

WEB SEMANTICA

Gabriel del Reguero García

UO245002 58430888D

Contenido

Parte1: Datos en wikidata	2
Parte 2: Shape Expressions	3
Parte3: SPARQL	6
Query 1: Desarrollados por ACE TEAM	6
Query 2: En ordenador en 2020.....	7
Query3: Acción singleplayer entre 2010 y 2015	8
Parte 4: Ontología	9
Parte 5: Negociación de contenidos	11
Entities (videojuegos).....	12
Queries	14

Parte1: Datos en wikidata

En esta parte se procedieron a añadir datos de diversas entidades en wikidata. Se añadieron datos de algunos pequeños conjuntos (restaurantes con estrellas michelin, youtubers y empresas de videojuegos indie españolas) antes de la decisión final.

Detecte que wikidata no tenía los datos de algunos de los últimos juegos lanzados al mercado, pero más grave, no poseía ítems que representasen a la mayoría de los editores de videojuegos y a veces tampoco a equipos de desarrollo, aun conteniendo entradas para los videojuegos desarrollados o editados por estos.

Se procedió entonces a añadir datos de videojuegos recientes, editores y estudios de desarrollo.

Lista de entidades:

- <https://www.wikidata.org/wiki/Q94658604>
- <https://www.wikidata.org/wiki/Q94657742>
- <https://www.wikidata.org/wiki/Q94657658>
- <https://www.wikidata.org/wiki/Q94658505>
- <https://www.wikidata.org/wiki/Q94649502>
- <https://www.wikidata.org/wiki/Q94648063>
- <https://www.wikidata.org/wiki/Q94653248>

Entidades previas (restaurantes):

- <https://www.wikidata.org/wiki/Q94652546>

*En el caso particular de 3goo he de mencionar que fue interesante buscar la información dado a que el editor solo ofrecía la información en Japones.

Parte 2: Shape Expressions

Tras varios problemas se consiguió crear shape expressions capaces de validar lo datos introducidos. Me encontré con muchos problemas para corregirlas.

La primera shape expression diseñada:

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ps: <http://www.wikidata.org/prop/statement/>

start = @<Videojuego>

<Videojuego> EXTRA wdt:P1476 {
    wdt:P31 [ wd:Q7889 ] ; # Instance de videojuego
    wdt:P1476 LITERAL?; #Titulo (extra)
    wdt:P136 {ps:P31 wd:Q659563 }+ ; #generos del videojuego
    wdt:P123 (@<Editor> OR {ps:P31 wd:Q1137109})* ; #editores o publishers
    wdt:P577 xsd:dateTime?; #fecha de lanzamiento
    wdt:P750 {ps:P31 wd:Q81989119}+ ;#distribuidores
    wdt:P178 (@<Desarrollador> OR {ps:P31 wd:Q47157316})*;#desarrolladores
    wdt:P400 { ps:P31 wd:Q241317 }* ; # plataformas en las que esta disponible
    wdt:P404 { ps:P31 wd:Q1971694 }* ; #modo de juego
    wdt:P437 { ps:P31 wd:Q81941037 }* ; #formato de distribucion
    wdt:P1733 xsd:string ?; #identifier steamId
}

<Editor> {
    wdt:P31 [ wd:Q1137109 ]+;#instance
    wdt:P452 [wd:Q16023725]* ; # industria
    wdt:P17 { ps:P31 wd:Q6256 }? ; # pais
    wdt:P159 {ps:P31 wd:Q7540126}?; # headquarters
    wdt:P6375 {ps:P32 wd:Q24574749 }? ; #street address
```

