

Laboratorio 2 – OWL

Objetivos:

1. Entender el modelo semántico OWL.
2. Entender el razonamiento automático en triple stores y editores de ontologías.

Recursos necesarios:

- Editor de ontologías Protégé.
- Triple Store GraphDB en Ubuntu (Usuario/password: lsi/lsi) o servidor Google Cloud.

Índice:

1. Introducción.
 2. Instalar Protégé.
 3. Crear ontología.
 4. Razonamiento automático en Protégé.
 5. Razonamiento automático en GraphDB.
-

1.- Introducción

OWL (Web Ontology Language) es un estándar para crear vocabularios interoperables (Ontologías) en la web. En este laboratorio aprenderéis a crear ontologías y aplicar razonamiento automático, incluyendo consultas SPARQL.

2.- Instalar Protégé

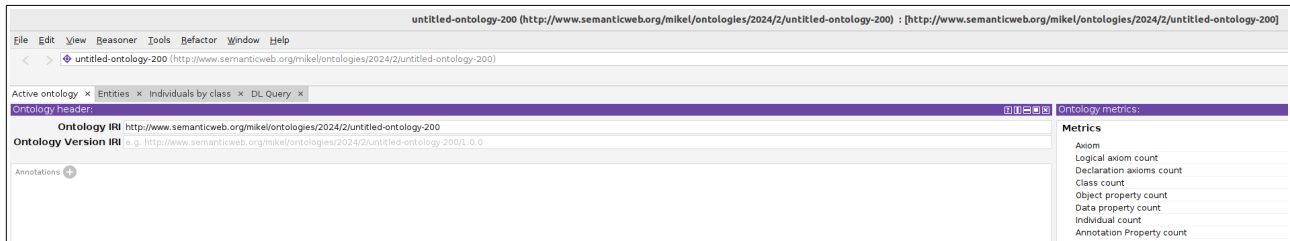
Protégé¹ es un editor de ontologías desarrollado por la prestigiosa universidad de Stanford. Para instalarlo, pinchar en *download now* y luego otra vez *download* (No es necesario registrarse). Es una aplicación Java.

3.- Crear una ontología

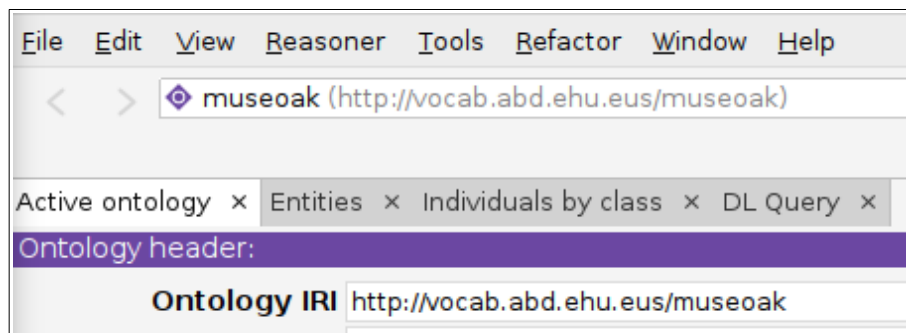
Para crear una ontología, ejecutar Protégé en la terminal y debería aparecer un interfaz de este estilo:

1 <https://protege.stanford.edu/>

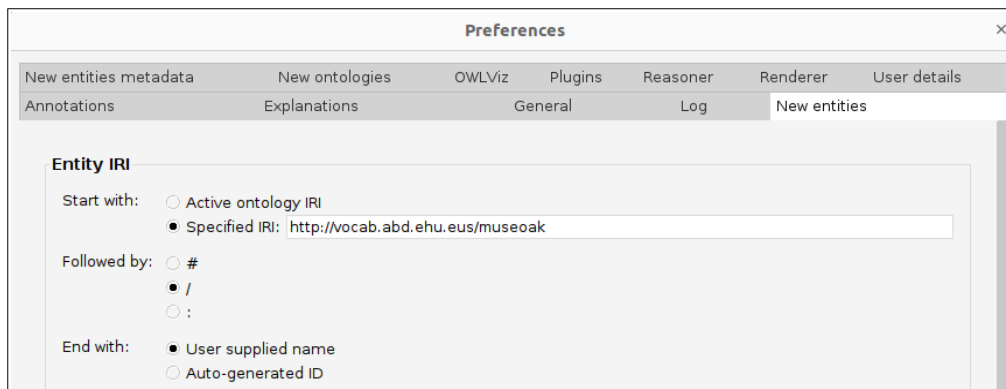
Administración de Bases de Datos 2023/2024



