
Ingenieria del Conocimiento – 61GIIN

Actividad 3 – Portafolio

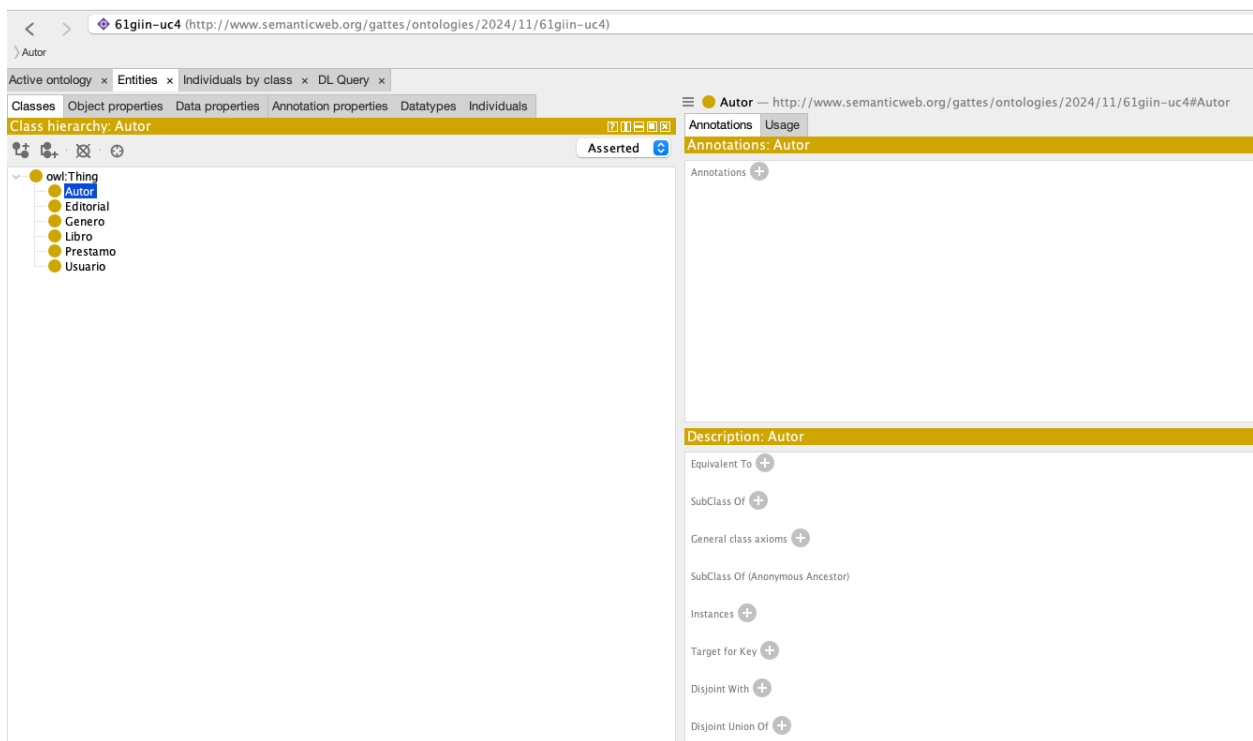
Gagliardo Miguel Angel

25 de Diciembre de 2024

Partiendo de la Ontología creada en la UC2 de organización y gestión de libros, se generará una Ontología ahora de manera práctica en **protégé** y a continuación mostraremos el paso a paso de los elementos más importantes