```
    wdt:P856 xsd:anyURI ?; # web oficial
}

<Desarrollador> {
    wdt:P31 [ wd:Q210167 ] +; #un estudio de desarrollo puede ser mas cosas
    wdt:P17 { ps:P31 wd:Q6256 }? ; # pais
    wdt:P159 {ps:31 wd:Q7540126}?; # headcuarters
    wdt:P6375 {ps:32 wd:Q24574749} ?; #street addres
    wdt:P856 xsd:anyURI ?; # web oficial
}
```

Esta define unos datos para validar y definir los ítems de desarrollador, videojuego y editor.

Tras consultar con el profesor se diseño la siguiente shape expresión

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ps: <http://www.wikidata.org/prop/statement/>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix : <http://example.org/>

start = @:Videojuego

:Videojuego {
    wdt:P31 [ wd:Q7889 ] ; # Instance de videojuego
    wdt:P1476 rdf:langString ?; #Titulo (extra)
    wdt:P136 . + ; #generos del videojuego
    wdt:P123 @:Editor * ; #editores o publishers
    wdt:P577 xsd:dateTime?; #fecha de lanzamiento
    wdt:P750 .+ ;#distribuidores
    wdt:P178 @:Desarrollador* ;#desarrolladores
    wdt:P400 . * ; # plataformas en las que esta disponible
    wdt:P404 . * ; #modo de juego
    wdt:P437 . * ; #formato de distribucion
    wdt:P1733 xsd:string ?; #identifier steamId
```

```
}

:Editor EXTRA wdt:P31 {
wdt:P31 [ wd:Q1137109 ] ; #instance-of video-game-publisher
wdt:P452 . * ; # industria
wdt:P17 . ? ; # pais
wdt:P159 . ? ; # headquarters
wdt:P6375 . ? ; #street address
wdt:P856 IRI ? ; # web oficial
}

:Desarrollador EXTRA wdt:P31 {
wdt:P31 [ wd:Q210167 ] ? ; #un estudio de desarrollo puede ser mas cosas
wdt:P17 . ? ; # pais
wdt:P159 . ? ; # headcuarters
wdt:P6375 . ? ; #street addres
wdt:P856 IRI ? ; # web oficial
}
```

Esta shape es mucho menos restrictiva y valida los datos que se reciben si empleamos una petición de estos en forma de rdf.

La pagina usada para evaluar y validar estos shapes devolvía errores no entendibles al intentar validar los ítems creados empleando estas shapes.

Sin embargo, la herramienta de visualización ayudaba a comprobar que ambos shapes no tienen ningún error de en su escritura o formato por lo que sé que ambos son validos como shapes pero debido al feedback de errores no se si validan los ítems creados.

Parte3: SPARQL

Se crearon algunas consultas SPARQL que se relacionan con los ítems incluidos.

Se emplearon tanto consultas para los ítems incluidos como para otros que compartían campos.

Es necesario destacar que muchos de los juegos aparentan tener varias labels para el mismo idioma y eso hace que aparezcan duplicados en los resultados de las queries que muestren las label aun usando DISTINCT.

Query 1: Desarrollados por ACE TEAM

En este caso buscamos todos los juegos desarrollados por ACE TEAM y su fecha de lanzamiento. Se puede observar toda la saga de rock of ages, entre otros juegos.

```
SELECT ?game ?label ?launchDate ?developer WHERE {  
  
  ?game wdt:P31 wd:Q7889 ;  
  rdfs:label ?label ;  
  wdt:P577 ?launchDate ;  
  wdt:P178 ?developer .  
  
  FILTER((?developer = wd:Q2080159) && (LANGMATCHES(LANG(?label), "en")))  
  
  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en". }  
}
```

Query 2: En ordenador en 2020

En este otro caso se realiza un query distinta, se llama a todos videojuegos lanzados para ordenador en 2020 que tengan etiqueta en inglés. Si se usa de plataforma “computadora” saldrán los introducidos por mi, pero resulta que si variamos el termino usado para referirnos a la misma plataforma nos saldrán distintos resultados. Esto implica que los juegos de pc no pueden ser correctamente identificados en su plenitud ya que existen varios ítems con las cualidades para cubrir el hueco de plataforma y significar lo mismo.

```
SELECT distinct ?game ?label ?launchDate WHERE {  
  
  ?game wdt:P31 wd:Q7889 ;  
  
  rdfs:label ?label ;  
  
  wdt:P577 ?launchDate ;  
  
  wdt:P178 ?developer ;  
  
  wdt:P400 ?plataform ;  
  
  FILTER((YEAR(?launchDate) = 2020) && (?plataform = wd:Q68) &&  
    (LANGMATCHES(LANG(?label), "en")))  
  
  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en". }  
}
```


