en *owl:class* (C_A), B es transformada en *owl:class* (C_B) y C_A es definida como subclase de C_B .
 - c. Relaciones compuestas: Son relaciones que no son ni relaciones básicas ni relaciones dependientes. Si todos los campos de la relación compuesta A son llaves foráneas, cada campo de A es transformado solamente a *owl:ObjectProperty*. De lo contrario la relación A es transformada a *owl:class*.
2. **Reglas para columnas:** Las columnas son mapeadas en atributos OWL:
 - a. *Owl:ObjectProperty*: indican relaciones entre instancias de clases como las llaves foráneas.
 - b. *Owl:DataTypeProperty*: indican relaciones entre instancias de clases y literales RDF y tipos de datos del Esquema XML. Debe tener las restricciones *rdf:domain* y *rdf:range*.

Cada atributo debe tener las restricciones de cardinalidad *owl:minCardinality* y *owl:maxCardinality* fijadas en el valor de uno (1).
3. **Reglas para tipos de datos:** Cada columna transformada en *owl:DataTypeProperty* debe estar acompañada por el tipo de dato. OWL usa la mayoría de tipos de datos del Esquema XML.

4. Reglas para restricciones:

- a. Restricciones de integridad de entidades (*Primary key*)
 - i. Una sola columna: la llave primaria es transformada en una *propiedad funcional inversa* y con una *cardinalidad mínima* de 1.
 - ii. Múltiples columnas: cada llave primaria debe ser transformada en dos *propiedades de objeto inversas*.
- b. Restricciones de integridad referencial (*foreign key*):
 - i. Si las llaves foráneas de A son sus mismas llaves primarias, A es transformada en una subclase de B.
 - ii. Si las llaves foráneas de A son diferentes o son un subconjunto de sus llaves primarias, las llaves foráneas de A son transformadas en una *propiedad de objeto*.
 - iii. Si las llaves foráneas de A son llaves foráneas a la misma tabla, las llaves foráneas de A son transformadas a una *propiedad simétrica*, y usa la *owl:class* (C_A) como dominio y rango.
 - iv. Si las llaves foráneas de A son llaves foráneas a la misma tabla, y las llaves foráneas de A están acompañadas por un *trigger on delete cascade*, las llaves foráneas de A son transformadas en una *propiedad transitiva*.
- c. Restricciones *UNIQUE*: son transformadas en una *owl:InverseFunctionalProperty*.
- d. Restricciones *NOT NULL*: son transformadas a *owl:minCardinality* de con el valor de 1.
- e. Restricciones *CHECK*: son transformadas a una *restricción de valor*. Si tiene una enumeración es transformada a un *tipo de dato enumerado*.

5. Reglas para filas: cada fila es transformada en una instancia de *owl:Class*.

Generating OWL Ontology from Relational Database (Choi & Kim, 2012).

Las reglas de transformación mostradas en este estudio, se detallan en una breve descripción de sus condiciones con su respectiva expresión.

1. **Regla 1:** Una tabla A es transformada en una *clase* (C_A).

Declaration(Class(:cls_{Ti}))

2. **Regla 2:** Si una columna de la tabla A es llave, la columna crea una *clase*.

Declaration(Class(:cls_{Ti.Ck}))

3. **Regla 3:** Si una columna de la tabla A es llave, la columna crea una *propiedad de objeto*.

Declaration(ObjectProperty(:op_{Ti.Ck}))

4. **Regla 4:** Toda columna crea una *propiedad de dato*.

Declaration(DataProperty(:dp_{Ti.Ck}))

5. **Regla 5:** La *propiedad de objeto* creada por una columna de la clase C_A define a la clase C_A como su *dominio*.

ObjectPropertyDomain(:op_{Ti.Ck}:cls_{Ti})

6. **Regla 6:** La *propiedad de objeto* creada por una columna de la clase C_A define a la clase C_A como su *rango*.

ObjectPropertyRange(:op_{Ti.Ck}:cls_{Ti.Ck})

7. **Regla 7:** Si una columna de la tabla A es llave, la *propiedad de dato* generada por esa columna define a la clase C_A como su *rango*.

DataPropertyDomain(:dp_{Ti.Ck}:cls_{Ti.Ck})

8. **Regla 8:** Si una columna de la tabla A no es llave, la *propiedad de dato* generada por esa columna define a la clase C_A como su *dominio*.

DataPropertyDomain(:dp_{Ti.Ck}:cls_{Ti})

9. **Regla 9:** Si una columna de la tabla A no tiene restricción *check* ni *enum*, la *propiedad de dato* generada por esa columna define el *tipo de dato*, correspondiente a esa columna, como su *rango*.

DataPropertyRange(:dp_{Ti.Ck} dt_{Ti.Ck})

10. **Regla 10:** Si una columna tiene *enumeración*, la *propiedad de dato* generada por esa columna, define la *enumeración de literales*, correspondiente a la *restricción check con enumeración* de esa columna, como su *rango*.

DataPropertyRange(:dp_{Ti.Ck} DataOneOf(lt₁ ... lt_n))

11. **Regla 11:** Si una columna tiene *restricción check*, la *propiedad de dato* generada por esa columna define la restricción de *tipo de dato*, correspondiente a la restricción *check* de esa columna, como su *rango*.

DataPropertyRange(:dp_{Ti.Ck} DatatypeRestriction(f₁ lt₁ ... f_n lt_n))

12. **Regla 12:** si una clase derivada de una tabla define propiedades de objeto, derivadas de cada columna que es llave primaria, como sus llaves, se definen como *HasKey* que recibe la clase y sus *propiedades de objeto* correspondientes.

HasKey(:cls_{Ti} (op_{Ti.C1} ... op_{Ti.Cm}) ())

13. **Regla 13:** Si una columna es llave, la *clase* derivada de esa columna define la *propiedad de dato*, derivada de la misma columna, como su *llave*.

HasKey(:cls_{Ti.Ck}) (:dp_{Ti.Ck}))

14. **Regla 14:** Si una columna tiene una restricción *unique*, la *propiedad de objeto* derivada de esa columna es definida como una *propiedad funcional inversa*.

InverseFunctionalObjectProperty(:op_{Ti.Ck})

15. **Regla 15:** Si una columna T es llave foránea, la *clase* derivada de T es definida como *subclase* de la *clase* derivada de la columna referenciada por T.

SubClassOf(:cls_{Ti.Ck} :cls_{ref(Ti.Ck)})

16. **Regla 16:** Si una columna es llave, la *clase* derivada de esa columna es definida como una *subclase* de expresión de *clase universal DataAllValuesFrom*.

SubClassOf(:cls_{Ti.Ck} DataAllValuesFrom(:dp_{Ti.Ck} dr_{Ti.Ck}))

17. **Regla 17:** Si una columna es llave, la *clase* derivada de esa columna es definida como una *subclase* de la expresión de *clase cardinalidad exact DataExactCardinality*.

SubClassOf(:cls_{Ti.Ck} DataExactCardinality(:dp_{Ti.Ck} dr_{Ti.Ck}))

18. **Regla 18:** Si una tabla tiene padre, la *clase* derivada de dicha tabla es definida como una *subclase* de la *clase* derivada de la tabla padre.

SubClassOf(:cls_{Ti} :cls_{parent(Ti)})

19. **Regla 19:** Si una columna es llave, la *clase* derivada de la tabla a la cual pertenece esa columna es definida como una *subclase* de la expresión de *clase universal ObjectAllValuesFrom*.

SubClassOf(:cls_{Ti} ObjectAllValuesFrom(:op_{Ti.Ck} :cls_{Ti.Ck}))

20. **Regla 20:** Si una columna es llave y no nula, la *clase* derivada de la tabla a la cual pertenece es definida como una *subclase* de la expresión de *clase cardinalidad exacta ObjectExactCardinality*.

SubClassOf(:cls_{Ti} ObjectExactCardinality(1 :op_{Ti.Ck} :cls_{Ti.Ck}))

21. **Regla 21:** Si una columna es llave y nula, la *clase* derivada de la tabla a la cual pertenece es definida como una *subclase* de la expresión de *clase cardinalidad máxima ObjectMaxCardinality*.

SubClassOf(:cls_{Ti} ObjectMaxCardinality(1 :op_{Ti.Ck} :cls_{Ti.Ck}))

22. **Regla 22:** Si una columna no es llave, la *clase* derivada de la tabla a la cual pertenece es definida como una *subclase* de la expresión de *clase universal DataAllValuesFrom*.

SubClassOf(:cls_{Ti} DataAllValuesFrom(:dp_{Ti.Ck} dr_{Ti.Ck}))

23. **Regla 23:** Si una columna no es llave primaria y es no nula, la *clase* derivada de la tabla a la cual pertenece es definida como una *subclase* de la expresión de *clase cardinalidad exacta* *DataExactCardinality*.

SubClassOf(:cls_{Ti} DataExactCardinality(1 :dp_{Ti.Ck} dr_{Ti.Ck}))

24. **Regla 24:** Si una columna no es llave y es nula, la *clase* derivada de la tabla a la cual pertenece es definida como *subclase* de la expresión de *clase cardinalidad máxima* *DataMaxCardinality*.

SubClassOf(:cls_{Ti} DataMaxCardinality(1 :dp_{Ti.Ck} dr_{Ti.Ck}))

25. **Regla 25:** Si la tabla T tiene padre y una columna de T es llave primaria, la *propiedad de objeto* derivada de esa columna es definida como una *subproperty* de la *propiedad de objeto* derivada de la columna referenciada por la llave primaria.

SubObjectPropertyOf(:op_{Ti.Ck} :op_{ref(Ti.Ck)})

26. **Regla 26:** Si una columna es llave foránea, la *propiedad de objeto* derivada de esa columna es definida como una *subproperty* de la *propiedad de dato* derivada de la columna referenciada por la llave foránea.

SubDataPropertyOf(:dp_{Ti.Ck} :dp_{ref(Ti.Ck)})

27. **Regla 27:** Si la tabla T no tiene padre y una columna que no es llave foránea es llave primaria o tiene restricción *unique*, todas las *clases* derivadas de la tabla T o dicha columna son definidas para ser *pairwise disjoint*.

DisjointClasses(:cls₁ ... :cls_n)

Rules for Mapping SQL Relational Databases to OWL Ontologies (Astrova, 2009).

Se consideran cuatro conjuntos de reglas que son:

1. **Regla para tablas:** Se identifican dos casos:

- a. Una tabla es mapeada a una *clase* a menos que todas sus columnas sean llaves foráneas.
- b. Si todas las columnas de una tabla son llaves foráneas
 - i. Si la tabla tiene solo dos columnas que son llaves foráneas que a su vez son llaves primarias de la tabla (lo cual indica una relación binaria de muchos a muchos), esta es mapeada a dos *propiedades de objeto* (una como inversa de la otra)

- ii. Si la tabla tiene tres llaves foráneas que a su vez son llaves primarias de la tabla (lo cual indica una relación ternaria), esta es mapeada a una clase.

2. Reglas para columnas: Una columna es mapeada a una propiedad tipo dato con una máxima cardinalidad de uno (1) a menos que esta sea (parte de) una llave foránea.

3. Reglas para restricciones:

- a. Regla para restricciones *unique*: Esta restricción de una columna en una tabla es mapeada a una propiedad funcional inversa.
- b. Regla para restricciones *not null*: Esta restricción de una columna en una tabla es mapeada a un mínimo de cardinalidad de uno.
- c. Regla para restricciones *primary key*: Esta restricción es mapeada como una propiedad funcional inversa con un mínimo de cardinalidad de 1.
- d. Regla para restricciones de referencia y *foreign key*: Una restricción de referencia es una restricción de columna mientras que una *foreign key* es una restricción de tabla. Ambas son usadas para especificar llaves foráneas.
 - i. Si una columna es una llave foránea en una tabla A indicando una relación binaria (uno-a-cero-o-uno, uno-a-uno o muchos-a-uno) con B entonces es mapeada a una *propiedad de objeto*, usando las clases A como su dominio y B como su rango, esta propiedad está restringida para todos los valores de B (*allValuesFrom*). Este mapeo se hace a menos que la llave foránea sea también una llave principal.
 - ii. Si una restricción de referencia esta sobre una columna de la tabla A indicando una relación binaria con la tabla B siendo esta llave foránea parte de la llave primaria entonces esta columna es mapeada a una *propiedad de objeto* con una cardinalidad de 1.
 - iii. Si una llave foránea en una tabla A que indica una relación binaria con la tabla B y a su vez es llave principal entonces se mapea a una clase herencia. Donde A es una subclase de B (*subClassOf*).
- e. Regla para restricciones *check*:
 - i. Una restricción *check* es mapeada a una restricción de valor (*hasValue*) si no tiene una enumeración.
 - ii. Una restricción *check* que especifica un rango de valores en un lista es mapeada a un *tipo de dato enumeración*, con un elemento para cada valor de la lista.

