

Representação e Processamento de Conhecimento na Web (RPCW2024)

2024-04-09

Objetivos

O objetivo desta ficha é aferir os conhecimentos adquiridos na UC até ao momento.

Quando tiver terminado coloque os resultados (ficheiros) no GIT que criou para a UC numa pasta **"Aferição"**.

Recursos de apoio

- [Validador oficial \(W3C\) para documentos RDF no formato RDF/XML.](#)
- [Validador para documentos RDF no formato Turtle.](#)
- [Conversor de RDF para outros formatos.](#)
- [Conceitos e sintaxe abstrata do RDF.](#)
- [Linguagem Turtle.](#)
- [RDF Schema.](#)
- [SPARQL.](#)
- [Ficheiro JSON com o dataset para o exercício da avaliação dos alunos.](#)
- [Ficheiro JSON com o dataset para o exercício do mapa virtual.](#)

Exercício 1: Avaliação de alunos

1. Cria uma ontologia OWL para modelar o universo da avaliação de alunos registado no dataset fornecido:
 - Cria as classes: Aluno, TPC Projeto e Exame;
 - Analisa o dataset que te é dado e cria as relações e propriedades que achares convenientes;
 - Povoia a tua ontologia com o dataset.
2. Cria um repositório no teu GraphDB local de nome **"Alunos"** e importa para lá a tua ontologia;
3. Sobre a ontologia especifica as queries SPARQL capazes de gerar os seguintes resultados:
 - Quantos alunos estão registados? (inteiro)
 - Quantos alunos frequentam o curso "LCC"? (inteiro)
 - Que alunos tiveram nota positiva no exame de época normal? (lista ordenada alfabeticamente por nome com: idAluno, nome, curso, nota do exame);
 - Qual a distribuição dos alunos pelas notas do projeto? (lista com: nota e número de alunos que obtiveram essa nota)
 - Quais os alunos mais trabalhadores durante o semestre? (lista ordenada por ordem decrescente do total: idAluno, nome, curso, total = somatório dos resultados dos TPC)
 - Qual a distribuição dos alunos pelos vários cursos? (lista de cursos, ordenada alfabeticamente por curso, com: curso, número de alunos nesse curso)

- Coloca as queries completas, com os prefixos necessários, num ficheiro de texto e coloca este na pasta do Git.
4. Cria uma API de dados com as seguintes rotas/pedidos (todas as respostas deverão ser em JSON):
- **GET /api/alunos** - Devolve a lista dos alunos, ordenada alfabeticamente por nome, com os campos: idAluno, nome e curso;
 - **GET /api/alunos/:id** - Devolve a informação completa de um aluno (nesta rota, considere para **id** o campo idAluno);
 - **GET /api/alunos?curso=X** - Devolve apenas uma lista, ordenada alfabeticamente por nome, com os alunos do curso X;
 - **GET /api/alunos/tpc** - Devolve a lista de alunos (com idAluno, nome e curso), ordenada alfabeticamente por nome, e um quarto campo correspondente ao número de TPC realizados;
 - **GET /api/alunos?groupBy=curso** - Devolve a lista de cursos, ordenada alfabeticamente, e para cada um indica quantos alunos estão registados;
 - **GET /api/alunos?groupBy=projeto** - Devolve uma lista de notas registadas no projeto e para cada um indica o total de alunos que a obtiveram;
 - **GET /api/alunos?groupBy=recurso** - Devolve a lista de alunos, ordenada alfabeticamente por nome, que realizaram o exame de recesso: idAluno, nome, curso, recesso;
 - **GET /api/alunos/avaliados** - Devolve uma lista de alunos, ordenada alfabeticamente por nome, com o resultado final: idAluno, nome, curso e notaFinal, em que notaFinal poderá ser "R" ou um valor entre 10 e 20 calculado da seguinte forma:
 - Se a nota do Projeto for inferior a 10 o resultado é "R";
 - Se o máximo das notas obtidas em exame for inferior a 10 o resultado é "R";
 - A nota final é calculada somando todos os resultados obtidos nos TPC, e somando a este resultado 40% da nota do projeto e 40% da nota máxima obtida em exame; se esta nota final for inferior a 10 o resultado é "R" caso contrário o resultado é a nota calculada.

Exercício 2: Mapa Virtual

Analisa o dataset do mapa virtual disponibilizado em cima e desenvolve as seguintes alíneas:

- Especifica uma ontologia OWL para este dataset que inclua as classes: Cidade, Distrito e Ligação;
- Acrescenta as Object Properties e Data Properties que achares necessárias;
- Povoia a tua ontologia com o dataset;
- Cria um repositório no GraphDB de nome **MAPA** e carrega lá a tua ontologia;
- Sobre a ontologia especifica as queries SPARQL capazes de gerar os seguintes resultados:
 - Lista de cidades, ordenada alfabeticamente pelo nome;
 - Distribuição das cidades por distrito: lista de distritos ordenada alfabeticamente em que para cada um se indica quantas cidades tem;
 - Que cidades têm ligações diretas com Braga? (Considera Braga como origem mas também como destino)
 - Partindo de Braga, que cidades se conseguem visitar? (Apresenta uma lista de cidades ordenada alfabeticamente)
 - Através duma query CONSTRUCT cria uma ligação direta entre Braga e todas as cidades que se conseguem visitar a partir dela.
 - Através duma query CONSTRUCT cria uma ligação direta entre cada uma das cidades e todas as cidades que se conseguem visitar a partir dela.

Parte 2:

Constrói um programa, numa linguagem à tua escolha, para enviar ao graphdb uma das queries CONSTRUCT anteriores (à tua escolha) e de seguida com o resultado obtido enviar novo pedido para aumentar a ontologia (realizar um "insert" dos triplos obtidos).

Coloca num ficheiro de texto as queries completas e a lista de resultados que obtiveste da sua execução.

Coloca este ficheiro e o programa da alínea anterior no Git desta ficha.