

**ISEL**  
**DEETC - Mestrado em Engenharia Informática**  
**Guia Aula Prática**

<b>Representação e Processamento de Conhecimento (RPC)</b>
--

## 1. Explorar o objectivo da “Linked Data” – suporte RDF(S)


*“(…) a widely held view is that the Semantic Web is made up of Linked Data; i.e. the Semantic Web is the whole, while Linked Data is the parts. Tim Berners-Lee, inventor of the Web and the person credited with coining the terms Semantic Web and Linked Data has frequently described Linked Data as “the Semantic Web done right.”; este excerto foi retirado de um site que atualmente não está ativo <http://linkeddata.org/faq> (in “What is the relationship between Linked Data and the Semantic Web?”). Este site foi integrado no W3C (a ver no próximo parágrafo). Por considerar o excerto relevante mantenho aqui esse registo.*

*“(…) to make the Web of Data a reality, it is important to have the huge amount of data on the Web available in a standard format, reachable and manageable by Semantic Web tools. Furthermore, not only does the Semantic Web need access to data, but relationships among data should be made available, too, to create a Web of Data (as opposed to a sheer collection of datasets). This collection of interrelated datasets on the Web can also be referred to as Linked Data.”; o site do parágrafo anterior foi integrado no W3C, cf., <https://www.w3.org/standards/semanticweb/data> (in “What is Linked Data?”).*

Outras questões frequentes (e interessantes) em “<https://www.w3.org/RDF/FAQ>”.

- a) Ver o (TED) [http://www.ted.com/talks/tim\\_berniers\\_lee\\_on\\_the\\_next\\_web.html](http://www.ted.com/talks/tim_berniers_lee_on_the_next_web.html)  
Ver o (TED) [http://www.ted.com/talks/ian\\_ritchie\\_the\\_day\\_i\\_turned\\_down\\_tim\\_berniers\\_lee](http://www.ted.com/talks/ian_ritchie_the_day_i_turned_down_tim_berniers_lee)

- b) Ver <http://www.w3.org/People/> e ver <http://www.w3.org/People/Berners-Lee/>.

Neste último link repare no símbolo  (RDF “Linked” Data) à direita no início da página. Selecciono o símbolo e note que é feita uma representação gráfica de uma estrutura FOAF (recorde aula anterior); a representação não está a funcionar bem em todos os Browsers.

- c) Ver <http://www.w3.org/People/Berners-Lee/card.rdf> (igual a `a00_card_local.rdf`) e note que a estrutura que representa é idêntica à que está apresentada no site.

## 2. “Linked Data” – Concretização na DBPedia “SPARQL end-point”

*“A SPARQL endpoint is a conformant SPARQL protocol service (...) enables users (human or other) to query a knowledge base via the SPARQL language. Results are typically returned in one or more machine-processable formats.”; este excerto foi retirado de um site que atualmente não está ativo [http://semanticweb.org/wiki/SPARQL\\_endpoint](http://semanticweb.org/wiki/SPARQL_endpoint)). Atualmente é melhor recorrer à especificação (cf., <https://www.w3.org/TR/sparql11-protocol/>) para ficar a descrição formal da noção de “SPARQL endpoint”.*

- a) Ver <http://wiki.dbpedia.org/about/> em “About – (...) DBpedia data is served as Linked Data, which is revolutionizing the way applications interact with the Web. One can navigate this Web of facts with standard Web browsers, automated crawlers or pose complex queries with SQL-like query languages (e.g. SPARQL). Have you thought of asking the Web about all cities with low criminality, warm weather and open jobs? That's the kind of query we are talking about.”

**ISEL**  
**DEETC - Mestrado em Engenharia Informática**  
**Guia Aula Prática**

<b>Representação e Processamento de Conhecimento (RPC)</b>
--

em “The English version of the DBpedia knowledge base – (...) *The DBpedia knowledge base allows you to ask quite surprising queries against Wikipedia, for instance “Give me all cities in New Jersey with more than 10,000 inhabitants” or “Give me all Italian musicians from the 18th century”. Altogether, the use cases of the DBpedia knowledge base are widespread and range from enterprise knowledge management, over Web search to revolutionizing Wikipedia search.*

- b) Ver <http://dbpedia.org/sparql/> Testar a instrução SPARQL que lá está (nesse site). Notar que usa uma sintaxe um pouco diferente (Turtle); mas a n-Triple continua válida. Substituir: “[[] a ?Concept” por “?\_ rdf:type ?Concept”. Atenção: se der erro escreva, no topo, o *namespace* (i.e., PREFIX rdf: <<http://www.w3.org/1999/02/22-rdf-syntax-ns#>>). Em vez de “?\_” escreva “?s” e execute a diretiva; note que nada se altera em relação à anterior.
- c) Projete agora sujeito “s” com “select distinct ?s ?Concept where {?s a ?Concept}”.

