



ugr

Universidad
de Granada

DESARROLLO DE SISTEMAS DE SOFTWARE BASADOS EN
COMPONENTES Y SERVICIOS
MÁSTER EN INGENIERÍA INFORMÁTICA

PRÁCTICA 4

Creación de una ontología de libros con Protégé

Autor:

Pablo Valenzuela Álvarez (pvalenzuela@correo.ugr.es)



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE TELECOMUNICACIÓN

Granada, 20 de diciembre de 2024

Índice

1. Estructura	3
1.1. Jerarquía de clases	3
1.2. Propiedades de los objetos	4
1.3. Propiedades de datos	6
2. Restricciones	7
2.1. LibroPopular y LibroDeLujo	7
2.2. LibroCienciaFiccionBarato	7
2.3. LibroEspecifico	7
3. Pruebas realizadas	8
3.1. Rellenando propiedades de libros	8
3.2. Datos inferidos en LibroPopular y LibroDeLujo	10
3.3. Datos inferidos en LibroCienciaFiccionBarato	11
3.4. Probando inconsistencias	12
4. Contenido adicional	13
4.1. Género Misterio	13
4.2. Clase Colección	13
4.2.1. Propiedad perteneceColeccion y esColeccionDe	14
5. Repositorio GitHub	15

Índice de figuras

1.	Estructura de la ontología.	3
2.	Gráfico con la estructura de la ontología.	4
3.	Lista de propiedades de objetos.	4
4.	Detalles de la propiedad escritoPor.	5
5.	Detalles de la propiedad haEscrito.	5
6.	Lista de propiedades de datos.	6
7.	Detalles de la propiedad TienePrecio.	6
8.	Detalles de la clase LibroPopular.	7
9.	Detalles de la clase LibroCienciaFiccionBarato.	7
10.	Detalles de la clase LibroEspecifico.	7
11.	Lista de individuos de Ebooks.	8
12.	Detalles del libro Las Dos Torres.	8
13.	Clase inferida del libro Las Dos Torres.	8
14.	Libros escritos inferidos por J.R.R. Tolkien.	9
15.	Libros con género Fantasia inferidos.	9
16.	Libros de la editorial Minotauro inferidos.	9
17.	Lista inferida de los libros populares.	10
18.	Lista inferida de los libros de lujo.	10
19.	Lista inferida de los libros de ciencia ficción baratos.	11
20.	Clase Libro con sus subclases inferidas.	11
21.	Detalles del libro inconsistente.	12
22.	Explicación de la inconsistencia.	12
23.	Libros del género Misterio.	13
24.	Lista de colecciones.	13
25.	Detalles de la propiedad perteneceColeccion.	14
26.	Libros de la colección ElSeñorDeLosAnillos.	14

1. Estructura

En esta práctica hemos de crear una ontología de libros. Para ello debemos hacer uso de una jerarquía de clases y crear relaciones entre ellas que tengan sentido y sean consistentes usando propiedades de objetos y datos.

El programa que vamos a usar durante la realización de esta práctica es **Protégé**, que es un editor de ontologías libre de código abierto y un sistema de adquisición de conocimiento.

1.1. Jerarquía de clases

La jerarquía que hemos definido es la que se muestra en la figura 1 y en forma gráfica en la figura 2.

Para las clases primarias: *Autor*, *Coleccion*, *Editorial*, *Genero* y *Libro*; se ha creado una **disjunción**, de tal forma que una instancia solo pueda pertenecer a una de estas clases. También se ha creado otra disjunción entre las subclases de los géneros.

Algunas de las subclases contenidas en la clase *Libro* son definidas y se explicarán al detalle en la sección 2.

Nota: Las clases *Coleccion* y *Misterio* son contenido extra y se explicarán en la sección 4.

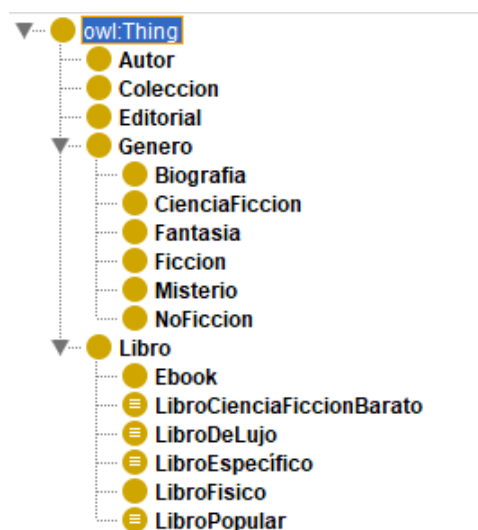


Figura 1: Estructura de la ontología.

