

Recomendaciones y buenas prácticas sobre D2RQ (Anexo 1)

Autor: Freddy Tapia León

1 Introducción.

D2RQ es una plataforma que permite acceder a bases de datos (BB.DD.) tradicionales, ver su estructura, y generar tripletas RDF. Adicionalmente, D2RQ permite (1) crear un Sparql Endpoint donde se almacenan las tripletas y consultarlas mediante SPARQL.

La versión utilizada para estas pruebas fue d2rq-0.8.1.

D2RQ es Open Source, publicado bajo los términos de Apache y desarrollado en Freie Universität Berlin entre otras organizaciones.

2 Cómo instalar.

- Descargar y descomprimir D2R Server. Depende de si es para Windows (.zip) o para Linux (.tar.gz).

<http://d2rq.org/>

Se puede descomprimir en cualquier ubicación.

- Descargar e instalar Java.

<http://www.java.com/es/download/>

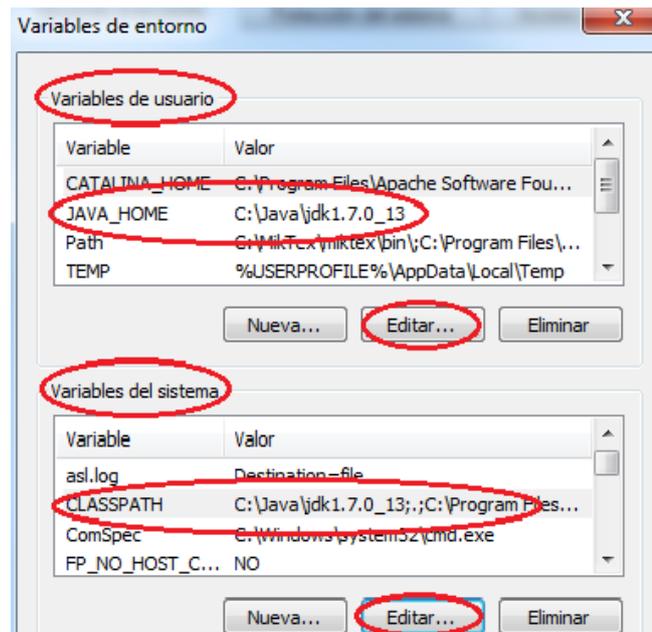
No es necesario instalar ningún otro componente adicional. Para verificar que podemos invocar los programas de java debemos abrir (en Windows) la consola de comandos (mediante el comando cmd) y teclear las siguientes instrucciones:

```
javac -version  
java -versión
```

El primer comando muestra la versión del compilador de java, y el segundo la versión de la máquina virtual.

Si aparece algún error de “Comando no encontrado”, esto quiere decir que hay que revisar las “variables de entorno”, PATH donde se encuentran instalado los ejecutables.

En la imagen abajo descrita podemos observar los parámetros que deben ser configurados de ser el caso.



- Descargar e instalar Apache Tomcat (Opcional).
Observación 1: Este punto es opcional. Y se aplica cuando la PC va a administrar más de un aplicativo Web (aplicativos que tengan la especificación Web Application Archive - .War).

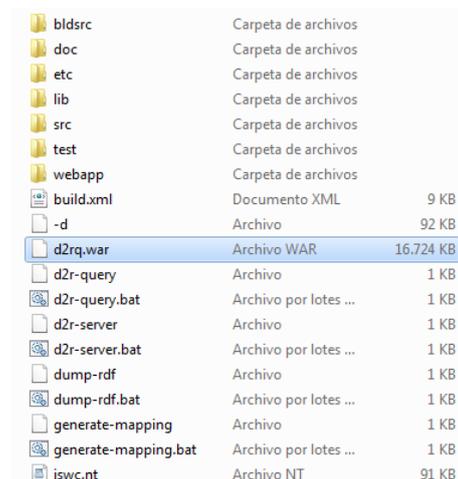
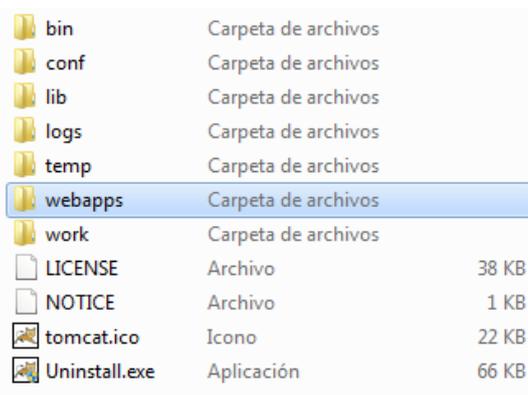
<http://tomcat.apache.org/>

Aparte de la instalación típica que maneja este tipo de software lo más importante es la conexión entre D2R Server y Apache. Para esto tenemos dos vías de configuración.

