

ECI046 – Ontologias em Organizações
Prof.: Renato Fabiano Matheus
Supervisão: Maurício Barcellos

Atividade Avaliativa 03 - Exercício individual

Versão 20181206 (modificações posteriores a versão inicial **marcadas e marcadas**)

Prazo de entrega: 09/12/2018 até 23h55 Valor: 40 pontos Entrega via Moodle.

Obs: entrega com atraso não serão possíveis. Entrega em 09/12/2018 até 23h55 valendo 40 pontos.

Este documento atualizado encontra-se em:

<https://docs.google.com/document/d/12ywy1WGtmP44SGp48QoyqMpSfEOXd5wI2Jc0OclmZg/edit>

(mudar para endereço do seu documento no Google Drive)

Aluno: **Yuri Augusto Blanco**

Descrição da atividade

1. Implementar as ontologias OWL e consultas SPARQL necessárias para solução especificada na Atividade Avaliativa 02.

Passos preliminares

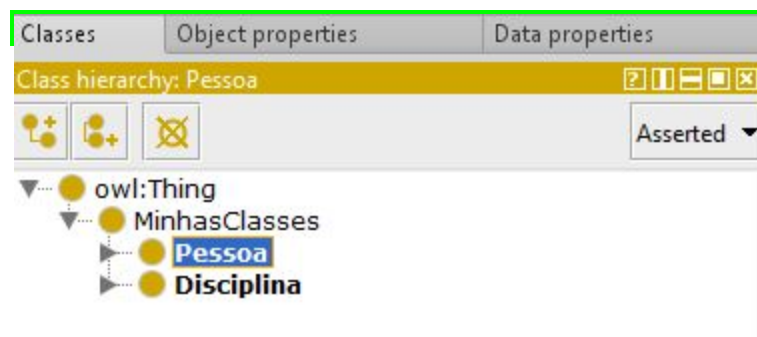
2. Criar uma cópia deste documento no Google Drive e editar a cópia como seu documento de entrega
 - URL do documento:
(URL do [documento base](#))
(Usar menu File ⇒ Make a Copy do Google Drive com usuário Google conectado e depois fazer SHARE ⇒ “Get Shareable Link” ⇒ “Done” e depois copiar endereço do documento a seguir)
<Endereço deste documento no Google Drive>

- **ENTREGA:** documento com respostas em formato PDF juntamente com uma cópia da <ontologia com nome específico.owl> via Moodle.
- Colocar link para <ontologia com nome específico.owl> neste documento e publicar no Github.
- Sugere-se colocar também uma cópia do PDF no seu Github: <endereço github>

Requisitos de arquitetura da ontologia (a implementação das ontologias será objetos da Atividade 03)

3. Criar uma nova ontologia OWL básica em RDF/XML usando Protégé e/ou [Protégé Web](#), cujo nome deve estar relacionado com a organização e o problema cuja solução você irá modelar e implementar. Os nomes das classes e propriedades de sua ontologia base devem ser em português.
4. Sua ontologia principal deve ser criada no Protégé e ter uma classe inicial de nível mais alto denominada **MinhasClasses**; as demais classes que você criar, exceto aquelas oriundas de ontologias externas (e.g. SKOS, SCHEMA.ORG) deverão ser criadas como **subClassOf** a partir de **MinhasClasses**.

Como exemplo, a ontologia usada anteriormente como exemplo, universidade.owl, ficaria como apresentada no diagrama de classes abaixo:



5. Agregar à sua ontologia básica pelo menos outras 2 (duas) ontologias vistas durante o curso ou disponíveis na Web, e.g.: Schema.org, FOAF, DBpedia Linked Data, SKOS, BFO e OBO-Foundry, ... (ver [slides](#) usados em aulas). (NÃO É REQUISITO NECESSÁRIO)
6. Sua ontologia base deve conter pelo menos 5 classes, cada classe pelo menos 3 atributos e 3 consultas SPARQL. As consultas SPARQL devem consultar preferencialmente pelo menos 2 classes.
7. Lembre-se de usar restrições de propriedades OWL (InverseOf, SameAs, DistinctWith,

Min/Max) (ver apresentações sobre OWL).

8. Procure usar outras características para propriedades de dados (“lang”, com diferentes línguas “en”, “pt”; tipos de dados “string”, “integer”, outros).
9. Não utilizar como base a ontologia universidade.owl.

Especificação básica

10. Sugere-se associar os itens deste documento com o documento da Atividade Avaliativa 02, possivelmente numerando itens e requisitos para facilitar associação.
11. Nomear sua ontologia base <ontologia>.owl, buscando dar à ontologia um nome que especifique o negócio ou instituição.
12. Descreva sua ontologia base identificando nome, cada uma das propriedades de objeto e de dados e exemplificando pelo menos um **Requisito de software** ou **Requisito de interface** ao qual a classe e cada propriedade / atributo estão associados.
13. Mesclar com ontologias RDF selecionadas, descrevendo quais ontologias você usou e os passos para integração (criar tabela): <ontologia usada> ⇔ <passos para integração> (**NÃO É NECESSÁRIO**)
14. Cadastre dados de instâncias dos objetos das classes, incluindo **pelo menos 3 instâncias para cada classe**.
15. Mostre as consultas SPARQL e indique quais os **Requisitos** cada uma atende.
16. Inclua diagrama de classes com OntoGraf neste documento.
17. Publique no Github a ontologia OWL e este documento.

