CC3201-1

Bases de Datos

Otoño 2020

Clase 13: Datos Semiestructurados: Grafos

Aidan Hogan

aidhog@gmail.com

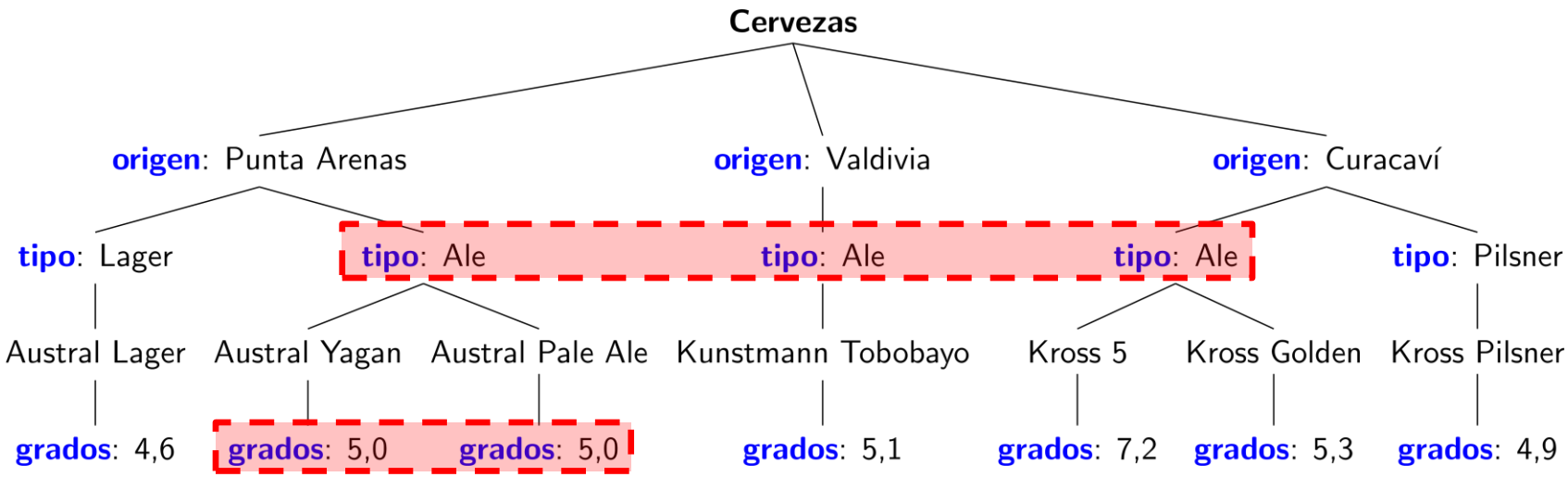
Modelos de Datos

Modelo de datos (tabla)

Cervezas

nombre	tipo	grados	ciudad-origen
Austral Lager	Lager	4,6	Punta Arenas
Austral Yagan	Ale	5,0	Punta Arenas
Austral Pale Ale	Ale	5,0	Punta Arenas
Kuntsmann Torobayo	Ale	5,1	Valdivia
Kross 5	Ale	7,2	Curacaví
Kross Golden	Ale	5,3	Curacaví
Kross Pilsner	Pilsner	4,9	Curacaví

Modelo de datos (árbol/jerarquía)



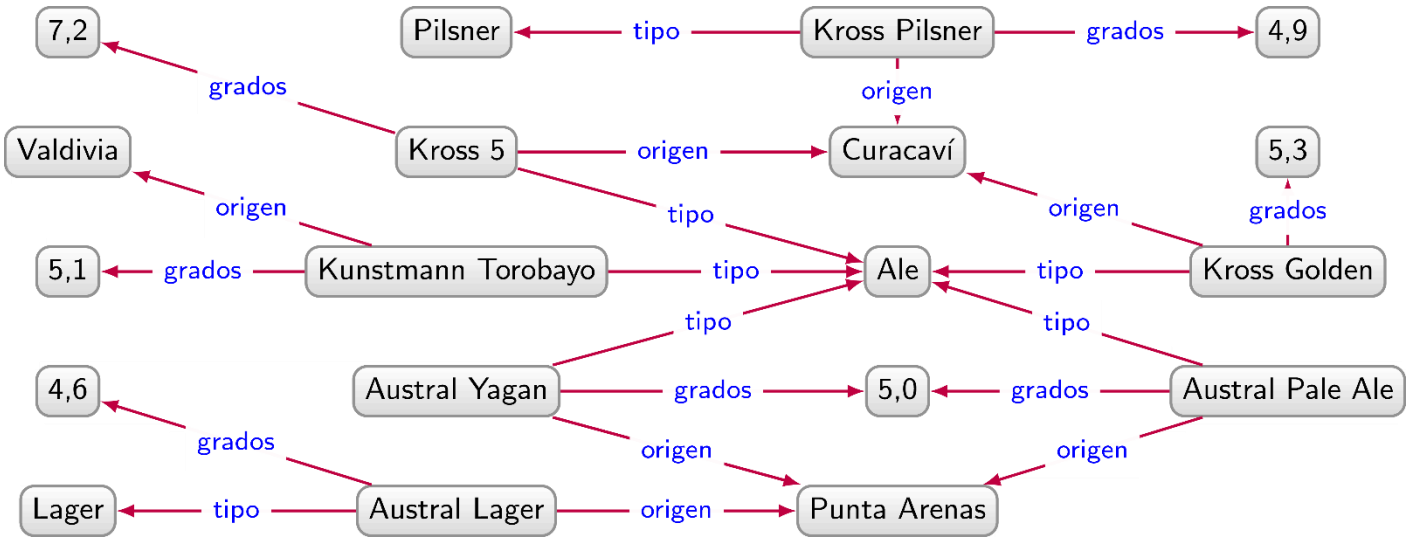
¿Es un "verdadero" árbol?

Umm … sí y no.

Es feo tener tres nodos para Ale

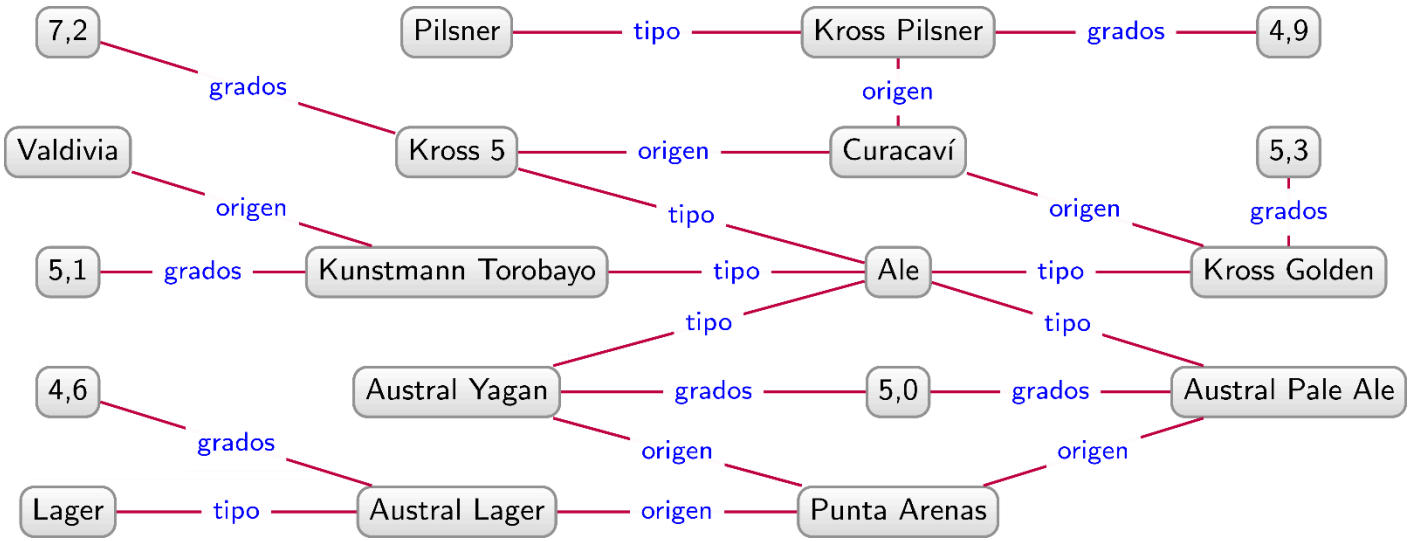
(y es más difícil de consultar)

Modelo de datos (grafo)



Un grafo dirigido con arcos etiquetados

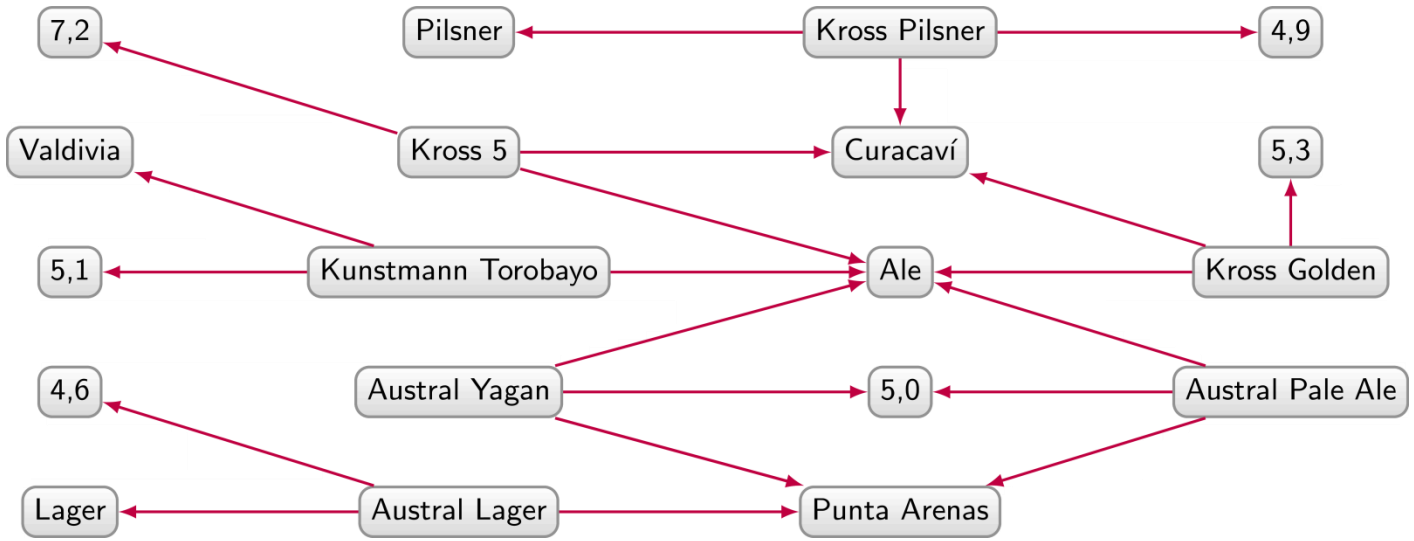
Grafo dirigido ...



¿El problema sin dirección?

¿Curacaví es el origen de Kross 5 o Kross 5 es el origen de Curacavía?

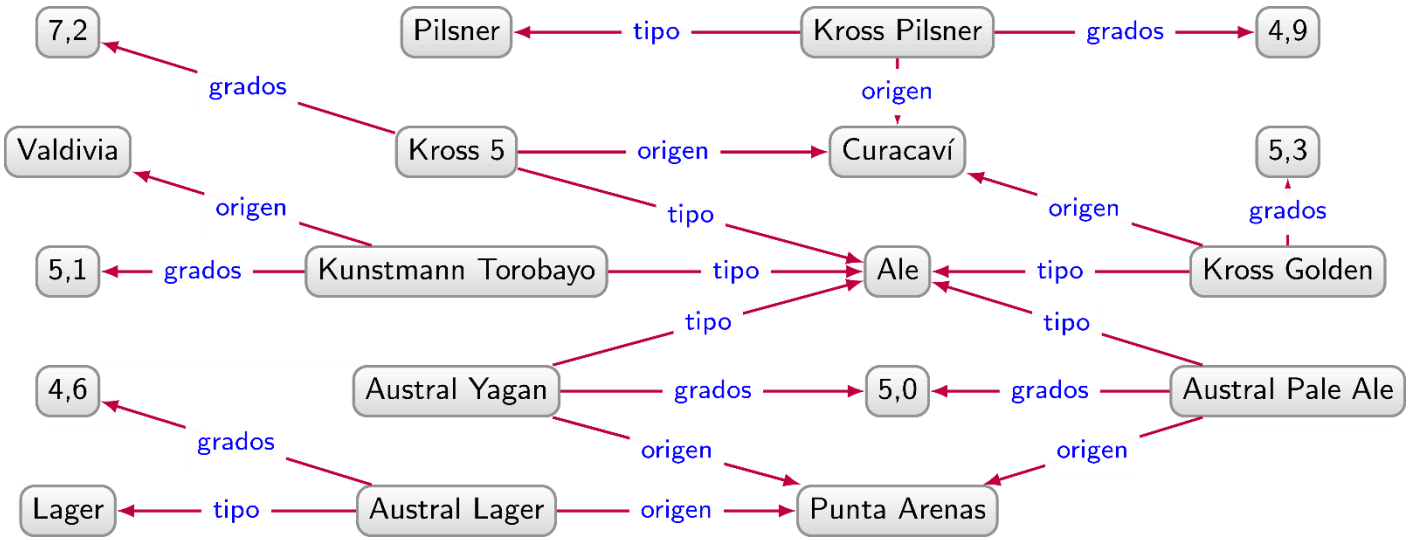
Grafo con arcos etiquetados



¿El problema sin etiquetas sobre los arcos?

No sabemos la relación entre Ale y Kross Golden, por ejemplo.

Un grafo dirigido con arcos etiquetados



Un grafo dirigido con arcos etiquetados

✓ **Muy flexible / simple**

✗ **Difícil de resumir los datos**

(no hay un esquema obvio)

El espectro de la estructura de datos

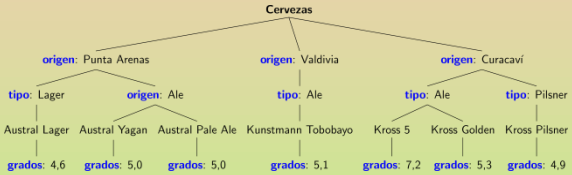
Texto Plano

Hay mucha variedad en las cervezas locales de Chile. La sede de la marca "Austral" se encuentra en Punta Arenas. Austral fabrica una amplia gama de cervezas, incluyendo Larger (4,6%), Yagan (un ale de 5%) y un Pale Ale (5%). La cerveza de marca "Kunstmann" también es popular, en particular su cerveza "Torobayo" (un ale de 5,1% elaborado en Valdiva). La marca Kross, basada en Curacaví, también tiene una gama de cervezas populares como, por ejemplo, Kross 5 (un ale fuerte de 7,2%) Kross Golden (un ale de 5,3%) y Kross Pilsner (4,9%).

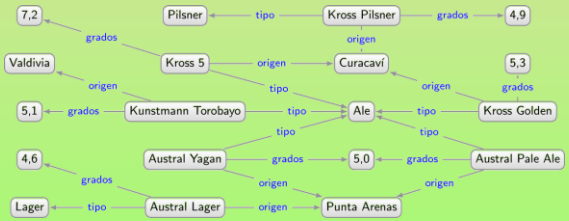
Texto Enriquecido
(HTML, Word, ...)

```
<ul>
  <li>Austral Lager [Lager] 4,6% de Punta Arenas</li>
  <li>Austral Yagan [Ale] 5% de Punta Arenas</li>
  <li>Austral Pale Ale [Ale] 5% de Punta Arenas</li>
  <li>Kunstsmann Torobayo [Ale] 5,1% de Valdivia</li>
  <li>Kross 5 [Ale] 7,2% de Curacaví</li>
  <li>Kross Golden [Ale] 5,3% de Curacaví</li>
  <li>Kross Pilsner [Pilsner] 4,9% de Curacaví</li>
</ul>
```

Árboles
(XML, JSON, ...)



Grafos
(RDF, Prop. Gs, ...)



Relacional
(SQL, CSV, ...)

Cervezas

nombre	tipo	grados	ciudad-origen
Austral Lager	Lager	4,6	Punta Arenas
Austral Yagan	Ale	5,0	Punta Arenas
Austral Pale Ale	Ale	5,0	Punta Arenas
Kuntsmann Torobayo	Ale	5,1	Valdivia
Kross 5	Ale	7,2	Curacaví
Kross Golden	Ale	5,3	Curacaví
Kross Pilsner	Pilsner	4,9	Curacaví

No estructurados

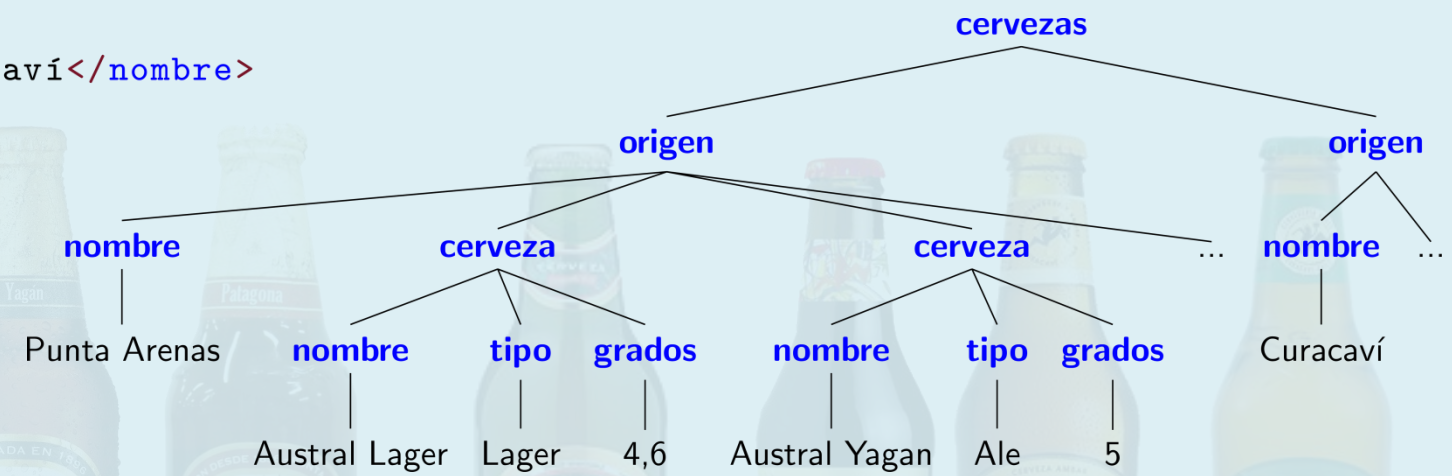
D A T O S

Semiestructurados

Estructurados

Para árboles, tenemos formatos como XML

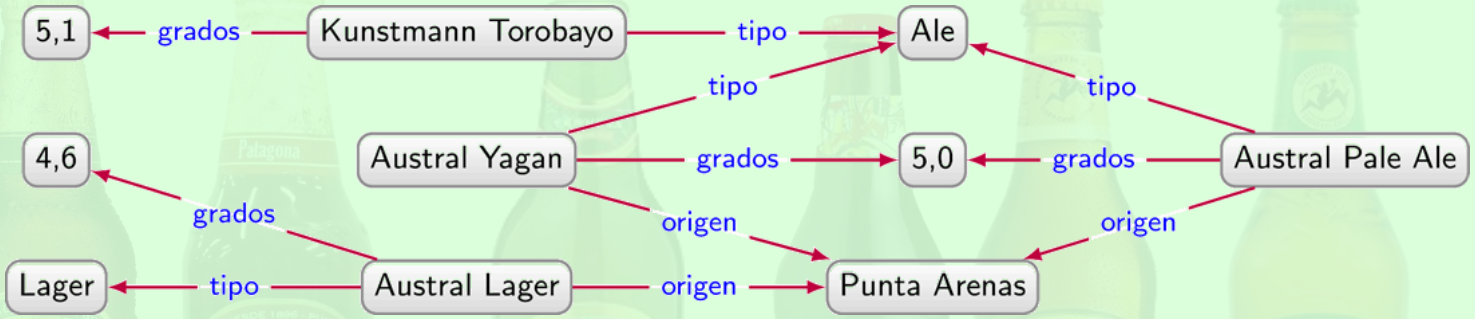
```
<cervezas>
  <origen>
    <nombre>Punta Arenas</nombre>
    <cerveza>
      <nombre>Austral Lager</nombre>
      <tipo>Lager</tipo>
      <grados>4.6</grados>
    </cerveza>
    <cerveza>
      <nombre>Austral Yagan</nombre>
      <tipo>Ale</tipo>
      <grados>5</grados>
    </cerveza>
    ...
  </origen>
  <origen>
    <nombre>Curacaví</nombre>
    ...
  </origen>
</cervezas>
```



Para grafos, podemos usar triples

Kunstmann_Torobayo	tipo	Ale
Kunstmann_Torobayo	grados	5,1
Austral_Yagan	tipo	Ale
Austral_Yagan	grados	5,0
Austral_Yagan	origen	Punta_Arenas
Austral_Pale_Ale	tipo	Ale
Austral_Pale_Ale	grados	5,0
Austral_Pale_Ale	origen	Punta_Arenas
Austral_Lager	tipo	Lager
Austral_Lager	grados	4,6
Austral_Lager	origen	Punta_Arenas

No hay orden.



