

Ejercicio 1. Ejemplo Básico en Turtle para modelar una familia.

Modelar la información de una familia y sus miembros utilizando el formato Turtle (TTL). Los alumnos deberán crear un vocabulario personalizado que permita representar conceptos clave, como los miembros de la familia, las relaciones familiares y atributos personales.

Desarrollo:

El código generado deberá respetar la sintaxis del formato Turtle y los alumnos pueden utilizar herramientas de validación para comprobar que no contenga errores. El alumno puede usar herramientas como, por ejemplo:

- <https://felixlohmeier.github.io/turtle-web-editor/>
- <https://github.com/IDLabResearch/TurtleValidator>
- <https://www.w3.org/RDF/Validator/>

Indicaciones:

- Definir clases para representar conceptos como Persona, Familia, y cualquier otra entidad que consideren relevante para el modelo.
- Crear propiedades que representen las relaciones familiares y los atributos de cada persona. Algunos ejemplos de estas propiedades podrían ser:
 - **nombre:** El nombre de la persona.
 - **edad:** La edad de la persona.
 - **direccion:** La dirección de la persona.
 - **esPadreDe:** Relación que indica que una persona es el padre de otra.
 - **esMadreDe:** Relación que indica que una persona es la madre de otra.
 - **esHijoDe:** Relación que indica que una persona es hijo de otra.
 - **perteneceAFamilia:** Relación que indica a qué familia pertenece una persona.

El alumno puede agregar más relaciones y atributos según lo considere necesario para reflejar adecuadamente la estructura de una familia.

Ejercicio 2. Ejemplo Básico en Turtle para diseñar un Vocabulario Propio que modele la relación entre autores y libros.

Modelar información relacionada con libros y autores utilizando el formato Turtle (TTL). El alumno debe diseñar un vocabulario personalizado que permita representar tanto a los libros como a los autores, incluyendo información relevante sobre ellos y la relación que existe entre ambos. El objetivo es que los estudiantes comprendan cómo estructurar datos semánticos mediante la creación de vocabularios propios.

Desarrollo:

El alumno deberá crear al menos tres instancias de la clase Libro, cada una con diferentes atributos, e instanciar al menos dos individuos de la clase Autor. Además, se deben definir propiedades para establecer las relaciones entre libros y autores.

- **Requisitos:**
 - Modelar relaciones como la autoría de un libro, la fecha de publicación, y cualquier otra información relevante que el alumno considere necesaria.
 - El vocabulario y las relaciones deberán seguir el formato Turtle, y se pueden utilizar herramientas de verificación para comprobar que el código generado no contenga errores de sintaxis.

Pautas:

- **Clases para definir:**
 - Autor: Representa a los autores de los libros.
 - Libro: Representa a los libros.
- **Propiedades sugeridas:**
 - tieneNombre: Para definir el nombre de los autores y libros.
 - escritoPor: Relación entre un libro y el autor que lo escribió.
 - publicadoEn: Fecha de publicación de un libro.
 - escribe: Relación inversa, que indica que un autor ha escrito un libro.

Herramientas: Se recomienda usar validadores online para asegurar que el código generado no contiene errores.

Ejercicio 3. Modelado de relaciones entre entidades compuestas por estudiantes y universidad.

El alumno deberá modelar la información de una persona utilizando un vocabulario personalizado en formato Turtle (TTL) para representar conceptos clave. La información modelada debe incluir datos como el nombre, edad, correo electrónico, lugar de residencia, universidad en la que estudia, y la ubicación de dicha universidad. Se invita al alumno a añadir información adicional que considere relevante para enriquecer el modelo.

Desarrollo:

El alumno deberá crear al menos tres individuos de la clase Persona, con atributos diversos como el nombre, la edad, el correo electrónico, y la residencia. Además, deberá instanciar al menos dos individuos de la clase Universidad, especificando su ubicación geográfica mediante la clase Ciudad u otra entidad relacionada.

El modelo deberá establecer relaciones entre las personas y las universidades, así como entre las universidades y sus respectivas ciudades. Se anima al alumno a agregar más información relevante, como hobbies, idiomas hablados, o cualquier otro atributo que enriquezca la representación de los individuos.

Recomendaciones:

- **Clases sugeridas:**
 - Persona: Para modelar a las personas.
 - Universidad: Para modelar las universidades.
 - Ciudad: Para modelar las ciudades en las que se encuentran las universidades.
- **Propiedades sugeridas:**
 - tieneNombre: Representa el nombre de la persona o universidad.
 - tieneEdad: Representa la edad de una persona.
 - viveEn: Relaciona una persona con su lugar de residencia.
 - estudiaEn: Relaciona una persona con la universidad en la que estudia.
 - ubicadaEn: Relaciona una universidad con la ciudad donde se encuentra.
- **Espacio de nombres:** Define un espacio de nombres adecuado para tu vocabulario, por ejemplo, <http://ejemplo.com/vocab#>.

Indicaciones:

1. Crear al menos **tres individuos** de la clase Persona, asignándoles atributos diversos como nombre, edad, email, y lugar de residencia.

2. Crear al menos **dos individuos** de la clase Universidad, especificando sus ubicaciones utilizando instancias de la clase Ciudad.
3. Utilizar las propiedades definidas para **relacionar a las personas con las universidades** en las que estudian, y a las universidades con sus respectivas ciudades.
4. Se permite añadir información adicional, como hobbies, idiomas hablados, o cualquier otra relación o atributo que se considere útil para enriquecer el modelo.

