

Práctica 5: Consultas SPARQL para la ontología de libros

1. Introducción e instancias creadas.....	2
2. Consultas simples en SPARQL.....	3
2.1. Lectores que compraron un libro del género CienciaFicción.....	3
2.2. Títulos de libros del género NoFicción.....	3
2.3. Libros con valoración 4.....	4
2.4. Libros comprados por cada lector.....	4
3. Consultas avanzadas en SPARQL.....	5
3.1. Total de libros comprados.....	5
3.2. Suma de los precios de libros comprados por cada lector.....	5

1. Introducción e instancias creadas

SPARQL es una herramienta útil y versátil para consultar y explorar datos estructurados en formato *RDF*, ampliamente utilizada en el ámbito de la web semántica.

Su capacidad para trabajar con ontologías permite explorar clases, propiedades y patrones complejos. Sin embargo, una de sus limitaciones es que no incluye automáticamente las clases e instancias inferidas por los razonadores. Por ello, se han creado algunas instancias explícitas de **Libro** y **Lector** previamente.

Concretamente, se han definido las siguientes instancias:

- **Lectores:** *LectorJulio* y *LectoraManuela*.
- **Libros:**
 - *LibroCienciaFicción:*
 - **Propiedades de objetos:**
 - tieneGénero CienciaFicción
 - compradoPor *LectoraManuela*
 - **Propiedades de datos:**
 - tieneTítulo “Dune”
 - tienePrecio 12.99
 - tieneValoración 4
 - *LibroNoFicción:*
 - **Propiedades de objetos:**
 - tieneGénero NoFicción
 - compradoPor *LectorJulio*
 - **Propiedades de datos:**
 - tieneTítulo “El hombre en busca del sentido”
 - tienePrecio 15.5
 - tieneValoración 4
 - *LibroNoFicción2:*
 - **Propiedades de objetos:**
 - tieneGénero NoFicción
 - compradoPor *LectoraManuela*
 - **Propiedades de datos:**
 - tieneTítulo “Las mejores recetas de cocina”
 - tienePrecio 6.25
 - tieneValoración 3

2. Consultas simples en SPARQL

2.1. Lectores que compraron un libro del género CienciaFicción

La siguiente consulta permite listar los lectores relacionados con libros del género Ciencia Ficción.

SPARQL query:	
PREFIX ex: <http://www.bookstore.com/ontology.owl#>	
SELECT ?lector	
WHERE {	
?libro ex:tieneGénero ex:géneroCienciaFicción .	
?libro ex:compradoPor ?lector .	
}	
lector	
LectoraManuela	

2.2. Títulos de libros del género NoFicción

Se consultan los títulos de aquellos libros de género No Ficción.

SPARQL query:	
PREFIX ex: <http://www.bookstore.com/ontology.owl#>	
SELECT ?titulo	
WHERE {	
?libro ex:tieneGénero ex:géneroNoFicción .	
?libro ex:tieneTítulo ?titulo .	
}	
titulo	
"El hombre en busca del sentido" ^{^^} <http://www.w3.org/2001/XMLSchema#string>	
"Las mejores recetas de cocina" ^{^^} <http://www.w3.org/2001/XMLSchema#string>	

2.3. Libros con valoración 4

Se obtienen los títulos de aquellos libros con una valoración específica, en este caso 4.

SPARQL query:	
<pre>PREFIX ex: <http://www.bookstore.com/ontology.owl#> SELECT ?titulo WHERE { ?libro ex:tieneValoración 4 . ?libro ex:tieneTítulo ?titulo . }</pre>	
titulo	
"El hombre en busca del sentido"^^<http://www.w3.org/2001/XMLSchema#string>	
"Dune"^^<http://www.w3.org/2001/XMLSchema#string>	

2.4. Libros comprados por cada lector

Relaciona lectores con los libros que han comprado, mostrando tres columnas (nombre de la instancia, título del libro y lector que lo ha comprado).

SPARQL query:		
<pre>PREFIX ex: <http://www.bookstore.com/ontology.owl#> SELECT ?libro ?titulo ?lector WHERE { ?libro ex:compradoPor ?lector . ?libro ex:tieneTítulo ?titulo . }</pre>		
libro	titulo	lector
LibroCienciaFicción	"Dune"^^<http://www.w3.org/2001/XMLSchema#string>	LectoraManuela
LibroNoFicción	"El hombre en busca del sentido"^^<http://www.w3.org/	LectorJulio

3. Consultas avanzadas en SPARQL

3.1. Total de libros comprados

Calcula el número total de libros comprados por algún lector a través del operador *COUNT*.

SPARQL query:	
<pre>PREFIX ex: <http://www.bookstore.com/ontology.owl#> SELECT (COUNT(?libro) AS ?totalLibros) WHERE { ?libro ex:compradoPor ?lector . }</pre>	
	totalLibros
"3"	<http://www.w3.org/2001/XMLSchema#integer>

3.2. Suma de los precios de libros comprados por cada lector

Para cada lector, se suman los precios de los libros que han comprado mediante el operador *SUM*.

SPARQL query:	
<pre>PREFIX ex: <http://www.bookstore.com/ontology.owl#> SELECT ?lector (SUM(?precio) AS ?totalGasto) WHERE { ?libro ex:compradoPor ?lector . ?libro ex:tienePrecio ?precio . } GROUP BY ?lector</pre>	
lector	totalGasto
LectoraManuela	"19.24"<http://www.w3.org/2001/XMLSchema#decimal>
LectorJulio	"15.5"<http://www.w3.org/2001/XMLSchema#decimal>