### 3. “RDF4J” (ex “Sesame”) – gerir (meus) contributos “Linked Data”

- a) “*Eclipse RDF4J (formerly known as Sesame) is a powerful Java framework for processing and handling RDF data. This includes creating, parsing, scalable storage, reasoning and querying with RDF and Linked Data*”.

Ver <https://rdf4j.org/>

- b) Descarregue `eclipse-rdf4j-3.6.3-sdk.zip`, ou use o da pasta “distribuicao\_02”.  
(<https://www.eclipse.org/downloads/download.php?file=/rdf4j/eclipse-rdf4j-3.6.3-sdk.zip>)
- c) Descomprimir o .zip (isto pode demorar algum tempo) na pasta desta aula prática. Identifique os dois ficheiros em `eclipse-rdf4j-3.6.3\war` (\*.war – Web application Archive ) ou use os da pasta “distribuicao\_02”.

Note que RDF4J substitui o “Sesame” e que esta versão 3.6.3 é bastante recente (24.abr.2021).

Documentação (geral e utilização via Web-application):

cf., <https://rdf4j.org/documentation/tools/server-workbench/>.

### 4. “Jetty – Servlet Container” – um servidor (http) para o “RDF4J”

- a) Ver <https://www.eclipse.org/jetty/download.html/>

Descarregue `jetty-distribution-9.4.8.v20171121.zip`, ou “distribuicao\_02”. Pode descarregar a versão mais recente do Jetty (a 11.\*) e seguir as instruções em `README.adoc` para efetuar o processo de instalar um módulo onde iniciar a WebApp. No entanto esse processo exige versão de Java mais recente que a minha instalada pelo que não testei esse processo.

- b) Descomprimir o .zip na pasta desta aula prática. Na pasta construída (Jetty) ler `README.txt`.
- c) Copiar os dois ficheiros que estão em `eclipse-rdf4j-3.6.3\war` para dentro da pasta: `jetty-distribution-9.4.8.v20171121\webapps\`.
- d) Inicie um ambiente de consola e aí vá à pasta `jetty-distribution-9.4.8.v20171121\`.

**ISEL**  
**DEETC - Mestrado em Engenharia Informática**  
**Guia Aula Prática**

<b>Representação e Processamento de Conhecimento (RPC)</b>
--

- e) Inicie o servidor “Jetty” no porto 8280 executando:

```
java -jar start.jar jetty.http.port=8280
```

Caso o “firewall” do Windows seja ativado surge pedido para atribuir acesso ao “Jetty”; caso não surja e o “firewall” do Windows esteja ativado deve atribuir acesso ao “Jetty”.

A iniciação do servidor http pode apresentar muita informação. No final, o servidor http inicia e na janela de consola surge algo como “(...) INFO:oejs.Server:main: Started (...)”.

- f) Teste o “Jetty” apontando o Browser a `http://localhost:8280/rdf4j-workbench/`

*Nota:* para iniciar o servidor “Jetty” pode usar (depois de ajustar) o script `jetty_start.bat` fornecido nesta aula.

**ISEL**  
**DEETC - Mestrado em Engenharia Informática**  
**Guia Aula Prática**

<b>Representação e Processamento de Conhecimento (RPC)</b>
--

## 5. Gerir repositórios RDF com o “RDF4J” via “linha-de-comando”

Encontra documentação de suporte adicional em:

> <https://rdf4j.org/documentation/tools/console/>

- a) Inicie um ambiente de consola e aí vá à pasta `eclipse-rdf4j-3.6.3\bin`.

Note que o servidor http (“Jetty”) deve continuar em execução”!

- b) Inicie o ambiente de interação com o “RDF4J” executando:

`console.bat` (ou `./onsole.sh`, em Unix).

- c) Execute cada uma das seguintes instruções e analise cada resultado:

```
help
connect http://localhost:8280/rdf4j-server
show repositories
create native-rdfs (nota: defina ID como “repo-con” e “title” como “repo-con-titulo”
e aceite os valores de omissão para os restantes)
open repo-con
sparql SELECT ?s WHERE { ?s rdf:type rdfs:Class }
verify "<...>\Aula_04_LinkedData\a00z_amigo.rdf" (nota: coloque “path” completo)
```

- d) Analise o conteúdo do ficheiro “a00z\_amigo.rdf” e desenhe (no papel) o grafo aí descrito.

*Nota:* os ficheiros “a00z\_amigo.rdf” e “a00z\_amigo.xml” são exatamente iguais, no entanto a extensão “.rdf” é importante para a verificação (e carregamento) em comando linha.

- e) Teste a verificação (“verify”) do “a00z\_amigo.xml” e note que “*File contains 0 statements*”!

- f) Execute cada uma das seguintes instruções e analise cada resultado:

```
load "<...>\Aula_04_LinkedData\a00z_amigo.rdf" (nota: coloque “path” completo)
show repositories
sparql
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?s WHERE { ?s rdf:type foaf:Person }
. (note que este comando – “multi-line sparql” - termina com um ponto “.”)
quit
```

- g) Identifique, no grafo que desenhou no papel, o resultado obtido pela interrogação anterior.