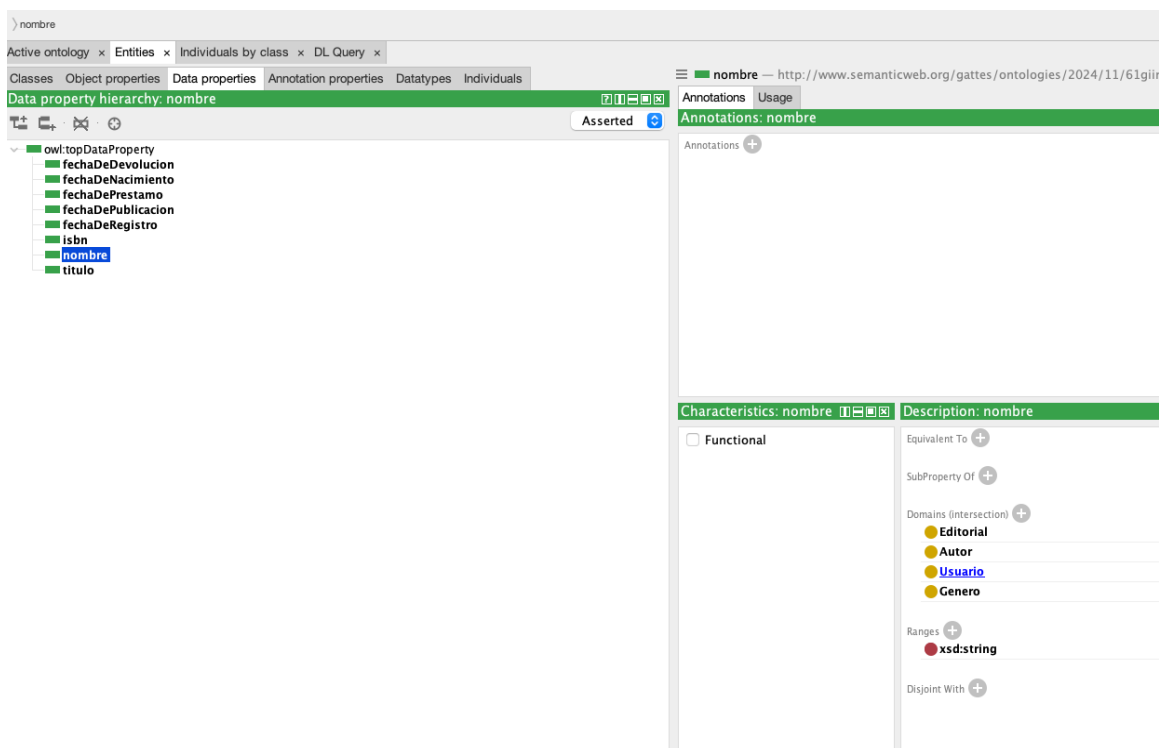
Clases

Lo primero es **crear las clases** mencionadas en la UC2: **Autor**, **Editorial**, **Libro**, **Género**, **Préstamo** y **Usuario**.



Propiedades

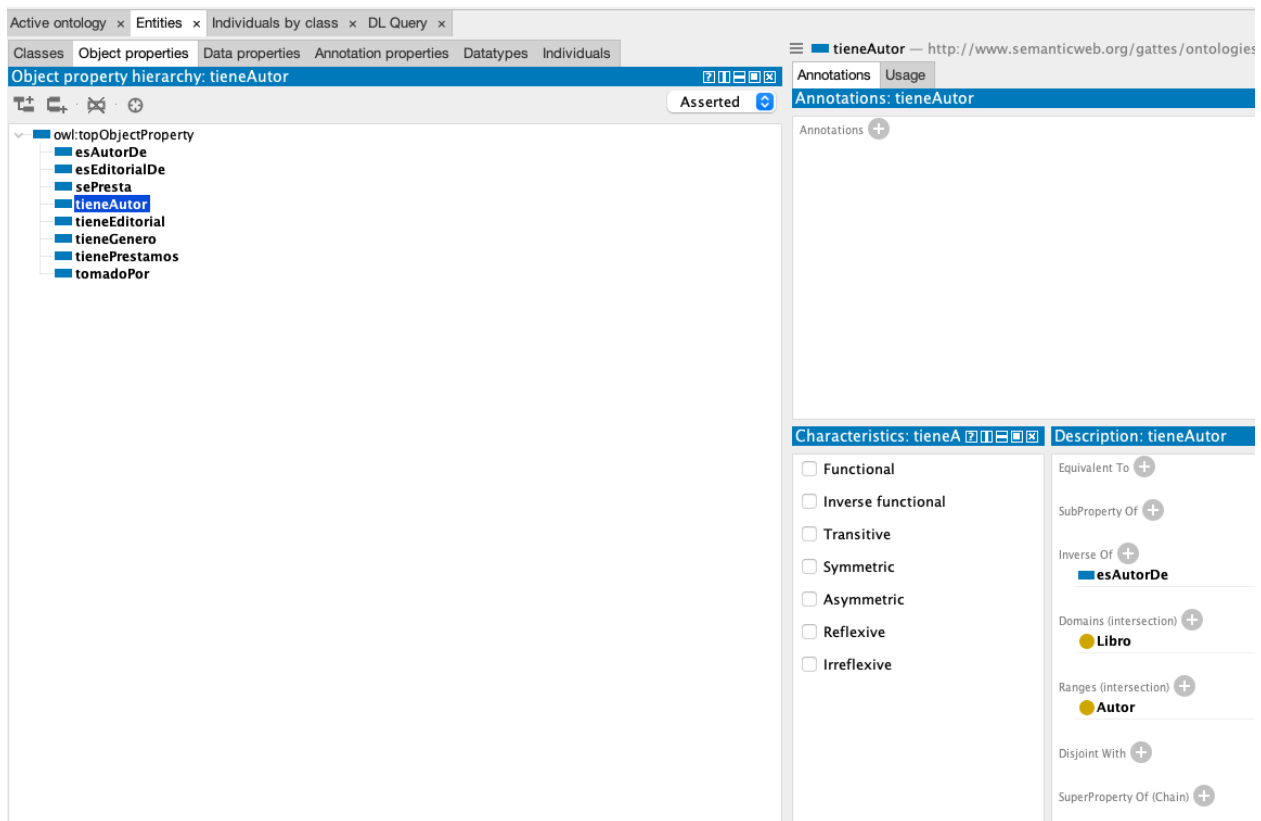
Luego, **para cada una de estas clases, vamos a definir sus propiedades**. Como podemos ver en detalle, hay algunas propiedades compartidas como por ejemplo **nombre**, dado que un Libro, un Autor, una Editorial y un Usuario, todos poseen una propiedad **nombre** de tipo **xsd:string** que los va a identificar, por tanto podemos tomar como “dominio” de la misma todas estas clases. En otros casos como **fechaDePublicacion (Libro)**, **fechaDeRegistro (Usuario)** o **isbn (Libro)** son unívocas a cada clase. Si, es verdad que para fecha podríamos haber utilizado una sola, pero poner **fecha** como propiedad sólo no es muy clara de por sí, es una fecha de publicación (de un libro)? Es una fecha de devolución (de un préstamo)? Por eso he preferido indicar cada propiedad **de tipo fecha** de forma única.