JUEGO DE DATOS

Texto Plano

Texto Enriquecido
(HTML, Word, ...)

Árboles
(XML, JSON, ...)

Grafos
(RDF, Prop. Gs, ...)

Relacional
(SQL, CSV, ...)

No estructurados

D
A
T
O
S

Semiestructurados

Estructurados

¿Pero dónde de usan grafos?

La Web de Datos / La Web Semántica

¡La Web es magnífica!

¿Pero puede ser mejor?

… a veces es difícil encontrar información relevante.

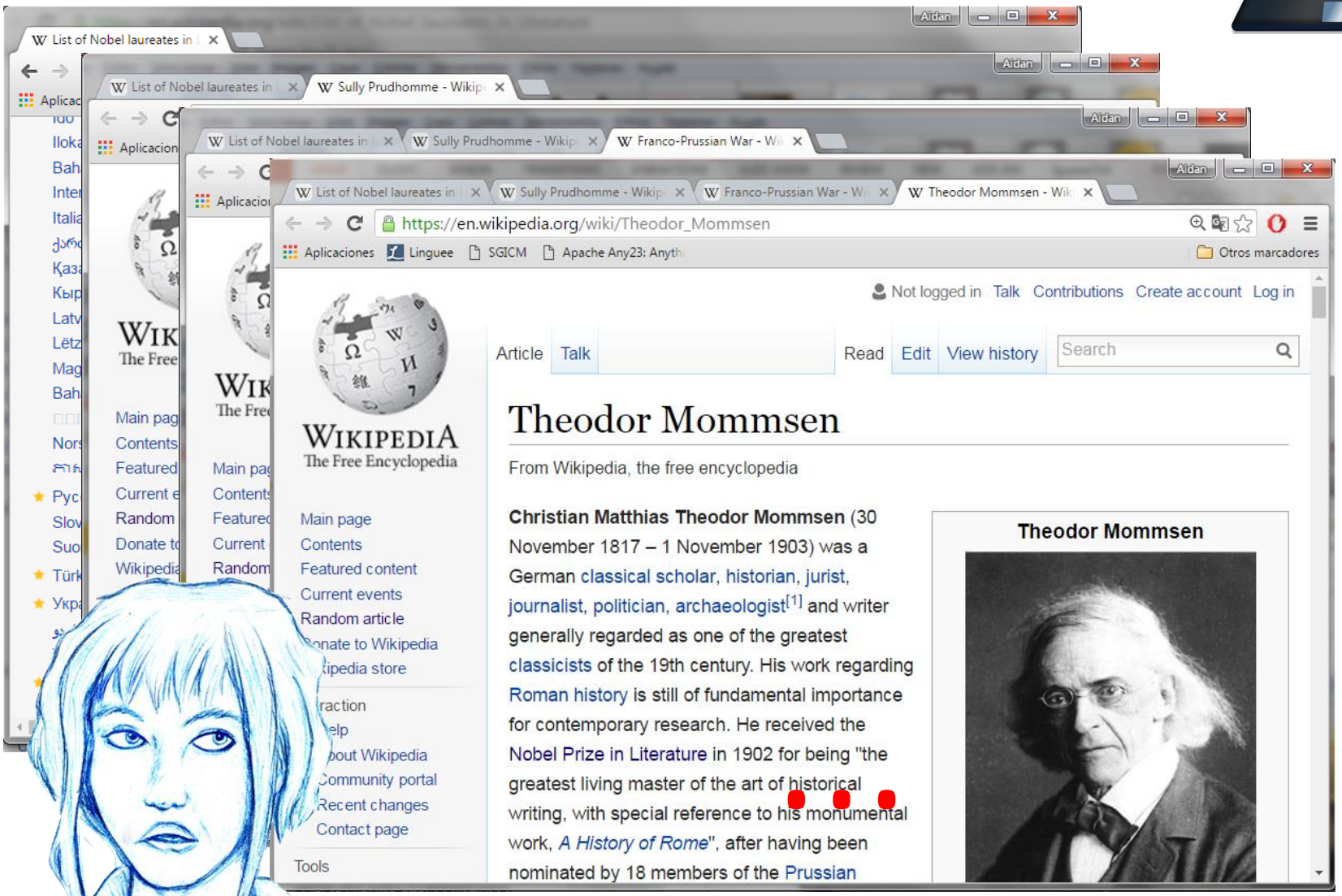
Haciendo un informe para su clase …

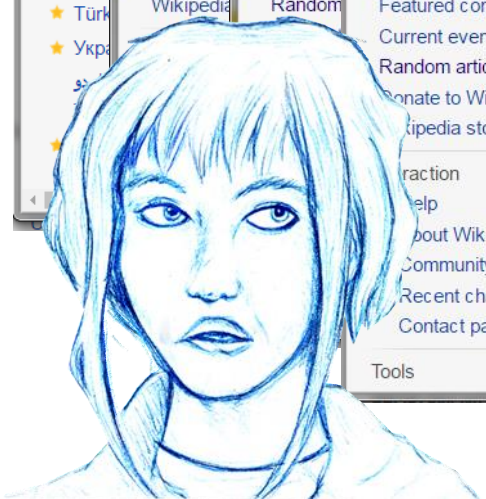
Quiere encontrar *cada*:

- Ganadores del Premio Nobel en Literatura
- Que han luchado en una guerra
- El año que ganaron el premio
- Y el año que comenzó la guerra

¿Cómo se puede encontrar esta información con la Web actual?

Muchas pestañas de Wikipedia …





La Web en el espectro de datos

Texto Plano

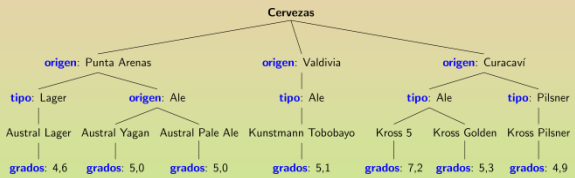
Hay mucha variedad en las cervezas locales de Chile. La sede de la marca "Austral" se encuentra en Punta Arenas. Austral fabrica una amplia gama de cervezas, incluyendo Larger (4,6%), Yagan (un ale de 5%) y un Pale Ale (5%). La cerveza de marca "Kunstmann" también es popular, en particular su cerveza "Torobayo" (un ale de 5,1% elaborado en Valdiva). La marca Kross, basada en Curacaví, también tiene una gama de cervezas populares como, por ejemplo, Kross 5 (un ale fuerte de 7,2%) Kross Golden (un ale de 5,3%) y Kross Pilsner (4,9%).

Texto Enriquecido
(HTML, Word, ...)

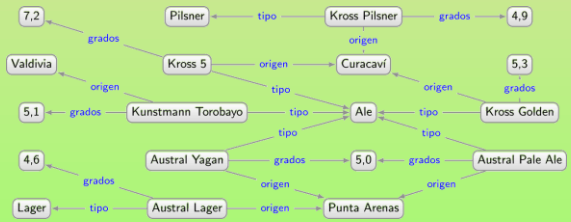
```
<ul>
  <li>Austral Lager [Lager] 4,6% de Punta Arenas</li>
  <li>Austral Yagan [Ale] 5% de Punta Arenas</li>
  <li>Austral Pale Ale [Ale] 5% de Punta Arenas</li>
  <li>Kunstsmann Torobayo [Ale] 5,1% de Valdivia</li>
  <li>Kross 5 [Ale] 7,2% de Curacaví</li>
  <li>Kross Golden [Ale] 5,3% de Curacaví</li>
  <li>Kross Pilsner [Pilsner] 4,9% de Curacaví</li>
</ul>
```

⇐ La Web

Árboles
(XML, JSON, ...)



Grafos
(RDF, Prop. Gs, ...)



Relacional
(SQL, CSV, ...)

Cervezas

nombre	tipo	grados	ciudad-origen
Austral Lager	Lager	4,6	Punta Arenas
Austral Yagan	Ale	5,0	Punta Arenas
Austral Pale Ale	Ale	5,0	Punta Arenas
Kuntsmann Torobayo	Ale	5,1	Valdivia
Kross 5	Ale	7,2	Curacaví
Kross Golden	Ale	5,3	Curacaví
Kross Pilsner	Pilsner	4,9	Curacaví

No estructurados

D A T O S

Semiestructurados

Estructurados

La Web Semántica …

… una base de datos global usando la Web

La Web de Datos en el espectro de datos

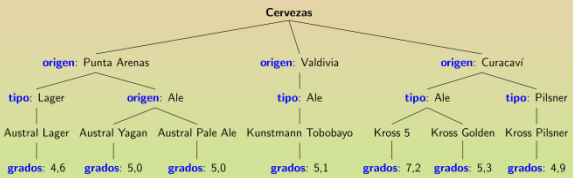
Texto Plano

Hay mucha variedad en las cervezas locales de Chile. La sede de la marca "Austral" se encuentra en Punta Arenas. Austral fabrica una amplia gama de cervezas, incluyendo Larger (4,6%), Yagan (un ale de 5%) y un Pale Ale (5%). La cerveza de marca "Kunstmann" también es popular, en particular su cerveza "Torobayo" (un ale de 5,1% elaborado en Valdiva). La marca Kross, basada en Curacaví, también tiene una gama de cervezas populares como, por ejemplo, Kross 5 (un ale fuerte de 7,2%) Kross Golden (un ale de 5,3%) y Kross Pilsner (4,9%).

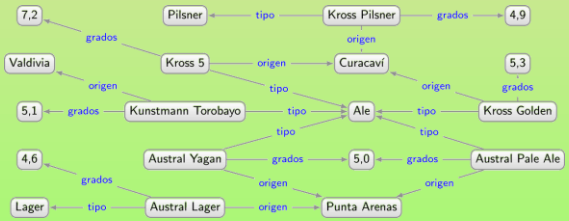
Texto Enriquecido
(HTML, Word, ...)

```
<ul>
  <li>Austral Lager [Lager] 4,6% de Punta Arenas</li>
  <li>Austral Yagan [Ale] 5% de Punta Arenas</li>
  <li>Austral Pale Ale [Ale] 5% de Punta Arenas</li>
  <li>Kunstsmann Torobayo [Ale] 5,1% de Valdivia</li>
  <li>Kross 5 [Ale] 7,2% de Curacaví</li>
  <li>Kross Golden [Ale] 5,3% de Curacaví</li>
  <li>Kross Pilsner [Pilsner] 4,9% de Curacaví</li>
</ul>
```

Árboles
(XML, JSON, ...)



Grafos
(RDF, Prop. Gs, ...)



Relacional
(SQL, CSV, ...)

Cervezas

nombre	tipo	grados	ciudad-origen
Austral Lager	Lager	4,6	Punta Arenas
Austral Yagan	Ale	5	Punta Arenas
Austral Pale Ale	Ale	5,[illegible]	Punta Arenas
Kunstsmann Torobayo	Ale	5,1	Valdivia
Kross 5	Ale	7,2	Curacaví
Kross Golden	Ale	5,3	Curacaví
Kross Pilsner	Pilsner	4,9	Curacaví

No estructurados

¿Y si quisiéramos estructurar los datos de la Web y crear una Web de Datos, donde la pondríamos en el espectro?

Imaginemos el esquema relacional. :/

Estructurados

¿La Web de Datos?

La Web de Datos en el espectro de datos

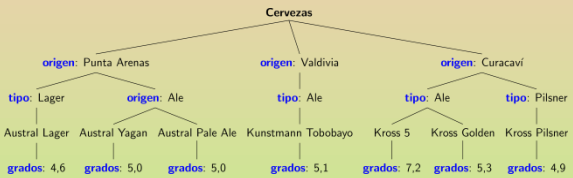
Texto Plano

Hay mucha variedad en las cervezas locales de Chile. La sede de la marca "Austral" se encuentra en Punta Arenas. Austral fabrica una amplia gama de cervezas, incluyendo Larger (4,6%), Yagan (un ale de 5%) y un Pale Ale (5%). La cerveza de marca "Kunstmann" también es popular, en particular su cerveza "Torobayo" (un ale de 5,1% elaborado en Valdiva). La marca Kross, basada en Curacaví, también tiene una gama de cervezas populares como, por ejemplo, Kross 5 (un ale fuerte de 7,2%) Kross Golden (un ale de 5,3%) y Kross Pilsner (4,9%).

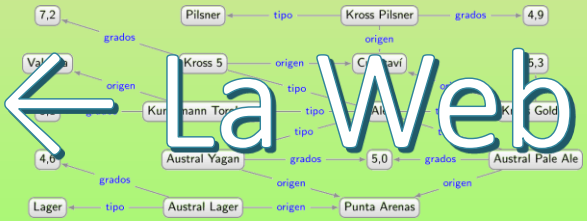
Texto Enriquecido
(HTML, Word, ...)

```
<ul>
  <li>Austral Lager [Lager] 4,6% de Punta Arenas</li>
  <li>Austral Yagan [Ale] 5% de Punta Arenas</li>
  <li>Austral Pale Ale [Ale] 5% de Punta Arenas</li>
  <li>Kunstsmann Torobayo [Ale] 5,1% de Valdivia</li>
  <li>Kross 5 [Ale] 7,2% de Curacaví</li>
  <li>Kross Golden [Ale] 5,3% de Curacaví</li>
  <li>Kross Pilsner [Pilsner] 4,9% de Curacaví</li>
</ul>
```

Árboles
(XML, JSON, ...)



Grafos
(RDF, Prop. Gs, ...)



Relacional
(SQL, CSV, ...)

Cervezas

nombre	tipo	grados	ciudad-origen
Austral Lager	Lager	4,6	Punta Arenas
Austral Yagan	Ale	5,0	Punta Arenas
Austral Pale Ale	Ale	5,0	Punta Arenas
Kuntsmann Torobayo	Ale	5,1	Valdivia
Kross 5	Ale	7,2	Curacaví
Kross Golden	Ale	5,3	Curacaví
Kross Pilsner	Pilsner	4,9	Curacaví

← La Web

← La Web de Datos

No estructurados

¿Y si quisiéramos estructurar los datos de la Web y crear una Web de Datos, donde la pondríamos en el espectro?

Parece más razonable usar grafos dado su flexibilidad

¿Pero qué <u>formato</u> de datos podemos ocupar entonces?

Estructura

...

Formato de Descripción de Recursos

Resource Description Framework (RDF)

Modelando el mundo con triples

sujeto	*predicado*	*objeto*
Irlanda	parteDe	Europa
Irlanda	tipo	País
Irlanda	capital	Dublín

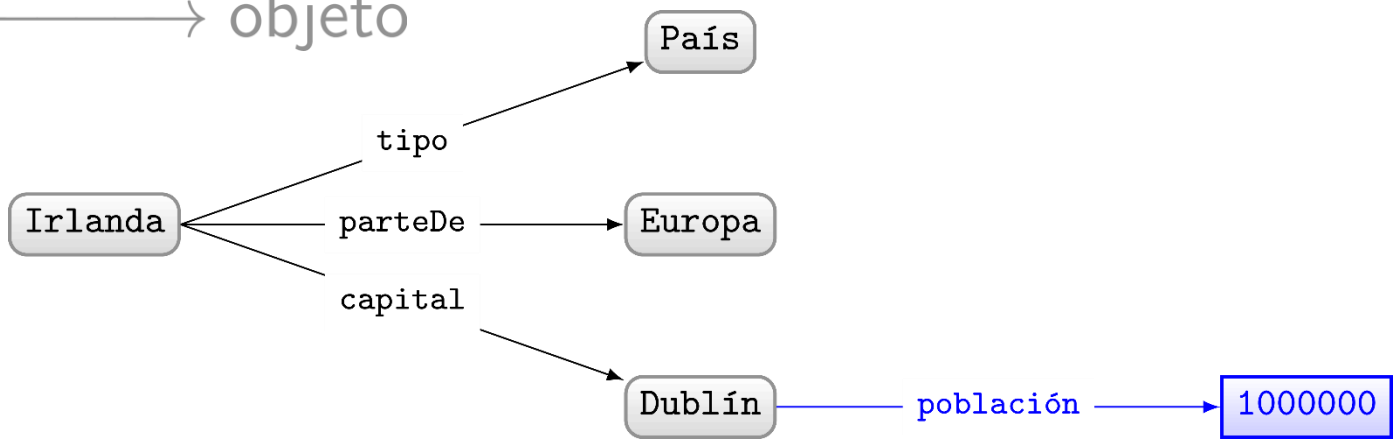
¿Agregar información? Concatenar triples.

sujeto	*predicado*	*objeto*
Irlanda	parteDe	Europa
Irlanda	tipo	País
Irlanda	capital	Dublín
Dublín	población	1,000,000

Los triples representan un grafo

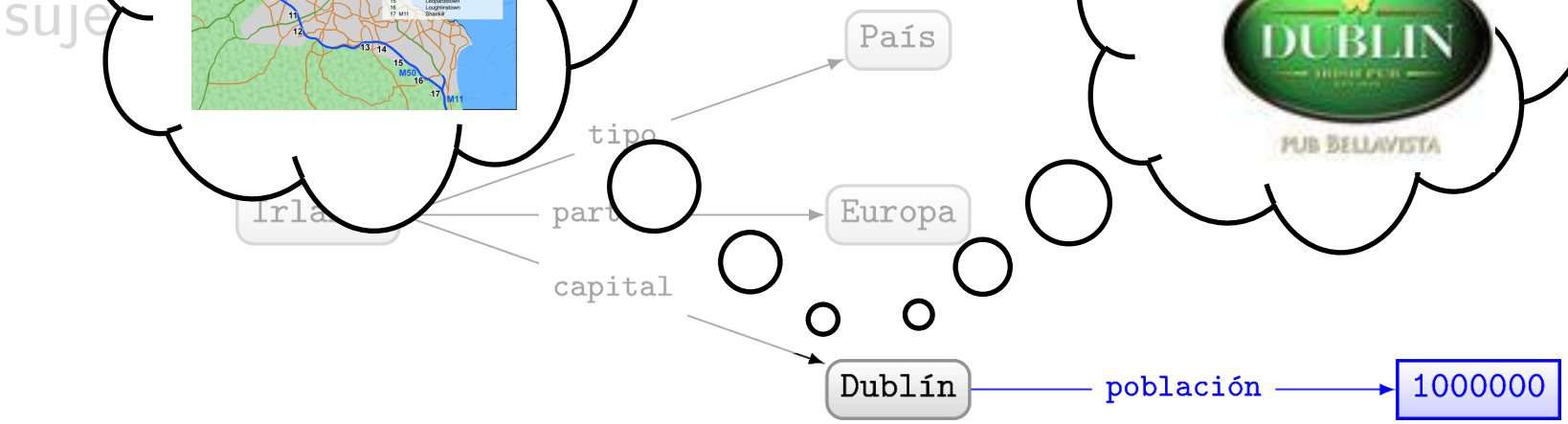
sujeto	*predicado*	*objeto*
Irlanda	parteDe	Europa
Irlanda	tipo	País
Irlanda	capital	Dublín
Dublín	población	1,000,000

sujeto $\xrightarrow{\text{predicado}}$ objeto



Los triples representan un grafo

sujeto	*predicado*	*objeto*
Irlanda	parteDe	Europa
Irlanda	tipo	País
[illegible]	capital	Dublín
[illegible]	población	1[illegible]