4. Regla para las filas: Una fila es mapeada a una instancia de la Ontología.

Towards Integration Rules of Mapping from Relational Databases to Semantic Web Ontology (Hazber, Yang, & Jin, 2010).

Se presentan cinco conjuntos de reglas que son:

1. **Reglas para tablas:** Cada tabla de una base de datos relacional debe ser mapeada a una clase de la ontología con el mismo nombre correspondiente a la tabla, y el comentario de la tabla ser transformado en un comentario de clase. Se presentan dos casos

- a. Cuando una tabla tiene una llave primaria que a su vez no es llave foránea entonces esta tabla se mapea a una *clase owl:Class*.
- b. Cuando una tabla tiene su llave primaria compuesta por dos campos que a su vez son llaves foráneas lo cual indica una relación binaria (m:n). Entonces estos dos campos serán mapeados a dos *owl:ObjectProperty* y se adiciona *inverseOf*, lo cual indica que la relación es bidireccional.

2. **Reglas para columnas:**

- a. Columna Simple: Una columna en un esquema relacional que no es ni llave primaria ni llave foránea, podría ser mapeada a una *propiedad tipo dato* con el mismo nombre de la columna y acompañado por un máximo de cardinalidad con el valor de 1 a menos que sea una llave foránea. El *rdfs:label* corresponde al nombre de la columna, y *rdfs:comment* corresponde a la descripción de la columna. El dominio es la clase creada por su tabla y el rango de la *propiedad tipo dato* es el Esquema XSD correspondiente al tipo de dato equivalente al tipo de dato original de la columna en la base de datos.
- b. Columna Compuesta: Hay dos maneras de mapear columnas compuestas:
 - i. Mapear solo sus columnas de componentes simples a *propiedades tipo dato* e ignorar las columnas compuestas.
 - ii. Mapear las columnas compuesta a *propiedades tipo dato* y luego mapear sus columnas de componentes simples a *sub propiedades* de las correspondientes *propiedades tipo dato*.

3. **Reglas para los tipos de datos:** Se usa el mapeo de tipos de datos SQL a XSD.

4. **Reglas para restricciones:**

- a. Regla para restricciones *not null*: Son mapeadas a un *minCardinality* de 1.

- b. Regla para restricciones *unique*: Son mapeadas a una *propiedad funcional inversa* en la ontología.
- c. Regla para restricciones *primary key*: Son mapeadas a la vez como *propiedad funcional inversa* y como *minCardinality* de 1.
- d. Reglas para restricciones *foreign key* y *reference*
 - i. Una columna C en una tabla A es una llave foránea a otra tabla B indicando una relación binaria (uno-a-cero-o-muchos, uno-a-uno o muchos-a-uno). Y considerando que C no es parte de la llave primaria de la tabla A. Entonces C será mapeada a una *propiedad de objeto* que usa la clase A y B como sus respectivos dominio y rango.
 - ii. Las llaves foráneas son mapeadas a *propiedades de objeto* la cual está restringida a todos los valores de la clase B (*allValuesFrom*).
 - iii. Las llaves foráneas son mapeadas a *propiedades de objeto* con un *minCardinality* de 1
 - iv. Una columna C en una tabla A es una llave foránea a otra tabla B indicando una relación binaria (uno-a-cero-o-muchos, uno-a-uno o muchos-a-uno). Y considerando que C es parte de la llave primaria de la tabla A. Entonces C será mapeada a una *propiedad de objeto* con una cardinalidad de 1.
 - v. Una columna C en la tabla A es una llave foránea a otra tabla B indicando una relación binaria (uno-a-uno). Y considerando que C es llave primaria de A entonces se mapea una clase herencia donde A es una subclase de B.
 - vi. Una columna C es una llave foránea de la su misma tabla indicando una relación recursiva. Entonces C es mapeada a una *propiedad simétrica* que usa su clase como rango y dominio.
 - vii. Una columna C es una llave foránea de la su misma tabla indicando una relación recursiva. Y considerando que C es acompañada por un *trigger on delete cascade*. (esta relación se compone de un todo y una parte, en donde la parte no puede existir sin el todo). Entonces C es mapeada a una *propiedad transitiva* que usa su clase como dominio y rango.

5. Reglas para restricciones *check*

- a. Restricción *check* en columna simple: Una columna C de una tabla A que tiene una restricción *check* que implica la verificación de un solo valor (igual a) es mapeada a una propiedad tipo dato restringida a tener el mismo valor para todas las instancias de la clase A.
- b. Restricción *check* en columna múltiple: Si una columna C y D de la tabla A que tienen una restricción *check* que implica la verificación de un valor (igual a) y la verificación de un valor (mayor que) respectivamente es mapeada a una propiedad tipo dato restringida a tener el mismo valor para todas las instancias de la clase A y un *minCardinality* con el valor que limita la condición (mayor que).
- c. Restricción *check between*: Es mapeada con un *minCardinality* y un *maxCardinality* con los respectivos valores para contener el intervalo, para casos como el intervalo de edad que pueden permitirse en una columna.
- d. Restricción *check* con enumeración: En una tabla se especifica un rango para la columna *Acc_currency* a través de la lista de valores 'USD', 'RMB' Entonces esta restricción es mapeada a un tipo de dato enumerado con un elemento para cada valor de la lista.
- e. Restricción *check like*: Una tabla tiene una restricción en el campo nacionalidad lo cual indica que solo se permiten valores que empiecen con la letra 'Y' esta restricción puede ser mapeada con *allValuesFrom* que use una clase *beginning_with_Y*.

6. **Reglas para las filas:** Se mapean los datos de las filas a instancias de la ontología.

Survey of directly mapping SQL databases to the Semantic Web (Sequeda, Tirmizi, Corcho, & Miranker, 2011).

Se presentan seis conjuntos de reglas que son:

1. Conjunto de reglas 1:

- a. Una relación R es una relación binaria entre las relaciones S y T si R es una relación que tiene llaves foráneas para S y T, y R no tiene otras llaves foráneas o atributos.

2. Conjunto de reglas 2:

- a. Una relación R es mapeada a un *owl:Class* si R no es una relación binaria.

3. Conjunto de reglas 3:

- a. Si R es una relación binaria entre S y T, y S y T no son relaciones binarias, R es transformado en un *objectProperty* con dominio S y rango T. Esto implica que S y T son clases owl. Además se crea la inversa de esa propiedad es decir, se crea un *objectProperty* con dominio T y rango S.

4. Conjunto de reglas 4:

- a. Si x es una llave foránea en la relación S que referencia la relación T, S y T no son relaciones binarias, x puede ser nulo y x no es único, x es transformado en un *objectProperty* con dominio S y rango T y en una *propiedad funcional*. Además se crea una *cardinalidad mínima* para la inversa del *objectProperty* creado con T igual a 0.
- b. Si x es una llave foránea en la relación S que referencia la relación T, S y T no son relaciones binarias, x es *not null* y x no es único, x es transformado en un *objectProperty* con dominio S y rango T, *cardinalidad máxima y mínima* para S igual a 1, y en una *propiedad funcional*. Además se crea una *cardinalidad mínima* para la inversa del *objectProperty* creado con T igual a 0.
- c. Si x es una llave foránea en la relación S que referencia la relación T, S y T no son relaciones binarias, x puede ser nulo y x es único, x es transformado en un *objectProperty* con dominio S y rango T y en una *propiedad funcional*. Además se crea una *propiedad funcional* para la inversa del *objectProperty* creado.
- d. Si x es una llave foránea en la relación S que referencia la relación T, S y T no son relaciones binarias, x es *not null*, x es único y x no es *llave primaria* en la relación S, x es transformado en un *objectProperty* con dominio S y rango T, en una *propiedad funcional y cardinalidad máxima y mínima* para S igual a 1. Además se crea una *propiedad funcional* para la inversa del *objectProperty* creado.

5. Conjunto de Reglas 5:

- a. Si x no es *llave foránea* en la relación R, x es transformado en una *propiedad de tipo de dato* con dominio R y rango igual al tipo de dato OWL apropiado para el campo x, y en una *propiedad funcional*.
- b. Si x no es *llave foránea* en la relación R y es *not null*, x es transformado en una *propiedad de tipo de dato* con dominio R y rango igual al tipo de dato OWL apropiado para el campo x, en una *propiedad funcional, y cardinalidad máxima y mínima* para R igual a 1.

- c. Si x no es *llave foránea* en la relación R y tiene una *restricción de lista enumerada*, x es transformado en una *propiedad de tipo de dato* con dominio R y rango igual al tipo de dato OWL apropiado para el campo x *unido* a la lista de valores permitidos para el campo x , y en una *propiedad funcional*.

6. Conjunto de reglas 6:

- a. Si R y S son relaciones, x es *llave primaria* en la relación R y x es *llave foránea* en la relación R que referencia a la relación S , R es transformada en una *subclase* de la clase S .

ANEXO 2. MANUAL DE USUARIO

Se construyó el prototipo de un servicio que permite realizar la transformación de los RDBS a OWL2, el cual se basa en las reglas de transformación propuestas por (Choi & Kim, 2012) e incluimos tres nuevas reglas que permiten representar las restricciones asociadas a los diferentes tipos de herencia como son: total, exclusiva, total y exclusiva, que se presentan a continuación:

Herencia total → **EquivalentClasses(:Parent ObjectUnionOf(:Child1, Child2, ... Childn))**

Herencia exclusiva → **DisjointClasses(:Child1, Child2, ... Childn)**

Herencia total y exclusiva → **DisjointUnion(:Parent :Child1, Child2, ... Childn)**

Donde Parent y Child son clases owl.

Requerimientos

- ✓ MySql versión 5.0.2 o posterior.
- ✓ La base de datos debe tener permisos de conexión remota para la ip: 54.224.36.189

Nota: Por motivos de seguridad se recomienda usar una cuenta de usuario temporal.

Instrucciones para el Usuario

Transformación del RDBS a OWL2

1. Visitar el cliente para rdfs2owl2 ubicado en la url: <http://rdfs2owl2client-tripplannertest.rhcloud.com> y seleccionar la pestaña HOME. Se mostrará el formulario presentado en la Figura 1.
2. Ingresar los datos solicitados en el formulario, presionar el botón enviar y esperar por la respuesta. Ver Figura 2.
3. Si el RDBS tiene relaciones de herencia aparecerá la sección "Ontology Information" y la tabla "Inheritances" como lo muestra la Figura 3.

Nota: Si el RDBS No tiene relaciones de herencia, la tabla "Inheritance" solo mostrara los encabezados.

4. Llenar la información de la ontología, ingresar nuevamente el password, seleccionar el tipo de herencia, presionar el botón enviar y esperar la respuesta que será presentada como lo muestra la Figura 4.

The screenshot shows a web application interface with a navigation bar at the top containing three buttons: "HOME", "DOCUMENTATION", and "ABOUT". The "DOCUMENTATION" button is highlighted in orange. Below the navigation bar, the title "RDBS2OWL2" is displayed in a large, bold font. Underneath the title, the text "Information Database Schema" is shown. The form consists of four input fields labeled "Server:", "Database:", "User:", and "Password:", each followed by a text input box. A "send" button is located at the bottom right of the form area.

Figura 1. Formulario para transformación de RDBS a OWL2

This screenshot shows the same web application interface as Figure 1, but with the form fields populated with example data. The "Server:" field contains "db4free.net:3306", the "Database:" field contains "proyectosinv", the "User:" field contains "userinv", and the "Password:" field contains six asterisks. The "send" button remains at the bottom right.

Figura 2. Ejemplo - Datos para transformación

HOME DOCUMENTATION ABOUT

RDBS2OWL2

Information Database Schema

Server:

Database:

User:

Password:

Ontology Information

xml:base

ontologyIRI

Inheritances

Parent	Childs	Inheritance Type
profesores	doctores no_doctores	partial & overlap ▼
publicaciones	congresos revistas	partial & overlap ▼

Figura 3. Información adicional para RDBS con herencia

HOME DOCUMENTATION ABOUT

RDBS2OWL2

Information Database Schema

Server:

Database:

User:

Password:

Result

```

<!DOCTYPE Ontology [
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
]
<Ontology
  xml:base="proyectosinv"
  ontologyIRI="proyectosinv"
  xmlns="http://www.w3.org/2002/07/owl#"
  <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#" />
  <Declaration>
    <Class IRI="cls_table_congresos" />
  <Declaration>
  <Declaration>
    <Class IRI="cls_key_congresos_proyecto" />
  <Declaration>
  <Declaration>
    <Class IRI="cls_key_congresos_num_publicacion" />
  <Declaration>

```

Figura 4. Resultado de la transformación

Documentación del prototipo

Seleccionar la pestaña Descripción y se mostrará la información relacionada con el prototipo RDB2OWL2. Ver Figura 5.