Primero ubicamos el archivo `d2rq.war`, este archivo está junto al resto de todos los archivos instalados anteriormente (D2R Server).

Este archivo debe ser cambiado de extensión a `.zip`.

Este paso nos permite visualizar todos los archivos que contiene el fichero `d2rq.zip`, de esta forma se los puede copiar.



Los archivos deben ser copiados dentro del directorio `webapps`, el cual se encuentra dentro del directorio raíz del Apache Tomcat.



Y la segunda forma, es hacer la llamada al archivo `d2rq.war`, dentro del “Tomcat Administration”.
Esto ya depende del usuario.

Observación 2: Si en nuestra PC solo vamos a tener D2RQ instalado y ningún otro aplicativo web no necesitamos instalar Tomcat.

- Descargar e instalar un motor de B.D. (D2RQ soporta Oracle, MySql, SQL server, entre otros), particularmente utilicé MySql.
<http://dev.mysql.com/downloads/installer/5.5.html>

Existen muchos videos tutoriales sobre cómo instalar y configurar MySql, así como también sobre las instrucciones básicas (crear, eliminar, modificar, etc.) que nos permitirían manejar y movernos adecuadamente dentro del ambiente MySql, en especial a través de su línea de comandos.

Existe la posibilidad de utilizar interfaz gráfica, al igual que en el caso anterior existen algunos tutoriales al respecto.
<http://www.mysql.com/products/workbench/>

- Descargar e instalar un archivo .jar (`mysql-connector-java-version-bin.jar`), este archivo tiene por finalidad realizar la conexión entre MySql y Java.
Para la práctica solo se requiere descargar y copiar el archivo .jar, ya que D2RQ incorpora todo el proceso de vinculación que usualmente se haría manualmente. Este archivo debe ser copiado dentro del directorio `/lib` ubicado dentro del directorio D2RQ.
<http://dev.mysql.com/doc/refman/5.6/en/connector-j-installing.html>

3 Cómo mapear

- a) Cargar el ejemplo propuesto en MySql³.

<http://d2rq.org/example/iswc-mysql.sql>

Este script SQL (`iswc-mysql.sql`) contiene las sentencias que permitirían cargar y poblar una B.D. tradicional de una forma automática.

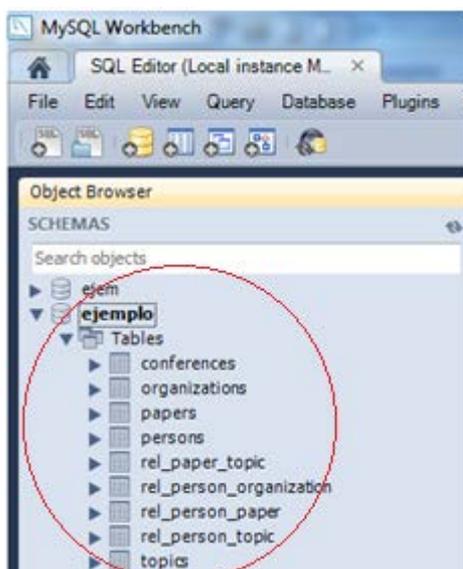
³ <http://d2rq.org/d2rq-language>

Este script debe ser cargado a MySQL (**SOURCE**) para lo cual se deberían seguir los siguientes pasos:

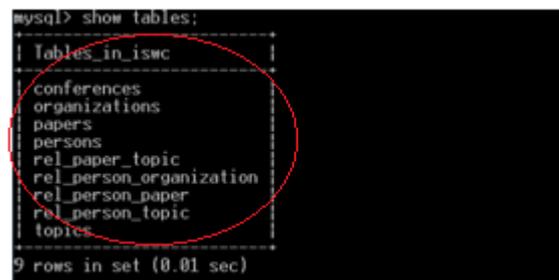
- `create nombreBD;` (Crear la B.D.)
- `use nombreBD;` (Usar o activar la B.D. antes creada)
- `source c:/nombreBD.sql;` (Importamos automáticamente toda la B.D. a MySQL)

Sí al finalizar el proceso de cargar la B.D. a MySQL no obtenemos ningún error, significa que la B.D. fue creada correctamente, de lo contrario se debería revisar los parámetros con los cuales trabaja la versión de MySQL instalada, por ejemplo: comillas, espacios en blanco, entre otros.