Idea: Usar los identificadores de la Web

- URL: *Uniform Resource Location*
  - La ubicación de un recurso de la Web
  - (p.ej., una pagina Web)
  - `http://ex.org/Dubl%C3%ADn.html`

- URI: *Uniform Resource Identifier* (RDF 1.0)
  - Un identificador de un recurso general
  - (p.ej., una ciudad)
  - `http://ex.org/Dubl%C3%ADn`

- IRI: *Internationalised Resource Identifier* (RDF 1.1)
  - Una URI que permite caracteres en *Unicode*
  - `http://ex.org/Dublín`

Usaremos IRIs con prefijos

`http://ex.org/Dublín` ↔ `ex:Dublín`

- "`ex:`" denota un prefijo para `http://ex.org/`
- "`Dublín`" en un nombre local

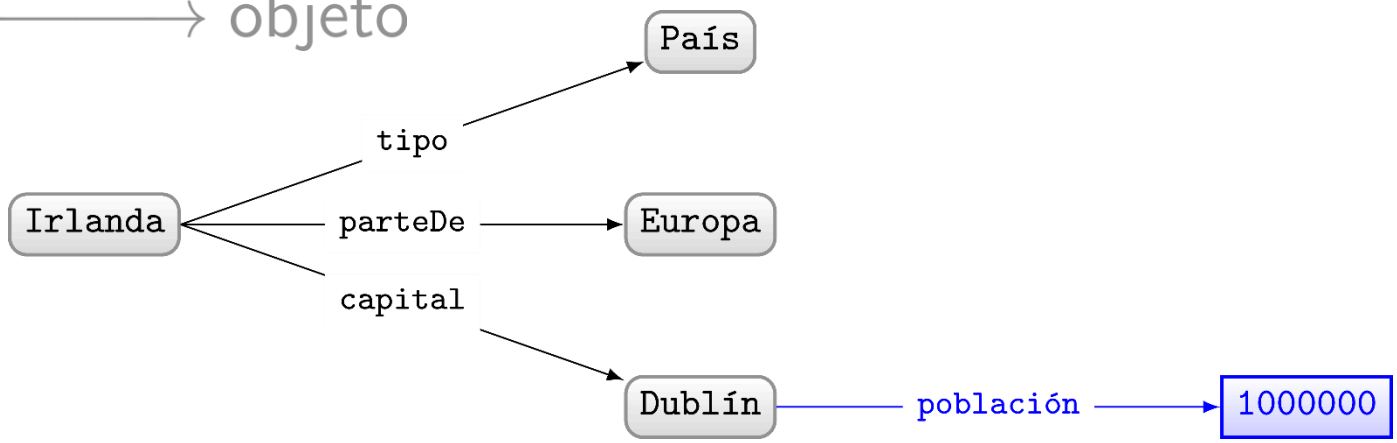
Prefijos comunes:

Prefix	Value
rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#
xsd:	http://www.w3.org/2001/XMLSchema#
rdfs:	http://www.w3.org/2000/01/rdf-schema#
owl:	http://www.w3.org/2002/07/owl#

Desde strings ...

sujeto	*predicado*	*objeto*
Irlanda	parteDe	Europa
Irlanda	tipo	País
Irlanda	capital	Dublín
Dublín	población	1,000,000

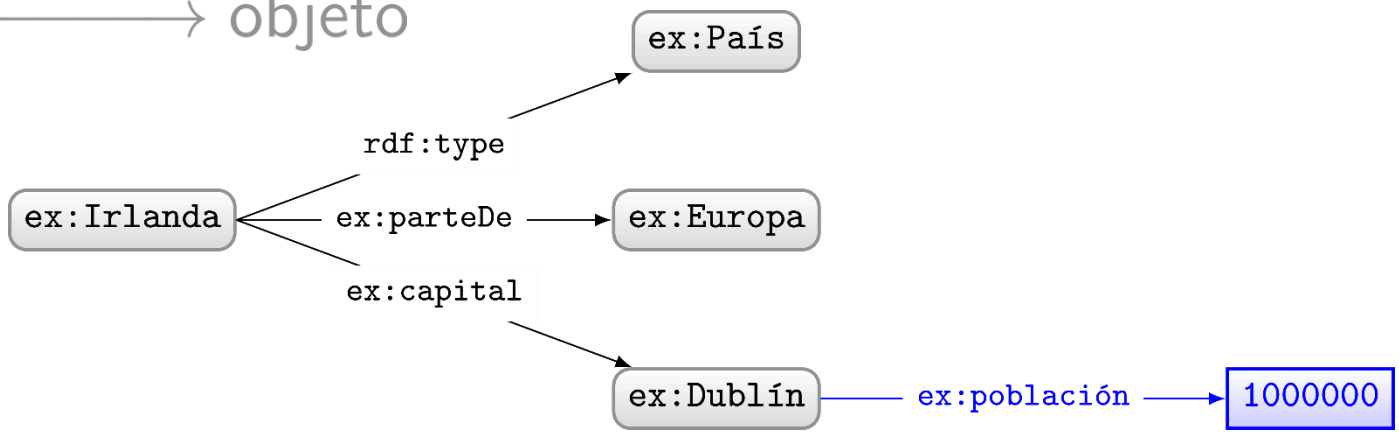
sujeto $\xrightarrow{\text{predicado}}$ objeto



… hasta identificadores de la Web …

sujeto	*predicado*	*objeto*
ex:Irlanda	ex:parteDe	ex:Europa
ex:Irlanda	rdf:type	ex:País
ex:Irlanda	ex:capital	ex:Dublín
ex:Dublín	ex:población	1,000,000

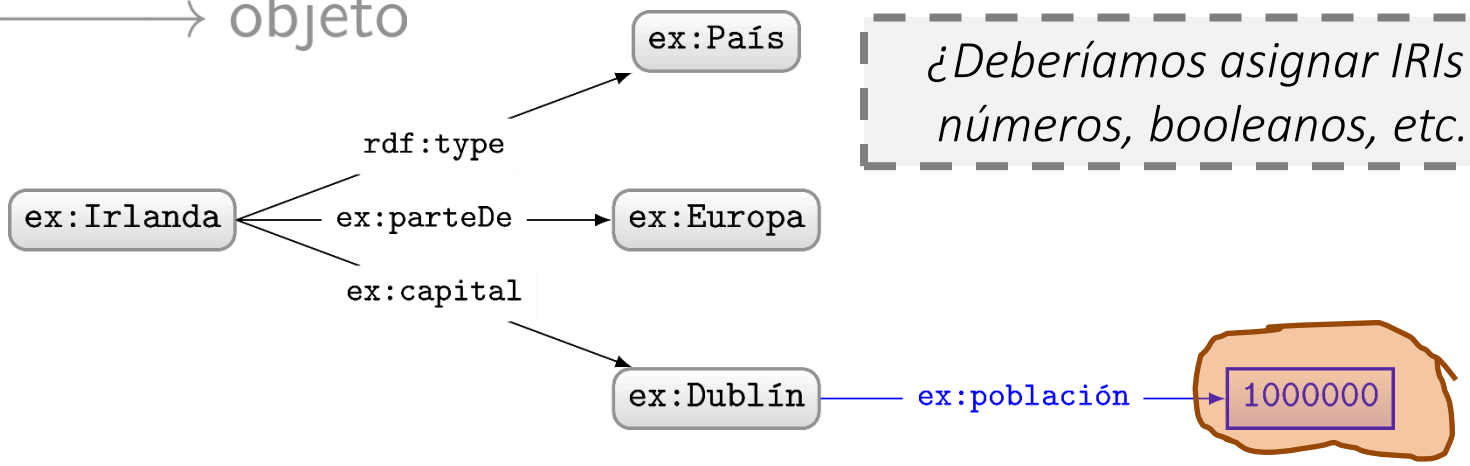
sujeto $\xrightarrow{\text{predicado}}$ objeto



… ¿y qué pasa con números, etc.? …

sujeto	*predicado*	*objeto*
ex:Irlanda	ex:parteDe	ex:Europa
ex:Irlanda	rdf:type	ex:País
ex:Irlanda	ex:capital	ex:Dublín
ex:Dublín	ex:población	1,000,000

sujeto $\xrightarrow{\text{predicado}}$ objeto



¿Deberíamos asignar IRIs a números, booleanos, etc.?

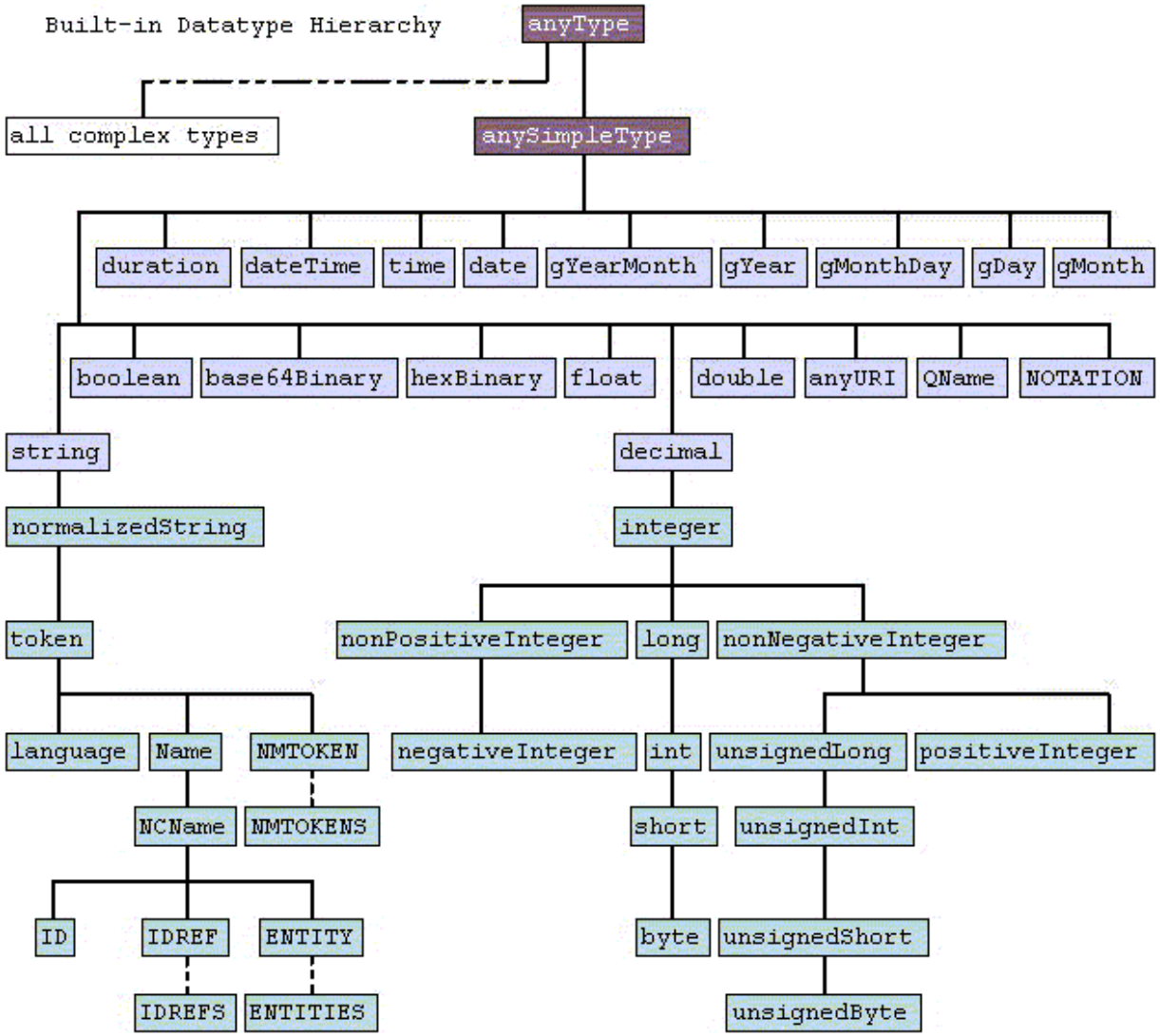
Literals representan “valores” …

… como strings, números, booleanos, fechas …

- Pero solo se permiten *literals* como objetos

sujeto	*predicado*	*objeto*	
ex:Dublín	ex:población	1,000,000	CORRECT
1,000,000	ex:población	ex:Dublín	INCORRECT
ex:Dublín	1,000,000	ex:población	INCORRECT

Se usan los tipos de XML Schema



Se usan los tipos de XML Schema

"200"^^xsd:int

"true"^^xsd:boolean

"4.065"^^xsd:decimal

"2020-02-20"^^xsd:date

"2020-02-20T12:41:10Z"^^xsd:dateTimeStamp

"Sharknado"

Por defecto un string

...

Se pueden usar etiquetas de idioma

"200"^^xsd:int

"true"^^xsd:boolean

"4.065"^^xsd:decimal

"2020-02-20"^^xsd:date

"2020-02-20T12:41:10Z"^^xsd:dateTimeStamp

"Sharknado"

"Dublin"@en "Dublín"@es-CL "Baile Átha Cliath"@ga

Etiquetas de idioma

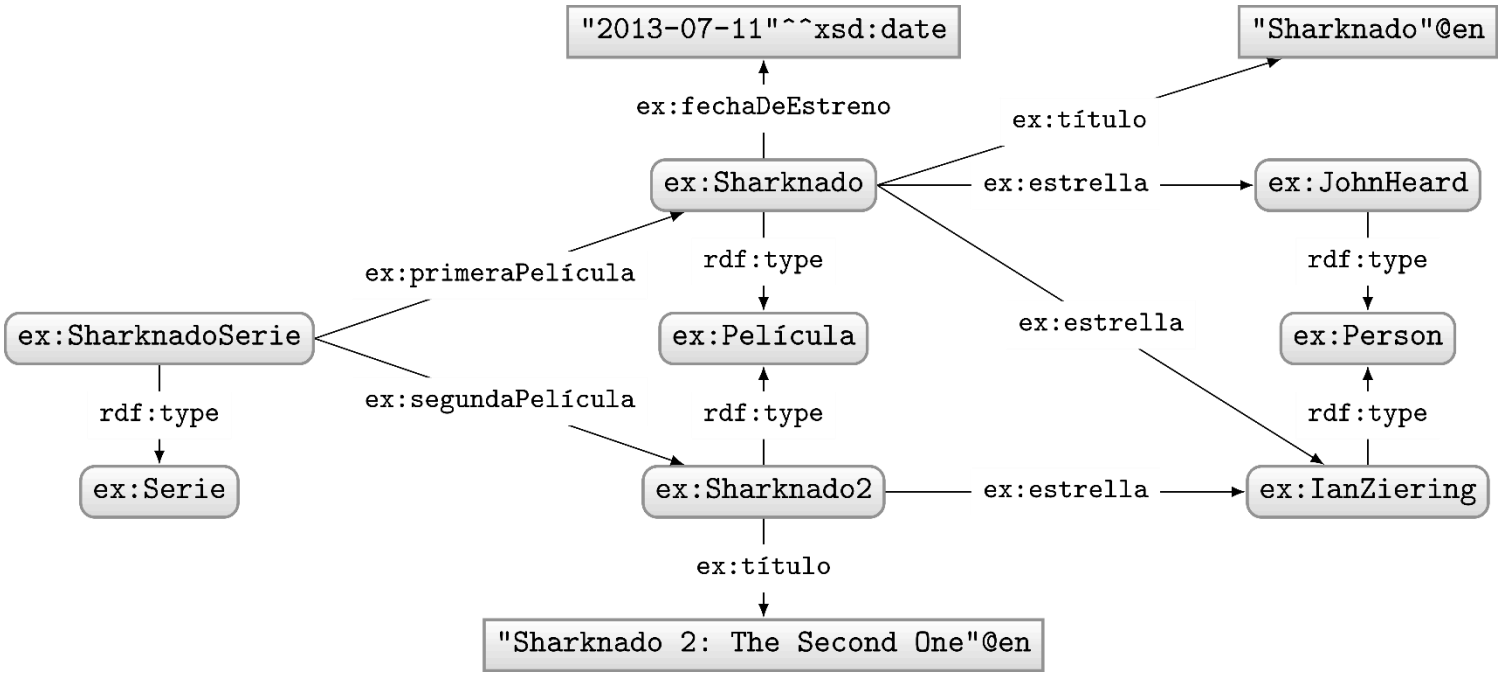
...

Protocolo y Lenguaje de Consulta de RDF:

SPARQL: SPARQL Protocol and RDF Query Langauge

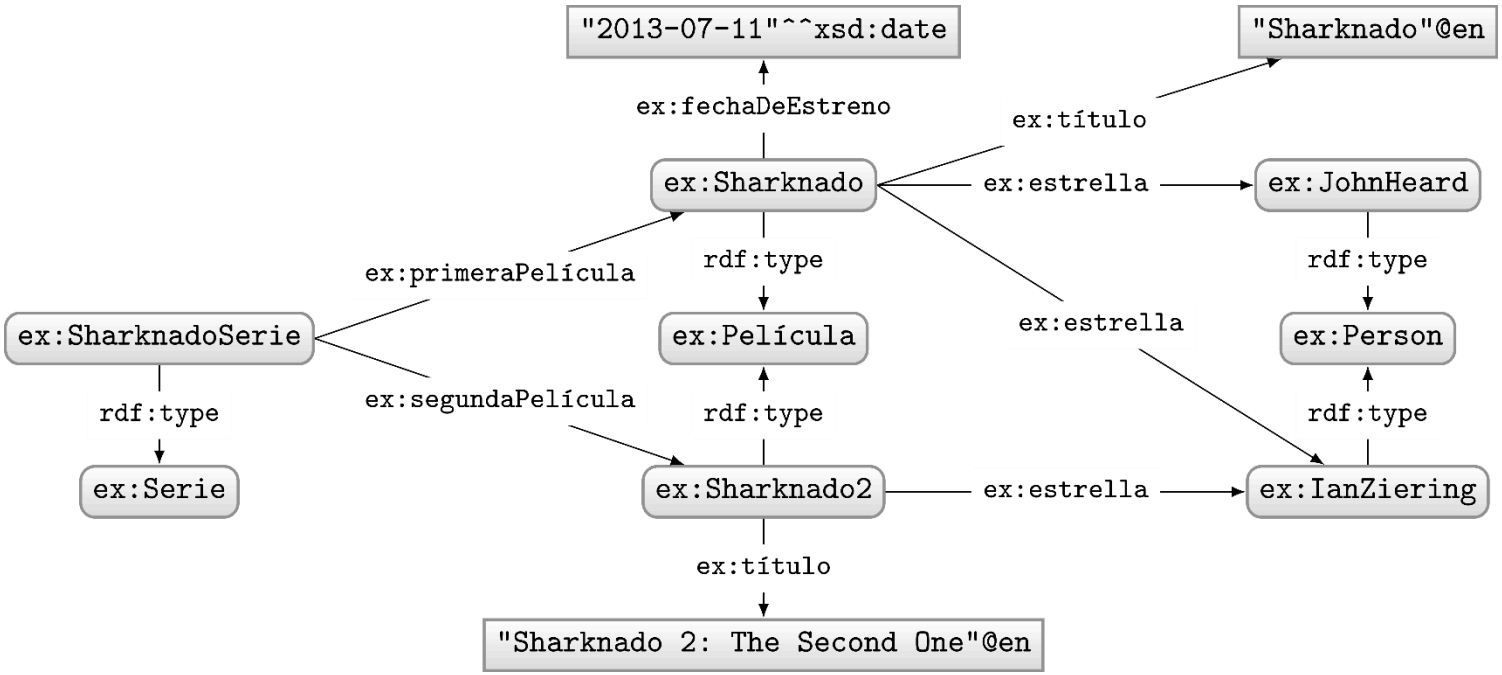
(1.1)
QL

SPARQL: Consultar Grafos en RDF



Consulta: “¿Quién protagoniza en la película ‘Sharknado’?”