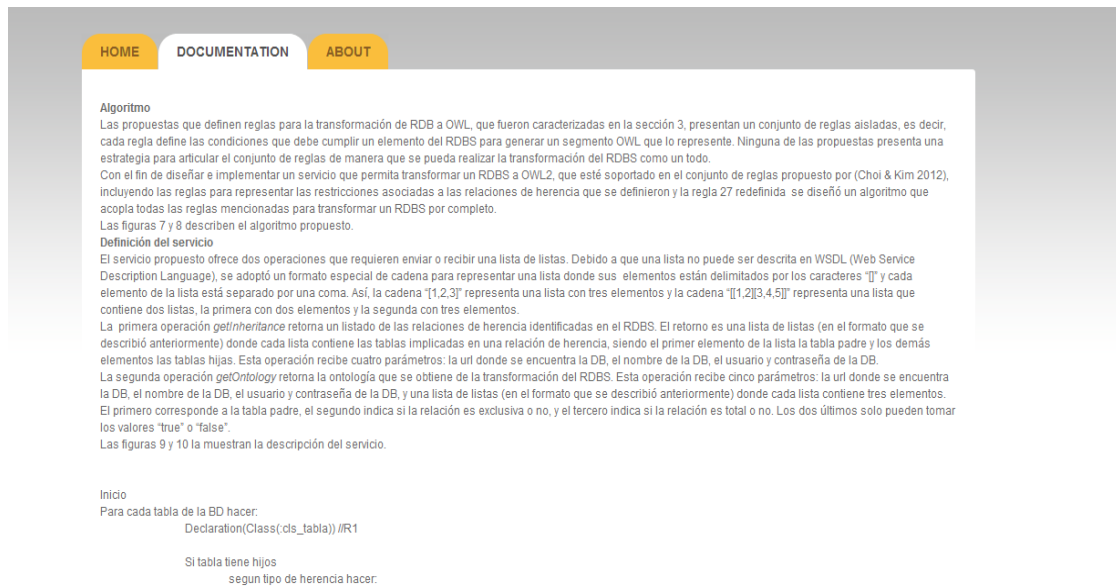


Figura 5. Documentación del prototipo

Información sobre los autores

Seleccionar la pestaña **Acerca de** y se mostrará la información relacionada con los autores de la herramienta. Ver



Figura 6. Información de los autores

ANEXO 3. MANUAL TÉCNICO

Especificación técnica del sistema

Arquitectura

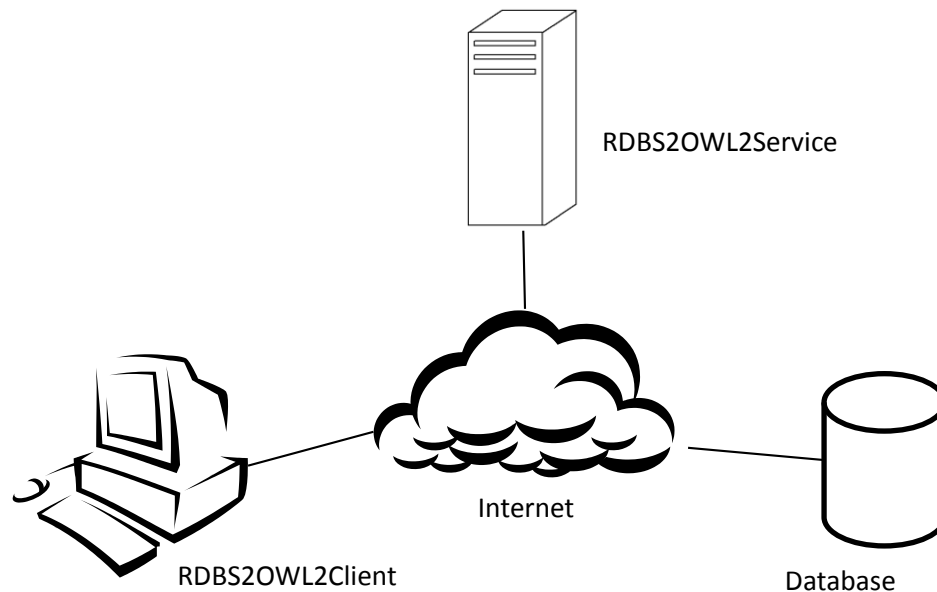


Figura 7. Arquitectura del sistema

La Figura 7 describe la arquitectura un servicio web que permite usar una conexión remota a través de internet para extraer el RDBS con los datos del cliente y realizar la petición al servicio RDB2OWL2, para transformar el RDBS a OWL2.

Diagrama de despliegue del sistema

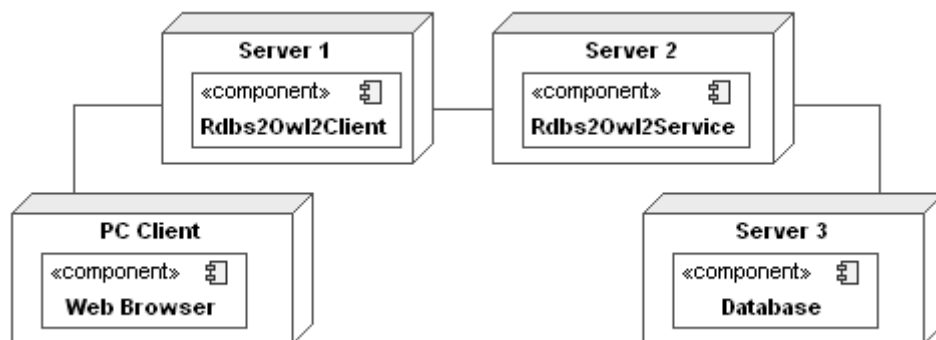


Figura 8. Diagrama de despliegue del sistema

Tecnologías

Lenguaje: Java

Entorno: NetBeans

Plataforma: Java EE (Java Enterprise Edition).

Servidor y Cliente: JAX-WS (Java API for XML Web Service)

Conexión con la BD: JDBC (Java Database Connectivity)

Entorno de despliegue

Se despliega tanto el servidor RDBS2OWL2Service como el cliente RDBS2OWL2Client en la plataforma de OPENSIFT (OpenShift es un producto de computación en la nube de plataforma como servicio de Red Hat), la cual permite que las aplicaciones servidor y cliente, estén disponibles en la web para diferentes usuarios.

Especificación técnica del Servicio

Diagrama de paquetes y clases

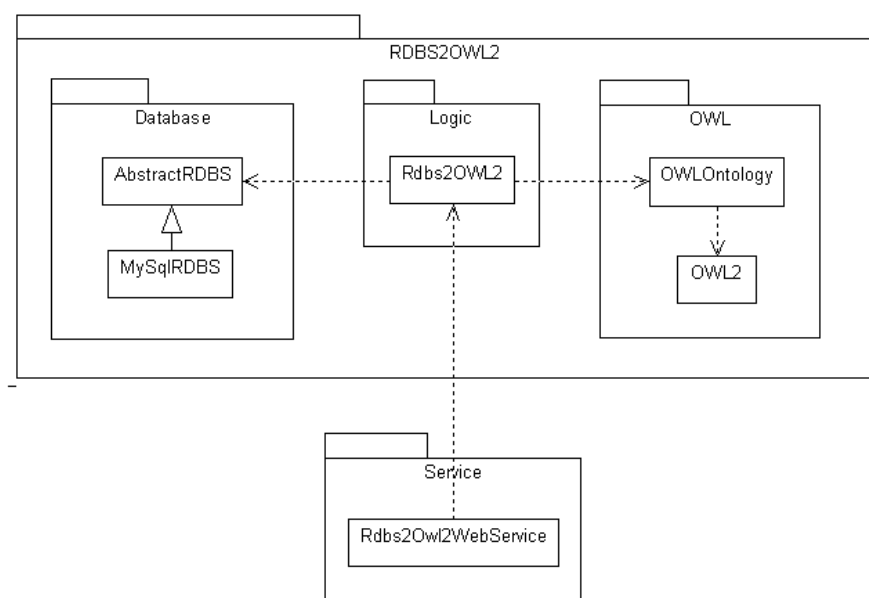


Figura 9. Diagrama de paquetes y clases del servicio

A continuación se presenta una breve descripción de cada paquete:

RDBS2OWL2: contiene todos los paquetes que ofrecen las funcionalidades necesarias para la transformación de un RDBS a OWL2. Estos paquetes son:

Database: contiene las clases necesarias para extraer la información del RDBS requerida por el algoritmo.

OWL: contiene las clases necesarias para crear los elementos OWL2 requeridos por el algoritmo y generar la ontología.

Logic: contiene la lógica necesaria para transformar un RDBS a OWL2. Hace uso de los paquetes Database y OWL para cumplir su objetivo.

Service: contiene una clase que actúa como interfaz entre el cliente y el servicio.

A continuación se presenta una breve descripción de las clases:

AbstractRDBS: es una clase abstracta que define los métodos con los parámetros y retornos necesarios para el correcto funcionamiento del algoritmo. Esta clase permite extender el prototipo para transformar bases de datos diferentes a MySQL.

MySQLRDBS: extiende la clase AbstractRDBS e implementa sus métodos abstractos para extraer información del esquema de una DB MySQL.

OWLontology: crea y organiza la estructura de la ontología generada por el algoritmo.

OWL2: ofrece las funciones necesarias para crear los elementos OWL2 requeridos por la ontología.

Rdbs2Owl2: implementa la lógica del algoritmo propuesto en el presente proyecto.

Rdbs2Owl2WebService: contiene funciones que responden a las solicitudes del cliente.

Diagrama de componentes

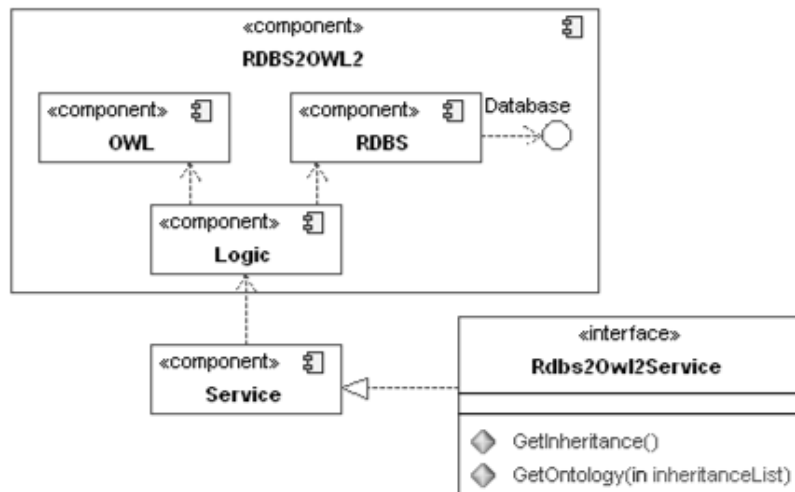


Figura 10. Diagrama de componentes del servicio

Especificación técnica del Cliente

Diagrama de paquetes y clases

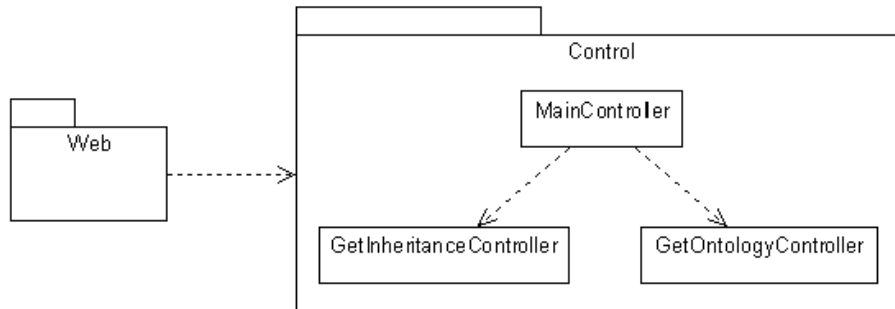


Figura 11. Diagrama de paquetes y clases del cliente

A continuación se presenta una breve descripción de cada paquete:

Web: contiene las vistas de la página web.

Controller: contiene las clases necesarias para gestionar el flujo de mensajes entre las vistas del cliente y las operaciones del servicio.

A continuación se presenta una breve descripción de las clases:

MainController: recibe los mensajes del cliente, verifica que la información sea correcta e invoca los métodos correspondientes a la petición del cliente.

GetInheritanceController: invoca la operación *getInheritance* del servicio enviando los parámetros necesarios, recibe la respuesta y la pasa a las vistas del cliente.

GetOntologyController: invoca la operación *getOntology* del servicio enviando los parámetros necesarios, recibe la respuesta y la pasa a las vistas del cliente.