Solução (coloque suas respostas a partir daqui)

Apresentação da solução

1. A ontologia feita para ambas as atividades foi a acougue.owl.
2. <https://docs.google.com/document/d/12ywy1WGtmP44SGp48QoyqMpSfEOXd5wI2Jc0OclmZg/edit>
3. O problema em questão seria uma forma eficiente de retornar ao administrador informações a respeito de seus produtos e de seus fornecedores, bem como auxiliá-lo apresentado um sistema de fácil implantação de novos produtos. Para tal, o sistemas de

ontologias se mostrariam extremamente eficientes, portanto o nome da ontologia será `acougue.owl`.

4. As classes foram todas criadas abaixo de minhas classes, após análise foi verificado que não existe uma ontologia disponível em português de caráter comercial que poderia ser usada, dado a possibilidade de não haver necessidade de realizar essa mesclagem com ontologias externas.
5. Visto que será uma ontologia simples para fins práticos, não houve necessidade de agregar outras.
6. Existem as classes “minhas classes”, “produto”, “bovinos”, “suínos”, “aves”, “fornecedor” além de 8 instâncias para cumprir os requisitos.
7. As propriedades “fornece” e “fornpor” possuem caráter inverse of. As classes “bovinas”, “suínas” e “aves” são disjuntas para garantir o cumprimento dos requisitos.
8. Estranhamente não consigo definir idioma na propriedade onde ela é mais relevante: `string`.
9. Não foi utilizado nenhum elemento da ontologia `universidade.owl`.
10. Será feito.
11. O nome da ontologia é `acougue.owl`, para indicar que a ontologia ilustra questões relacionadas ao ambiente e aos produtos de um açougue, a `ç` foi omitida para evitar que haja incompatibilidade com a consulta `sparql`.
12. O nome da ontologia é `acougue`(<http://www.semanticweb.org/jigsa/ontologies/2018/11/acougue>), as propriedades são “preço”, “nome”, “contato” e “CNPJ”, os relacionamentos são “fornece” e “efornecidopor”. Um requisitos de software é “demonstrar o preço ao digitar determinada peça de carne(ID01)” e o requisito de interface não discrimina usuários por tipo, visto que é algo exclusivo ao administrador da empresa.
13. Visto que essa etapa se tornou opcional, preferi não adicionar quaisquer outras ontologias, principalmente devido à diferença de língua.

14. Foram registradas um total de 8 instâncias para 4 classes finais, cada uma com pelo menos duas características e um relacionamento.

15. Revelar o nome da peca1 quando consultado

PREFIX acougue: <<http://www.semanticweb.org/jigsa/ontologies/2018/11/acougue#>>

SELECT ?peca1 ?nome

WHERE {

 ?peca1 acougue:nome ?nomepeca1.

}

ORDER BY ?nome

Revelar o preço da peca5 quando consultado:

PREFIX acougue: <<http://www.semanticweb.org/jigsa/ontologies/2018/11/acougue#>>

SELECT ?peca1 ?preco

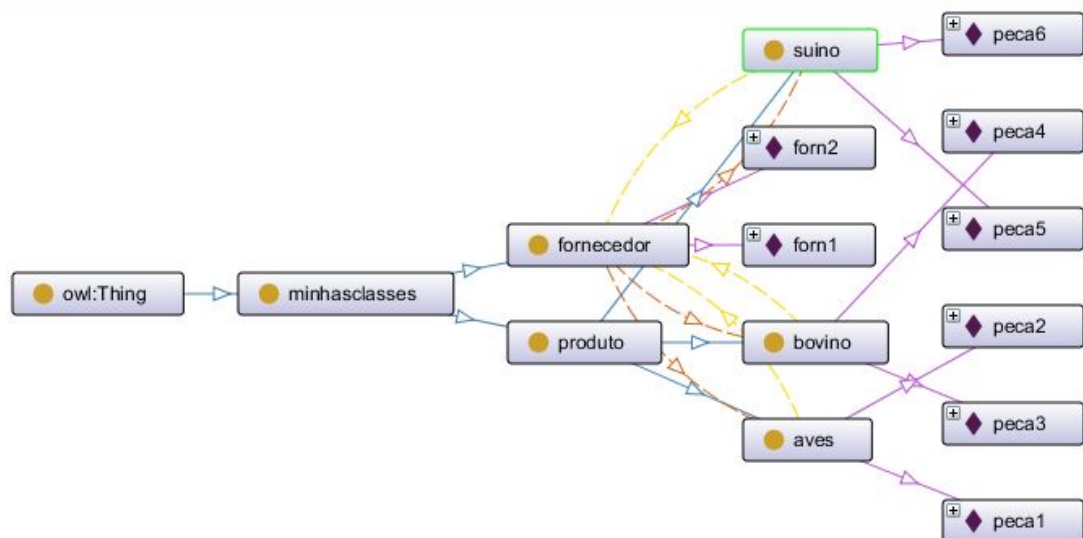
WHERE {

 ?peca1 acougue:nome ?precopeca1.

}

ORDER BY ?preco

16.



17. <https://github.com/yuriblanco/Augusto/blob/master/acougue.zip>