SPARQL: Consultar Grafos en RDF



Consulta:

```
PREFIX ex: <http://ex.org/voc#>
SELECT *
WHERE {
  ex:Sharknado ex:estrella ?e .
}
```

Soluciones:

?e
ex:JohnHeard
ex:IanZiering

SPARQL Prefijos: Abreviaturas de IRIs

```
PREFIX ex: <http://ex.org/voc#>
SELECT *
WHERE {
  ex:Sharknado ex:estrella ?e .
}
```

SPARQL: clausula de WHERE

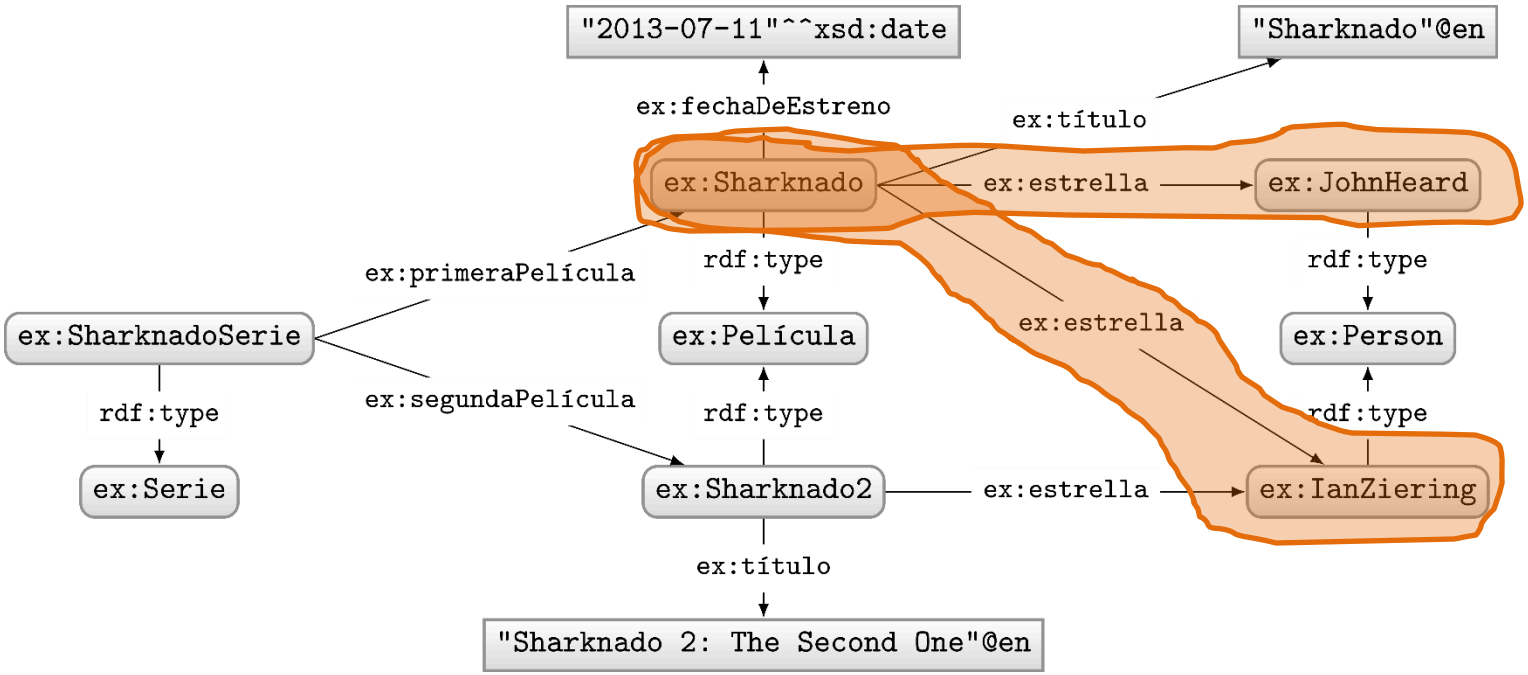
- Especifica un grafo de consulta

```
PREFIX ex: <http://ex.org/voc#>
SELECT *
WHERE {
  ex:Sharknado ex:estrella ?e .
}
```

"Patrón triple"

(un triple con variables)

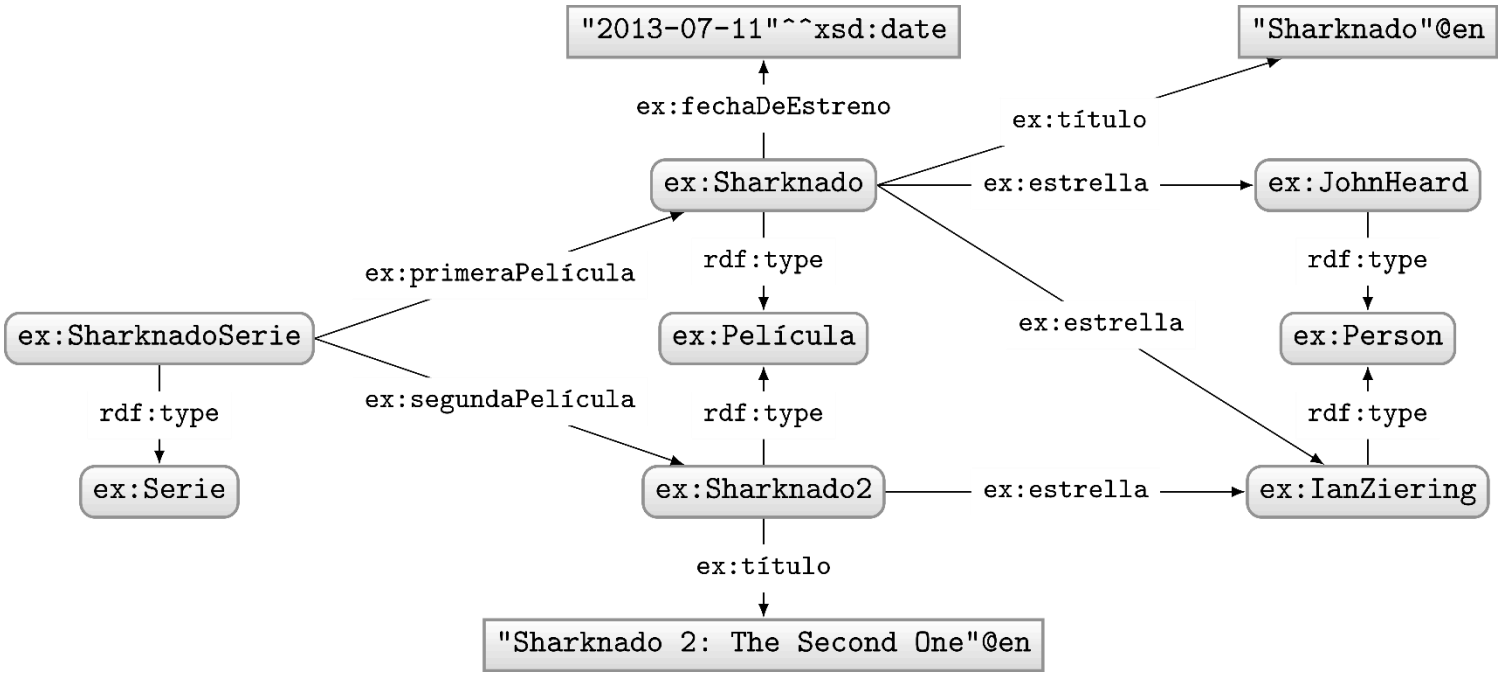
SPARQL: clausula de WHERE




```
PREFIX ex: <http://ex.org/voc#>
SELECT *
WHERE {
  ex:Sharknado ex:estrella ?e .
}
```

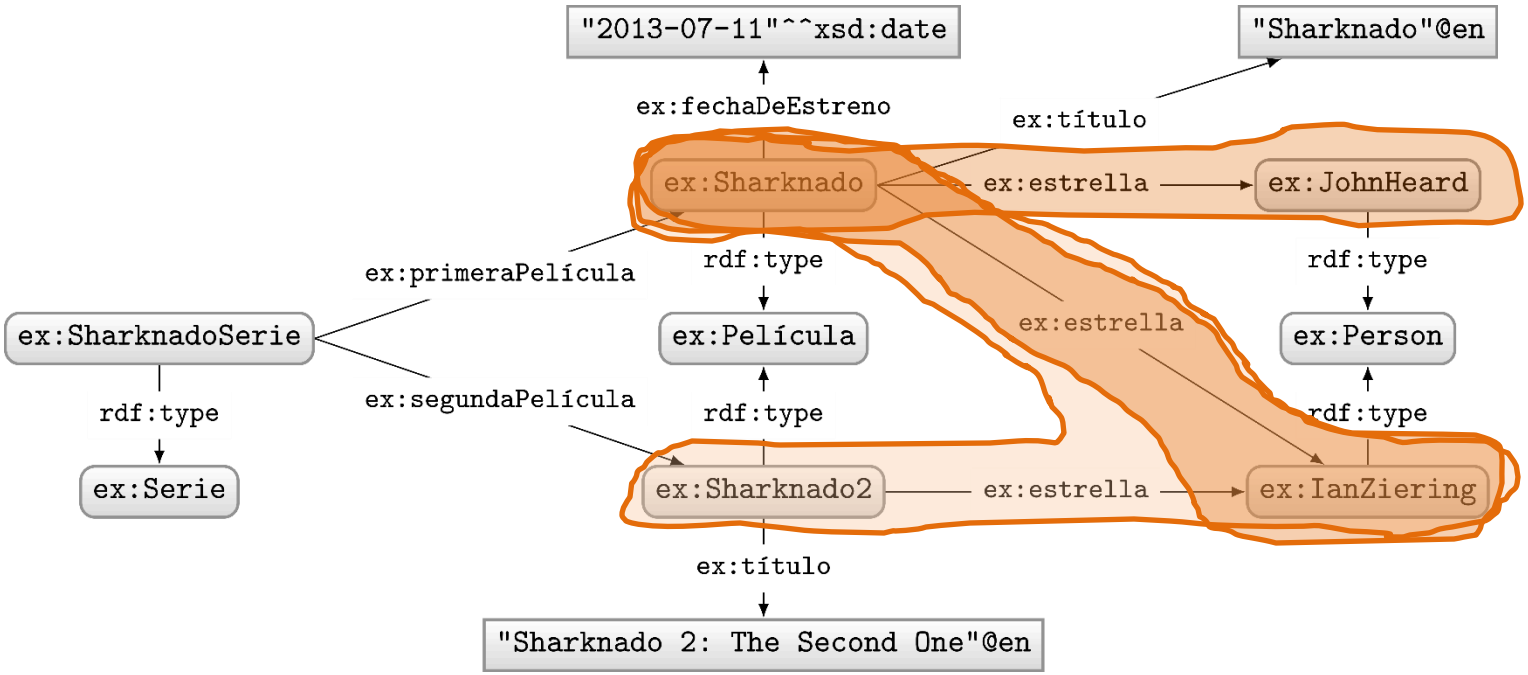
?e
ex:JohnHeard
ex:IanZiering

SPARQL: clausula de WHERE



Consulta: “¿En cuáles (otras) películas han actuado los actores de Sharknado?”

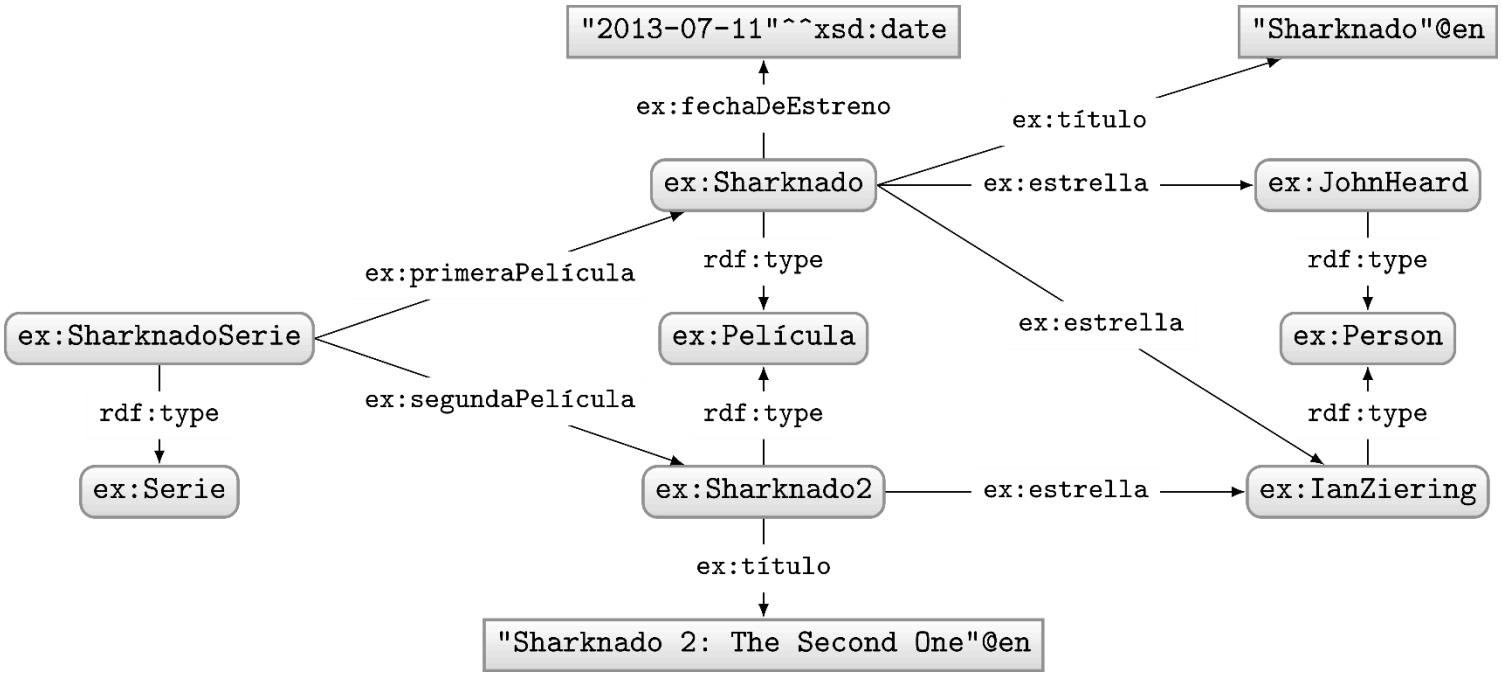
SPARQL: clausula de WHERE

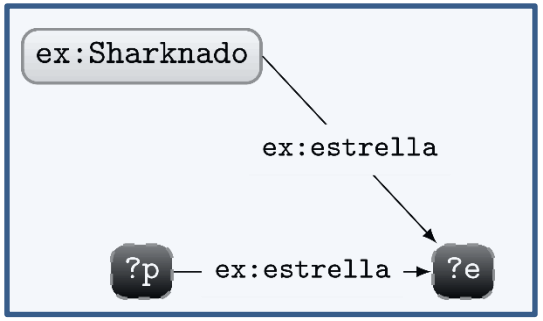



```
PREFIX ex: <http://ex.org/voc#>
SELECT *
WHERE {
  ex:Sharknado ex:estrella ?e .
  ?p ex:estrella ?e .
}
```

?e	?p
ex:IanZiering	ex:Sharknado2
ex:IanZiering	ex:Sharknado
ex:JohnHeard	ex:Sharknado

SPARQL: clausula de WHERE



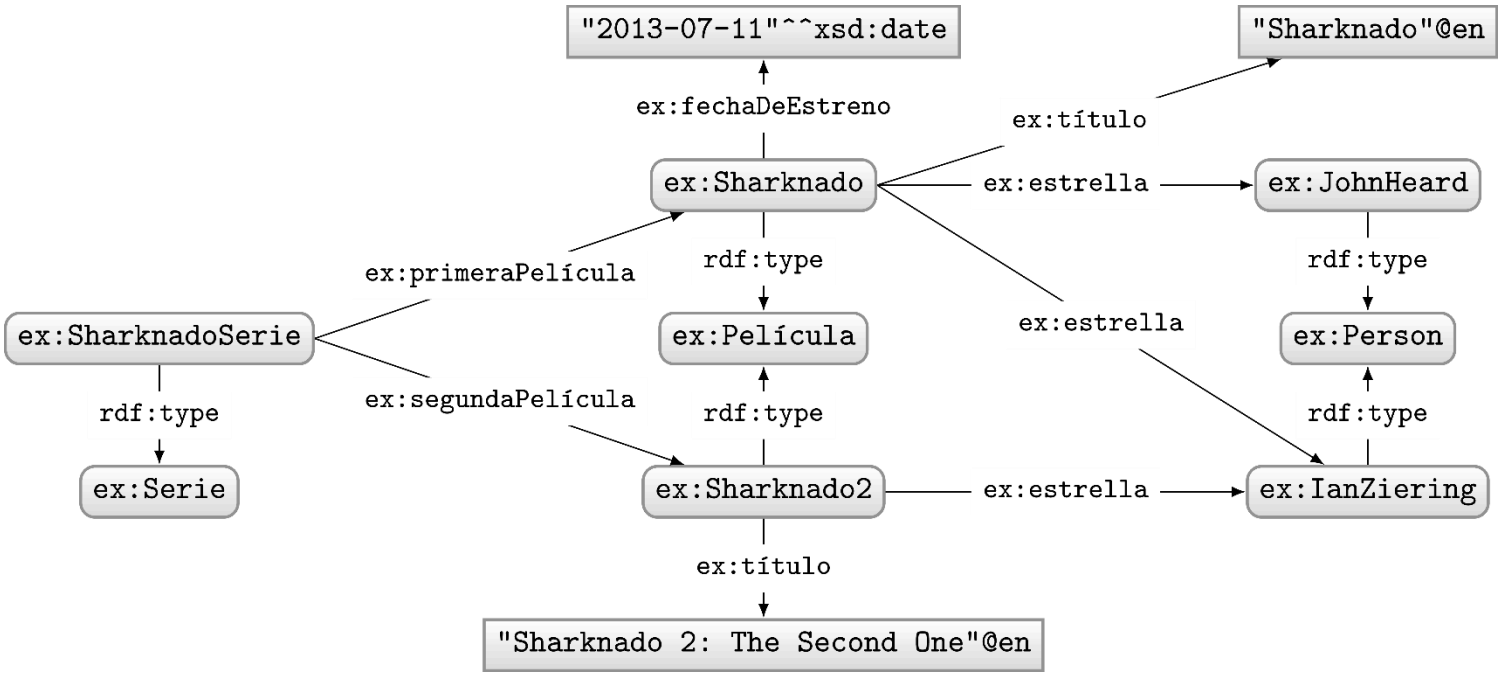



```
PREFIX ex: <http://ex.org/voc#>
SELECT *
WHERE {
  ex:Sharknado ex:estrella ?e .
  ?p ex:estrella ?e .
}
```