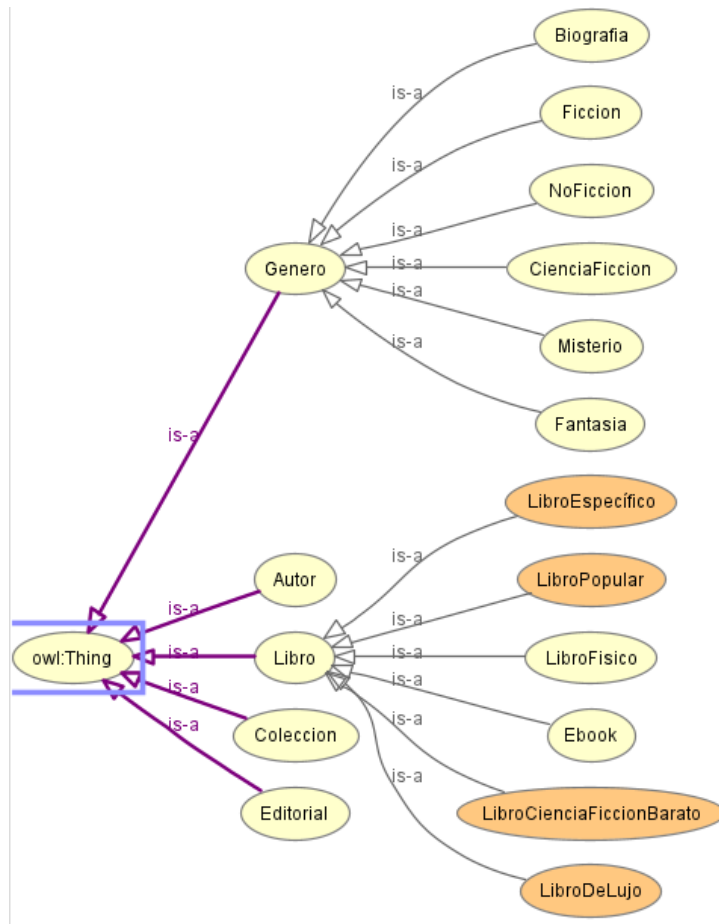


Figura 2: Gráfico con la estructura de la ontología.

1.2. Propiedades de los objetos

Las propiedades de los objetos definen como se relacionan las clases de la ontología. La lista de la figura 3 muestra todas las que se han establecido.

Nota: Las propiedades *esColeccionDe* y *perteneceColeccion* son contenido extra y se explicarán en la sección 4.

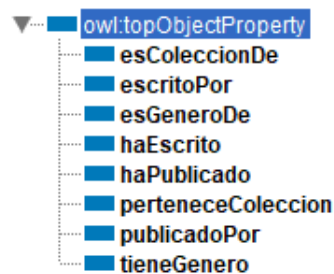


Figura 3: Lista de propiedades de objetos.

Para cada propiedad se ha definido su inversa, como por ejemplo se puede ver en las figuras 4 y 5. Para el caso de la propiedad *escritoPor*, su **dominio** es *Libro* y su **rango** *Autor*, siendo la

inversa (*haEscrito*) al contrario.



Figura 4: Detalles de la propiedad escritoPor.