Diagrama de Componentes

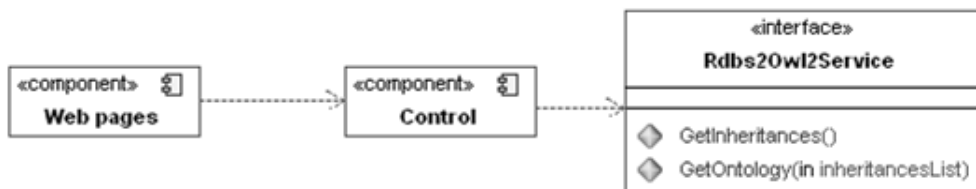


Figura 12. Diagrama de componentes del cliente

ANEXO 4. SCRIPTS DE LOS RDBS

En esta sección se muestran los scripts usados en la sección 5.3.3 del Capítulo 5 del presente proyecto. Se presenta el título con su correspondiente script en SQL.

1. Alojamientos rurales.

Script:

```
-- phpMyAdmin SQL Dump
-- version 3.5.2.2
-- http://www.phpmyadmin.net
--
-- Host: 127.0.0.1
-- Generation Time: Sep 06, 2013 at 05:05 AM
-- Server version: 5.5.27
-- PHP Version: 5.4.7

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Database: `alojamientos_rurales`
--

-----

--
-- Table structure for table `actividades`
--

CREATE TABLE IF NOT EXISTS `actividades` (
  `codigo_activ` char(3) NOT NULL,
  `nombre_activ` char(30) NOT NULL,
  `descripcion` char(50) NOT NULL,
  `nivel` char(2) NOT NULL,
  PRIMARY KEY (`codigo_activ`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-----

--
-- Table structure for table `alojamientos`
--

CREATE TABLE IF NOT EXISTS `alojamientos` (
  `nombre_aloja` char(20) NOT NULL,
  `direccion` char(25) NOT NULL,
  `telefono` char(9) NOT NULL,
  `cotacto` char(3) NOT NULL,
  `no_hab` int(11) NOT NULL,
  PRIMARY KEY (`nombre_aloja`),
  UNIQUE KEY `cotacto` (`cotacto`),
  UNIQUE KEY `telefono` (`telefono`),
  KEY `cotacto_2` (`cotacto`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-----

--
-- Table structure for table `habitaciones`
--

CREATE TABLE IF NOT EXISTS `habitaciones` (
```

```

`alojamiento` char(30) NOT NULL,
`n_habita` int(3) NOT NULL,
`tipo` char(1) NOT NULL,
`banio` char(2) DEFAULT NULL,
`precio` int(11) NOT NULL,
PRIMARY KEY (`alojamiento`,`n_habita`),
KEY `alojamiento` (`alojamiento`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-----
--
-- Table structure for table `personal`
--

```

```

CREATE TABLE IF NOT EXISTS `personal` (
`codigo_p` char(3) NOT NULL,
`nombre_p` char(30) NOT NULL,
`direccion` char(25) NOT NULL,
`NIF` char(10) NOT NULL,
`nombre_aloja` char(30) NOT NULL,
PRIMARY KEY (`codigo_p`),
UNIQUE KEY `NIF` (`NIF`),
KEY `nombre_aloja` (`nombre_aloja`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-----
--
-- Table structure for table `realiza_actividad`
--

```

```

CREATE TABLE IF NOT EXISTS `realiza_actividad` (
`codigo_act` char(3) NOT NULL,
`nombre_aloja` char(30) NOT NULL,
`dia_semana` char(10) NOT NULL,
PRIMARY KEY (`codigo_act`,`nombre_aloja`,`dia_semana`),
KEY `nombre_aloja` (`nombre_aloja`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

--
-- Constraints for dumped tables
--

```

```

--
-- Constraints for table `alojamientos`
--

```

```

ALTER TABLE `alojamientos`
ADD CONSTRAINT `FK_alojamientos_personal` FOREIGN KEY (`cotacto`) REFERENCES `personal` (`codigo_p`) ON UPDATE CASCADE;

```

```

--
-- Constraints for table `habitaciones`
--

```

```

ALTER TABLE `habitaciones`
ADD CONSTRAINT `FK_habitaciones_alojamientos` FOREIGN KEY (`alojamiento`) REFERENCES `alojamientos` (`nombre_aloja`) ON DELETE CASCADE ON UPDATE CASCADE;

```

```

--
-- Constraints for table `personal`
--

```

```

ALTER TABLE `personal`
ADD CONSTRAINT `FK_personal_alojamientos` FOREIGN KEY (`nombre_aloja`) REFERENCES `alojamientos` (`nombre_aloja`) ON UPDATE CASCADE;

```

```

--
-- Constraints for table `realiza_actividad`
--

```

```

ALTER TABLE `realiza_actividad`
ADD CONSTRAINT `FK_realiza_actividad_alojamientos` FOREIGN KEY (`nombre_aloja`) REFERENCES `alojamientos` (`nombre_aloja`) ON UPDATE CASCADE,
ADD CONSTRAINT `FK_realiza_actividad_actividad` FOREIGN KEY (`codigo_act`) REFERENCES `actividades` (`codigo_activ`) ON UPDATE CASCADE;

```

```

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

2. Observación de aves.

Script:

```
-- phpMyAdmin SQL Dump
-- version 3.3.9
-- http://www.phpmyadmin.net
--
-- Servidor: localhost
-- Tiempo de generación: 13-11-2013 a las 04:52:26
-- Versión del servidor: 5.5.8
-- Versión de PHP: 5.3.5

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Base de datos: `observacion_aves_db`
--
-----

--
-- Estructura de tabla para la tabla `asociaciones`
--

CREATE TABLE IF NOT EXISTS `asociaciones` (
  `nombre_aso` char(60) NOT NULL,
  `direccion` char(50) NOT NULL,
  `telefono` char(9) NOT NULL,
  PRIMARY KEY (`nombre_aso`),
  UNIQUE KEY `telefono` (`telefono`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Volcar la base de datos para la tabla `asociaciones`
--

-----

--
-- Estructura de tabla para la tabla `especies`
--

CREATE TABLE IF NOT EXISTS `especies` (
  `nombre_cientifico` char(60) NOT NULL,
  `nombre_vulgar` char(60) NOT NULL,
  `descripcion` char(100) NOT NULL,
  `nombre-g` char(5) DEFAULT NULL,
  PRIMARY KEY (`nombre_cientifico`),
  UNIQUE KEY `nombre_vulgar` (`nombre_vulgar`),
  KEY `nombre-g` (`nombre-g`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Volcar la base de datos para la tabla `especies`
--

-----

--
-- Estructura de tabla para la tabla `grupos_ornitologicos`
--

CREATE TABLE IF NOT EXISTS `grupos_ornitologicos` (
```

```

`nombre` char(60) NOT NULL,
`patas` char(20) DEFAULT NULL,
`dedos` char(20) DEFAULT NULL,
`pico` char(20) DEFAULT NULL,
PRIMARY KEY (`nombre`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Volcar la base de datos para la tabla `grupos_ornitologicos`
--

-----

--
-- Estructura de tabla para la tabla `observaciones`
--

CREATE TABLE IF NOT EXISTS `observaciones` (
  `codigo_obs` char(5) NOT NULL,
  `nombre_cientifico` char(60) NOT NULL,
  `fecha` date NOT NULL,
  PRIMARY KEY (`codigo_obs`,`nombre_cientifico`,`fecha`),
  KEY `nombre_cientifico` (`nombre_cientifico`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Volcar la base de datos para la tabla `observaciones`
--

-----

--
-- Estructura de tabla para la tabla `observadores`
--

CREATE TABLE IF NOT EXISTS `observadores` (
  `codigo_obs` char(5) NOT NULL,
  `dni` char(9) NOT NULL,
  `nombre` char(40) NOT NULL,
  `direccion` char(50) NOT NULL,
  `nombre_aso` char(60) DEFAULT NULL,
  `codigo_z` char(5) NOT NULL,
  `situacion` char(8) NOT NULL,
  PRIMARY KEY (`codigo_obs`),
  UNIQUE KEY `dni` (`dni`),
  KEY `nombre_aso` (`nombre_aso`),
  KEY `codigo_z` (`codigo_z`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Volcar la base de datos para la tabla `observadores`
--

-----

--
-- Estructura de tabla para la tabla `provincias`
--

CREATE TABLE IF NOT EXISTS `provincias` (
  `nombre` char(60) NOT NULL,
  `comunidad` char(60) NOT NULL,
  PRIMARY KEY (`nombre`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Volcar la base de datos para la tabla `provincias`
--

-----

--
-- Estructura de tabla para la tabla `zonas_observacion`

```

```

--
CREATE TABLE IF NOT EXISTS `zonas_observacion` (
  `codigo_z` char(5) NOT NULL,
  `nombre` char(15) NOT NULL,
  `provincia` char(20) NOT NULL,
  `tipo` char(15) NOT NULL,
  PRIMARY KEY (`codigo_z`),
  KEY `provincia` (`provincia`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Volcar la base de datos para la tabla `zonas_observacion`
--

--
-- Filtros para las tablas descargadas (dump)
--

--
-- Filtros para la tabla `especies`
--
ALTER TABLE `especies`
  ADD CONSTRAINT `FK_especies_grupos_ornitologicos` FOREIGN KEY (`nombre-g`) REFERENCES `grupos_ornitologicos` (`nombre`)
  ON UPDATE CASCADE;

--
-- Filtros para la tabla `observaciones`
--
ALTER TABLE `observaciones`
  ADD CONSTRAINT `FK_observaciones_observadores` FOREIGN KEY (`codigo_obs`) REFERENCES `observadores` (`codigo_obs`) ON
  UPDATE CASCADE,
  ADD CONSTRAINT `FK_observaciones_especies` FOREIGN KEY (`nombre_cientifico`) REFERENCES `especies` (`nombre_cientifico`)
  ON UPDATE CASCADE;

--
-- Filtros para la tabla `observadores`
--
ALTER TABLE `observadores`
  ADD CONSTRAINT `FK_observadores_asociaciones` FOREIGN KEY (`nombre_aso`) REFERENCES `asociaciones` (`nombre_aso`) ON
  DELETE SET NULL ON UPDATE CASCADE,
  ADD CONSTRAINT `FK_observadores_zonas_observacion` FOREIGN KEY (`codigo_z`) REFERENCES `zonas_observacion` (`codigo_z`)
  ON UPDATE CASCADE;

--
-- Filtros para la tabla `zonas_observacion`
--
ALTER TABLE `zonas_observacion`
  ADD CONSTRAINT `FK_zonas_observacion_provincias` FOREIGN KEY (`provincia`) REFERENCES `provincias` (`nombre`) ON UPDATE
  CASCADE;

```

3. Gestión de proyectos informáticos.

Script:

```

-- phpMyAdmin SQL Dump
-- version 3.5.2.2
-- http://www.phpmyadmin.net
--
-- Host: 127.0.0.1
-- Generation Time: Nov 22, 2013 at 08:21 PM
-- Server version: 5.5.27
-- PHP Version: 5.4.7

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;

```

```
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
```

```
--
-- Database: `gestion-proyectos_informaticos_bd`
--
```

```
-----
```

```
--
-- Table structure for table `fases`
--
```

```
CREATE TABLE IF NOT EXISTS `fases` (
  `cod_proy` char(6) NOT NULL,
  `num_fase` int(11) NOT NULL,
  `nombre` char(50) NOT NULL,
  `fecha_i` date NOT NULL,
  `fecha_f` date NOT NULL,
  `estado` enum('en curso','finalizado') NOT NULL,
  PRIMARY KEY (`cod_proy`,`num_fase`),
  KEY `num_fase` (`num_fase`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
-----
```

```
--
-- Table structure for table `generan`
--
```

```
CREATE TABLE IF NOT EXISTS `generan` (
  `cod_proy` char(6) NOT NULL,
  `num_fase` int(11) NOT NULL,
  `cod_pr` char(6) NOT NULL,
  PRIMARY KEY (`cod_proy`,`num_fase`,`cod_pr`),
  KEY `cod_pr` (`cod_pr`),
  KEY `num_fase` (`num_fase`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
-----
```

```
--
-- Table structure for table `productos`
--
```

```
CREATE TABLE IF NOT EXISTS `productos` (
  `cod_pr` char(6) NOT NULL,
  `nombre` char(50) NOT NULL,
  `descripcion` text NOT NULL,
  `estado` enum('si','no') NOT NULL,
  `tipo` enum('software','prototipo','informe tecnico') NOT NULL,
  PRIMARY KEY (`cod_pr`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
-----
```

```
--
-- Table structure for table `prototipos`
--
```

```
CREATE TABLE IF NOT EXISTS `prototipos` (
  `cod_pr` char(6) NOT NULL,
  `version` char(20) NOT NULL,
  `ubicacion` char(50) NOT NULL,
  PRIMARY KEY (`cod_pr`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
-----
```

```
--
-- Table structure for table `proyectos`
--
```

```
CREATE TABLE IF NOT EXISTS `proyectos` (
  `cod_proy` char(5) NOT NULL,
  `nombre` char(50) NOT NULL,
```