"*Basic Graph Pattern*"

(un conjunto de patrones triples)

SPARQL: *Joins*

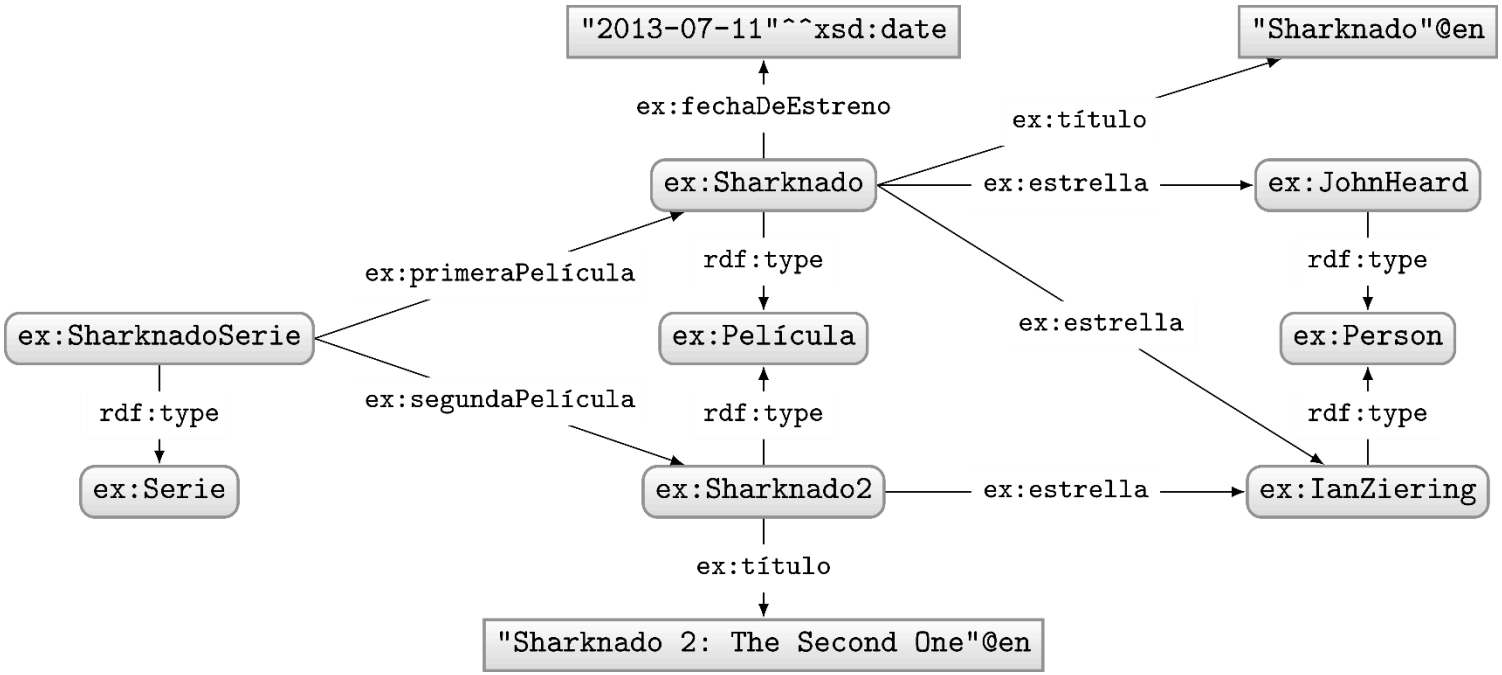



```
PREFIX ex: <http://ex.org/voc#>
SELECT *
WHERE {
  ex:Sharknado ex:estrella ?e .
  ?p ex:estrella ?e .
}
```

"Variable de Join"

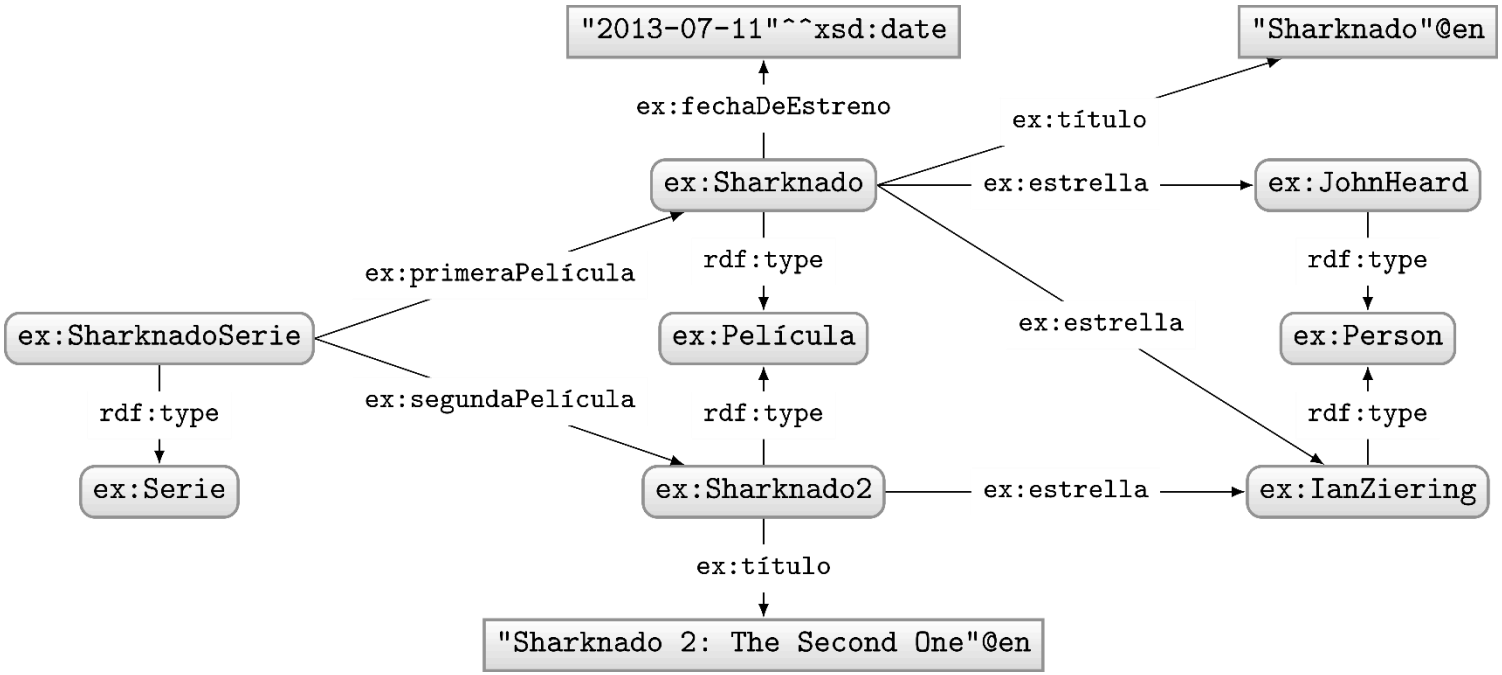
(una variable en múltiples lugares)

SPARQL: Unión



Consulta: “¿Qué son los títulos de las dos primeras películas en la serie Sharknado?”

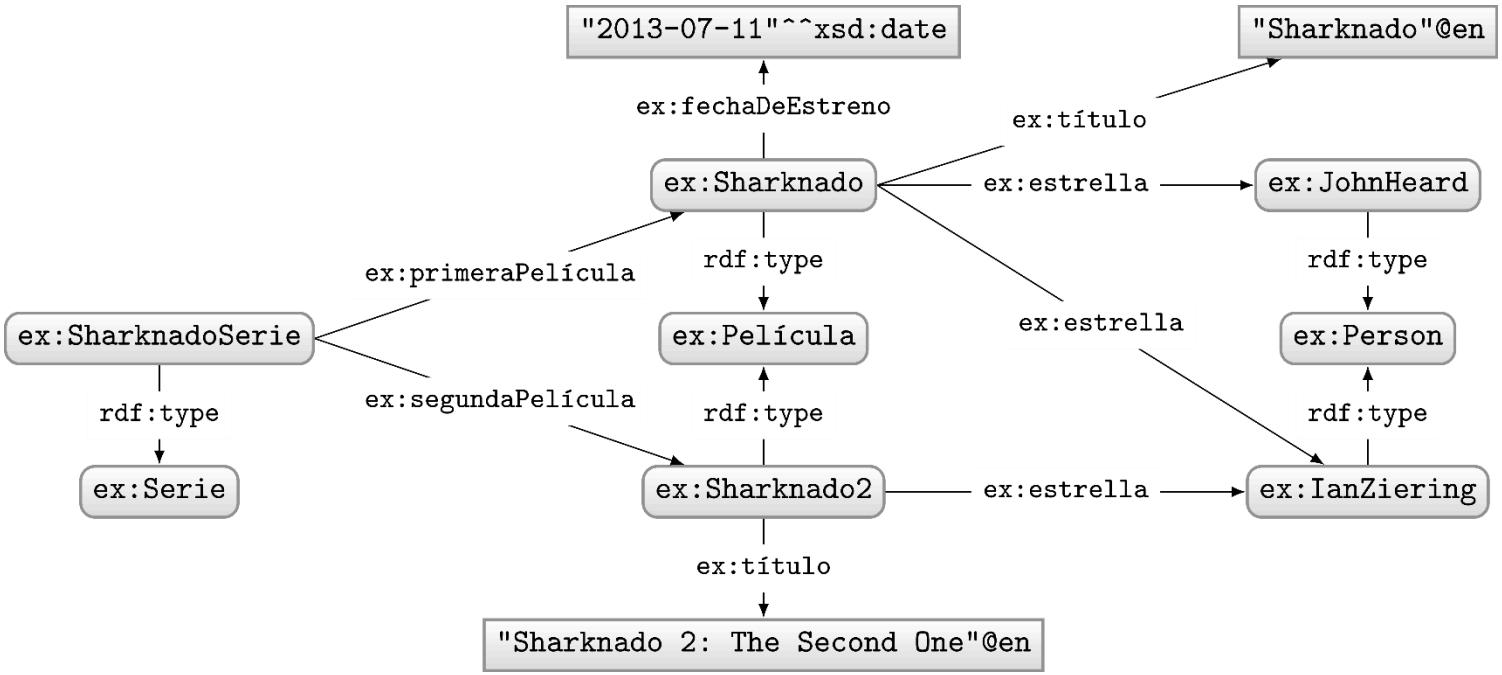
SPARQL: Unión




```
PREFIX ex: <http://ex.org/voc#>
SELECT *
WHERE {
  { ex:SharknadoSeries ex:primeraPelícula ?p . }
  UNION
  { ex:SharknadoSeries ex:segundaPelícula ?p . }
  ?p ex:título ?t .
}
```

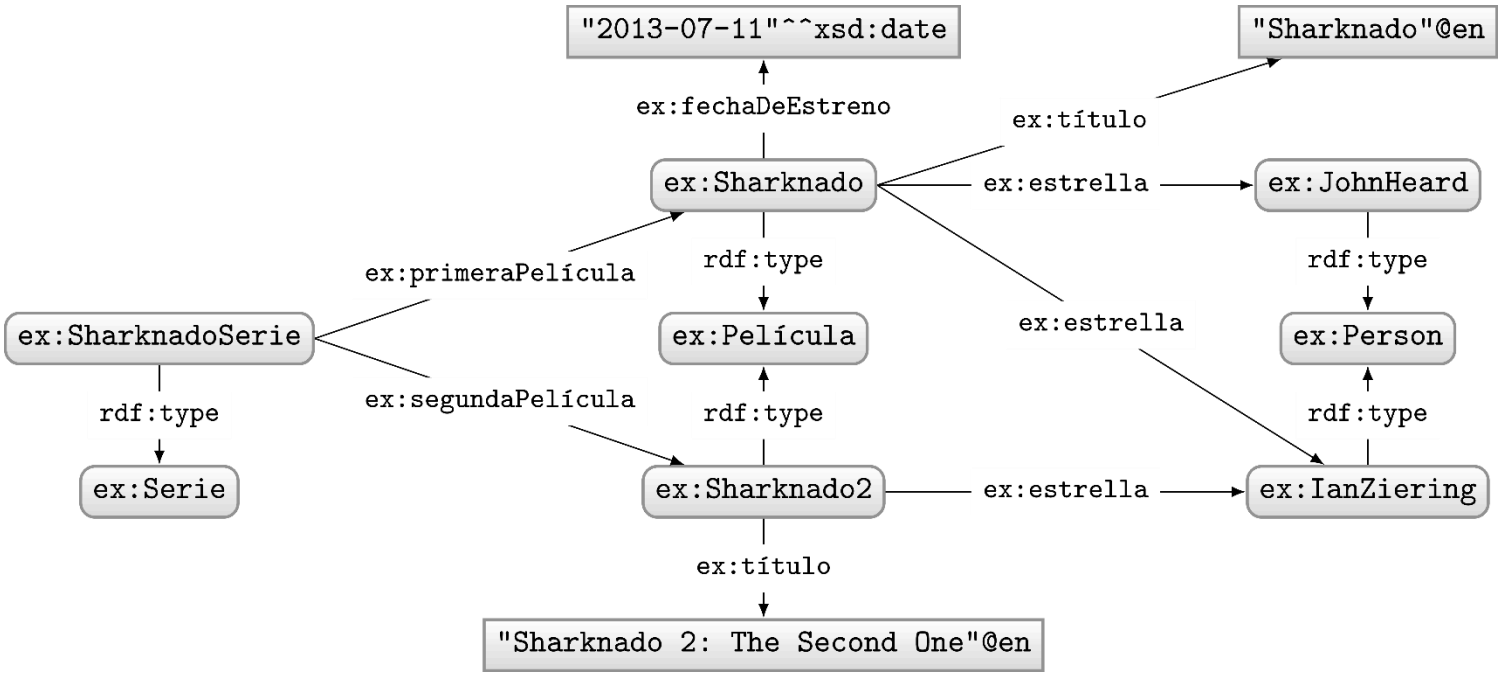
?p	?t
ex:Sharknado	"Sharknado"@en
ex:Sharknado2	"Sharknado 2: The Second One"@en

SPARQL: *Left-join* (OPTIONAL)



Consulta: “¿Los títulos de películas y (de estar disponible) sus fechas de estreno?”

SPARQL: *Left-join* (OPTIONAL)



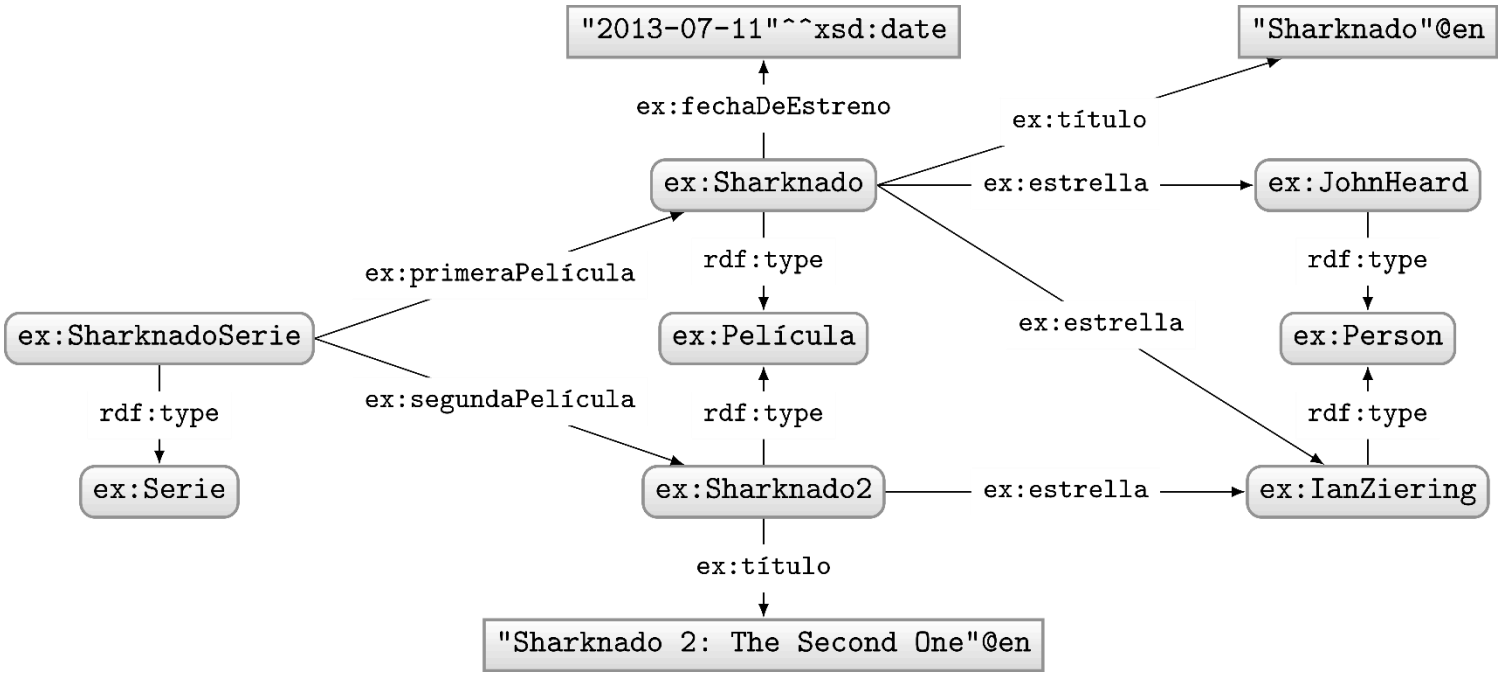

```
PREFIX ex: <http://ex.org/voc#>
SELECT *
WHERE {
  ?p a ex:Película .
  ?p ex:título ?t .
  OPTIONAL
     { ?p ex:fechaDeEstreno ?f }
}
```

?p	?t	?f
ex:Sharknado	"Sharknado"@en	"2013-07-11"^^xsd:date
ex:Sharknado2	"Sharknado 2: The Second One"@en	

"Variable UNBOUND"

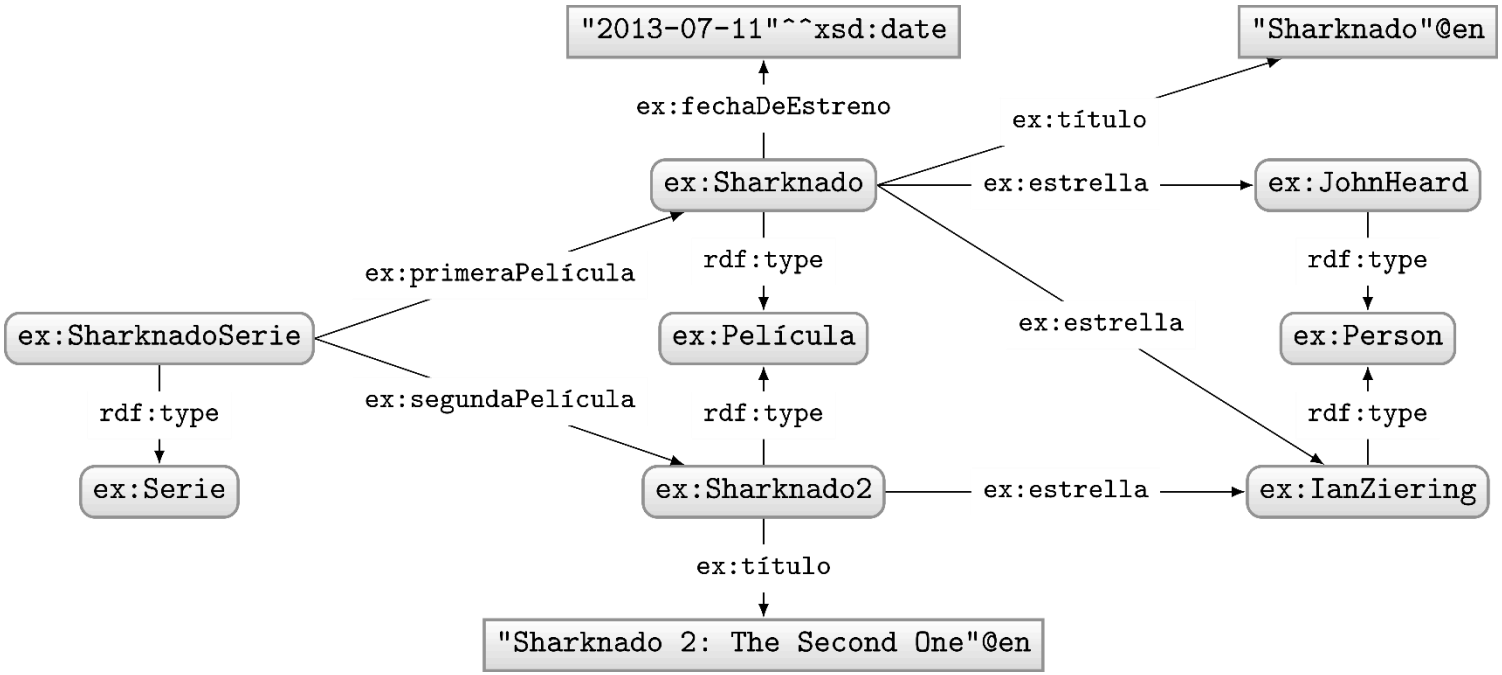
(una variable sin una solución)

SPARQL: Filtros



Consulta: “¿Cuáles películas estrenaron en 2014?”

