Universidade Federal de Minas Gerais (UFMG)
Escola de Ciência da Informação (ECI)
Colegiado de Biblioteconomia

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/18ZNI7TVodz0Jjt2tLjGq7zrRaLyyuNLLudVtJjo-sCU **(mudar para endereço do seu documento no Google Drive)**

Aluno: <escreva seu nome>

Descrição da atividade

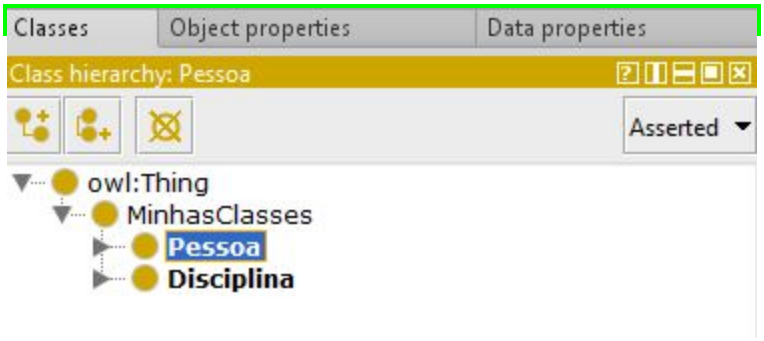
1. Implementar as ontologias OWL e consultas SPARQL necessárias para solução especificada na Atividade Avaliativa 02.

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) Criada nova ontologia com a mesma ideia, chamada de TurismoCLowl(Turismo Clean), uma reimplatação da ontologia turismo.owl levaria a cometer alguns erros na hora de gerar as consultas SPARQL, inclusive, a ontologia original não possuía nenhuma consulta devido ao fato de que esse recurso só foi mais ou menos explicado na última aula.
2) **Ontologia:** turismocl.owl
   linkdrive:https://docs.google.com/document/d/1T-v-t-lKr3LdFumD7a0r6LBCbSmrS5tCYh0KxoAH3jY/edit?usp=sharing
   GITHUB: https://github.com/unrelatedgarbage
   Nome: Ian Naor Amaru Romeu
3) Checar ontologia.
4) Checar ontologia.
5) Não foi feito por causar muita entropia na ontologia, ainda acredito que a melhor forma de utilizar mesclagem de ontologia é utilizar a estrutura como base e ir montando a partir de então, para efeitos desse exercício, até por conta da diferença de língua, isso é impraticável.
6) Checar ontologia.
7) Checar ontologia.
8) Checar ontologia.
9) .
10) Explicado na número 1.
11) turismocl.owl

12) Nome da ontologia é turismocl.owl.

Classes são todas subclasses de minhas classes, são as seguintes e servem para:

Pacote: Registrar pacotes de turismo.

Acomodação: Registrar hotel, pousadas, etc.

Atividade > Esportes: Registro de provedores de atividades esportivas, identificação das atividades.

Atividade > Relaxamento: Registro de provedores de atividades relaxantes, identificação das atividades.

Estado > Cidade: Registro de estados e cidades.

Relacionamentos são e servem para:

PossuiAcomodação: Serve para relacionar pacotes a suas acomodações.

PossuiAtividade: Serve para relacionar pacotes aos provedores de atividades.

PossuiLocal: Serve para relacionar pacotes, acomodações ou provedores de atividades às cidades em que são executados.

PossuiCapital: Serve para indicar a capital de determinado estado.

PossuiEstado: Serve para indicar o estado de determinada capital, e para servir de exemplo de inverseOf,

PossuiPacote: Serve para indicar a qual pacote(s) pertence(m) uma atividade, cidade ou acomodação.

Propriedades são e servem para:

TipoAcomodacao: Indica o tipo de acomodação, tipo string.

NomeAcomodacao: Indica o nome da acomadação, tipo string.

NomeAtividade: Serve para nomear as atividades que um provedor de atividades dá.

NomeCidade: Serve para nomear instâncias de cidade.

NomeEstado: Serve para nomear instâncias de estado.

Instâncias são:

CD1: Cidade 1, possui NomeCidade “Recife”; PossuiPacote PC1 e PossuiEstado ES1.

CD2: Cidade 1, possui NomeCidade “Natal”; PossuiPacote PC2 e PossuiEstado ES2.

ES1: Estado 1, possui NomeEstado “Pernambuco” e PossuiCapital CD1.

ES1: Estado 2, possui NomeEstado “Rio Grande do Norte” e PossuiCapital CD1.