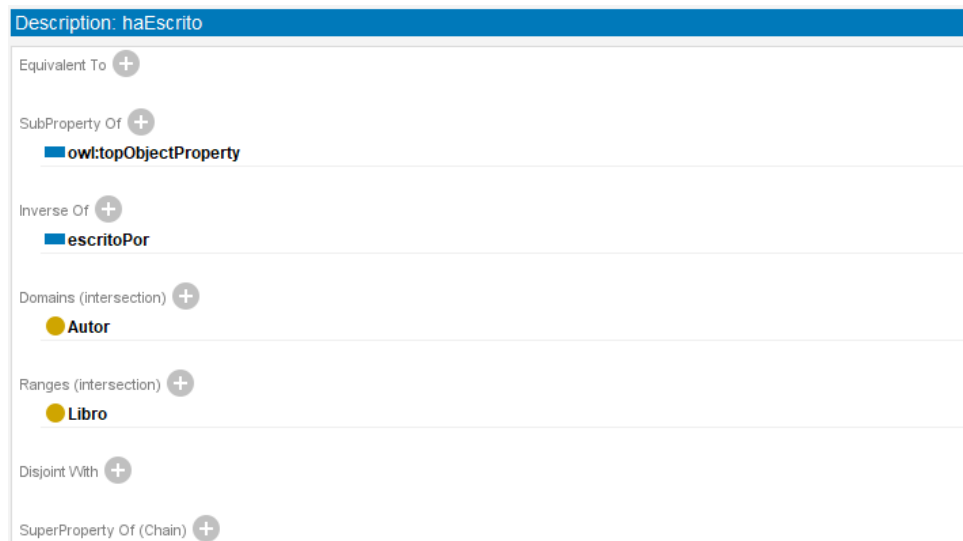


Figura 5: Detalles de la propiedad haEscrito.

Los detalles de las demás propiedades vistas en la figura 3 son:

- **publicadoPor:**
 - Dominio: *Libro*
 - Rango: *Editorial*
 - Inversa: *haPublicado*
- **tieneGenero:**
 - Dominio: *Libro*
 - Rango: *Genero*
 - Inversa: *esGeneroDe*

1.3. Propiedades de datos

Las propiedades de datos asignan valores “literales” a propiedades que puedan tener los objetos. Por ejemplo, la propiedad *tienePrecio* da un valor decimal a un libro, que es correspondiente a su precio (ver figura 7).

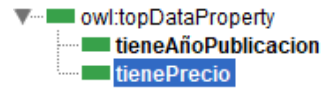


Figura 6: Lista de propiedades de datos.

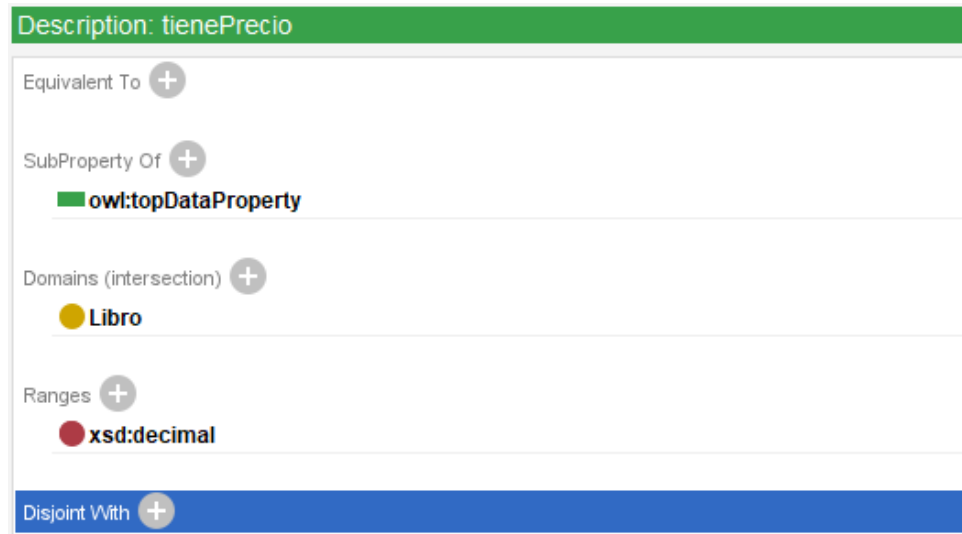


Figura 7: Detalles de la propiedad TienePrecio.

Y la propiedad *tieneAñoPublicacion* da un valor entero relativo al año de publicación del libro.

2. Restricciones

En esta sección veremos las restricciones que hemos aplicado para las clases definidas *LibroPopular*, *LibroDeLujo*, *LibroCienciaFiccionBarato* y *LibroEspecifico*.

2.1. LibroPopular y LibroDeLujo

La clase *LibroPopular* integra a los libros con un precio menor a 20 euros. En la figura 8 se puede observar la forma que tenemos para definir dicha restricción.

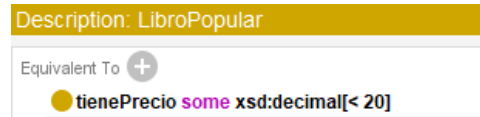


Figura 8: Detalles de la clase LibroPopular.

En el caso de la clase *LibroDeLujo* tendremos que aplicar una fórmula similar:

tienePrecio some xsd:decimal[≥ 50]

2.2. LibroCienciaFiccionBarato

Esta clase engloba a los libros del género ciencia ficción que cuesten menos de 20 euros. La fórmula equivalente tiene que considerar estas dos restricciones, por eso debemos hacer dos condiciones (ver figura 9) y unir las con el axioma **AND**.