SPARQL: Filtros

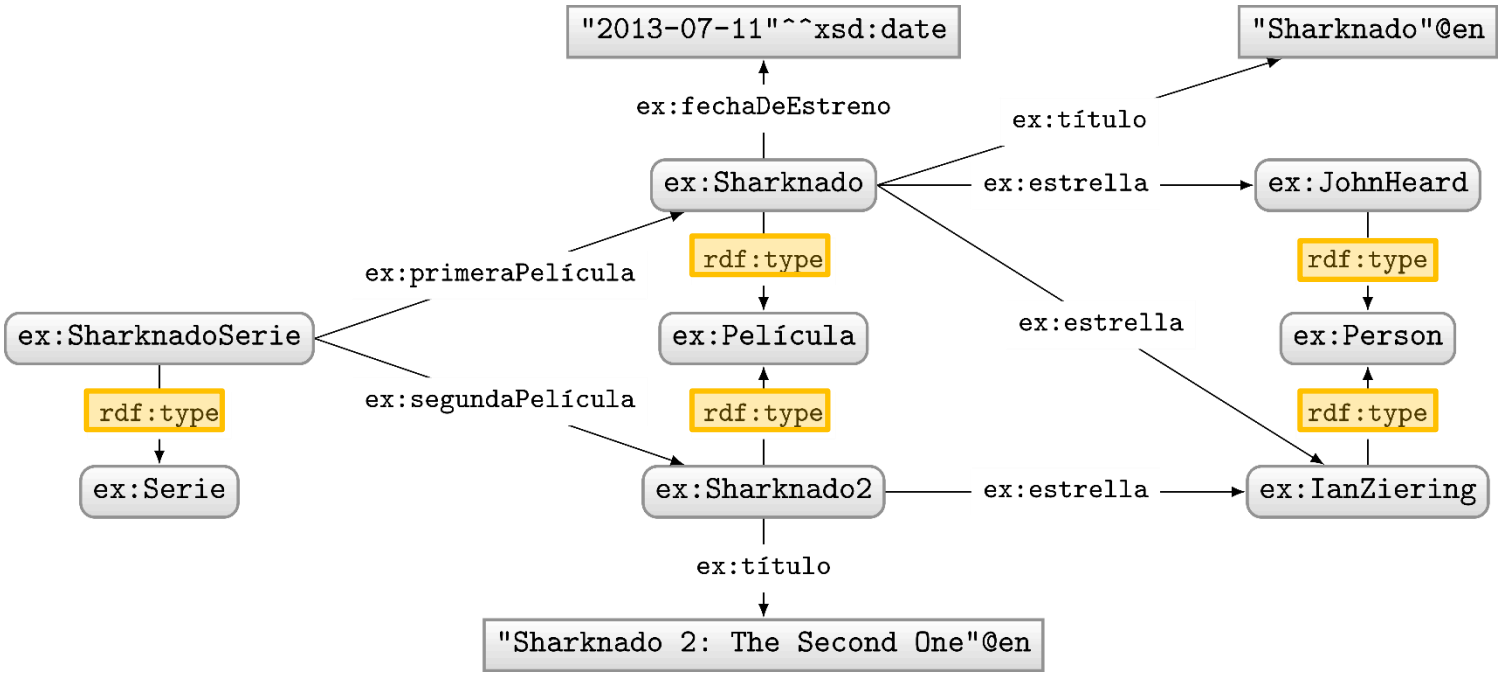



```
PREFIX ex: <http://ex.org/voc#>
SELECT *
WHERE {
  ?p a ex:Película .
  ?p ex:fechaDeEstreno ?f .
  FILTER(?f > "2013-12-31"^^xsd:date
     && ?f <= "2014-12-31"^^xsd:date)
}
```

?p	?f

Resultados vacíos

SPARQL: Filtros

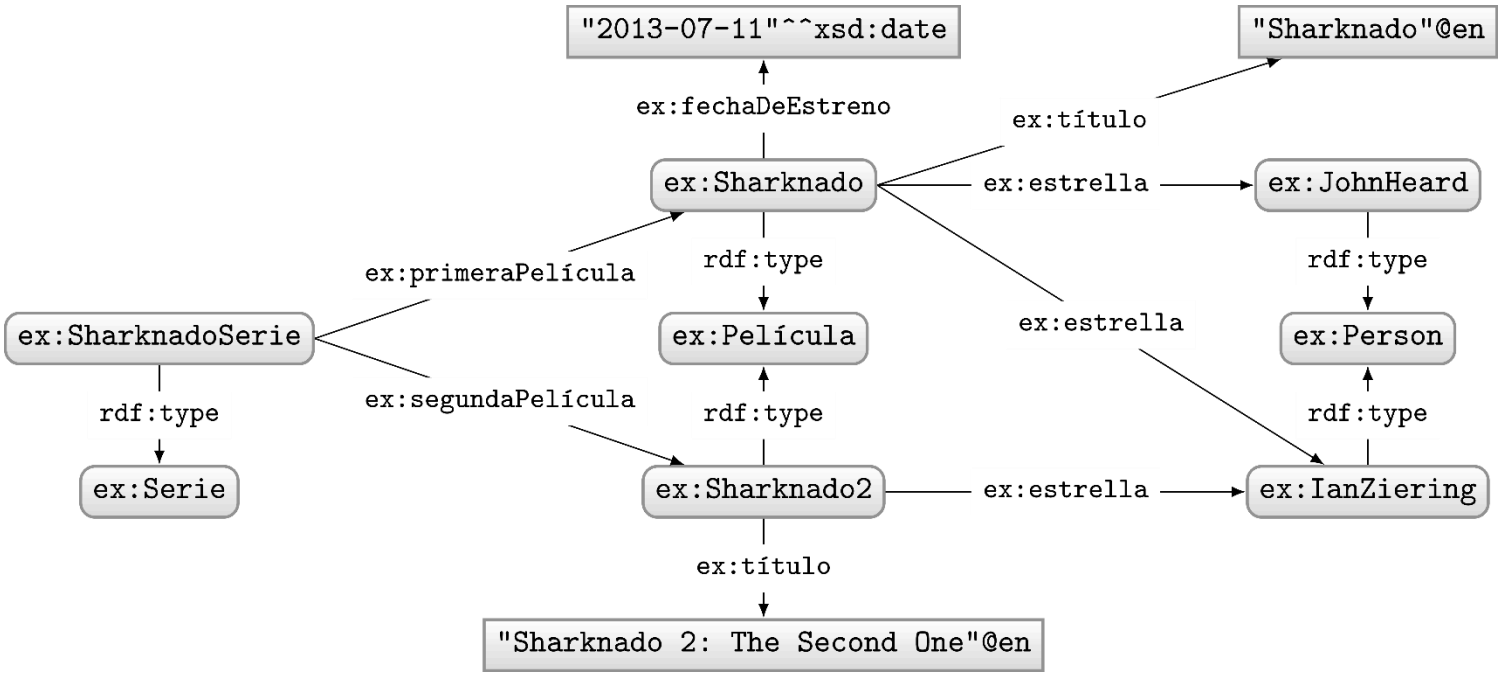



```
PREFIX ex: <http://ex.org/voc#>
SELECT *
WHERE {
  ?p a ex:Película .
  ?p ex:fechaDeEstreno ?f .
  FILTER(?f > "2013-12-31"^^xsd:date
     && ?f <= "2014-12-31"^^xsd:date)
}
```

?p	?f

Una abreviatura

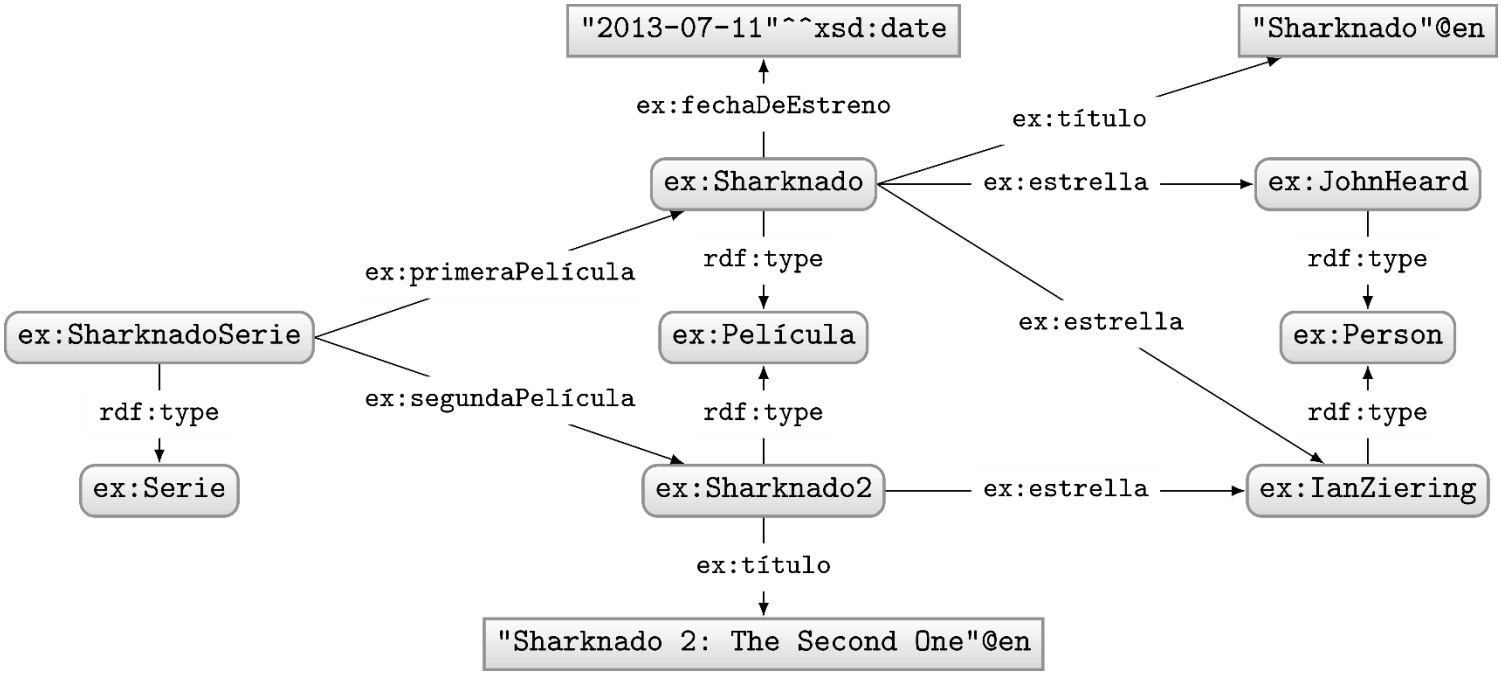
SPARQL: clausula de WHERE (otro ejemplo)




```
PREFIX ex: <http://ex.org/voc#>
SELECT *
WHERE {
  { ex:SharknadoSerie ex:primeraPelícula ?p . }
  UNION
  { ex:SharknadoSerie ex:segundaPelícula ?p . }
  OPTIONAL
  { ?p ex:fechaDeEstreno ?f . }
  ?p ex:título ?t .
  FILTER(REGEX(STR(?t),"*[0-9]*"))
}
```

?p	?t	?f
ex:Sharknado2	"Sharknado 2: The Second One"@en	

SPARQL: clausula de WHERE (otro ejemplo)

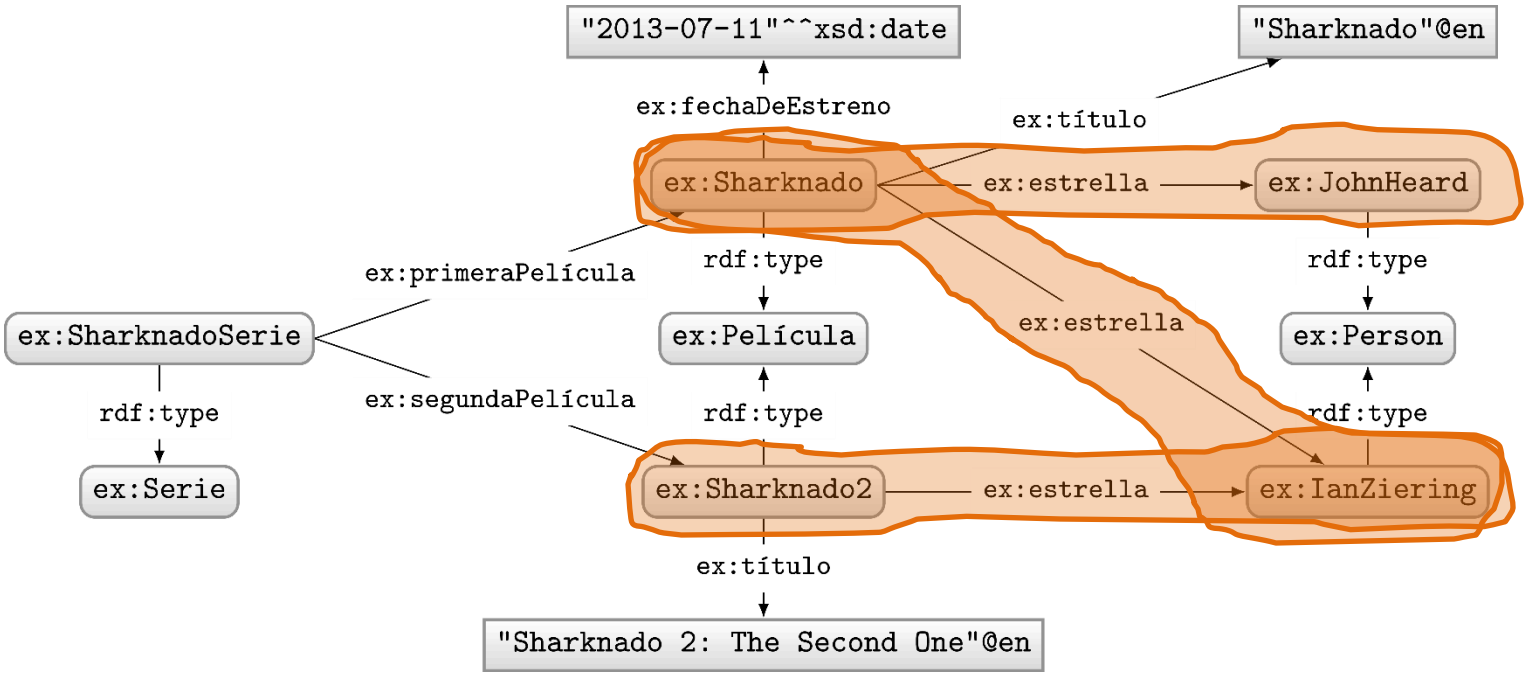



```
PREFIX ex: <http://ex.org/voc#>
SELECT *
WHERE {
  ?p a ex:Película .
  OPTIONAL
  { ?p ex:fechaDeEstreno ?f . }
  FILTER(!BOUND(?f))
}
```

?p	?f
ex:Sharknado2	

Como NOT EXISTS/EXCEPT!