Ahora bien, en las imágenes abajo mostradas podemos ver que la B.D. fue cargada adecuadamente, tal es así que ambos casos (ambiente gráfico y consola) podemos observar las tablas creadas en MySQL.



Ambiente gráfico (Workbench)



Ambiente consola (línea de comandos MySQL)

b) Generación del fichero de mapeos⁴

El proceso de mapeo (mapping) consiste en transformar los datos contenidos en la B.D. en tripletas. Generate-mapping, genera un “borrador” del fichero de mapeos (mapping.ttl)

```
generate-mapping -o mapping_borrador.ttl -d driver.class.name -u db-user -p db-password jdbc:url:...
```

⁴ <http://d2rq.org/generate-mapping>

-o `mapping_borrador.ttl` = Crea un fichero llamado `mapping_borrador.ttl`. Si omitimos el parámetro `-o` solo visualizaríamos el mapeo realizado dentro de la ventana de comandos del Windows, es decir, no se crearía ningún archivo de extensión `.ttl`

-d `driver.class.name` = Se refiere al driver de conexión entre Java y la B.D. Específicamente el driver usado para MySQL es: `com.mysql.jdbc.Driver`

-u `db-user` = Es el nombre de usuario con el cual se configuró la B.D. al momento de instalarla (MySQL).

-p `db-password` = Es la clave de usuario con el cual se configuró la B.D. al momento de instalarla (MySQL).

`jdbc:url:...` = Es el URL de la B.D.

Particularmente el URL usado para MySQL es:

`jdbc:mysql://localhost/nombreBD`

Si al ejecutar la instrucción (`generate-mapping -o mapping.ttl.....`), obtenemos un mensaje `"Too many arguments"`, en la mayoría de casos se refiere a errores de digitación.

Una vez ejecutado el comando anterior, deberíamos obtener el fichero de mapeos (`mapping.ttl`). El cual contiene una estructura básica sobre la que se puede trabajar dependiendo de la ontología a usar.

Observemos principalmente los prefijos en la siguiente imagen:

```
@prefix map: <#> .
@prefix db: <> .
@prefix vocab: <vocab/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix d2rq: <http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/0.1#> .
@prefix jdbc: <http://d2rq.org/terms/jdbc/> .
```

Por otro lado tenemos el fichero que muestra el tutorial de D2RQ, el cual se encuentra en la siguiente dirección:

<http://d2rq.org/example/mapping-iswc.ttl>

Este fichero ha sido concebido para trabajar con la ontología ISWC (International Semantic Web Conference). El cual incorpora otros vocabularios adicionales como por ejemplo: foaf (usado para las personas), vcard (usado para tarjetas de presentación), etc. Todo esto con el objetivo de darle un mayor nivel semántico a los datos.

Si comparamos los dos ficheros podemos evidenciar que existen muchas diferencias, es por ello que no debemos olvidar que el primero solo era un borrador, y el segundo es un poco más completo.

Observemos los prefijos del fichero `mapping-iswc.ttl`:

```
@prefix map:
<file:/Users/richard/D2RQ/workspace/D2RQ/doc/example/mapping-iswc.ttl#> .
@prefix vocab: <vocab/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix d2rq: <http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/0.1#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix iswc: <http://annotation.semanticweb.org/iswc/iswc.daml#> .
@prefix vcard: <http://www.w3.org/2001/vcard-rdf/3.0#> .
@prefix jdbc: <http://d2rq.org/terms/jdbc/> .
```

Con estos antecedentes vamos a crear una fusión entre ambos ficheros, con el objetivo de dotarles de semántica a las tablas y campos del archivo SQL, adicionalmente usaríamos la ontología DBpedia como enlace principal.