Club1: Clube 1, possui NomeAtividade “Basquete”, “Voley” e “Futebol”; PossuiPacote PC1 e PossuiLocal CD1

SPA1: Spa 1, possui NomeAtividade “Tratamento térmico”, “Massagem” e “Sauna”; PossuiPacote PC2 e PossuiLocal CD2

Hot1: Possui NomeAcomodacao “Hotel Lua Cheia”, TipoAcomodacao “Hotel”; PossuiPacote PC1 e PossuiLocal CD1

Pou1: Possui NomeAcomodacao “Pousada Vida Longa”, TipoAcomodacao “Pousada”; PossuiPacote PC2 e PossuiLocal CD2.

PC1: PossuiLocal CD1; PossuiAcomodacao Hot1; PossuiAtividade Club1

PC2: PossuiLocal CD2; PossuiAcomodacao Pou1; PossuiAtividade Spa1

Detalhe: Muito trabalhoso e distrativo querer descrever de acordo com a atividade 2, especialmente quando a maior parte da lição passada foi sobre mecânica do Protege, recomendo cortar isso em futuras lições.

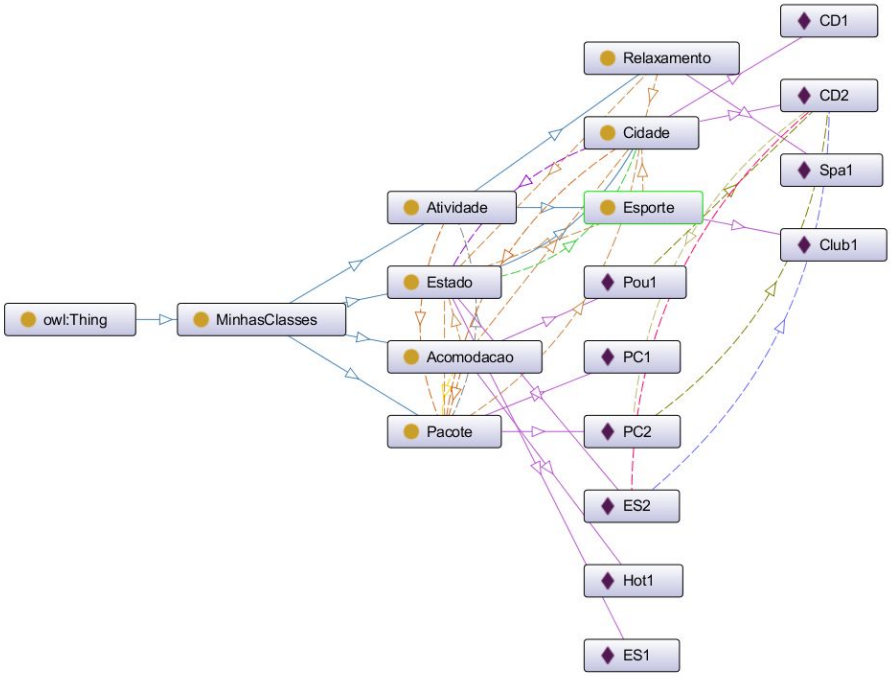
13) Desconsiderado.
14) Impraticável, quais instâncias entram na classe “minhaclasse”? Em um exemplo que existe a classe pessoa e as subclasses pessoas físicas e jurídicas, quais seriam as instâncias da classe pessoas? E a repetição é desnecessária. Foram feitas duas instâncias para cada classes finais com diversas características e relacionamentos, julgo ser o bastante.
15) Erros e problemas constantes de programação e sintaxe, não consegui fazer a sua rodar como exemplo(só apresentava campos vazios, mesmo declarando o this ou o sisbacen nos prefixos de ontologia), abaixo as tentativas falhas na minha ontologia.

Retornar quais pacotes possuíam a cidade de Recife.

```
PREFIX turismocl: <http://www.semanticweb.org/iannaor/ontologies/2018/11/turismocl#>
SELECT ?subject ?valorPossuiPacote
WHERE {
        ?subject turismocl:NomeCidade”Recife”^^<http://www.w3.org/2001/XMLSchema#string>;
         turismocl:PossuiPacote .
      }
```

Retornar quais atividades um pacote tem:

```
PREFIX turismocl: <http://www.semanticweb.org/iannaor/ontologies/2018/11/turismocl#>
SELECT ?Pacote ?Esporte ?Relaxamento ?NomeAtividade
WHERE {
        ?Pacote turismocl: PossuiAtividade ?Atividade .
        ?Esporte turismocl: NomeAtividade ?NomeAtividade .
        ?Relaxamento turismocl: NomeAtividade ?NomeAtividade .
        }
ORDER BY ?NomeAtividade.
```


CD1
Relaxamento
CD2
Cidade
Spa1
Atividade
Esporte
Club1
Estado
Pou1
owl:Thing
MinhasClasses
Acomodacao
PC1
Pacote
PC2
ES2
Hot1
ES1

16)

17)