**ISEL**  
**DEETC - Mestrado em Engenharia Informática**  
**Guia Aula Prática**

<b>Representação e Processamento de Conhecimento (RPC)</b>
--

## 6. Gerir repositórios RDF com o “RDF4J” via “Browser”

*Nota:* tive problemas ao utilizar o “Internet Explorer” (não apresenta resultado de interrogações); não tive problemas ao utilizar o Chrome. Mais informação sobre compatibilidade Browser em: [http://docs.rdf4j.org/server-workbench-console/#\\_installing\\_rdf4j\\_server\\_and\\_rdf4j\\_workbench](http://docs.rdf4j.org/server-workbench-console/#_installing_rdf4j_server_and_rdf4j_workbench). (cf., secção 3.1)

- a) Aceda ao RDF4J apontando o Browser a <http://localhost:8280/rdf4j-workbench/>
- b) Seleccionar “Repositories”. Seleccionar “New repository” e definir:  
**Type:** Native Java Store RDF Schema // **ID:** repo-web // **Title:** repo-web-titulo  
**(confirme estas escolhas pois têm dependência com o que se segue)**  
depois seleccione: **[Next]** e em seguida **[Create]**
- c) Seleccionar “System \ Information”. Seleccionar “Explore \ Summary”.
- d) Seleccionar “Modify \ Add”; em “Data Format” escolher “RDF/XML”.
- e) Seleccionar [Browse] e escolher o ficheiro “a00z\_amigo.xml” (na pasta desta aula prática).  
**ATENÇÃO:** confirme que em “Data Format” se mantém “RDF/XML” (houve casos em que a opção se alterou após escolha ficheiro passando a “(autodetect)” e depois não funcionou corretamente.
- f) Seleccionar [Upload].
- g) Explore o repositório em “Explore \ **Query**” com directivas SPARQL tal como a que se segue.  
Caso surja o erro “Lexical error at line 1, ...” remova as aspas que envolvem a interrogação.  
Antes de carregar em <Execute> retire a seleção da opção “Include inferred statements”  

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?person ?name
WHERE { ?person rdf:type foaf:Person . ?person foaf:name ?name }
```
- h) Em “Modify \ SPARQL **Update**” teste a seguinte directiva:  

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
DELETE { ?person foaf:name 'o meu nome' }
INSERT { ?person foaf:name 'PPPPP' }
WHERE { ?person rdf:type foaf:Person . ?person foaf:name 'o meu nome' }
```
- i) Construa uma interrogação que lhe permita confirmar a atualização que realizou (alínea anterior).
- j) Limpe o repositório em “Modify \ Clear” \ <Clear Context(s)>. Volte a fazer uma interrogação e veja o número de asserções em “Explore \ Summary”.
- k) Elimine todo o repositório repo-web (em “Delete repository”).

**ATENÇÃO:** convém verificar se ele foi apagado do disco. Em geral (no Windows 10) o repositório está em pasta de nome igual ao ID (atribuído no momento em que foi criado) em:

“C:\Users\<UserName>\AppData\Roaming\RDF4J\Server\repositories” (se precisar de remover aí uma pasta deve primeiro terminar a execução do servidor “Jetty” <ctr-c>).

**ISEL**  
**DEETC - Mestrado em Engenharia Informática**  
**Guia Aula Prática**

<b>Representação e Processamento de Conhecimento (RPC)</b>
--

---

## 7. Inferência usando SPARQL sobre repositório “RDF4J”

Garanta que tem o repositório com o grafo em “a00z\_amigo.xml” (ou “a00z\_amigo.rdf”).

- a) Em “Explore \ Query” **desative** a opção “Include inferred statements”, execute a diretiva abaixo e analise o resultado:

```
SELECT ?s ?o WHERE { ?s rdf:type ?o }
```

- b) Em “Explore \ Query” **ative** a opção “Include inferred statements”, execute a diretiva abaixo e analise o resultado:

```
SELECT ?s ?o WHERE { ?s rdf:type ?o }
```

*Nota:* a diferença entre as duas respostas anteriores resulta da aplicação das “regras semânticas” do RDFS com as quais se obtêm asserções que não estão explicitamente descritas no grafo (as regras e processo de obtenção dessas “novas” asserções são analisadas na aula teórica).

## 8. Usar o “RDF4J” para gerir os “meus repositórios” RDF

- c) Construa um repositório de “LinkedData” adicionando ao RDF4J os grafos (documentos RDF) que construiu nas aulas (práticas) anteriores.
- d) Construa 2 interrogações para cada um dos repositórios que construiu.
- e) Utilize o ambiente interativo do RDF4J para executar as diretivas construídas na alínea anterior.
- f) Construa 2 diretivas SPARQL que atualizem os repositórios que construiu.
- g) Utilize o ambiente interativo do RDF4J para executar as diretivas construídas na alínea anterior.