A continuación algunos aspectos incluidos en la nueva propuesta del nuevo archivo mapping.ttl:

Vocabularios: Facilita la interoperabilidad entre la ontología y los datos, es decir, determinar que vocabularios se van a usar. A continuación la descripción de los vocabularios más destacables:

“dc” (Dublin Core), describe solo propiedades, por ejemplo “source”, “date”.

“foaf” (friend of a Friend), describe información de personas y actividades.

“dbpedia”, facilita el acceso a esa ontología.

“vcard”, gestiona la información de contactos (tarjetas personales).

Recordemos que no es necesario la utilización de todos los vocabularios descritos, ni su posterior uso. A continuación los vocabularios usados:

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix vcard: <http://www.w3.org/2001/vcard-rdf/3.0#> .
@prefix dbpedia: <http://dbpedia.org/ontology/> .
@prefix dbpedia-owl: <http://dbpedia.org/ontology/> .
@prefix dbpedia-prop: <http://dbpedia.org/property/> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
```

Continuando con el proceso de **mapeo** se debe observar las tablas y campos de la B.D. (Conferences, Organizations, Papers, Persons, entre otras) para así poder determinar a qué clases/propiedades de la DBpedia son equiparables.

Por ejemplo, se podría decir que la tabla Papers es equiparable en ciertos aspectos con la clase “Academic Journal” de la ontología DBpedia, para esto se debería comprobar que el significado semántico de las dos clases es similar. Hay que destacar que (1) no todas las tablas, ni todos los atributos, tienen que ser mapeados obligatoriamente y en (2) caso de usar una única ontología (DBpedia), no todas las tablas/campos tienen un equivalente.

Por ejemplo: Observemos cómo se mapea la tabla “papers”, así como uno de sus campos “papers_Title”.

```
# Table papers

map:papers a d2rq:ClassMap;
  d2rq:dataStorage map:database;
  d2rq:uriPattern "papers/@@papers.PaperID@";
  d2rq:class dbpedia-owl:AcademicJournal;
  d2rq:classDefinitionLabel "papers";
  d2rq:class foaf:Document

map:papers_Title a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:papers;
  # d2rq:property vocab:papers_Title;
  d2rq:property dc:title;
  d2rq:propertyDefinitionLabel "papers Title";
  d2rq:column "papers.Title";
```

Cada registro es de un type, por ejemplo:

“AcademicJournal” (d2rq:class dbpedia-owl:AcademicJournal) pertenece a la ontología DBpedia.

Los registro del campo “title” se convertirán en tripletas de la propiedad “title” de la ontología dublin core (d2rq:property dc:title).

Ahora se deben analizar qué mapear, es decir, analizar la correspondencia entre tablas y campos y ver su equivalente.

A continuación el detalle de los cambios realizados:

Un ejemplo de mapeo de tabla otro de campo

mapping_borrador.ttl	mapping_final.ttl (corregido)
<pre>@prefix iswc: <http://annotation.semanticweb.org /iswc/iswc.daml#></pre>	<p>Se eliminó. Ya que se va a trabajar con lenguaje OWL.</p>
	<p>Se incorporaron estos cuatro prefijos esenciales:</p> <pre>@prefix dbpedia: <http://dbpedia.org/ontology/> . @prefix dbpedia-owl: <http://dbpedia.org/ontology/> . @prefix dbpedia-prop: <http://dbpedia.org/property/> . @prefix dbpedia: <http://dbpedia.org/resource/> .</pre>

# Table conferences d2rq:class iswc:Conference; d2rq:property iswc:location;	# Table conferences d2rq:class dbpedia-owl:Conference; d2rq:property dbpedia-prop:location;
# Table organizations d2rq:class iswc:Organization;	# Table organizations d2rq:class dbpedia-owl:Organization;

c) Creación de tripletas RDF⁶

Dump permite el volcamiento de los datos mapeados hacia tripletas RDF.