Query3: Acción singleplayer entre 2010 y 2015

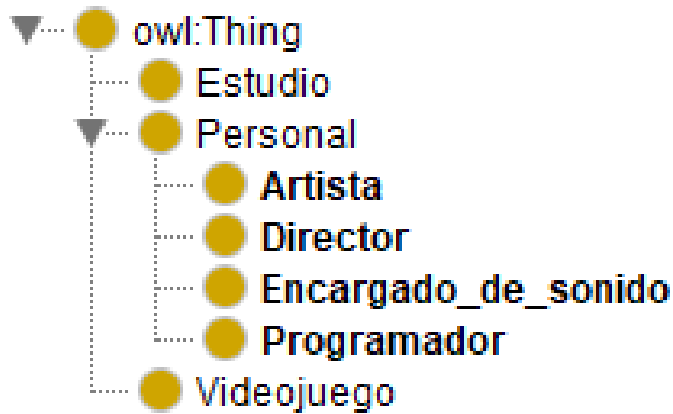
En esta query buscamos listar todos los juegos dentro del género acción con un modo de juego individual que se lanzasen entre 2010 y 2015.

```
SELECT distinct ?game ?label ?launchDate WHERE {  
  
  ?game wdt:P31 wd:Q7889 ;  
  
  rdfs:label ?label ;  
  
  wdt:P577 ?launchDate ;  
  
  wdt:P178 ?developer ;  
  
  wdt:P136 ?genre;  
  
  wdt:P404 ?gamemode ;  
  
  FILTER((?gamemode = wd:Q208850) && (?genre= wd:Q270948) &&(YEAR(?launchDate)  
    >=2010) &&(YEAR(?launchDate) <=2015) && (LANGMATCHES(LANG(?label), "es")))  
  
  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en". }  
  
}
```

Parte 4: Ontología

Se crea una pequeña ontología centrada en videojuegos y equipos de desarrollo empleando Protege 5.5.

Existirán como clases básicas Estudio (de desarrollo), Personal y Videojuego, con varias subclases dentro de personal.

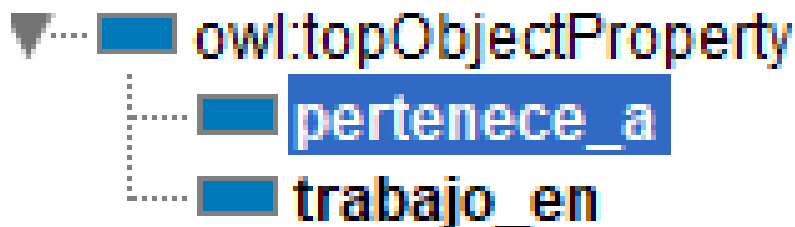


Se crearon tanto propiedades de objeto como propiedades de datos.

En el caso de las propiedades de objeto se crearon para Videojuego y para Personal.

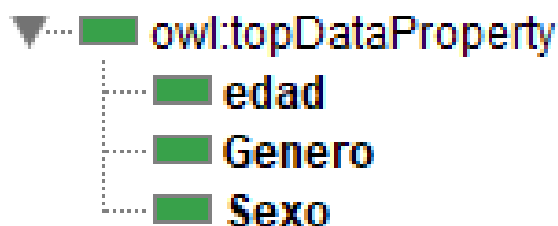
Las propiedades serían de una sola dirección y puede haber varias iguales por objeto:

- "pertenece_a" (que une un videojuego con estudio).
- "Trabajo_en" (que une un personal con videojuego).



Se crean también algunas propiedades de datos. En este caso 3:

- Edad: perteneciente al Personal se trata de un "int".
- Género: perteneciente a los Videojuegos se trata de un "string".
- Sexo: perteneciente al Personal, se trata de un "string".



Se crearon unos cuantos objetos para crear un ejemplo de la ontología y se establecieron las relaciones y propiedades.