```

`descripcion` text NOT NULL,
`fecha_i` date NOT NULL,
`fecha_f` date NOT NULL,
`presupuesto` float NOT NULL,
`horas` int(11) NOT NULL,
PRIMARY KEY (`cod_proy`),
UNIQUE KEY `nombre` (`nombre`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-----

--
-- Table structure for table `software`
--

CREATE TABLE IF NOT EXISTS `software` (
  `cod_pr` char(6) NOT NULL,
  `tipo_sw` char(50) NOT NULL,
  PRIMARY KEY (`cod_pr`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Constraints for dumped tables
--

--
-- Constraints for table `fases`
--
ALTER TABLE `fases`
  ADD CONSTRAINT `FK_fases_proyectos` FOREIGN KEY (`cod_proy`) REFERENCES `proyectos` (`cod_proy`);

--
-- Constraints for table `generan`
--
ALTER TABLE `generan`
  ADD CONSTRAINT `FK_generan_fases1` FOREIGN KEY (`cod_proy`) REFERENCES `fases` (`cod_proy`),
  ADD CONSTRAINT `FK_generan_productos` FOREIGN KEY (`cod_pr`) REFERENCES `productos` (`cod_pr`),
  ADD CONSTRAINT `FK_generan_fases2` FOREIGN KEY (`num_fase`) REFERENCES `fases` (`num_fase`);

--
-- Constraints for table `prototipos`
--
ALTER TABLE `prototipos`
  ADD CONSTRAINT `FK_prototipos_productos` FOREIGN KEY (`cod_pr`) REFERENCES `productos` (`cod_pr`);

--
-- Constraints for table `software`
--
ALTER TABLE `software`
  ADD CONSTRAINT `FK_software_productos` FOREIGN KEY (`cod_pr`) REFERENCES `productos` (`cod_pr`);

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

4. Proyectos de investigación.

Script:

```

-- phpMyAdmin SQL Dump
-- version 3.5.2.2
-- http://www.phpmyadmin.net
--
-- Host: 127.0.0.1
-- Generation Time: Nov 23, 2013 at 02:07 AM
-- Server version: 5.5.27
-- PHP Version: 5.4.7

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

```

```

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

```

```

--
-- Database: `proyectos-investigacion`
--

```

```

-----

```

```

--
-- Table structure for table `congresos`
--

```

```

CREATE TABLE IF NOT EXISTS `congresos` (
  `proyecto` char(9) NOT NULL,
  `num_publicacion` int(11) NOT NULL,
  `tipo` varchar(20) NOT NULL,
  `lugar` varchar(50) NOT NULL,
  `fecha_i` date NOT NULL,
  `fecha_f` date NOT NULL,
  `pais` varchar(50) NOT NULL,
  `editorial` varchar(50) NOT NULL,
  PRIMARY KEY (`proyecto`,`num_publicacion`),
  KEY `num_publicacion` (`num_publicacion`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-----

```

```

--
-- Table structure for table `doctores`
--

```

```

CREATE TABLE IF NOT EXISTS `doctores` (
  `nombre` char(20) NOT NULL,
  PRIMARY KEY (`nombre`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-----

```

```

--
-- Table structure for table `no_doctores`
--

```

```

CREATE TABLE IF NOT EXISTS `no_doctores` (
  `nombre` char(20) NOT NULL,
  PRIMARY KEY (`nombre`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-----

```

```

--
-- Table structure for table `participan`
--

```

```

CREATE TABLE IF NOT EXISTS `participan` (
  `profesor` char(20) NOT NULL,
  `proyecto_inv` char(9) NOT NULL,
  `fecha_i` date NOT NULL,
  `fecha_f` date NOT NULL,
  PRIMARY KEY (`profesor`,`proyecto_inv`,`fecha_i`),
  KEY `proyecto_inv` (`proyecto_inv`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-----

```

```

--
-- Table structure for table `profesores`
--

```

```

CREATE TABLE IF NOT EXISTS `profesores` (
  `nombre` char(20) NOT NULL,
  `despacho` char(20) NOT NULL,
  `telefono` char(9) NOT NULL,

```



```

`tipo` int(11) NOT NULL,
PRIMARY KEY (`nombre`),
UNIQUE KEY `telefono` (`telefono`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

--
-- Table structure for table `publicaciones`
--

```

```

CREATE TABLE IF NOT EXISTS `publicaciones` (
  `proyecto` char(9) NOT NULL,
  `num` int(11) NOT NULL,
  `titulo` varchar(50) NOT NULL,
  `nombre` varchar(50) NOT NULL,
  `tipo` varchar(20) NOT NULL,
  PRIMARY KEY (`proyecto`,`num`),
  KEY `num` (`num`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

--
-- Table structure for table `p_investigacion`
--

```

```

CREATE TABLE IF NOT EXISTS `p_investigacion` (
  `cod_ref` char(9) NOT NULL,
  `nombre` char(50) NOT NULL,
  `presupuesto` int(11) NOT NULL,
  `acronimo` char(20) NOT NULL,
  `prog_i+d` char(20) NOT NULL,
  `descripcion` char(60) NOT NULL,
  `fecha_i` date NOT NULL,
  `fecha_f` date NOT NULL,
  `inv_ppal` char(20) NOT NULL,
  PRIMARY KEY (`cod_ref`),
  KEY `inv_ppal` (`inv_ppal`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

--
-- Table structure for table `revistas`
--

```

```

CREATE TABLE IF NOT EXISTS `revistas` (
  `proyecto` char(9) NOT NULL,
  `num_publicacion` int(11) NOT NULL,
  `volumen` int(11) NOT NULL,
  `numero` int(11) NOT NULL,
  `pagina_ini` int(11) NOT NULL,
  `pagina_fin` int(11) NOT NULL,
  PRIMARY KEY (`proyecto`,`num_publicacion`),
  KEY `num_publicacion` (`num_publicacion`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

--
-- Table structure for table `supervisan`
--

```

```

CREATE TABLE IF NOT EXISTS `supervisan` (
  `doctor` char(20) NOT NULL,
  `no_doctor` char(20) NOT NULL,
  `f_ini` date NOT NULL,
  `f_fin` date NOT NULL,
  PRIMARY KEY (`no_doctor`,`f_ini`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

--
-- Constraints for dumped tables
--