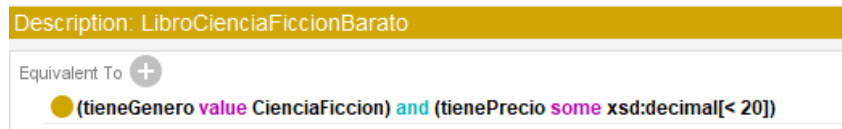


Figura 9: Detalles de la clase LibroCienciaFiccionBarato.

2.3. LibroEspecifico

Esta es una clase con cierre que comprende los libros que poseen un solo género. Para esta ocasión el axioma que debemos usar es **EXACTLY** seguido del número de elementos, en nuestro caso 1.



Figura 10: Detalles de la clase LibroEspecifico.

3. Pruebas realizadas

En la sección de pruebas veremos lo consistente que es nuestra ontología. Para ello necesitamos introducir individuos en las clases tal y como se puede ver en la figura 11.



Figura 11: Lista de individuos de Ebooks.

También hemos introducido individuos en todas las clases salvo en las que son definidas (*LibroPopular*, *LibroDeLujo*, *LibroEspecifico* y *LibroCienciaFiccionBarato*). La razón a esto, es que las clases definidas se rellenarán automáticamente al ejecutar el **razonador**, siempre y cuando hayamos definido bien las propiedades de los libros.

3.1. Rellenando propiedades de libros

Cuando rellenamos las propiedades de objeto y de datos de los libros como en el ejemplo de la figura 12 y ejecutamos el **razonador**, podemos tener clases inferidas como la que se observa en la figura 13. En este caso, se cumple la condición para que el libro sea considerado *LibroPopular* y de forma automática Protégé le asigna esa clase.



Figura 12: Detalles del libro Las Dos Torres.

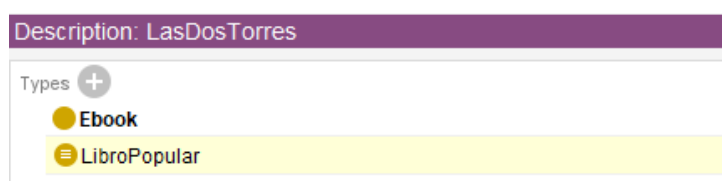


Figura 13: Clase inferida del libro Las Dos Torres.

A su vez, si miramos al autor o a la editorial o al género del ejemplo de la figura 12 podemos comprobar si se ha inferido bien este libro. Y como se muestra en las figuras 14, 15 y 16, se ha hecho correctamente.

Property assertions: JRRTolkien	
Object property assertions +	
■ haEscrito	LasDosTorres
■ haEscrito	ElRetornoDelRey
■ haEscrito	LaComunidadDelAnillo

Figura 14: Libros escritos inferidos por J.R.R. Tolkien.

Property assertions: Fantasia	
Object property assertions +	
■ esGeneroDe	LaCiudadDeLasBestias
■ esGeneroDe	ElBosqueDeLosPigmeos
■ esGeneroDe	LasDosTorres
■ esGeneroDe	ElRetornoDelRey
■ esGeneroDe	ParqueJurásico
■ esGeneroDe	Neuromante
■ esGeneroDe	LaComunidadDelAnillo

Figura 15: Libros con género Fantasia inferidos.

Property assertions: Minotauro	
Object property assertions +	
■ haPublicado	LasDosTorres
■ haPublicado	ElRetornoDelRey
■ haPublicado	Neuromante
■ haPublicado	LaComunidadDelAnillo

Figura 16: Libros de la editorial Minotauro inferidos.

3.2. Datos inferidos en LibroPopular y LibroDeLujo

La figura 17 muestra los libros que cumplen la propiedad necesaria para ser instancias de esta clase (Libros que cuestan menos de 20 euros).

Description: LibroPopular	
Instances	+
◆	ElBosqueDeLosPigmeos
◆	ElDiamanteDeJerusalen
◆	ElDragonRojo
◆	ElRetornoDelRey
◆	ElSilencioDeLosInocentes
◆	LaCiudadDeLasBestias
◆	LaComunidadDelAnillo
◆	LasDosTorres
◆	MuerteEnLaFenice
◆	Patrimonio
◆	PrimaveraSilenciosa
◆	TreceRosasRojas

Figura 17: Lista inferida de los libros populares.