- ◆ Aunuma
- ◆ Breath_of_the_wild
- ◆ Iwatasi_Imoto
- ◆ Jo-Un_Sun_Yo
- ◆ Library_of_Ruina
- ◆ Lobotomy_corp
- ◆ Nintendo
- ◆ Pokemon_Sword_and_shield
- ◆ ProjectMoon
- ◆ William_Griever
- ◆ Yun-Sang_Loo

Types +

Encargado_de_sonido

? @ x o

Same Individual As +

Different Individuals +

Object property assertions +

trabajo_en Library_of_Ruina

trabajo_en Lobotomy_corp

Data property assertions +

edad "38"^^xsd:int

Sexo "Mujer"^^xsd:string

Negative object property assertions +

Negative data property assertions +

Parte 5: Negociación de contenidos

En esta ultima parte se diseñó una aplicación empleando angular para negociar contenido con wikidata y poder mostrarlo. En este caso se trata de mostrar los datos de los 3 juegos añadidos y permitir emplear las queries usadas anteriormente. Los datos devueltos son tratados de diferente manera para su visualización en la página web.

Se decidió crear la pagina empleando angular ya que este fue recomendado por varios sitios web para realizar páginas de este estilo rápidamente, sobre todo empleando plantillas de estilos.

Para explicar lo que se hace dentro de la aplicación es necesario mostrar primero como se reciben los campos de wikidata al negociar el contenido.

```
0:
  ▶ propLabel: {xml:lang: "en", type: "literal", value: "publisher"}
  ▶ propUrl: {type: "uri", value: "http://www.wikidata.org/prop/direct/P123"}
  ▶ vallabel: {xml:lang: "en", type: "literal", value: "Modus Games"}
  ▶ valUrl: {type: "uri", value: "http://www.wikidata.org/entity/Q94657742"}
  ▶ __proto__: Object
1:
  ▶ propLabel: {xml:lang: "en", type: "literal", value: "publisher"}
  ▶ propUrl: {type: "uri", value: "http://www.wikidata.org/prop/direct/P123"}
  ▶ vallabel: {xml:lang: "en", type: "literal", value: "3goo"}
  ▶ valUrl: {type: "uri", value: "http://www.wikidata.org/entity/Q94658505"}
  ▶ __proto__: Object
2:
  ▶ propLabel: {xml:lang: "en", type: "literal", value: "genre"}
  ▶ propUrl: {type: "uri", value: "http://www.wikidata.org/prop/direct/P136"}
```

Como se puede observar en la imagen los datos de cada campo del item llegan distribuidos entre vallabel , valUrl , propUrl y propLabel.

Cabe resaltar que este contenido se obtiene mediante la petición a wikidata del elemento especifico empleando su identificador.

La idea de la página será representar un listado de los videojuegos añadidos por mi y si es posible mostrar también los resultados de las queries realizadas anteriormente.

Con este fin se representarán los datos negociados con wikidata y se transformarán en json personalizados.

Entities (videojuegos)