Para cambiar la URI de la ontología nueva, ir a *Active Ontology* y meter el valor de la URI en *Ontology IRI*, por ejemplo `http://vocab.abd.ehu.eus/museoak`:

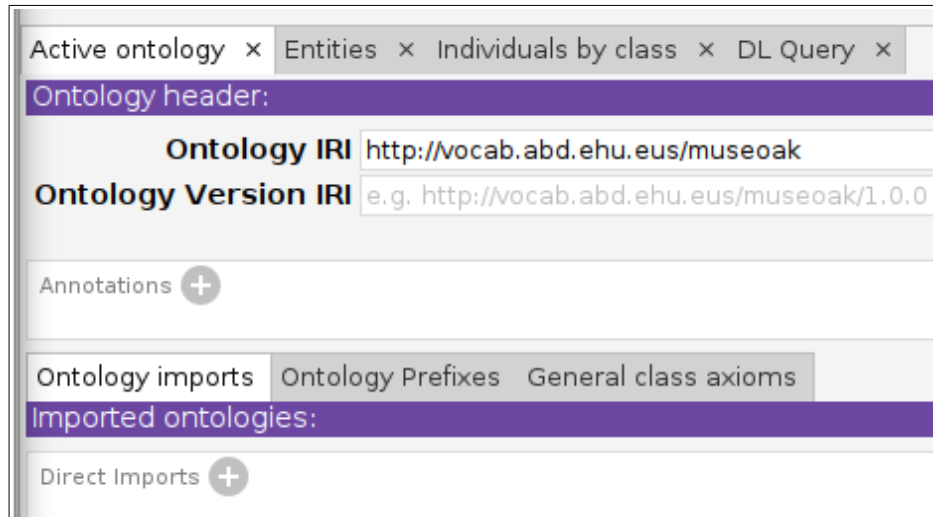


Guardar la ontología (*File > save*). Para que las URIs de las nuevas entidades tengan la misma estructura, ir a *File > Preferences > New entities* y añadir la URI en *Specified IRI*:

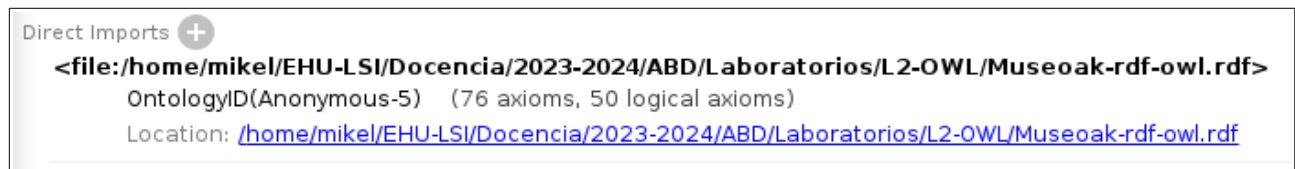


Ahora vamos a importar un ontología que tiene varias entidades que se van a usar en los siguientes ejercicios, pinchar en *Active ontology > Direct imports*:

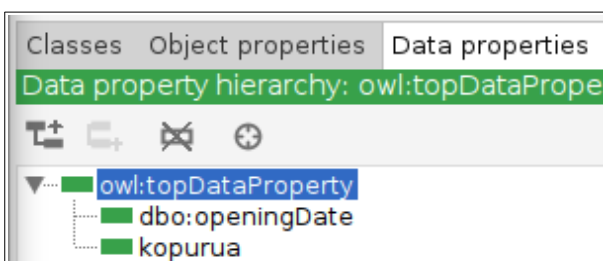
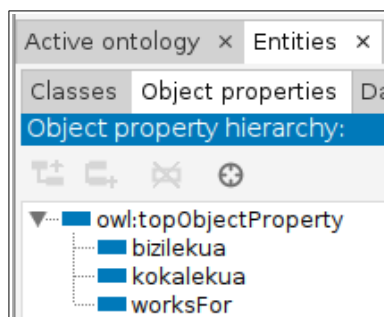
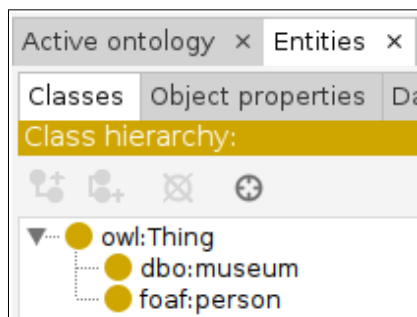
Administración de Bases de Datos 2023/2024