O más de 50 euros.

Description: LibroDeLujo	
◆	ElRabino
◆	LouisVuitton

Figura 18: Lista inferida de los libros de lujo.

3.3. Datos inferidos en LibroCienciaFiccionBarato

La figura 19 muestra los libros que cumplen la propiedad necesaria para ser instancias de esta clase: ser del género ciencia ficción y no costar más de 20 euros.

Notamos también que ha agregado la clase *LibroPopular* como si esta clase fuese su subclase.

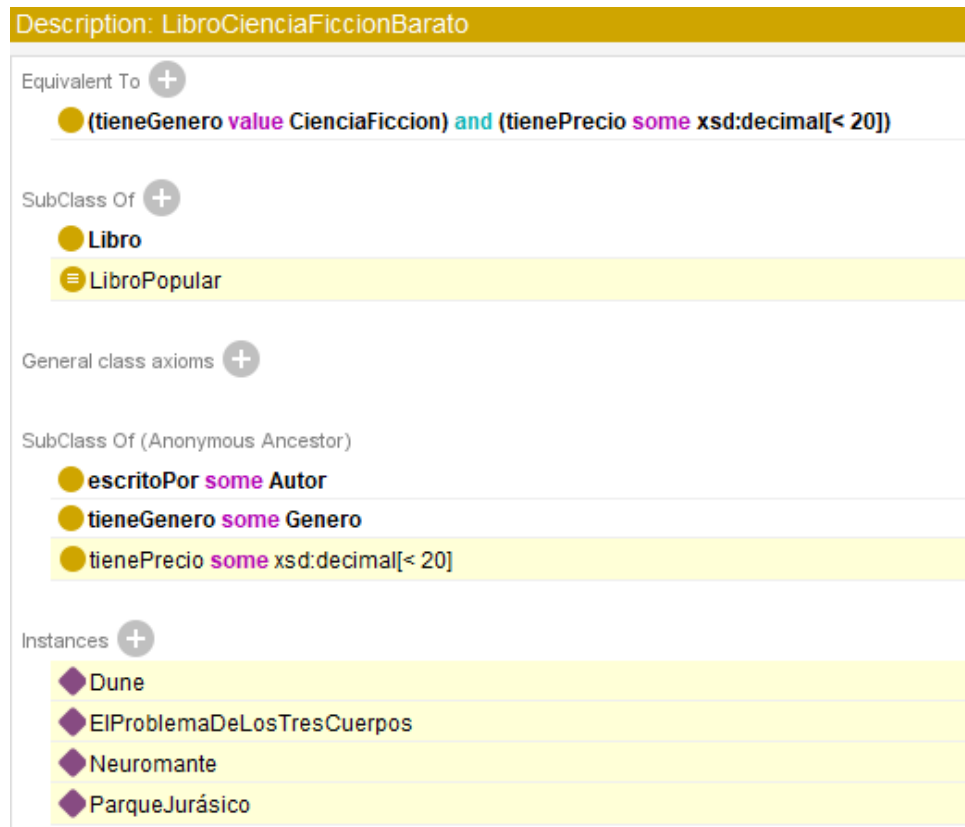


Figura 19: Lista inferida de los libros de ciencia ficción baratos.

Esto es debido a que una de las condiciones de la clase *LibroCienciaFiccionBarato* es la misma condición de precio que tiene *LibroPopular*. La gráfica de la figura 20 nos muestra como Protégé ha deducido que esta clase es una subclase de *LibroPopular*, lo cual tiene sentido.

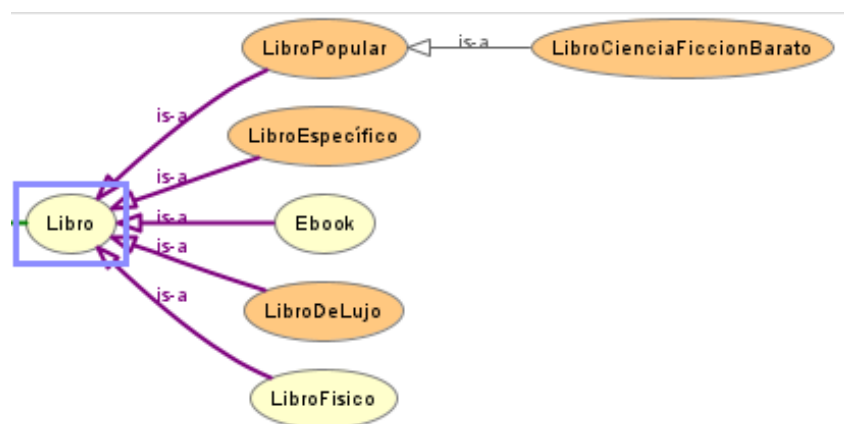


Figura 20: Clase Libro con sus subclases inferidas.