The screenshot shows the Protégé ontology editor interface. The left pane displays the 'Data property hierarchy: nombre' with a tree structure including properties like `fechaDeDevolucion`, `fechaDeNacimiento`, `fechaDePrestamo`, `fechaDePublicacion`, `fechaDeRegistro`, `isbn`, `nombre` (highlighted), and `titulo`. The right pane shows the configuration for the `nombre` property. The 'Annotations' tab is active, showing 'Annotations: nombre'. Below this, the 'Characteristics: nombre' section includes a 'Functional' checkbox. The 'Description: nombre' section shows 'Equivalent To', 'SubProperty Of', and 'Domains (intersection)' (containing `Editorial`, `Autor`, `Usuario`, and `Genero`). The 'Ranges' section shows `xsd:string`.

Relaciones

Por otro lado, de la misma manera que definimos propiedades lo hacemos con las **relaciones entre clases**. Como se ve a continuación en este ejemplo, la clase **Libro** tiene una relación con **autor** (**tieneAutor**) y viceversa (**esAutorDe**). Esto en la ontología serviría para correlacionar un autor con uno o más libros y un libro con su autor. Esto mismo pasa para casi todas las otras relaciones.



The screenshot shows the Protégé ontology editor interface. The top menu bar includes 'Active ontology', 'Entities', 'Individuals by class', and 'DL Query'. The left sidebar displays the 'Object property hierarchy: tieneAutor' with a tree view showing properties like 'esAutorDe', 'esEditorialDe', 'sePresta', 'tieneAutor', 'tieneEditorial', 'tieneGenero', 'tienePrestamos', and 'tomadoPor'. The main workspace shows the 'tieneAutor' property configuration. The 'Annotations' tab is active, displaying 'Annotations: tieneAutor'. The 'Characteristics' section on the right includes checkboxes for 'Functional', 'Inverse functional', 'Transitive', 'Symmetric', 'Asymmetric', 'Reflexive', and 'Irreflexive'. The 'Description' section on the right shows the property's domain and range, with 'Libro' as the domain and 'Autor' as the range, both indicated by yellow circles.

Active ontology x

Entities x

Individuals by class x

DL Query x

Classes

Object properties

Data properties

Annotation properties

Datatypes

Individuals

esAutorDe — http://www.semanticweb.org/gattes/ontologies/2011/09/2011-09-20_01-01-01.owl

Annotations Usage

Annotations: esAutorDe

Annotations +

Object property hierarchy: esAutorDe

Asserted

owl:topObjectProperty

esAutorDe

esEditorialDe

sePresta

tieneAutor

tieneEditorial

tieneGenero

tienePrestamos

tomadoPor

Characteristics: esAutorDe

☐ Functional
 ☐ Inverse functional
 ☐ Transitive
 ☐ Symmetric
 ☐ Asymmetric
 ☐ Reflexive
 ☐ Irreflexive