Las entidades recuperadas de wikidata son videojuegos que llegan en el formato antes mostrado. Para procesarlas se extraen los datos relevantes y se crea una clase o modelo nuevo que actuara como nueva representación de dichos datos:

```
export class Videojuego {
  genre: WikiPair[];
  publisher: WikiPair[];
  distributedby: WikiPair[];
  developer: WikiPair[];
  platform: WikiPair[];
  gamemode: WikiPair[];
  identity: WikiPair;

  toJSONLD(): string {
    let jsonld = `{
      "@context": "http://schema.org/",
      "@type": "videogame";
      jsonld += `"name": "${this.identity.value}";`
      jsonld += `"sameAs": "${this.identity.url}";`
      jsonld += `"platforms": [;
      this.platform.forEach(plat => {
        jsonld += `{"@type": "Platform",
          "name": "${plat.value}",
          "sameAs": "${plat.url}";`
        if ( this.platform[this.platform.length - 1].value !== plat.value ) {
          jsonld += `';`
        }
      });

      jsonld += `], "Distributors": [;
      this.distributedby.forEach(d => {
        jsonld += `{"@type": "DistributedBy",
          "name": "${d.value}",
          "sameAs": "${d.url}";`
        if ( this.distributedby[this.distributedby.length - 1].value !== d.value ) {
          jsonld += `';`
        }
      });

      jsonld += `], "Genres": [;
      this.genre.forEach(g => {
        jsonld += `{"@type": "Genre",
          "name": "${g.value}",
          "sameAs": "${g.url}";`
        if ( this.genre[this.genre.length - 1].value !== g.value ) {
          jsonld += `';`
        }
      });

      jsonld += `], "GameModes": [;
      this.gamemode.forEach(gm => {
        jsonld += `{"@type": "GameMode",
          "name": "${gm.value}",
          "sameAs": "${gm.url}";`
        if ( this.gamemode[this.gamemode.length - 1].value !== gm.value ) {
          jsonld += `';`
        }
      });

      jsonld += `], "Developers": [;`
  }
}
```

```
this.developer.forEach(dev => {
  jsonld += `{"@type": "Developer",
    "name": "${dev.value}",
    "sameAs": "${dev.url}"}`;
  if ( this.developer[this.developer.length - 1].value !== dev.value ) {
    jsonld += ',';
  }
});
jsonld += `], "Publishers": [`;
this.publisher.forEach(pb => {
  jsonld += `{"@type": "Publisher",
    "name": "${pb.value}",
    "sameAs": "${pb.url}"}`;
  if ( this.publisher[this.publisher.length - 1].value !== pb.value ) {
    jsonld += ',';
  }
});
jsonld += `]`;

jsonld = jsonld.replace('[,]', '[').replace('},]', '},').replace('"', '""');

console.log(jsonld);
return JSON.parse(jsonld);
}
```

La extracción de dichos datos se realiza campo a campo en typescript. Se obtiene la respuesta de wikidata y se extraen los datos según los campos anteriormente vistos.

Como se puede ver de ejemplo en la siguiente imagen:

```
let aux = res.filter(vgame => vgame.propLabel.value === 'identity');
game.identity = new WikiPair(aux[0].valUrl.value, aux[0].valLabel.value);
```

Una vez unidos los datos al modelo (habiendo generado un nuevo objeto por cada objeto recibido). Se procede a representarlos en la página empleando html.

La representación de html se ha realizado mediante plantillas de tablas modificadas para acelerar la velocidad de desarrollo de la página.

Queries

Se ha añadido otra parte a la página para usar las queries mostradas anteriormente, la idea es simple, se reciben los datos empleando la misma petición que las queries de wikidata designadas y luego se representan en forma de lista.

Se decidió seguir los pasos de una plantilla de angular para la representación en forma de tabla.

La transformación se realiza de una manera similar a la sección principal de videojuegos, pero se detecta de manera automática los campos devueltos y se añade el número de columnas necesarias con las label de dichos campos.

Desgraciadamente las transformaciones necesarias hacen que se tarde en obtener los resultados un poco (el tiempo que tarda wikidata+transformaciones). Dado que wikidata ya se toma su tiempo para alguna de las queries pueden llegar a tardar 20 segundos hasta mostrar los resultados.



The screenshot shows a web application interface with a header bar containing the Wikidata logo and navigation links for 'Ultimos videojuegos' and 'Queries de wikidata'. The main section is titled 'Prepared queries' and features a sidebar on the left with a list of queries. The selected query, 'Juegos en ordenador en 2020', is highlighted in blue. Below the sidebar is a table with three columns: 'GAME', 'LABEL', and 'LAUNCHDATE'. The table displays three results, each with a Wikidata ID (Q94653248, Q94657658, Q94648063), a game title, and a launch date. The interface includes pagination controls and a status bar at the bottom indicating 'Hay 3 resultados. Mostrando 3.'

Queries de wikidata	GAME	LABEL	LAUNCHDATE
Juegos desarrollados por ACE TEAM			
Juegos en ordenador en 2020	Q94653248	XCOM: Chimera Squad	2020-04-24T00:00:00Z
Accion singelplayer entre 2010-2015	Q94657658	Rock of ages III: Make & Break	2020-07-21T00:00:00Z
	Q94648063	Library of Ruina	2020-05-15T00:00:00Z

Link

Enlace a la dirección donde esta desplegado

<http://156.35.98.133:4200>