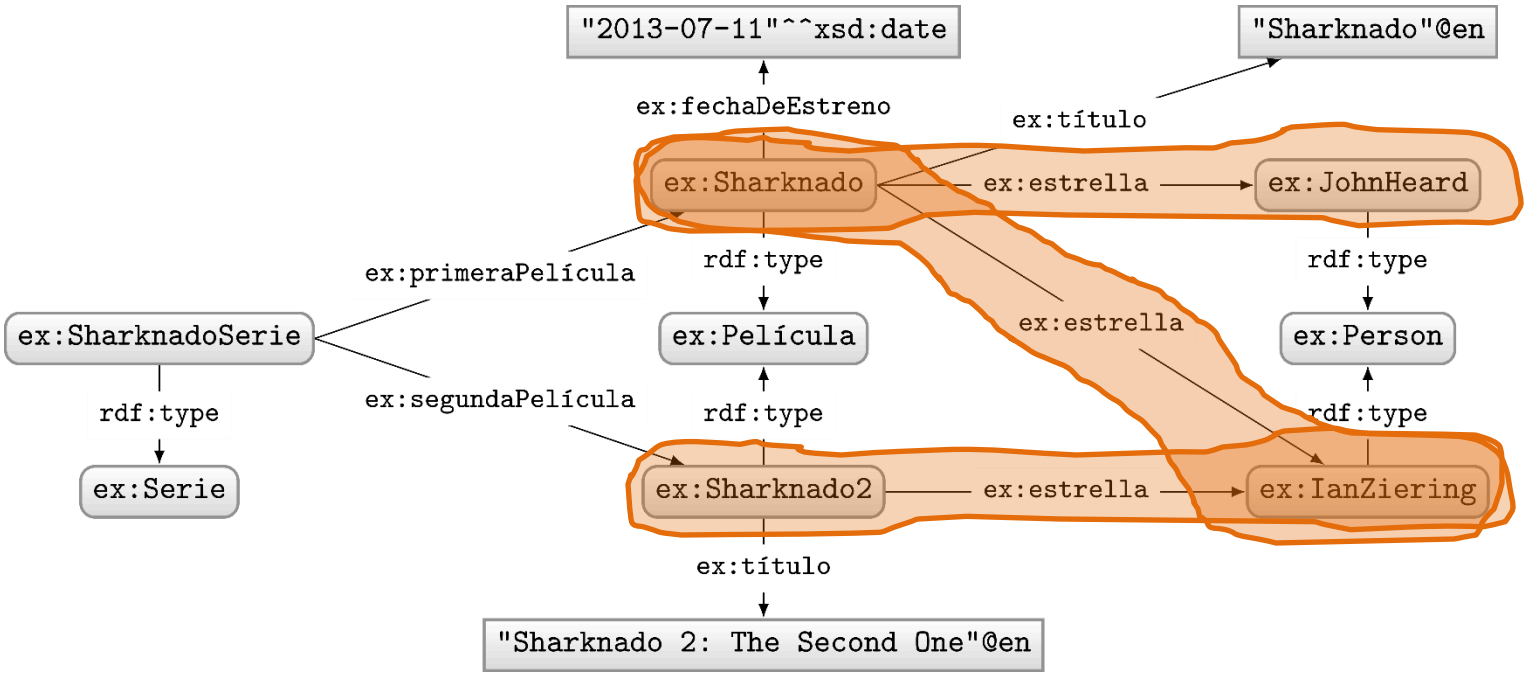
SPARQL: SELECT con *




```
PREFIX ex: <http://ex.org/voc#>
SELECT *
WHERE {
  ?p a ex:Película .
  ?p ex:estrella ?e .
}
```

?p	?e
ex:Sharknado	ex:JohnHeard
ex:Sharknado	ex:IanZiering
ex:Sharknado2	ex:IanZiering

SPARQL: SELECT con proyección

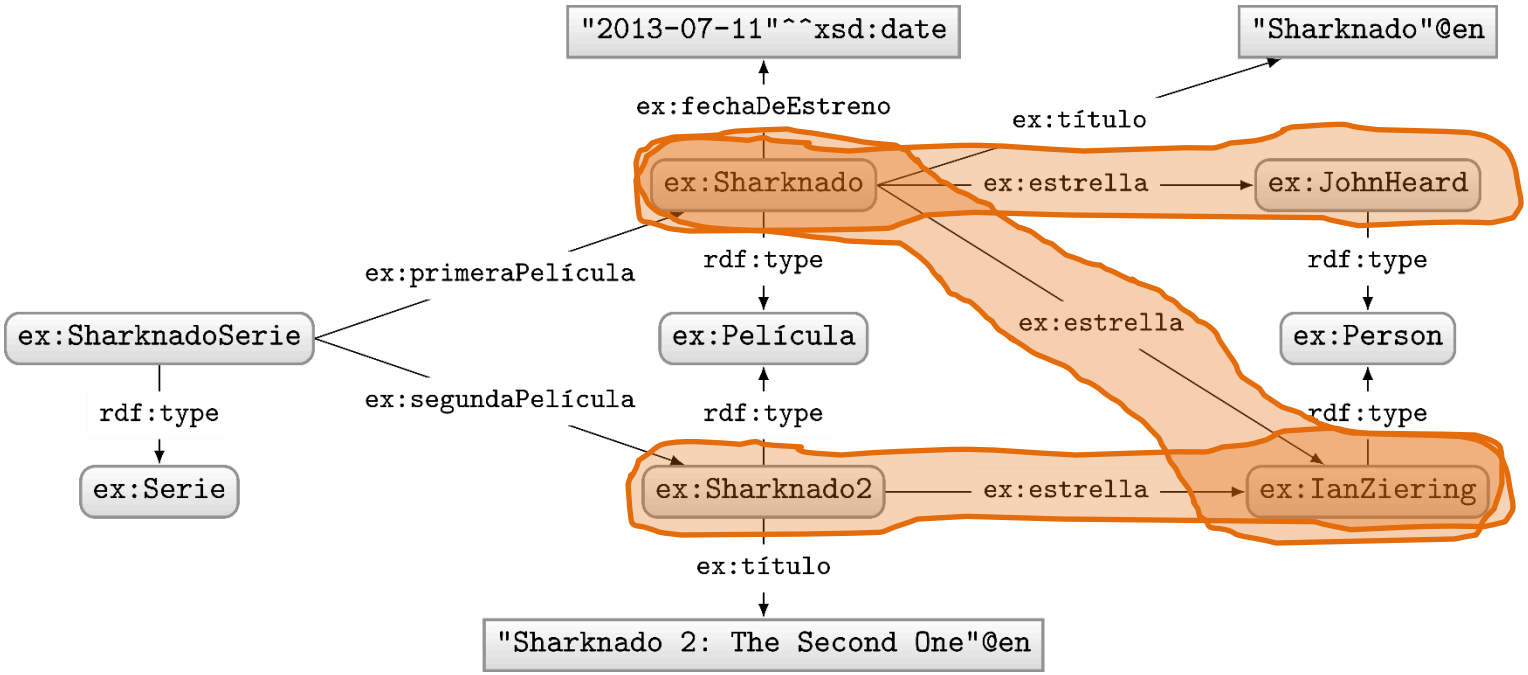



```
PREFIX ex: <http://ex.org/voc#>
SELECT ?e
WHERE {
  ?p a ex:Película .
  ?p ex:estrella ?e .
}
```

?e
ex:JohnHeard
ex:IanZiering
ex:IanZiering

Devuelve duplicados

SPARQL: SELECT con DISTINCT

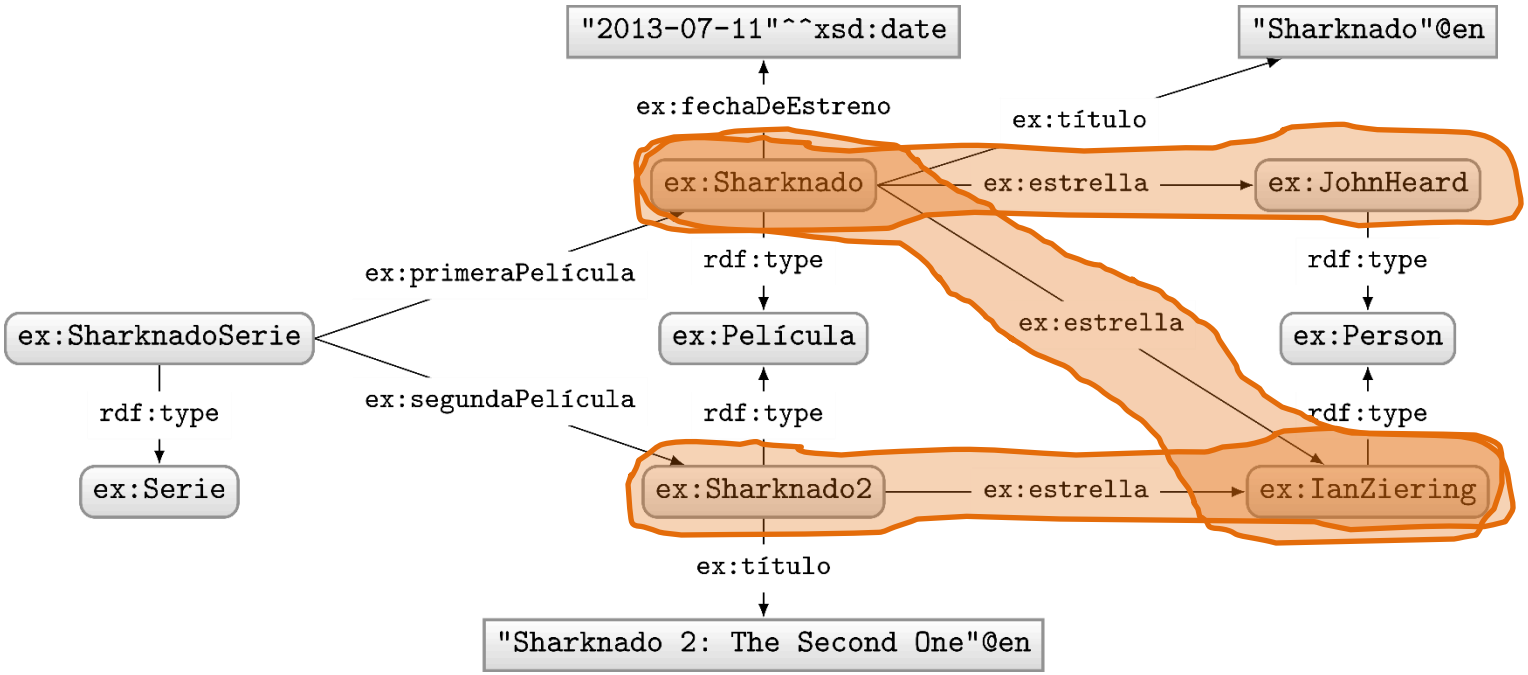



```
PREFIX ex: <http://ex.org/voc#>
SELECT DISTINCT ?e
WHERE {
  ?p a ex:Película .
  ?p ex:estrella ?e .
}
```

?e
ex:JohnHeard
ex:IanZiering

(no hay duplicados)

SPARQL: ASK



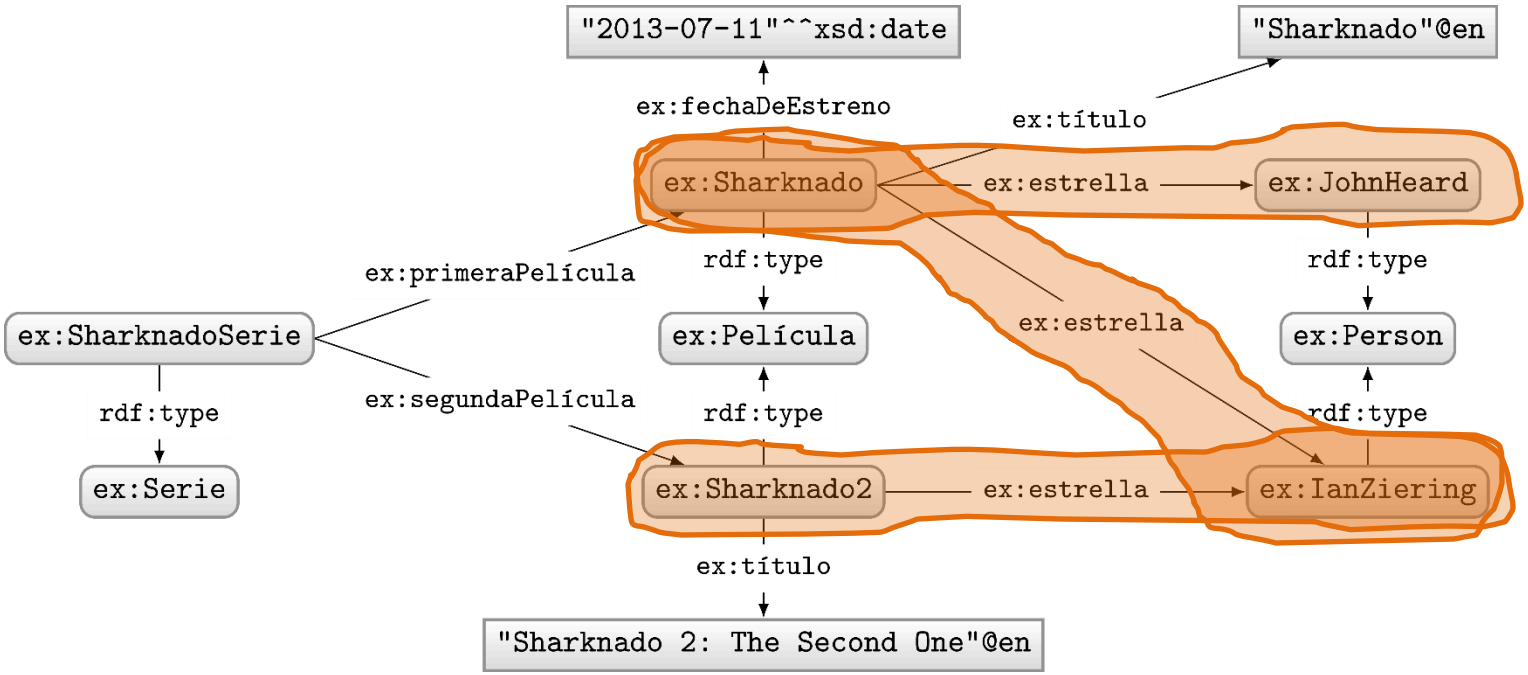

```
PREFIX ex: <http://ex.org/voc#>
ASK
WHERE {
  ?p a ex:Película .
  ?p ex:estrella ?e .
}
```

true

true si hay al menos un resultado,

false si no.

SPARQL: CONSTRUCT




```
PREFIX ex: <http://ex.org/voc#>
CONSTRUCT { ?e ex:empleo ex:Actor }
WHERE {
  ?p a ex:Película .
  ?p ex:estrella ?e .
}
```

```
@prefix ex: <http://ex.org/voc#> .
ex:JohnHeard ex:empleo ex:Actor .
ex:IanZiering ex:empleo ex:Actor .
```

Devuelve un grafo de RDF

Modificadores: ORDER BY, LIMIT, OFFSET

Consulta: “La segunda película y la tercera película más recientes”

```
PREFIX ex: <http://ex.org/voc#>
SELECT ?p
WHERE { ?p ex:fechaDeEstreno ?f . }
ORDER BY DESC(?f)
LIMIT 2
OFFSET 1
```

Expresiones de Caminos: *"Property paths"*

e es definido recursivamente como	
p	un predicado simple
^e	un camino inverso
e_1/e_2	e_1 seguido por e_2
$e_1 \mid e_2$	e_1 o e_2
$e*$	cero o más de e
$e+$	uno o más de e
$e?$	cero o uno de e
!p	no p
(e)	parénthesis indican precedencia

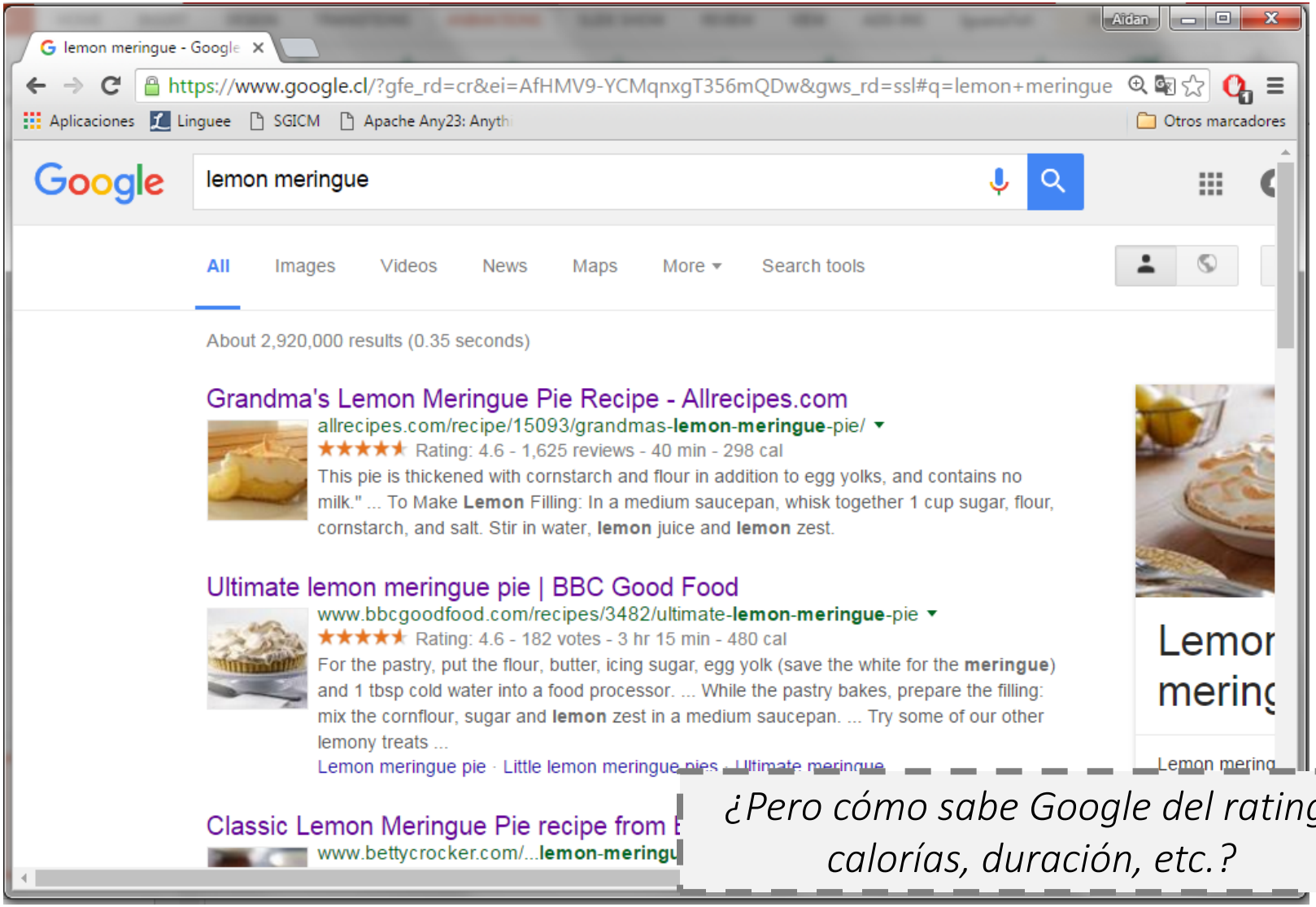
Consulta: "Los actores con un 'Número de Bacon' finito"

```
PREFIX ex: <http://ex.org/voc#>
SELECT ?k
WHERE {
  ex:KevinBacon (ex:estrella/^ex:estrella)* ?k .
}
```

La Web de Datos... hoy en día

Google's "*Rich Snippets*"

Resultados lujosos …



¿Pero cómo sabe Google del rating, calorías, duración, etc.?

Datos anidados en los documentos

```
...
<ul class="nutrition" itemprop="nutrition" itemscope
   itemtype="http://schema.org/NutritionInformation">
  <li>
    <span class="nutrition__label">kcal</span>
    <span class="nutrition__value" itemprop="calories">480</span>
    ...
  </li>
</ul>
...
```

- ✓ La pública recibe más clics
- ✓ Google puede crear resultados lujosos

schema.org (Bing, Google, Yahoo!, Yandex)

schema.rdfs.org

Home Learn Tools Mappings FAQ About

What is Schema.RDFS.org?

In early June 2011, the three big search engines Bing, Google and Yahoo! introduced Schema.org, a collection of terms that webmasters can use to markup their pages to improve the display of search results. This site is a complementary effort by people from the Linked Data community to support Schema.org deployment and usage with a special focus on Linked Data:

- We provide markup examples and tutorials about publishing & consuming data with Schema.org terms.
- We maintain mappings from Web Data vocabularies such as the DBpedia ontology to Schema.org terms.
- We list tools and libraries that are able to consume or produce Schema.org-based data.
- We automatically scrape the Schema.org terms on a daily basis and generate the following formats:

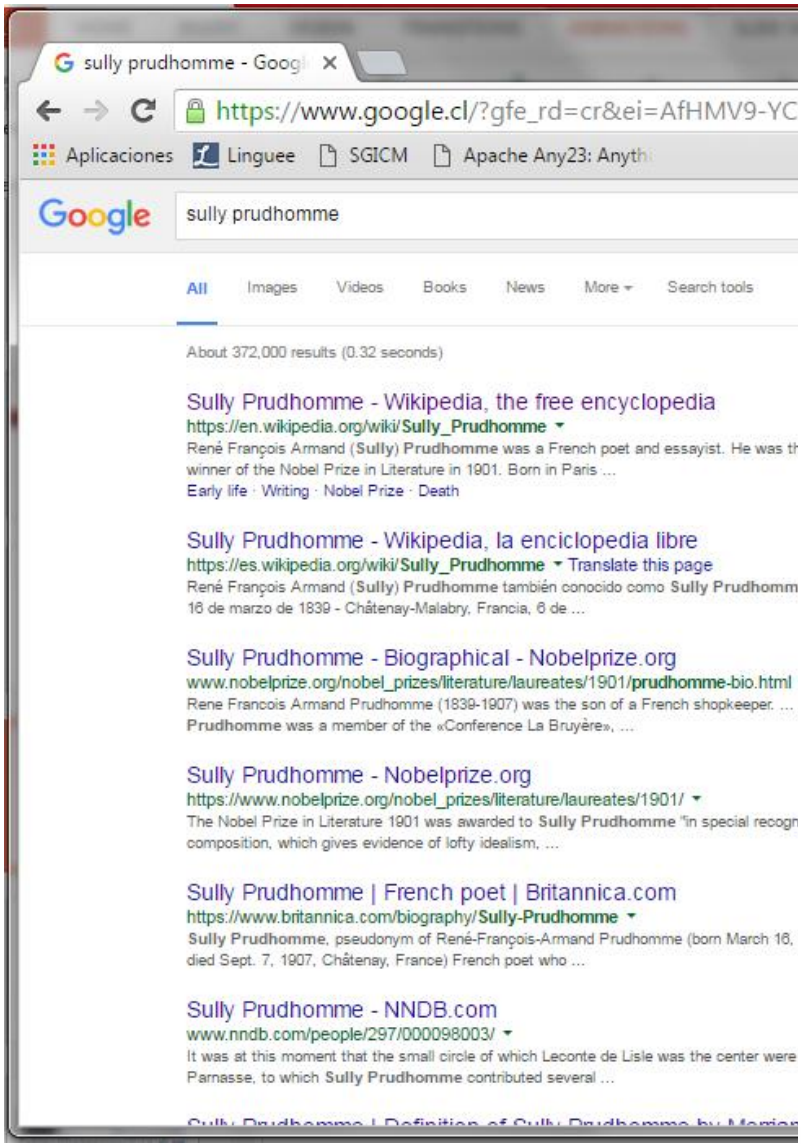
RDF/Turtle RDF/XML RDF/NTriples

JSON CSV: all classes, all properties

Note that the official OWL version of the terms is directly maintained at Schema.org and independent from the above presented formats. Various tools that use or produce Schema.org terms are already available or in preparation. Stay tuned!