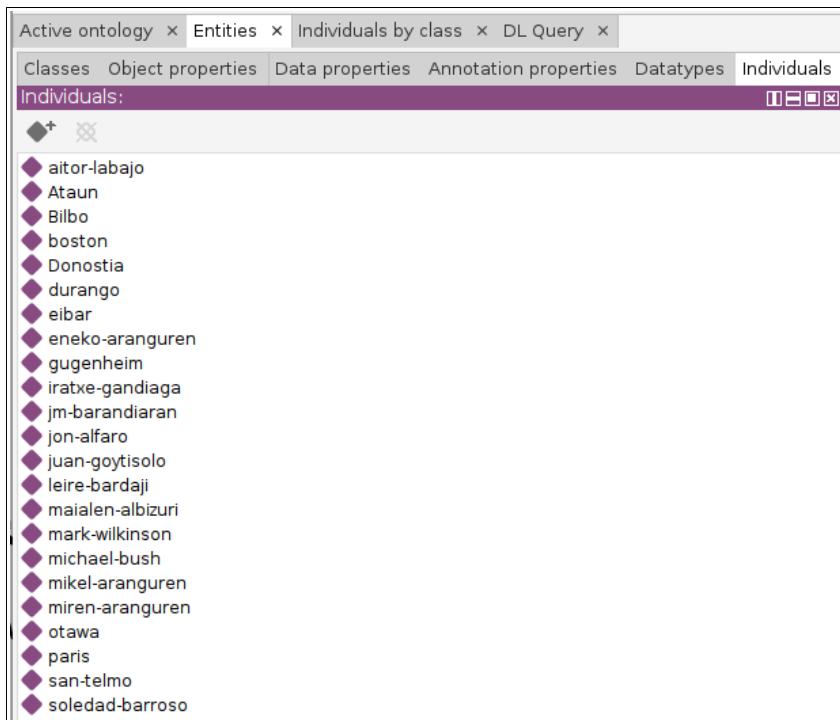


Elegir *Import an ontology contained in a local file*, y en el path elegir el archivo **museoak-rdf-owl.rdf** (Bajado de eGela):



La ontología importada incluye clases, object properties, data properties, e individuos, algunos de los cuales son de otras ontologías (Fijate en las URIs de **dbo:museum** y **foaf:person**):





Entrega:

- Ontología en OWL.

4.- Razonamiento automático en Protégé

Para entender el razonamiento automático, vamos a crear nuevas entidades y axiomas.

Crea una nueva clase llamada **langilea** con los siguientes atributos:

- `rdfs:label` "trabajador" (es)
- `equivalentTo` `foaf:person` and `worksFor` some `dbo:museum`

Ir a *Reasoner* y ahí elegir el razonador Hermit: pinchar en *Synchronize reasoner*. ¿Qué cambios ha habido en la clase **langilea**? Añade la respuesta en una anotación `rdfs:comment` en la clase misma.

Crea otra clase llamada **trabajadorExplotado** con el siguiente atributo:

- `equivalentTo` `foaf:person` and `worksFor` min 2 `dbo:museum`

Al razonar debería clasificar a los individuos aitor-labajo y mikel-aranguren como miembros de esta clase, pero no lo hace. ¿Por qué? [Pista: OWL no tiene Unique Name

Administración de Bases de Datos 2023/2024

Assumption] ¿Que cambiarías en la ontología para que se dé esa inferencia? Añade la respuesta en una anotación `rdfs:comment` en la clase misma.

Crea otra clase llamada **bilbotarLangilea** con el siguiente atributo:

- `equivalentTo foaf:person and worksFor some dbo: museum and bizilekua value Bilbo`

¿Cuál es la inferencia? Añade la respuesta en una anotación `rdfs:comment` en la clase misma.

En la pestaña Object Properties, en **bizilekua**, añade la siguiente cadena en SuperPropertyOf (Chain):

- `worksFor o kokalekua = bizilekua`

¿Al aplicar la inferencia, que implica este axioma para los individuos?