```

--

```

-- Constraints for table `congresos`
--
ALTER TABLE `congresos`
  ADD CONSTRAINT `congresos_ibfk_1` FOREIGN KEY (`proyecto`) REFERENCES `publicaciones` (`proyecto`) ON DELETE CASCADE
  ON UPDATE CASCADE,
  ADD CONSTRAINT `congresos_ibfk_2` FOREIGN KEY (`num_publicacion`) REFERENCES `publicaciones` (`num`) ON DELETE
  CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `doctores`
--
ALTER TABLE `doctores`
  ADD CONSTRAINT `FK_doctores_profesores` FOREIGN KEY (`nombre`) REFERENCES `profesores` (`nombre`) ON DELETE CASCADE
  ON UPDATE CASCADE;

--
-- Constraints for table `no_doctores`
--
ALTER TABLE `no_doctores`
  ADD CONSTRAINT `FK_no_doctores_profesores` FOREIGN KEY (`nombre`) REFERENCES `profesores` (`nombre`) ON DELETE
  CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `participan`
--
ALTER TABLE `participan`
  ADD CONSTRAINT `FK_participan_profesores` FOREIGN KEY (`profesor`) REFERENCES `profesores` (`nombre`) ON DELETE NO
  ACTION ON UPDATE CASCADE,
  ADD CONSTRAINT `FK_participan_p_investigacion` FOREIGN KEY (`proyecto_inv`) REFERENCES `p_investigacion` (`cod_ref`) ON
  DELETE NO ACTION ON UPDATE CASCADE;

--
-- Constraints for table `publicaciones`
--
ALTER TABLE `publicaciones`
  ADD CONSTRAINT `publicaciones_ibfk_1` FOREIGN KEY (`proyecto`) REFERENCES `p_investigacion` (`cod_ref`) ON DELETE
  CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `p_investigacion`
--
ALTER TABLE `p_investigacion`
  ADD CONSTRAINT `FK_p_investigacion_no_doctores` FOREIGN KEY (`inv_ppal`) REFERENCES `no_doctores` (`nombre`) ON DELETE
  NO ACTION ON UPDATE CASCADE;

--
-- Constraints for table `revistas`
--
ALTER TABLE `revistas`
  ADD CONSTRAINT `revistas_ibfk_1` FOREIGN KEY (`num_publicacion`) REFERENCES `publicaciones` (`num`),
  ADD CONSTRAINT `revistas_publicaciones1` FOREIGN KEY (`proyecto`) REFERENCES `publicaciones` (`proyecto`) ON DELETE
  CASCADE ON UPDATE CASCADE;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

ANEXO 5. RESULTADOS DE LAS TRANSFORMACIONES

A continuación se presentan los resultados de cada una de las transformaciones realizadas durante la evaluación del prototipo y el porcentaje de error respectivo.

Transformación I – RDBS Observación de aves

Tabla 1. Resultado transformación I – RDBS Observación de aves

Elemento OWL	Cantidad esperada	Cantidad Obtenida	% Error
Class	23	23	0
ObjectProperty	16	16	0
DataProperty	27	27	0
ObjectPropertyDomain	16	16	0
ObjectPropertyRange	16	16	0
DataPropertyDomain	27	27	0
HasKey con ObjectProperty	7	7	0
HasKey con DataProperty	16	16	0
SubClassOf con DataAllValuesFrom	27	27	0
SubClassOf con DataExactCardinality	24	24	0
SubClassOf con ObjectAllValuesFrom	16	16	0
SubClassOf	6	6	0
SubDataPropertyOf	6	6	0
InverseFunctionalObjectProperty	3	3	0
DataPropertyRange con DatatypeRestriccion	0	0	0
DataPropertyRange con DataOneOf	0	0	0
DataPropertyRange	27	27	0
SubClassOf con ObjectExactCardinality	14	14	0
SubClassOf con ObjectMaxCardinality	2	2	0
SubClassOf con DataMaxCardinality	3	3	0
SubObjectPropertyOf	0	0	0
DisjoinClasses	7	7	0
Total	283	283	0

En esta transformación no se presentó ningún error entre los resultados esperados y los resultados obtenidos, por lo cual se procedió con la transformación del siguiente RDBS usando la versión 2 del prototipo.

Transformación II – RDBS Proyectos de investigación

Tabla 2. Resultado transformación II – RDBS Proyectos de investigación

Elemento OWL	Cantidad esperada	Cantidad Obtenida	% Error
--------------	-------------------	-------------------	---------

Class	17	17	0
ObjectProperty	11	11	0
DataProperty	23	23	0
ObjectPropertyDomain	11	11	0
ObjectPropertyRange	11	11	0
DataPropertyDomain	23	23	0
HasKey con ObjectProperty	6	6	0
HasKey con DataProperty	11	11	0
SubClassOf con DataAllValuesFrom	23	23	0
SubClassOf con DataExactCardinality	23	23	0
SubClassOf con ObjectAllValuesFrom	11	11	0
SubClassOf	7	7	0
SubDataPropertyOf	5	5	0
InverseFunctionalObjectProperty	1	1	0
DataPropertyRange con DatatypeRestriccion	0	0	0
DataPropertyRange con DataOneOf	0	0	0
DataPropertyRange	23	23	0
SubClassOf con ObjectExactCardinality	11	11	0
SubClassOf con ObjectMaxCardinality	0	0	0
SubClassOf con DataMaxCardinality	0	0	0
SubObjectPropertyOf	2	2	0
DisjoinClasses	4	4	0
Total	223	223	0

En esta transformación no se presentó ningún error entre los resultados esperados y los resultados obtenidos, por lo cual se procedió con la transformación del siguiente RDBS usando la versión 2 del prototipo.

Transformación III – RDBS Alojamiento rurales

Tabla 3. Resultado transformación III – RDBS Alojamiento rurales

Elemento OWL	Cantidad esperada	Cantidad Obtenida	% Error
Class	17	17	0
ObjectProperty	12	12	0
DataProperty	22	22	0
ObjectPropertyDomain	12	12	0
ObjectPropertyRange	12	12	0
DataPropertyDomain	22	22	0
HasKey con ObjectProperty	5	5	0
HasKey con DataProperty	12	12	0
SubClassOf con DataAllValuesFrom	22	22	0
SubClassOf con DataExactCardinality	21	21	0
SubClassOf con	12	12	0

ObjectAllValuesFrom			
SubClassOf	5	5	0
SubDataPropertyOf	5	5	0
InverseFunctionalObjectProperty	3	3	0
DataPropertyRange con DatatypeRestriccion	0	0	0
DataPropertyRange con DataOneOf	0	0	0
DataPropertyRange	22	22	0
SubClassOf con ObjectExactCardinality	12	12	0
SubClassOf con ObjectMaxCardinality	0	0	0
SubClassOf con DataMaxCardinality	1	1	0
SubObjectPropertyOf	0	0	0
DisjoinClasses	5	5	0
Total	222	222	0

En esta transformación no se presentó ningún error entre los resultados esperados y los resultados obtenidos, por lo cual se procedió con la transformación del siguiente RDBS usando la versión 2 del prototipo.

Transformación IV – RDBS Gestión de proyectos informáticos

Tabla 4. Resultados transformación IV – RDBS Gestión de proyectos informáticos

Elemento OWL	Cantidad esperada	Cantidad Obtenida	% Error
Class	16	16	0
ObjectProperty	10	10	0
DataProperty	26	26	0
ObjectPropertyDomain	10	10	0
ObjectPropertyRange	10	10	0
DataPropertyDomain	26	26	0
HasKey con ObjectProperty	6	6	0
HasKey con DataProperty	10	10	0
SubClassOf con DataAllValuesFrom	26	26	0
SubClassOf con DataExactCardinality	26	26	0
SubClassOf con ObjectAllValuesFrom	10	10	0
SubClassOf	8	9	0.01
SubDataPropertyOf	6	6	0
InverseFunctionalObjectProperty	1	1	0
DataPropertyRange con DatatypeRestriccion	0	0	0
DataPropertyRange con DataOneOf	3	3	0
DataPropertyRange	23	23	0
SubClassOf con ObjectExactCardinality	10	10	0
SubClassOf con ObjectMaxCardinality	0	0	0
SubClassOf con DataMaxCardinality	0	0	0

SubObjectPropertyOf	2	5	0.03
DisjoinClasses	3	3	0
Total	232	236	0.04

En esta transformación se presentó dos errores en los resultados de los elementos OWL *SubClassOf* y *SubObjectPropertyOf*. Estos errores fueron corregidos hasta que el error total se hizo igual a cero, se generó la versión 3 del prototipo, con el cual se ejecutó, nuevamente, la transformación de los anteriores RDBS.

Transformación V – RDBS Observación de aves

Tabla 5. Resultados transformación V – RDBS Observación de aves

Elemento OWL	Cantidad esperada	Cantidad Obtenida	% Error
Class	23	23	0
ObjectProperty	16	16	0
DataProperty	27	27	0
ObjectPropertyDomain	16	16	0
ObjectPropertyRange	16	16	0
DataPropertyDomain	27	27	0
HasKey con ObjectProperty	7	7	0
HasKey con DataProperty	16	16	0
SubClassOf con DataAllValuesFrom	27	27	0
SubClassOf con DataExactCardinality	24	24	0
SubClassOf con ObjectAllValuesFrom	16	16	0
SubClassOf	6	8	0.02
SubDataPropertyOf	6	6	0
InverseFunctionalObjectProperty	3	3	0
DataPropertyRange con DatatypeRestriccion	0	0	0
DataPropertyRange con DataOneOf	0	0	0
DataPropertyRange	27	27	0
SubClassOf con ObjectExactCardinality	14	14	0
SubClassOf con ObjectMaxCardinality	2	2	0
SubClassOf con DataMaxCardinality	3	3	0
SubObjectPropertyOf	0	2	0.02
DisjoinClasses	7	5	0.02
Total	283	285	0.06

En esta transformación se presentó tres errores en los resultados de los elementos OWL *SubClassOf*, *SubObjectPropertyOf* y *DisjoinClasses*. Estos errores fueron corregidos hasta que el error total se hizo igual a cero, se generó la versión 4 del prototipo, con el cual se ejecutó, nuevamente, la transformación de los anteriores RDBS.

Transformación VI – RDBS Proyectos de investigación

Tabla 6. Resultados transformación VI – RDBS Proyectos de investigación

Elemento OWL	Cantidad esperada	Cantidad Obtenida	% Error
Class	17	17	0
ObjectProperty	11	11	0
DataProperty	23	23	0
ObjectPropertyDomain	11	11	0
ObjectPropertyRange	11	11	0
DataPropertyDomain	23	23	0
HasKey con ObjectProperty	6	6	0
HasKey con DataProperty	11	11	0
SubClassOf con DataAllValuesFrom	23	23	0
SubClassOf con DataExactCardinality	23	23	0
SubClassOf con ObjectAllValuesFrom	11	11	0
SubClassOf	7	7	0
SubDataPropertyOf	5	5	0
InverseFunctionalObjectProperty	1	1	0
DataPropertyRange con DatatypeRestriccion	0	0	0
DataPropertyRange con DataOneOf	0	0	0
DataPropertyRange	23	23	0
SubClassOf con ObjectExactCardinality	11	11	0
SubClassOf con ObjectMaxCardinality	0	0	0
SubClassOf con DataMaxCardinality	0	0	0
SubObjectPropertyOf	2	2	0
DisjoinClasses	4	4	0
Total	223	223	0

En esta transformación no se presentó ningún error entre los resultados esperados y los resultados obtenidos, por lo cual se procedió con la transformación del siguiente RDBS usando la versión 4 del prototipo.

Transformación VII – RDBS Alojamiento rurales

Tabla 7. Resultados transformación VII – RDBS Alojamiento rurales

Elemento OWL	Cantidad esperada	Cantidad Obtenida	% Error
Class	17	17	0
ObjectProperty	12	12	0
DataProperty	22	22	0
ObjectPropertyDomain	12	12	0
ObjectPropertyRange	12	12	0
DataPropertyDomain	22	22	0

HasKey con ObjectProperty	5	5	0
HasKey con DataProperty	12	12	0
SubClassOf con DataAllValuesFrom	22	22	0
SubClassOf con DataExactCardinality	21	21	
SubClassOf con ObjectAllValuesFrom	12	12	0
SubClassOf	5	5	0
SubDataPropertyOf	5	5	0
InverseFunctionalObjectProperty	3	3	0
DataPropertyRange con DatatypeRestriccion	0	0	0
DataPropertyRange con DataOneOf	0	0	0
DataPropertyRange	22	22	0
SubClassOf con ObjectExactCardinality	12	12	0
SubClassOf con ObjectMaxCardinality	0	0	0
SubClassOf con DataMaxCardinality	1	1	0
SubObjectPropertyOf	0	0	0
DisjoinClasses	5	5	0
Total	222	222	0

En esta transformación no se presentó ningún error entre los resultados esperados y los resultados obtenidos, por lo cual se procedió con la transformación del siguiente RDBS usando la versión 4 del prototipo.

Transformación VIII – RDBS Gestión de proyectos informáticos

Tabla 8. Resultados transformación VIII – RDBS Gestión de proyectos informáticos

Elemento OWL	Cantidad esperada	Cantidad Obtenida	% Error
Class	16	16	0
ObjectProperty	10	10	0
DataProperty	26	26	0
ObjectPropertyDomain	10	10	0
ObjectPropertyRange	10	10	0
DataPropertyDomain	26	26	0
HasKey con ObjectProperty	6	6	0
HasKey con DataProperty	10	10	0
SubClassOf con DataAllValuesFrom	26	26	0
SubClassOf con DataExactCardinality	26	26	
SubClassOf con ObjectAllValuesFrom	10	10	0
SubClassOf	8	8	0
SubDataPropertyOf	6	6	0
InverseFunctionalObjectProperty	1	1	0
DataPropertyRange con DatatypeRestriccion	0	0	0

DataPropertyRange con DataOneOf	3	3	0
DataPropertyRange	23	23	0
SubClassOf con ObjectExactCardinality	10	10	0
SubClassOf con ObjectMaxCardinality	0	0	0
SubClassOf con DataMaxCardinality	0	0	0
SubObjectPropertyOf	2	2	0
DisjoinClasses	3	3	0
Total	232	232	0

En esta transformación no se presentó ningún error entre los resultados esperados y los resultados obtenidos. Con esta transformación se terminó las iteraciones debido a que la versión 4 del prototipo realizó la transformación de los cuatro RDBS destinados para la prueba, obteniendo un error total de cero (0) en cada transformación.

ANEXO 6. RESULTADOS OWL

Esta sección se muestra los resultados obtenidos de las transformaciones hechas con la versión 4 del prototipo, para cada una de las RDBs utilizadas en el proceso de evaluación del prototipo.

1. Alojamientos rurales.

OWL

```

<!DOCTYPE Ontology [
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
]>

<Ontology
xml:base="http://example.com/owl/alojamientos_rurales/"
ontologyIRI="http://example.com/owl/alojamientos_rurales"
xmlns="http://www.w3.org/2002/07/owl#">
<Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#">

<Declaration>
<Class IRI = "cls_table_actividades" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_actividades_codigo_activ" />
</Declaration>
<Declaration>
<Class IRI = "cls_table_alojamientos" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_alojamientos_nombre_aloja" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_alojamientos_telefono" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_alojamientos_cotacto" />
</Declaration>
<Declaration>
<Class IRI = "cls_table_habitaciones" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_habitaciones_alojamiento" />
</Declaration>
<Declaration>

<Class IRI = "cls_key_habitaciones_n_habita" />
</Declaration>
<Declaration>
<Class IRI = "cls_table_personal" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_personal_codigo_p" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_personal_NIF" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_personal_nombre_aloja" />
</Declaration>
<Declaration>
<Class IRI = "cls_table_realiza_actividad" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_realiza_actividad_codigo_act" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_realiza_actividad_nombre_aloja" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_realiza_actividad_dia_semana" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_codigo_activ" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_nombre_aloja" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_telefono" />
</Declaration>
<Declaration>

```



```

</ObjectMaxCardinality>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_habitaciones" />
<DataMaxCardinality cardinality = "1" >
<DataProperty IRI = "dp_column_habitaciones_banio" />
<DataPropertyRange>
<DataProperty IRI = "dp_column_habitaciones_banio" />
<DataType IRI = "&xsd:String" />
</DataPropertyRange>
</DataMaxCardinality>
</SubClassOf>
<DisjointClasses>
<Class IRI = "cls_table_actividades" />
<Class IRI = "cls_key_actividades_codigo_activ" />
</DisjointClasses>
<DisjointClasses>
<Class IRI = "cls_table_alojamientos" />
<Class IRI = "cls_key_alojamientos_nombre_aloja" />
<Class IRI = "cls_key_alojamientos_telefono" />

```

```

<Class IRI = "cls_key_alojamientos_cotacto" />
</DisjointClasses>
</DisjointClasses>
<Class IRI = "cls_table_habitaciones" />
<Class IRI = "cls_key_habitaciones_alojamiento" />
<Class IRI = "cls_key_habitaciones_n_habita" />
</DisjointClasses>
</DisjointClasses>
<Class IRI = "cls_table_personal" />
<Class IRI = "cls_key_personal_codigo_p" />
<Class IRI = "cls_key_personal_NIF" />
<Class IRI = "cls_key_personal_nombre_aloja" />
</DisjointClasses>
</DisjointClasses>
<Class IRI = "cls_table_realiza_actividad" />
<Class IRI = "cls_key_realiza_actividad_codigo_act" />
<Class IRI = "cls_key_realiza_actividad_nombre_aloja" />
<Class IRI = "cls_key_realiza_actividad_dia_semana" />
</DisjointClasses>
</Ontology>

```

2. Observación de aves.

OWL

```

<!DOCTYPE Ontology [
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
]>

<Ontology
xml:base="http://example.com/owl/observacion_aves/"
ontologyIRI="http://example.com/owl/observacion_aves"
xml:ns="http://www.w3.org/2002/07/owl#">
<Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#" />

<Declaration>
<Class IRI = "cls_table_asociaciones" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_asociaciones_nombre_aso" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_asociaciones_telefono" />
</Declaration>
<Declaration>
<Class IRI = "cls_table_especies" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_especies_nombre_cientifico" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_especies_nombre_vulgar" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_especies_nombre-g" />
</Declaration>
<Declaration>
<Class IRI = "cls_table_grupos_ornitologicos" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_grupos_ornitologicos_nombre" />
</Declaration>
<Declaration>
<Class IRI = "cls_table_observaciones" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_observaciones_codigo_obs" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_observaciones_nombre_cientifico" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_observaciones_fecha" />
</Declaration>
<Declaration>
<Class IRI = "cls_table_observadores" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_observadores_codigo_obs" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_observadores_dni" />
</Declaration>

```

```

<Declaration>
<Class IRI = "cls_key_observadores_nombre_aso" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_observadores_codigo_z" />
</Declaration>
<Declaration>
<Class IRI = "cls_table_provincias" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_provincias_nombre" />
</Declaration>
<Declaration>
<Class IRI = "cls_table_zonas_observacion" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_zonas_observacion_codigo_z" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_zonas_observacion_provincia" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_nombre_aso" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_telefono" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_nombre_cientifico" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_nombre_vulgar" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_nombre-g" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_nombre" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_codigo_obs" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_nombre_cientifico" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_fecha" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_codigo_obs" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_dni" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_nombre_aso" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_codigo_z" />

```