Google's Knowledge Graph

Google: *"Info-box"*



Sully Prudhomme

Poet

René François Armand Prudhomme was a French poet and essayist. He was the first ever winner of the Nobel Prize in Literature in 1901.
Wikipedia

Born: March 16, 1839, Paris, France

Died: September 6, 1907, Châtenay-Malabry, France

Books: Les vaines tendresses

Awards: Nobel Prize in Literature

People also search for

View 10+ more

Leconte de Lisle

Theodor Mommsen

Frédéric Mistral

Paul Verlaine

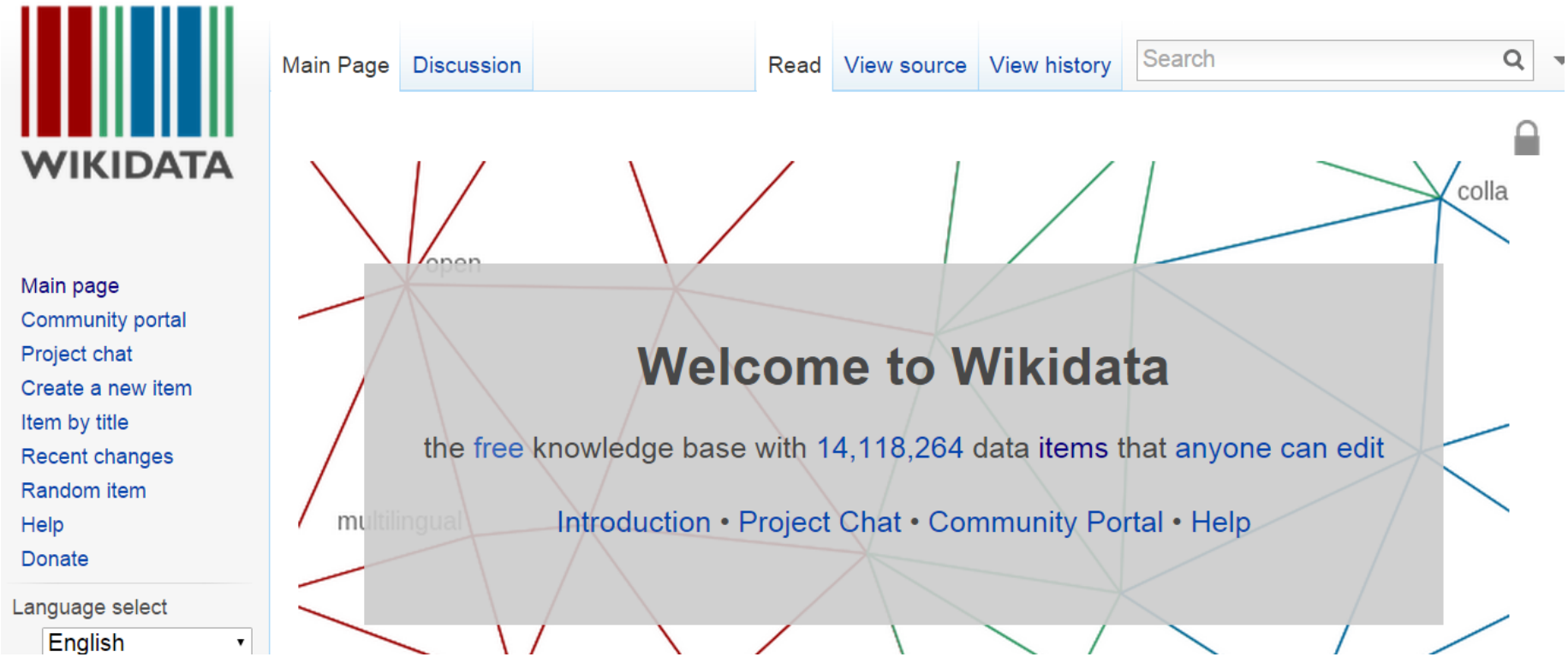
Gabriel Fauré

Datos estructurados como grafos …

WIKIDATA

La Wikipedia de Datos: Wikidata

¿Qué es Wikidata?



Wikipedia: Varios Idiomas

Alexis Sánchez

Sánchez with Chile in 2013

Personal information	
Full name	Alexis Alejandro Sánchez Sánchez[1]
Date of birth	19 December 1988 (age 26)[1][2]
Place of birth	Tocopilla, Chile[3][2]

National team‡			
2007	Chile U20	12	(2)
2006–	Chile	82	(26)

Alexis Sánchez

Datos personales	
Nombre completo	Alexis Alejandro Sánchez Sánchez

Carrera internacional	
Selección	Chile
Part. (goles)	82 (26)
Debut	2006

Alexis

Alexis Sánchez im Dezember 2011

Spielerinformationen	
Voller Name	Alexis Alejandro Sánchez Sánchez
Geburtstag	19. Dezember 1988
Geburtsort	Tocopilla, Chile

Nationalmannschaft[2]		
2007	Chile U-20	
2006–	Chile	76 (25)

Wikipedia: Listas, etc.

WIKIPEDIA
The Free Encyclopedia

Not logged in Talk Contributions Create account Log in

Article Talk Read Edit View history Search

Chile national football team

From Wikipedia, the free encyclopedia

Most capped players [edit]

As of September 1, 2016

*Players in **bold** are still active, at least at club level.*

#	Name	International Career	Caps	Goals
1.	**Claudio Bravo**	2004 –	106	0
2.	**Alexis Sánchez**	2006 –	102	34
3.	**Gary Medel**	2007 –	96	7
4.	**Gonzalo Jara**	2006 –	95	3

Top goalscorers [edit]

As of September 1, 2016

*Players in **bold** are still active, at least at club level.*

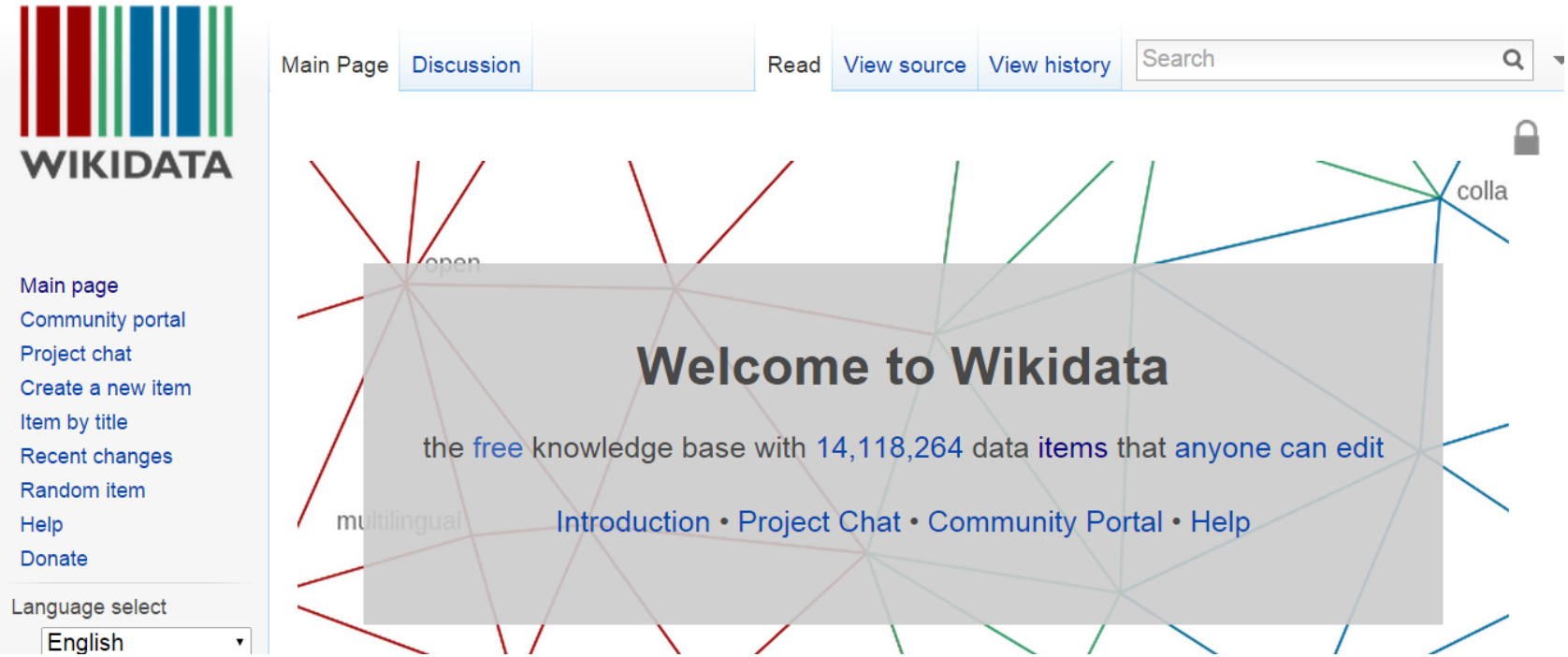
#	Name	International Career	Goals	Caps
1.	Marcelo Salas	1994–2007	37	70
2.	Iván Zamorano	1987–2001	34	69
2.	**Alexis Sánchez** (list)	2006 –	34	102

Alexis marca un gól …

Ahora un ejército de personas tienen que actualizar Wikipedia

(texto, listas, idiomas, etc.)

Una solución: Wikidata



Wikidata: datos estructurados

Abraham Lincoln (Q91)

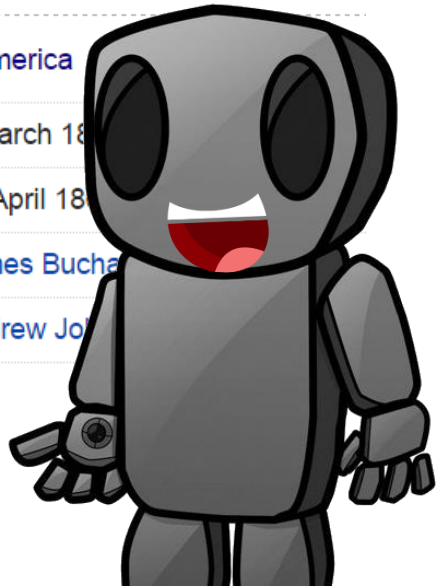
American politician, 16th President of the United States in office from 1861 to 1865

Abe Lincoln | Lincoln | Honest Abe

In more languages

place of death	Petersen House	[edit]
	1 reference	

position held	President of the United States of America	
	start time	4 March 18
	end time	15 April 18
	replaces	James Bucha
	succeeded by	Andrew Jo
	1 reference	



Servicio de consulta (SPARQL):

Wikidata Query Service

https://query.wikidata.org/#PREFIX%20wd%3A%20%3Chttp%3A%2F%2Fwww.wikidata.org%2Fentity%2F%3

Aplicaciones | Linguee | SGICM | Apache Any23: Anyth | Otros marcadores

Wikidata Query Service | Examples | Prefixes | Tools | Help | English

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX p: <http://www.wikidata.org/prop/>
PREFIX ps: <http://www.wikidata.org/prop/statement/>
PREFIX pq: <http://www.wikidata.org/prop/qualifier/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?laureateName ?awardYear ?warName ?warYear
WHERE {
  ?laureate p:P166 ?award .              # Winner of some prize
  ?award ps:P166 wd:Q37922 .             # Prize is Nobel Pr. in Lit.
  ?award pq:P585 ?awardDate .            # Get the date of the award
  BIND(YEAR(?awardDate) as ?awardYear) # Get the year of the award
  ?laureate wdt:P607 ?war .              # Find war(s) laureate was in
  ?war rdfs:label ?warName .             # Get name(s) of war(s)
  ?war wdt:P580 ?warStart  .             # Get date the war started
  BIND(YEAR(?warStart) as ?warYear)      # Get year from date
  ?laureate rdfs:label ?laureateName . # Get name of laureate
  FILTER(lang(?warName)="en"             # Filter for English
    && lang(?laureateName)="en")         #  ...  names only
} ORDER BY ?awardYear                    # Order by year
```

Find award received Nobel Prize in Literature

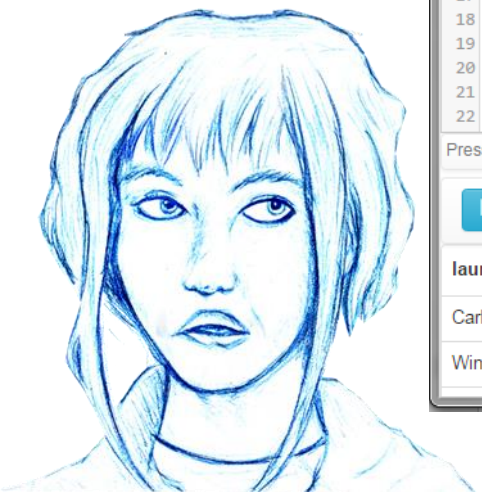
Show

Limit

Press [CTRL-SPACE] to activate auto completion. Data updated a few seconds ago

Run | Clear | 12 Results in 1589 ms | Display | Download | Link

laureateName	awardYear	warName	warYear
Carl Spitteler	1919	World War I	1914
Winston Churchill	1953	World War I	1914



Servicio de consulta (SPARQL):

Wikidata Query Service

https://query.wikidata.org/#PREFIX%20wd%3A%20%3Chttp%3A%2F%2Fwww.wikidata.org%2Fentity%2F%3

```
  ?war wdt:P580 ?warStart .          # Get date the war started
  BIND(YEAR(?warStart) as ?warYear)  # Get year from date
  ?laureate rdfs:label ?laureateName . # Get name of laureate
  FILTER(lang(?warName)="en"         # Filter for English
    && lang(?laureateName)="en")     #  ...  names only
} ORDER BY ?awardYear                # Order by year
```

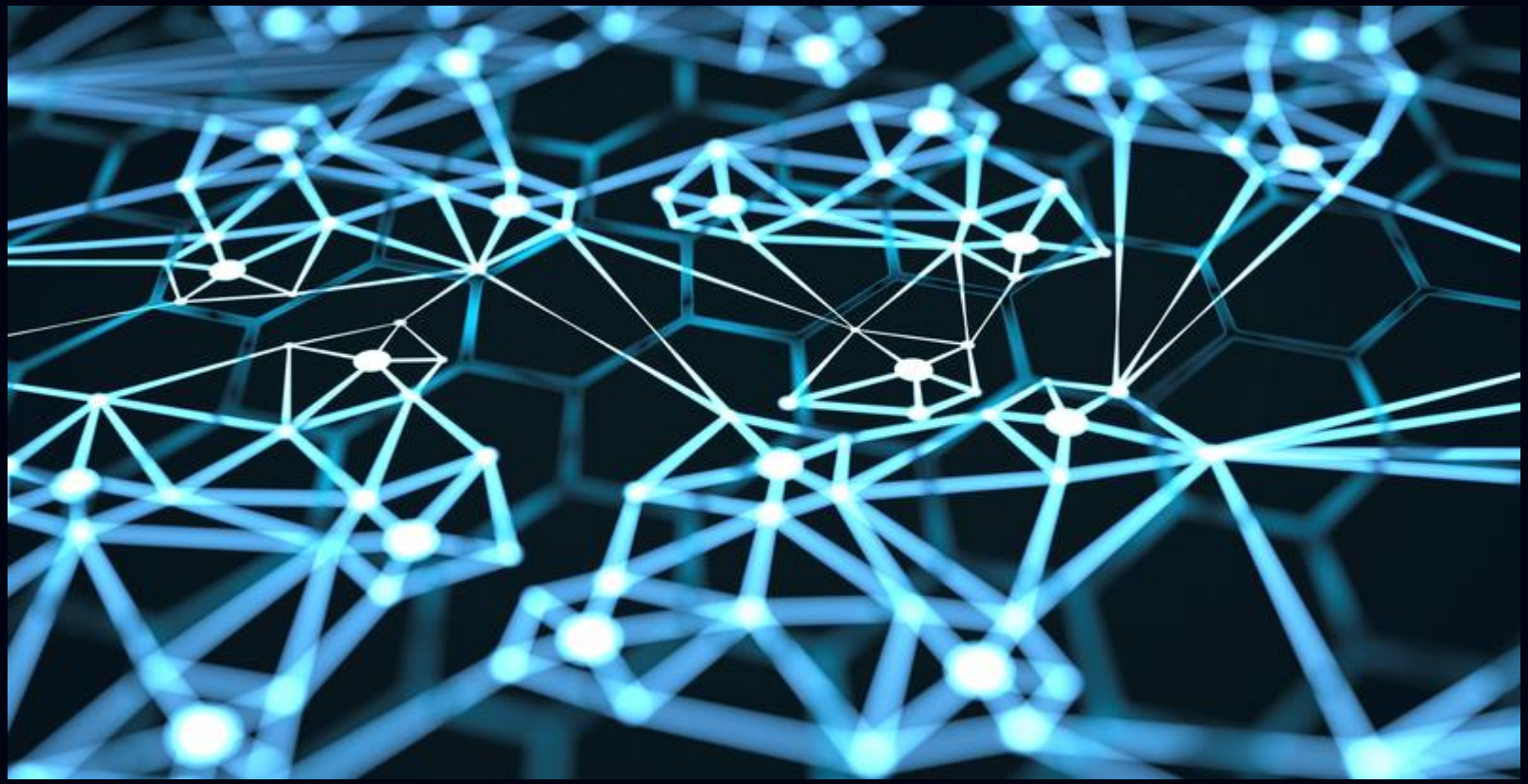
Press [CTRL-SPACE] to activate auto completion. Data updated a few seconds ago

Run | Clear | 12 Results in 1589 ms | Display | Download | Link

laureateName	awardYear	warName	warYear
Carl Spitteler	1919	World War I	1914
Winston Churchill	1953	World War I	1914
Ernest Hemingway	1954	World War I	1914
Ernest Hemingway	1954	World War II	1939
Jean-Paul Sartre	1964	Algerian War	1954
Jean-Paul Sartre	1964	World War II	1939
Heinrich Böll	1972	World War II	1939
Eugenio Montale	1975	World War I	1914
William Golding	1983	World War II	1939
Claude Simon	1985	Spanish Civil War	1936
Camilo José Cela	1989	Spanish Civil War	1936
Günter Grass	1999	World War II	1939

En conclusión

Una área de investigación aquí …

Datos ≠ Datos Relacionales

¿Preguntas?