Añade la respuesta en una anotación `rdfs:comment` en el axioma cadena.

Añade la siguiente relación al individuo **Bilbo** (Hay que crear el individuo **Euskadi**):

- `Bilbo kokalekua Euskadi`

Crea la clase **Euskalplace** con el siguiente atributo:

- `equivalentTo kokalekua value Euskadi`

Al aplicar la inferencia, ¿Qué individuos aparecen como miembros de esa clase? ¿Si el Guggenheim esta localizado en Bilbo, y Bilbo esta localizado en Euskadi, el Guggenheim no debería aparecer como miembro de esa clase? ¿Qué cambiarías en la ontología para obtener ese resultado al aplicar razonamiento automático? Añade la respuesta en una anotación `rdfs:comment` en la clase misma.

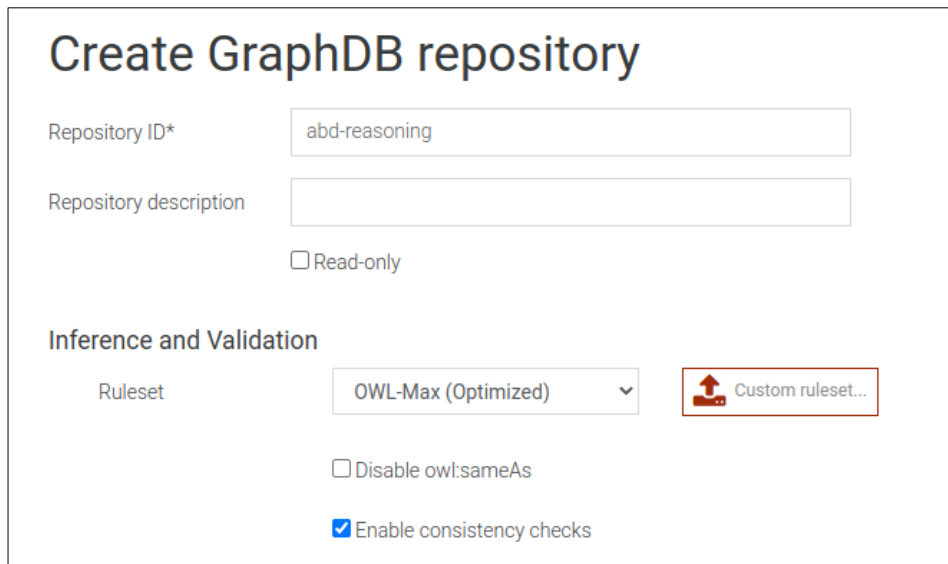
Modifica la propiedad **worksFor** con el atributo *Inverse functional* y aplica el razonamiento automático. ¿Que sucede con los individuos? Ahora haz que todos los individuos de la clase `foaf:person` sean diferentes entre sí y aplica el razonamiento automático. ¿Cuál es el resultado? Añade la respuesta en una anotación `rdfs:comment` en la clase misma.

Entrega:

- Ontología en OWL, incluyendo nuevos axiomas y explicaciones del resultado del razonamiento automático.

5.- Razonamiento automático en consultas SPARQL

El razonamiento automático también se puede usar en GraphDB en consultas SPARQL. Para conseguirlo, hay que crear un repositorio GraphDB con las características **OWL-Max (Optimized)** y **Enable consistency checks**:



Crear un Named Graph y subir el archivo **museoak-rdf-NO-owl.rdf**. Ejecuta la siguiente consulta, con las opciones **Include inferred data in the results** y **expand results over owl:sameAs**:

PREFIX owl: <http://www.w3.org/2002/07/owl#>

```
select * where {  
    ?person owl:sameAs ?other_person .  
}
```

¿Cuál es el resultado?

Crea una ontología nueva con la propiedad <http://vocab.data.gov/def/drm#worksFor> que sea **inverse functional** y súbela al mismo Named Graph que los datos anteriores. Vuelve a ejecutar la consulta, ¿Cuál es el resultado? ¿Por qué?

Administración de Bases de Datos 2023/2024

Añade más axiomas y entidades a la ontología (Por ejemplo propiedades transitivas) y prueba con distintas consultas, incluyendo consultas que tengan el predicado **rdf:type**.

Entrega:

- Ontología en OWL.
- Consultas SPARQL.