Notas:

El código generado deberá seguir el formato Turtle, y el alumno puede comprobar si contiene errores mediante el uso de herramientas de validación online o software especializado.

Ejercicio 4. Modelado de una red social.

Se pide diseñar y modelar una red social utilizando el formato Turtle (TTL). Los estudiantes deberán definir una ontología para representar usuarios, publicaciones, relaciones de amistad y otros conceptos clave dentro de la red social. Se busca que los alumnos apliquen los conceptos de modelado semántico mediante la creación de un vocabulario personalizado y la instancia de ejemplos concretos que simulen la interacción dentro de una red social.

Desarrollo:

Los estudiantes deberán realizar las siguientes tareas:

1. **Definir la ontología:** Crear las clases y propiedades necesarias para modelar la red social. Las clases mínimas que deben definirse incluyen:
 - Usuario: Clase que representa a los usuarios de la red social.
 - Publicación: Clase que modela las publicaciones realizadas por los usuarios.
 - Amistad: Propiedad que relaciona dos usuarios que son amigos.
2. **Instancias:** Crear instancias que muestren ejemplos de usuarios y sus interacciones en la red social. Se pide cumplir con los siguientes requisitos:
 - Crear al menos **10 usuarios** con los atributos de nombre, fecha de nacimiento, género y ubicación.
 - Definir al menos **5 publicaciones** realizadas por diferentes usuarios. Estas publicaciones deben incluir atributos como fecha de publicación, contenido, y número de "me gusta".
 - Crear al menos **15 relaciones de amistad** entre los usuarios, utilizando una propiedad que indique una relación de amistad.
3. **Enriquecimiento del modelo:** Se anima a los estudiantes a agregar información adicional para enriquecer el modelo, como:
 - Información sobre los hobbies o intereses de los usuarios.
 - Seguidores o interacciones más complejas como comentarios en publicaciones.
 - Información sobre la popularidad de las publicaciones (número de comentarios o me gusta).
4. **Formato y verificación:** El código generado debe seguir el formato Turtle. Los alumnos deben verificar la ontología para asegurarse de que no contenga errores utilizando herramientas de validación de RDF.

Se pide:

- Definir las siguientes clases: Usuario, Publicación, RedSocial y cualquier otra clase relevante.
- Propiedades propuestas:
 - nombre: Atributo que indica el nombre del usuario.
 - fechaNacimiento: Atributo que indica la fecha de nacimiento del usuario.
 - generaPublicacion: Relación entre un usuario y una publicación.
 - amigoDe: Relación que indica que un usuario es amigo de otro.
 - meGusta: Relación que indica cuántos "me gusta" ha recibido una publicación.

Ejercicios para SPARQL.

Descripción del entorno de trabajo.

Para los ejercicios que se presentan a continuación se puede utilizar los ficheros en python disponible en aulavirtual: procesarSparql.py, examplePersonaCompañia.ttl y sentenciaSparql.txt.

El fichero examplePersonaCompañia.ttl contiene una ontología de ejemplo sobre la que se pueden realizar diferentes sentencias Sparql para practicar. El archivo sentenciaSparql.txt contiene la sentencia Sparql que se quiere realizar sobre la ontología examplePersonaCompañia.ttl. Finalmente, el archivo procesarSparql.py el encargado de ejecutar la sentencia Sparql sobre la ontología cargada.

Antes de ejecutar por primera vez el código es necesario instalar las librerías necesarias. Para ello es necesario tener instalado python y además tener instalada la librería 'rdflib'. Esta librería puede ser instalada a través del comando "pip install rdflib".

Para ejecutar el archivo procesarSparql.py hay que escribir en consola el siguiente comando:

```
python3 procesarSparql.py examplePersonaCompañia.ttl querySparql.txt
```


Ejercicio 1

Partiendo del modelo creado en el ejercicio 1. Escribir las consultas sparql para responder lo que se pide en los siguientes puntos:

- Obtener todas las instancias de la ontología
- Obtener todas las intancias de libros
- Obtener el listado de los nombres de los libros
- Consulta SPARQL para obtener libros escritos por el autor1
- Consulta para obtener la lista de libros junto al autor

Ejercicio 2

Partiendo del modelo creado en el ejercicio 2. Escribir las consultas sparql para responder lo que se pide en los siguientes puntos:

- Consulta para obtener los miembros de la familia García
- Consulta para obtener la relación de hijos y padres (no madres)
- Consulta para obtener todas las personas y sus edades
- Consulta para obtener las personas de género masculino
- Consulta para obtener los nombres y apellidos de las personas casadas
- Consulta para obtener el nombre de los padres de Pedro García
- Consulta para obtener el nombre y apellido de las personas que tienen hijos

Ejercicio 3

Partiendo del modelo creado en el ejercicio 2. Escribir las consultas sparql para responder lo que se pide en los siguientes puntos:

- Obtener los nombres de todas las personas
- Obtener los nombres y edades de todas las personas
- Obtener las personas que viven en Madrid
- Obtener las personas y las universidades en las que estudian
- Obtener los correos electrónicos de todas las personas
- Obtener las personas que hablan más de un idioma
- Obtener las personas cuyo hobby es 'Leer'
- Obtener las ciudades en las que están ubicadas las universidades
- Obtener las personas que estudian en 'Universidad de Madrid'