3.4. Probando inconsistencias

Vamos a añadir un libro inconsistente para probar esta restricción. Para ello, debemos crear un individuo de la clase *Libro* y añadirle los géneros *Ficcion* y *NoFiccion* como se ve en la figura 21.

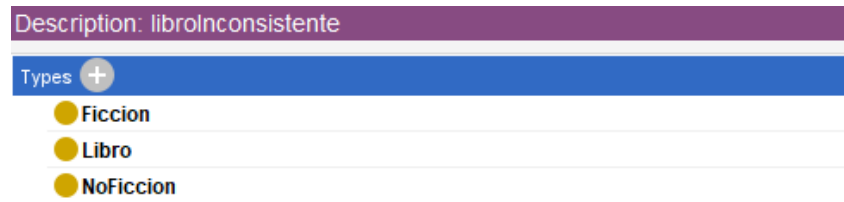


Figura 21: Detalles del libro inconsistente.

Hemos definido una **disjunción** entre estos dos géneros, por lo que al ejecutar el razonador nos muestra el aviso de la figura 22. Se aplica correctamente la restricción

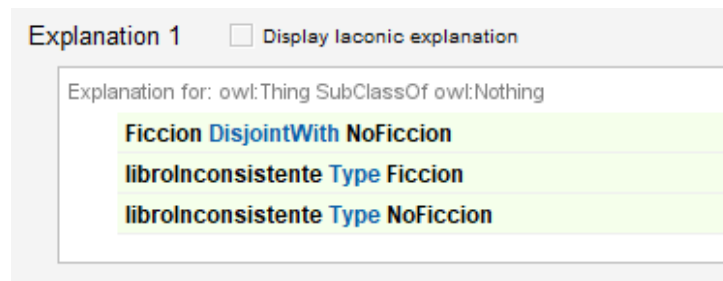


Figura 22: Explicación de la inconsistencia.

4. Contenido adicional

Como parte opcional, hemos añadido un par de clases y una propiedad como contenido adicional de la práctica.

4.1. Género Misterio

Se ha añadido este género literario y se han creado varios individuos con él a través de la propiedad *tieneGenero*. En la figura 23 podemos ver como se han inferido correctamente.



Figura 23: Libros del género Misterio.

4.2. Clase Colección

Esta clase se ha implementado para reunir los libros que son miembros de una saga o colección. Hemos definido dos individuos (ver figura 24) para usarlos con las propiedades adicionales definidas en el siguiente punto.



Figura 24: Lista de colecciones.

4.2.1. Propiedad perteneceColeccion y esColeccionDe

Estas propiedades unen las clases *Coleccion* y *Libro*. El uso de una de ellas (*perteneceColeccion*), se ha podido ver en la figura 12. Basicamente es la encargada de reunir los libros de una colección.

En los detalles de esta propiedad (ver figura 25), podemos ver como su dominio es *Libro*, su rango es *Coleccion* y su inversa es *EsColeccionDe*. Obviamente, su inversa tiene el dominio y rango intercambiados.



Figura 25: Detalles de la propiedad perteneceColeccion.

Si comprobamos una colección, por ejemplo “ElSeñorDeLosAnillos”, podremos los libros que pertenecen a esa colección (ver figura 26). Como podemos observar, están bien inferidos por Protégé.

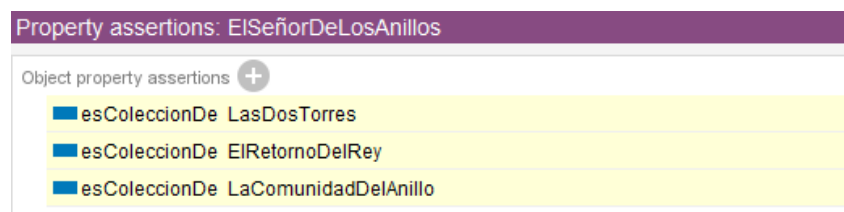


Figura 26: Libros de la colección ElSeñorDeLosAnillos.

5. Repositorio GitHub

Acceso al repositorio en GitHub de la asignatura: <https://github.com/Valenz23/DSS>