```
dump-rdf -f N-TRIPLE -b http://localhost:2020/ mapping.ttl>triples.nt
```

`-f N-TRIPLE` = Es el tipo de archivo RDF a obtener, entre los cuales tenemos: TTL (Turtle), RDF/XML y N-TRIPLE.

`-b http://localhost:2020/` = Es la base URI de nuestro servidor local (namespace base).

`mapping.ttl>triples.nt` = Es el direccionamiento de salida entre el archivo origen (.ttl) y el archivo destino (.nt)

La ejecución de esta instrucción me crea un archivo de extensión .nt, el cual contiene las tripletas RDF.

Nota: En algunos casos hay que probar las diversas opciones (ejemplos), que se incluyen dentro del tutorial, ya que el uso de algunos parámetros influye para la obtención de triples RDF que no podrían ser válidos (dependiendo de la necesidad).

Por ejemplo:

Cuando ejecutamos esta instrucción:

```
dump-rdf -f "n-triple" -o triples.nt mapping.ttl
```

Obtenemos estos triples

```
<file:///C:/Users/ddr2/d2rq-0.8.1/d2rq-0.8.1/triples.nt#people/7>  
<file:///C:/Users/ddr2/d2rq-0.8.1/d2rq-0.8.1/vocab/people_addr>  
<file:///C:/Users/ddr2/d2rq-0.8.1/d2rq-0.8.1/salida.nt#addresses/18>.  
<file:///C:/Users/ddr2/d2rq-0.8.1/d2rq-0.8.1/salida.nt#people/7>  
<file:///C:/Users/ddr2/d2rq-0.8.1/d2rq-0.8.1/vocab/people_fname> "Bob".  
<file:///C:/Users/ddr2/d2rq-0.8.1/d2rq-0.8.1/salida.nt#people/7>  
<file:///C:/Users/ddr2/d2rq-0.8.1/d2rq-0.8.1/vocab/people_id>
```

Este mismo resultado se obtiene si eliminamos el formato "n-triple"

```
dump-rdf -o triples.nt mapping.ttl
```

⁶ <http://d2rq.org/dump-rdf>

d) Ejecutar el servidor D2R⁷

D2R Server puede ser ejecutado desde la línea de comandos (cmd), por medio de la siguiente instrucción:

```
d2r-server -b http://localhost:2020/ mapping.ttl
```

`-b http://localhost:2020/` = Es la base URI de nuestro servidor local (namespace base).

`mapping.ttl` = Es el nombre del fichero de mapeos

Si la instrucción se ejecutó adecuadamente deberíamos tener el servidor inicializado a través de la ventana de comandos:

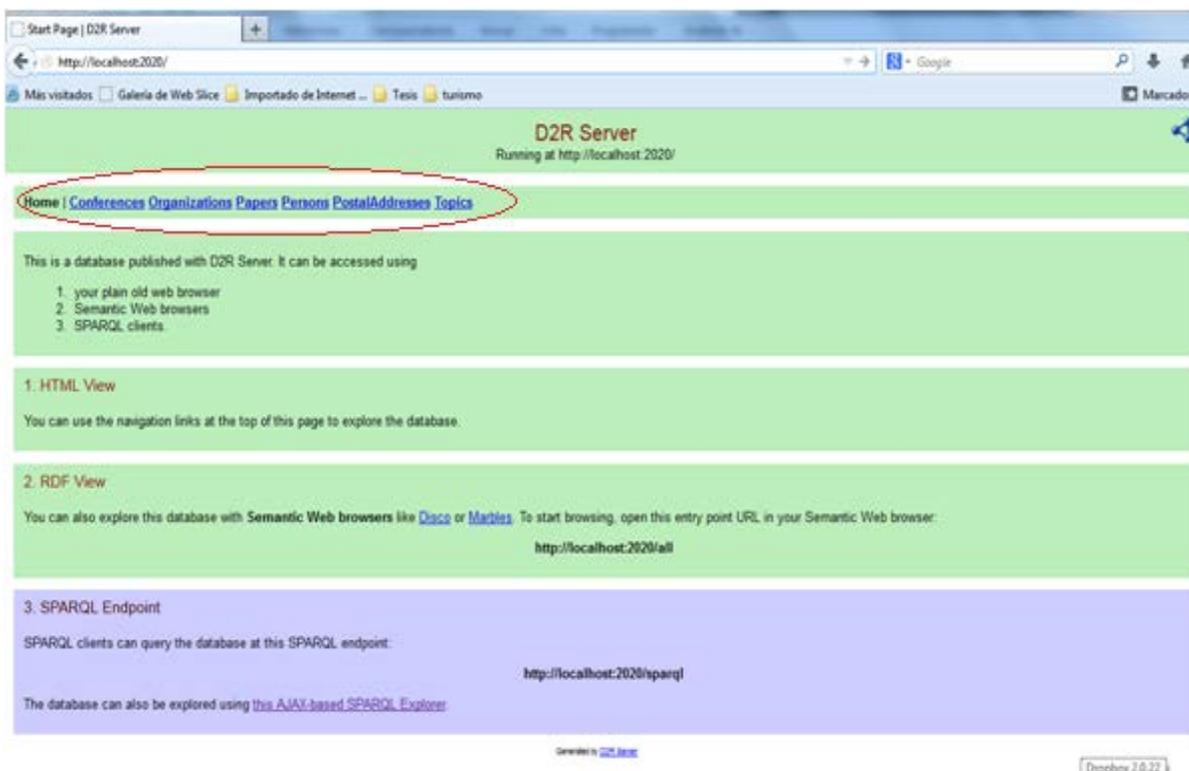
```
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\ddr2\d2rq-0.8.1\d2rq-0.8.1>d2r-server -b http://localhost:2020/ mapping.ttl
08:47:53 INFO JettyLauncher      : [[Server started at http://localhost:2020/]]
```

e) Abrir el browser y digitar la siguiente dirección:

`http://localhost:2020/`

El browser debería mostrar esta pantalla.



⁷ <http://d2rq.org/d2r-server>

4 Cómo consultar a través de sentencias SPARQL y ejemplo.

Una vez que tenemos ejecutándose el D2R Server podemos proceder a realizar las consultas respectivas, para lo cual damos clic en “Ajax-based SPARQL Explorer”, esta línea se encuentra dentro del punto 3 SPARQL Endpoint como podemos ver en la imagen arriba descrita.

Una vez dentro tenemos acceso al servidor Sparql.



En la imagen arriba descrita podemos observar dos ambientes, en la parte superior se detallan varios “prefix”, que son los vocabularios a usar, podríamos compararlas con las librerías que se definen en programación (Lenguaje C).

En la parte inferior, es donde realizamos las consultas (Query) respectivas.

Por ejemplo:

Deseo saber los nombres y direcciones de todas las personas registradas.