```

<SubClassOf>
<Class IRI = "cls_table_observaciones" />
<ObjectExactCardinality cardinality = "1" >
<ObjectProperty IRI = "op_key_nombre_cientifico" />
<Class IRI = "cls_key_observaciones_nombre_cientifico" />
</ObjectExactCardinality>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_observaciones" />
<ObjectExactCardinality cardinality = "1" >
<ObjectProperty IRI = "op_key_fecha" />
<Class IRI = "cls_key_observaciones_fecha" />
</ObjectExactCardinality>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_observadores" />
<ObjectExactCardinality cardinality = "1" >
<ObjectProperty IRI = "op_key_codigo_obs" />
<Class IRI = "cls_key_observadores_codigo_obs" />
</ObjectExactCardinality>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_observadores" />
<ObjectExactCardinality cardinality = "1" >
<ObjectProperty IRI = "op_key_dni" />
<Class IRI = "cls_key_observadores_dni" />
</ObjectExactCardinality>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_observadores" />
<ObjectExactCardinality cardinality = "1" >
<ObjectProperty IRI = "op_key_codigo_z" />
<Class IRI = "cls_key_observadores_codigo_z" />
</ObjectExactCardinality>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_provincias" />
<ObjectExactCardinality cardinality = "1" >
<ObjectProperty IRI = "op_key_nombre" />
<Class IRI = "cls_key_provincias_nombre" />
</ObjectExactCardinality>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_zonas_observacion" />
<ObjectExactCardinality cardinality = "1" >
<ObjectProperty IRI = "op_key_codigo_z" />
<Class IRI = "cls_key_zonas_observacion_codigo_z" />
</ObjectExactCardinality>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_zonas_observacion" />
<ObjectExactCardinality cardinality = "1" >
<ObjectProperty IRI = "op_key_provincia" />
<Class IRI = "cls_key_zonas_observacion_provincia" />
</ObjectExactCardinality>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_especies" />
<ObjectMaxCardinality cardinality = "1" >
<ObjectProperty IRI = "op_key_nombre-g" />
<Class IRI = "cls_key_especies_nombre-g" />
</ObjectMaxCardinality>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_observadores" />
<ObjectMaxCardinality cardinality = "1" >
<ObjectProperty IRI = "op_key_nombre_aso" />
<Class IRI = "cls_key_observadores_nombre_aso" />
</ObjectMaxCardinality>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_grupos_ornitologicos" />
<DataMaxCardinality cardinality = "1" >
<DataProperty IRI = "dp_column_grupos_ornitologicos_patas" />
<DataPropertyRange>
<DataProperty IRI = "dp_column_grupos_ornitologicos_patas" />
<DataType IRI = "&xsd:String" />
</DataPropertyRange>
</DataMaxCardinality>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_grupos_ornitologicos" />
<DataMaxCardinality cardinality = "1" >
<DataProperty IRI = "dp_column_grupos_ornitologicos_dedos" />
<DataPropertyRange>
<DataProperty IRI = "dp_column_grupos_ornitologicos_dedos" />
<DataType IRI = "&xsd:String" />
</DataPropertyRange>
</DataMaxCardinality>
</SubClassOf>
</Ontology>

```


3. Gestión de proyectos informáticos.

OWL

```
<!DOCTYPE Ontology [
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
]>

<Ontology
xml:base="http://example.com/owl/gestion-
proyectos_informaticos_bd/"
ontologyIRI="http://example.com/owl/gestion-
proyectos_informaticos_bd"
xmlns="http://www.w3.org/2002/07/owl#">
<Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#">

<Declaration>
<Class IRI = "cls_table_fases" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_fases_cod_proy" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_fases_num_fase" />
</Declaration>
<Declaration>
<Class IRI = "cls_table_generan" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_generan_cod_proy" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_generan_num_fase" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_generan_cod_pr" />
</Declaration>
<Declaration>
<Class IRI = "cls_table_productos" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_productos_cod_pr" />
</Declaration>
<Declaration>
<Class IRI = "cls_table_prototipos" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_prototipos_cod_pr" />
</Declaration>
<Declaration>
<Class IRI = "cls_table_proyectos" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_proyectos_cod_proy" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_proyectos_nombre" />
</Declaration>
<Declaration>
<Class IRI = "cls_table_software" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_software_cod_pr" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_cod_proy" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_num_fase" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_cod_proy" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_num_fase" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_cod_pr" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_cod_pr" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_cod_pr" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_cod_pr" />
</Declaration>

<ObjectProperty IRI = "op_key_cod_proy" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_nombre" />
</Declaration>
<Declaration>
<ObjectProperty IRI = "op_key_cod_pr" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_fases_cod_proy" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_fases_num_fase" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_fases_nombre" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_fases_fecha_i" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_fases_fecha_f" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_fases_estado" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_generan_cod_proy" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_generan_num_fase" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_generan_cod_pr" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_productos_cod_pr" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_productos_nombre" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_productos_descripcion" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_productos_estado" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_productos_tipo" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_prototipos_cod_pr" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_prototipos_version" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_prototipos_ubicacion" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_proyectos_cod_proy" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_proyectos_nombre" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_proyectos_descripcion" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_proyectos_fecha_i" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_proyectos_fecha_f" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_proyectos_presupuesto" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_proyectos_horas" />
</Declaration>
<Declaration>
<DataProperty IRI = "dp_column_software_cod_pr" />
</Declaration>
```



```

<DataProperty IRI = "dp_column_software_cod_pr" />
<Class IRI = "cls_key_software_cod_pr" />
</DataPropertyDomain>
<DataPropertyDomain>
<DataProperty IRI = "dp_column_software_tipo_sw" />
<Class IRI = "cls_table_software" />
</DataPropertyDomain>
<HasKey>
<Class IRI = "cls_table_fases" />
<ObjectProperty IRI = "op_key_cod_proy" />
<ObjectProperty IRI = "op_key_num_fase" />
</HasKey>
<HasKey>
<Class IRI = "cls_table_generan" />
<ObjectProperty IRI = "op_key_cod_proy" />
<ObjectProperty IRI = "op_key_num_fase" />
<ObjectProperty IRI = "op_key_cod_pr" />
</HasKey>
<HasKey>
<Class IRI = "cls_table_productos" />
<ObjectProperty IRI = "op_key_cod_pr" />
</HasKey>
<HasKey>
<Class IRI = "cls_table_prototipos" />
<ObjectProperty IRI = "op_key_cod_pr" />
</HasKey>
<HasKey>
<Class IRI = "cls_table_proyectos" />
<ObjectProperty IRI = "op_key_cod_proy" />
</HasKey>
<HasKey>
<Class IRI = "cls_table_software" />
<ObjectProperty IRI = "op_key_cod_pr" />
</HasKey>
<HasKey>
<Class IRI = "cls_key_fases_cod_proy" />
<DataProperty IRI = "dp_column_fases_cod_proy" />
</HasKey>
<HasKey>
<Class IRI = "cls_key_fases_num_fase" />
<DataProperty IRI = "dp_column_fases_num_fase" />
</HasKey>
<HasKey>
<Class IRI = "cls_key_generan_cod_proy" />
<DataProperty IRI = "dp_column_generan_cod_proy" />
</HasKey>
<HasKey>
<Class IRI = "cls_key_generan_num_fase" />
<DataProperty IRI = "dp_column_generan_num_fase" />
</HasKey>
<HasKey>
<Class IRI = "cls_key_generan_cod_pr" />
<DataProperty IRI = "dp_column_generan_cod_pr" />
</HasKey>
<HasKey>
<Class IRI = "cls_key_productos_cod_pr" />
<DataProperty IRI = "dp_column_productos_cod_pr" />
</HasKey>
<HasKey>
<Class IRI = "cls_key_prototipos_cod_pr" />
<DataProperty IRI = "dp_column_prototipos_cod_pr" />
</HasKey>
<HasKey>
<Class IRI = "cls_key_proyectos_cod_proy" />
<DataProperty IRI = "dp_column_proyectos_cod_proy" />
</HasKey>
<HasKey>
<Class IRI = "cls_key_proyectos_nombre" />
<DataProperty IRI = "dp_column_proyectos_nombre" />
</HasKey>
<HasKey>
<Class IRI = "cls_key_software_cod_pr" />
<DataProperty IRI = "dp_column_software_cod_pr" />
</HasKey>
<SubClassOf>
<Class IRI = "cls_key_fases_cod_proy" />
<DataAllValueFrom>
<DataProperty IRI = "dp_column_fases_cod_proy" />
<DataPropertyRange>
<DataProperty IRI = "dp_column_fases_cod_proy" />
<DataType IRI = "&xsd:String" />
</DataPropertyRange>
</DataAllValueFrom>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_key_fases_num_fase" />
<DataAllValueFrom>
<DataProperty IRI = "dp_column_fases_num_fase" />
<DataPropertyRange>
<DataProperty IRI = "dp_column_fases_num_fase" />
</DataPropertyRange>
</SubClassOf>
<DataProperty IRI = "dp_column_fases_num_fase" />
</DataPropertyRange>
<DataProperty IRI = "dp_column_fases_num_fase" />

```

```

<DataType IRI = "&xsd:int" />
</DataPropertyRange>
</DataAllValueFrom>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_fases" />
<DataAllValueFrom>
<DataProperty IRI = "dp_column_fases_nombre" />
<DataPropertyRange>
<DataProperty IRI = "dp_column_fases_nombre" />
<DataType IRI = "&xsd:String" />
</DataPropertyRange>
</DataAllValueFrom>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_fases" />
<DataAllValueFrom>
<DataProperty IRI = "dp_column_fases_fecha_i" />
<DataPropertyRange>
<DataProperty IRI = "dp_column_fases_fecha_i" />
<DataType IRI = "&xsd;Date" />
</DataPropertyRange>
</DataAllValueFrom>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_fases" />
<DataAllValueFrom>
<DataProperty IRI = "dp_column_fases_fecha_f" />
<DataPropertyRange>
<DataProperty IRI = "dp_column_fases_fecha_f" />
<DataType IRI = "&xsd;Date" />
</DataPropertyRange>
</DataAllValueFrom>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_fases" />
<DataAllValueFrom>
<DataProperty IRI = "dp_column_fases_estado" />
<DataPropertyRange>
<DataProperty IRI = "dp_column_fases_estado" />
<DataOneOf>
<Literal datatypeIRI = "http://www.w3.org/2001/XMLSchema#string">"en curso"</Literal>
<Literal datatypeIRI = "http://www.w3.org/2001/XMLSchema#string">"finalizado"</Literal>
</DataOneOf>
</DataPropertyRange>
</DataAllValueFrom>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_key_generan_cod_proy" />
<DataAllValueFrom>
<DataProperty IRI = "dp_column_generan_cod_proy" />
<DataPropertyRange>
<DataProperty IRI = "dp_column_generan_cod_proy" />
<DataType IRI = "&xsd:String" />
</DataPropertyRange>
</DataAllValueFrom>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_key_generan_num_fase" />
<DataAllValueFrom>
<DataProperty IRI = "dp_column_generan_num_fase" />
<DataPropertyRange>
<DataProperty IRI = "dp_column_generan_num_fase" />
<DataType IRI = "&xsd:int" />
</DataPropertyRange>
</DataAllValueFrom>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_key_generan_cod_pr" />
<DataAllValueFrom>
<DataProperty IRI = "dp_column_generan_cod_pr" />
<DataPropertyRange>
<DataProperty IRI = "dp_column_generan_cod_pr" />
<DataType IRI = "&xsd:String" />
</DataPropertyRange>
</DataAllValueFrom>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_key_productos_cod_pr" />
<DataAllValueFrom>
<DataProperty IRI = "dp_column_productos_cod_pr" />
<DataPropertyRange>
<DataProperty IRI = "dp_column_productos_cod_pr" />
<DataType IRI = "&xsd:String" />
</DataPropertyRange>
</DataAllValueFrom>
</SubClassOf>
<SubClassOf>

```