Description: esAutorDe

Equivalent To +

SubProperty Of +

Inverse Of +

tieneAutor

Domains (intersection) +

Autor

Ranges (intersection) +

Libro

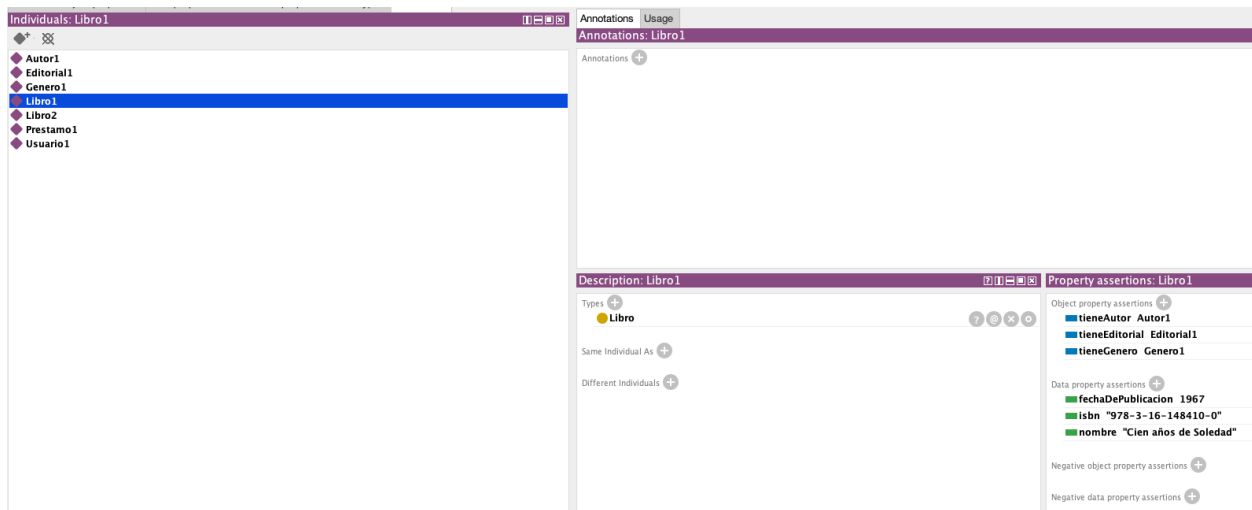
Disjoint With +

SuperProperty Of (Chain) +

Instancias de Clases

Por último, nos queda crear **instancias de clases**: Un Libro, un Autor (que escribe el libro), una Editorial (que lo publica), etc.

Como se ve a continuación, he creado las instancias **Libro1** y **Libro2** que pertenecen a los libros con **nombre** “Cien años de Soledad” y “Relato de un Náufrago”, respectivamente. Ambos enlazadas a la misma instancia de **autor** (**tieneAutor**) que es **Autor1**, con **nombre** **Gabriel García Marquez** y publicados por la misma **Editorial** (**tieneEditorial**) que es la instancia **Editorial1**, por último, ambos pertenecen al mismo **genero** (**tieneGenero**) **Genero1**, que es de tipo “**novela**”



The screenshot displays a Semantic Web editor interface. On the left, the 'Individuals: Libro1' panel lists several instances: Autor1, Editorial1, Genero1, Libro1 (selected), Libro2, Prestamo1, and Usuario1. The main area is divided into three tabs: 'Annotations', 'Usage', and 'Description: Libro1'. The 'Annotations' tab is active, showing a list of annotations for 'Libro1'. The 'Description: Libro1' tab shows the class hierarchy, with 'Libro' selected. The 'Property assertions: Libro1' tab shows the following assertions:

- Object property assertions:
 - tieneAutor Autor1
 - tieneEditorial Editorial1
 - tieneGenero Genero1
- Data property assertions:
 - fechaDePublicacion 1967
 - isbn "978-3-16-148410-0"
 - nombre "Cien años de Soledad"
- Negative object property assertions: (empty)
- Negative data property assertions: (empty)

CONCLUSIONES

Protégé es una herramienta poderosa y accesible para la creación de ontologías, que permite modelar y visualizar datos semánticos de manera intuitiva. En este caso, al crear una ontología sobre la gestión de libros y préstamos, se han utilizado diversas funcionalidades de Protégé, como la definición de clases, propiedades de objetos y propiedades de datos, así como la creación de individuos (Autor, Libro, Editorial). La interfaz gráfica facilita la organización y relación de los elementos, lo que ayuda a mantener la consistencia y claridad del modelo. Además, Protégé permite exportar la ontología en formatos estándar como **RDF/XML**, lo que facilita su integración con otras aplicaciones y sistemas semánticos. Sin embargo, un desafío común es asegurar que la ontología sea lo suficientemente flexible y escalable para adaptarse a necesidades futuras, por lo que es recomendable revisar y ajustar las relaciones y restricciones conforme se amplíen los datos.

BIBLIOGRAFIA UTILIZADA

Marlene Goncalves Da Silva. 61GIIN, Ingenieria del Conocimiento, Universidad Internacional de Valencia. Diapositivas.