```
SELECT ?name ?localidad WHERE{
  ?res foaf:name ?name .
  ?res dbpedia-prop:address ?localidad
}
```

Results:

SPARQL results:

name	localidad
"Andreas Eberhart"	"International University in Germany Campus 2 76646 Bruchsal Germany"
"Borys Omelayenko"	"Vrije Universiteit, Division of Mathematics and Computer Science, De Boelelaan 1081a,1081hv, Amsterdam, The Netherlands"
"Andy Seaborne"	"Hewlett-Packard Laboratories, Bristol, BS34 8QZ, UK"
"Sonia Bergamaschi"	"DII- Universita di Modena e Reggio Emilia via Vignolese 905 41100 Modena"
"Francesco Guerra"	"DII- Universita di Modena e Reggio Emilia via Vignolese 905 41100 Modena Italy"
"Christian Bizer"	"Freie Universität Berlin"

Bajo el mismo criterio de búsqueda anterior, se desea consultar cuántas personas están registradas tienen (al menos) una dirección:

```
SELECT count (?name)
WHERE{
  ?res foaf:name ?name .
  ?res dbpedia-prop:address ?loc
}
```

Results: Browse

SPARQL results:

1
6

Deseo saber qué conferencia se realizó en Edinburgh.

```
SELECT ?Conference ?loc
WHERE{
  ?Conference rdf:type <http://dbpedia.org/ontology/Conference> .
  ?Conference dbpedia-prop:location "Edinburgh"
}
```

Results: Browse

SPARQL results:

Conference	loc
<http://localhost:2020/resource/conferences/23542>	-

Existen otros sitios web que muestran más criterios de búsqueda (query), los cuales nos podrían ayudar y complementar los criterios (comandos) vistos en el tutorial D2RQ. A continuación algunos links :

Q papers tienen un autor cuyo nombre empieza con J

SPARQL Query Language for RDF

<http://www.w3.org/TR/rdf-sparql-query/>

Dbpedia Español

<http://es.dbpedia.org/Wiki.jsp?page=Ejemplos%20de%20consultas%20SPARQL>