```

<SubClassOf>
<Class IRI = "cls_key_prototipos_cod_pr" />
<Class IRI = "cls_key_productos_cod_pr" />
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_software" />
<Class IRI = "cls_table_productos" />
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_key_software_cod_pr" />
<Class IRI = "cls_key_productos_cod_pr" />
</SubClassOf>
<SubDataPropertyOf>
<DataProperty IRI = "dp_column_fases_cod_proy" />
<DataProperty IRI = "dp_column_proyectos_cod_proy" />
</SubDataPropertyOf>
<SubDataPropertyOf>
<DataProperty IRI = "dp_column_generan_cod_proy" />
<DataProperty IRI = "dp_column_fases_cod_proy" />
</SubDataPropertyOf>
<SubDataPropertyOf>
<DataProperty IRI = "dp_column_generan_num_fase" />
<DataProperty IRI = "dp_column_fases_num_fase" />
</SubDataPropertyOf>
<SubDataPropertyOf>
<DataProperty IRI = "dp_column_generan_cod_pr" />
<DataProperty IRI = "dp_column_productos_cod_pr" />
</SubDataPropertyOf>
<SubDataPropertyOf>
<DataProperty IRI = "dp_column_prototipos_cod_pr" />
<DataProperty IRI = "dp_column_productos_cod_pr" />
</SubDataPropertyOf>
<SubDataPropertyOf>
<DataProperty IRI = "dp_column_software_cod_pr" />
<DataProperty IRI = "dp_column_productos_cod_pr" />
</SubDataPropertyOf>
<InverseFunctionalObjectProperty>
<ObjectProperty IRI = "op_key_nombre" />
</InverseFunctionalObjectProperty>
<DataPropertyRange>
<DataProperty IRI = "dp_column_fases_cod_proy" />
<DataType IRI = "&xsd;String" />
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_fases_num_fase" />
<DataType IRI = "&xsd:int" />
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_fases_nombre" />
<DataType IRI = "&xsd;String" />
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_fases_fecha_i" />
<DataType IRI = "&xsd;Date" />
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_fases_fecha_f" />
<DataType IRI = "&xsd;Date" />
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_generan_cod_proy" />
<DataType IRI = "&xsd;String" />
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_generan_num_fase" />
<DataType IRI = "&xsd:int" />
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_generan_cod_pr" />
<DataType IRI = "&xsd;String" />
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_productos_cod_pr" />
<DataType IRI = "&xsd;String" />
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_productos_nombre" />
<DataType IRI = "&xsd;String" />
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_productos_descripcion" />
<DataType IRI = "&xsd;String" />
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_prototipos_cod_pr" />
<DataType IRI = "&xsd;String" />
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_prototipos_version" />
<DataType IRI = "&xsd;String" />
</DataPropertyRange>
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_prototipos_ubicacion" />
<DataType IRI = "&xsd;String" />
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_proyectos_cod_proy" />
<DataType IRI = "&xsd;String" />
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_proyectos_nombre" />
<DataType IRI = "&xsd;String" />
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_proyectos_descripcion" />
<DataType IRI = "&xsd;String" />
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_proyectos_fecha_i" />
<DataType IRI = "&xsd;Date" />
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_proyectos_fecha_f" />
<DataType IRI = "&xsd;Date" />
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_proyectos_presupuesto" />
<DataType IRI = "&xsd;Float" />
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_proyectos_horas" />
<DataType IRI = "&xsd:int" />
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_software_cod_pr" />
<DataType IRI = "&xsd;String" />
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_software_tipo_sw" />
<DataType IRI = "&xsd;String" />
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_fases_estado" />
<DataOneOf>
<Literal datatypeIRI = "http://www.w3.org/2001/XMLSchema#string">"en curso"</Literal>
<Literal datatypeIRI = "http://www.w3.org/2001/XMLSchema#string">"finalizado"</Literal>
</DataOneOf>
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_productos_estado" />
<DataOneOf>
<Literal datatypeIRI = "http://www.w3.org/2001/XMLSchema#string">"si"</Literal>
<Literal datatypeIRI = "http://www.w3.org/2001/XMLSchema#string">"no"</Literal>
</DataOneOf>
</DataPropertyRange>
<DataPropertyRange>
<DataProperty IRI = "dp_column_productos_tipo" />
<DataOneOf>
<Literal datatypeIRI = "http://www.w3.org/2001/XMLSchema#string">"software"</Literal>
<Literal datatypeIRI = "http://www.w3.org/2001/XMLSchema#string">"prototipo"</Literal>
<Literal datatypeIRI = "http://www.w3.org/2001/XMLSchema#string">"informe tecnico"</Literal>
</DataOneOf>
</DataPropertyRange>
<SubClassOf>
<Class IRI = "cls_table_fases" />
<ObjectExactCardinality cardinality = "1" >
<ObjectProperty IRI = "op_key_cod_proy" />
<Class IRI = "cls_key_fases_cod_proy" />
</ObjectExactCardinality>
<SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_fases" />
<ObjectExactCardinality cardinality = "1" >
<ObjectProperty IRI = "op_key_num_fase" />
<Class IRI = "cls_key_fases_num_fase" />
</ObjectExactCardinality>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_generan" />
<ObjectExactCardinality cardinality = "1" >
<ObjectProperty IRI = "op_key_cod_proy" />
<Class IRI = "cls_key_generan_cod_proy" />

```

```

</ObjectExactCardinality>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_generan" />
<ObjectExactCardinality cardinality = "1" >
<ObjectProperty IRI = "op_key_num_fase" />
<Class IRI = "cls_key_generan_num_fase" />
</ObjectExactCardinality>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_generan" />
<ObjectExactCardinality cardinality = "1" >
<ObjectProperty IRI = "op_key_cod_pr" />
<Class IRI = "cls_key_generan_cod_pr" />
</ObjectExactCardinality>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_productos" />
<ObjectExactCardinality cardinality = "1" >
<ObjectProperty IRI = "op_key_cod_pr" />
<Class IRI = "cls_key_productos_cod_pr" />
</ObjectExactCardinality>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_prototipos" />
<ObjectExactCardinality cardinality = "1" >
<ObjectProperty IRI = "op_key_cod_pr" />
<Class IRI = "cls_key_prototipos_cod_pr" />
</ObjectExactCardinality>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_proyectos" />
<ObjectExactCardinality cardinality = "1" >
<ObjectProperty IRI = "op_key_cod_proy" />
<Class IRI = "cls_key_proyectos_cod_proy" />
</ObjectExactCardinality>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_proyectos" />

```

```

<ObjectExactCardinality cardinality = "1" >
<ObjectProperty IRI = "op_key_nombre" />
<Class IRI = "cls_key_proyectos_nombre" />
</ObjectExactCardinality>
</SubClassOf>
<SubClassOf>
<Class IRI = "cls_table_software" />
<ObjectExactCardinality cardinality = "1" >
<ObjectProperty IRI = "op_key_cod_pr" />
<Class IRI = "cls_key_software_cod_pr" />
</ObjectExactCardinality>
</SubClassOf>
<SubObjectPropertyOf>
<ObjectProperty IRI = "op_key_cod_pr" />
<ObjectProperty IRI = "productos" />
</SubObjectPropertyOf>
<SubObjectPropertyOf>
<ObjectProperty IRI = "op_key_cod_pr" />
<ObjectProperty IRI = "productos" />
</SubObjectPropertyOf>
<DisjointClasses>
<Class IRI = "cls_table_fases" />
<Class IRI = "cls_key_fases_cod_proy" />
<Class IRI = "cls_key_fases_num_fase" />
</DisjointClasses>
<DisjointClasses>
<Class IRI = "cls_table_productos" />
<Class IRI = "cls_key_productos_cod_pr" />
</DisjointClasses>
<DisjointClasses>
<Class IRI = "cls_table_proyectos" />
<Class IRI = "cls_key_proyectos_cod_proy" />
<Class IRI = "cls_key_proyectos_nombre" />
</DisjointClasses>
<DisjointClasses>
<Class IRI = "prototipos" />
<Class IRI = "software" />
</DisjointClasses>
</Ontology>

```

4. Proyectos de investigación.

OWL

```

<!DOCTYPE Ontology [
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
]>

```

```

<Ontology
xml:base="http://example.com/owl/proyectos-investigacion/"
ontologyIRI="http://example.com/owl/proyectos-investigacion"
xmlns="http://www.w3.org/2002/07/owl#">
<Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#" />

```

```

<Declaration>
<Class IRI = "cls_table_congresos" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_congresos_proyecto" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_congresos_num_publicacion" />
</Declaration>
<Declaration>
<Class IRI = "cls_table_doctores" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_doctores_nombre" />
</Declaration>
<Declaration>
<Class IRI = "cls_table_no_doctores" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_no_doctores_nombre" />
</Declaration>
<Declaration>
<Class IRI = "cls_table_p_investigacion" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_p_investigacion_cod_ref" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_p_investigacion_inv_ppal" />

```

```

</Declaration>
<Declaration>
<Class IRI = "cls_table_participan" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_participan_profesor" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_participan_proyecto_inv" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_participan_fecha_i" />
</Declaration>
<Declaration>
<Class IRI = "cls_table_profesores" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_profesores_nombre" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_profesores_telefono" />
</Declaration>
<Declaration>
<Class IRI = "cls_table_publicaciones" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_publicaciones_proyecto" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_publicaciones_num" />
</Declaration>
<Declaration>
<Class IRI = "cls_table_revistas" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_revistas_proyecto" />
</Declaration>
<Declaration>
<Class IRI = "cls_key_revistas_num_publicacion" />
</Declaration>

```



```

<DisjointClasses>
<Class IRI = "cls_table_profesores" />
<Class IRI = "cls_key_profesores_nombre" />
<Class IRI = "cls_key_profesores_telefono" />
</DisjointClasses>
<DisjointClasses>
<Class IRI = "cls_table_publicaciones" />
<Class IRI = "cls_key_publicaciones_proyecto" />
<Class IRI = "cls_key_publicaciones_num" />
</DisjointClasses>
<DisjointClasses>
<Class IRI = "cls_table_supervisan" />
<Class IRI = "cls_key_supervisan_no_doctor" />
<Class IRI = "cls_key_supervisan_f_ini" />
</DisjointClasses>
<DisjointUnion>
<Class IRI = "profesores" />
<Class IRI = "doctores" />
<Class IRI = "no_doctores" />
</DisjointUnion>
</Ontology>

```

ANEXO 7. HISTORIAS DE USUARIO

A continuación se muestran las historias de usuario elaboradas en la fase de exploración, durante la implementación del prototipo propuesto.

Historia de Usuario		
Id: HU-1	Nombre: Mostrar relaciones de herencia	
Descripción corta El sistema brindará la opción de identificar y mostrar las relaciones de herencia presentes en el esquema de base de datos que se desea transformar.		
Descripción detallada El sistema mostrará un formulario donde se deben ingresar los datos necesarios para establecer una conexión remota con la base de datos que se desea transformar. El formulario solicitará la URL donde está alojada la base de datos, el nombre de la base de datos, la cuenta de usuario y la contraseña de la base de datos. Una vez el usuario llene el formulario y envíe la información, el sistema le mostrará, si existen, las relaciones de herencia identificadas en el esquema de base de datos relacionando cada clase padre con sus respectivas clases hijas.		
Excepciones El sistema no pudo establecer una conexión con la base de datos.		
Observaciones Ninguna.		
Elaborado por	Firma	Fecha
Carlos Bravo		Septiembre 11 de 2013
Carlos Suarez		Septiembre 11 de 2013

Figura 13. Historia de usuario - Mostrar relaciones de herencia

Historia de Usuario		
Id: HU-2	Nombre: Transformar RDBS a OWL2	
Descripción corta El sistema mostrará la ontología que resulte de la transformación del esquema de base de datos relacional.		
Descripción detallada El sistema mostrará un formulario donde se deben ingresar los datos necesarios para establecer una conexión remota con la base de datos que se desea transformar e información que se incluirá en la ontología. El formulario solicitará la URL donde está alojada la base de datos, el nombre de la base de datos, la cuenta de usuario y la contraseña de la base de datos, la base xml y el IRI de la ontología que se generará. Además, el usuario seleccionará el tipo de herencia que le corresponde a cada relación de herencia mostrada por el sistema. Una vez el usuario llene el formulario y envíe la información, el sistema mostrará la ontología que se obtuvo como resultado de la transformación.		
Excepciones El sistema no pudo establecer una conexión con la base de datos.		
Observaciones Ninguna.		
Elaborado por	Firma	Fecha
Carlos Bravo		Septiembre 11 de 2013
Carlos Suarez		Septiembre 11 de 2013

Figura 14. Historia de usuario - Transformar RDBS a OWL2

BIBLIOGRAFÍA

- Astrova, I. (2009). Rules for Mapping SQL Relational Databases to OWL Ontologies. *Springer Link*, 415–424.
- Chen, A., Liu, L., & Shang, J. (2012). A Hybrid Strategy to Construct Scientific Instrument Ontology from Relational Database Model. *2012 International Conference on Computer Distributed Control and Intelligent Environmental Monitoring*, 25–33. doi:10.1109/CDCIEM.2012.14
- Choi, J. W., & Kim, M. H. (2012). Generating OWL Ontology from Relational Database. *2012 Third FTRA International Conference on Mobile, Ubiquitous, and Intelligent Computing*, 53–59. doi:10.1109/MUSIC.2012.17
- Hazber, M., Yang, J., & Jin, Q. (2010). Towards Integration Rules of Mapping from Relational Databases to Semantic Web Ontology. *2010 International Conference on Web Information Systems and Mining*, 335–339. doi:10.1109/WISM.2010.21
- Sequeda, J. F., Tirmizi, S. H., Corcho, O., & Miranker, D. P. (2011). *Survey of directly mapping SQL databases to the Semantic Web*. *The Knowledge Engineering Review* (Vol. 26, pp. 445–486). doi:10.1017/S0269888911000208