Instituto de Matemática e Estatística
da Universidade de São Paulo

Relatório do Projeto de Sistemas Baseados em Conhecimento

Grupo:	Daniel Silva Lopes da Costa	NUSP 11302720
	Marília Takaguti Dicezare	NUSP 6818375
	Thomas Palmeira Ferraz	NUSP 9348985
	Victor Senoguchi Borges	NUSP 9298580

Professora: Renata Wassermann

Dezembro
2021

Conteúdo

1 Introdução

Neste projeto foi possível nos familiarizar com as ferramentas para construção de ontologias e consultas. Para isso, criamos uma ontologia, populamos as classes criadas e realizamos consultas em SPARQL.

1.1 Objetivos

O presente projeto tem como objetivo criar uma ontologia pertinente ao mundo do Cinema. A ontologia é baseada nos conceitos e propriedades da ontologia FOAF (Friend of a Friend), mas não se restringe a ela.

2 Método

Primeiramente, criamos uma ontologia no Protégé para representar conceitos relevantes ao mundo do cinema. Utilizamos como base para isso a ontologia FOAF (Friend of a Friend).

Dessa forma, criamos os conceitos Movie (Filme), Character (Personagem), Actor (Ator ou Atriz) e Director (Diretor). Além disso, também criamos as propriedades de objeto que representam relações entre atores e personagens (plays/playedBy), atores e filmes (appearsIn/featuring), e diretores e filmes (directs/directedBy). Ademais, foram também utilizadas propriedades dos dados da FOAF, como foaf:firstName, foaf:familyName e foaf:gender, e algumas criadas, como hasTitle para o título do filme, hasDurationOf para sua duração e wasReleasedIn para seu ano de lançamento.

Com isso, foi possível popular as classes criadas com os dados da planilha de filmes disponibilizada através da criação de regras no Plugin Cellfie. Por fim, utilizamos a ferramenta sugerida SPARQL GUI Wrapper para realizar as consultas referentes às questões de competência na ontologia inferida.

2.1 Conceitos e propriedades

Os conceitos utilizados se baseiam nos agentes que envolvem o Cinema e foram criados a partir da FOAF:

- **Movie**: uma classe que representa um filme.
- **Character**: uma subclasse de foaf:Agent que representa um personagem de um filme e pode ou não ser uma pessoa.

- **Actor**: uma subclasse de foaf:Person que representa um ator ou uma atriz de um filme. Seu gênero é determinado pela propriedade dos dados foaf:gender.

- **Director**: uma subclasse de foaf:Person que representa um diretor de um filme (pode ser do sexo masculino ou feminino, não sendo especificado).

As propriedades de objeto foram criadas como subpropriedades de foaf:maker e foaf:made:

- **plays**: uma subpropriedade de foaf:maker que representa a relação entre um ator/atriz e um personagem (Actor plays Character).

- **playedBy**: uma subpropriedade de foaf:made que representa a relação inversa de plays, ou seja, entre um personagem e um ator/atriz (Character playedBy Actor).

- **appearsIn**: uma subpropriedade de foaf:maker que representa a relação entre um ator/atriz e um filme (Actor appearsIn Movie).

- **featuring**: uma subpropriedade de foaf:made que representa a relação inversa de appearsIn, ou seja, entre um filme e um ator/atriz (Movie featuring Actor).

- **directs**: uma subpropriedade de foaf:maker que representa a relação entre um diretor e um filme (Director directs Movie).

- **directedBy**: uma subpropriedade de foaf:made que representa a relação inversa de directs, ou seja, entre um filme e um diretor (Movie directedBy Director).

As seguintes propriedades dos dados da FOAF foram utilizadas para caracterizar as subclasses de foaf:Person:

- **foaf:firstName**: representa o primeiro nome de uma pessoa (ator, atriz ou diretor).

- **foaf:familyName**: representa o sobrenome de uma pessoa (ator, atriz ou diretor).

- **foaf:gender**: uma string que representa o gênero da subclasse Actor, "male"para ator e "female"para atriz.

e outras propriedades relacionadas aos filmes foram criadas:

- **hasTitle**: uma subpropriedade de foaf:title que representa o título de um filme (Movie hasTitle xsd:string).
- **hasDurationOf**: representa o tempo de duração de um filme (Movie hasDurationOf xsd:integer).
- **wasReleasedIn**: representa o ano de lançamento de um filme (Movie wasReleasedIn xsd:integer).

2.2 Base de Dados

A base de dados consiste num arquivo xlsx disponibilizado na página da disciplina no Moddle USP: e-Disciplinas. O arquivo contém as planilhas de filmes, diretores, atores e atrizes.

Tais dados foram utilizados para povoar a ontologia acima, por meio do Cellfie. Foram criadas as regras necessárias para popular as classes criadas na etapa anterior (Movie, Actor, Director e Character), assim como estabelecer as relações e propriedades de cada indivíduo da ontologia.

3 Consultas e Resultados

Em todas as consultas feitas foi utilizado o seguinte cabeçalho, ela faz referência a nossa ontologia, bem como a foaf utilizada. Esses prefixos foram necessários para a execução de todas as queries, e são usados para fazer referência a uma busca por propriedade ou indivíduo da ontologia.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xml: <http://www.w3.org/XML/1998/namespace>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX onto: <http://www.semanticweb.org/familiatd/ontologies
/2021/11/cinema#>
```

Para algumas implementações além da querie pedida pelo enunciado com o formato de entrada e resultado requeridos pela disciplina, o grupo também desenvolveu queries adicionais, para busca por strings, ou que tem como saída os nomes como strings, a fim de aprender mais e trazer uma nova perspectiva sobre os problemas trabalhados.

Além disso, foram destacados os resultados de alguns exemplos implementados, para deixar explícito o formato da saída e corretude do resultado. No

arquivo texto com as queries, elas estão com as entradas iguais aos exemplos descritos no presente relatório.

3.1 Primeira Querie

Quais os títulos dos filmes que foram dirigidos pelo diretor D, em ordem lexicográfica?

```
SELECT      ?TituloDoFilme

WHERE       {
             ?movie onto:directedBy onto:D.
             ?movie onto:hasTitle ?TituloDoFilme.
            }

ORDER BY    ?TituloDoFilme
```

D seria a referência ao diretor na ontologia. Também foi implementada a querie para buscar pelo nome do diretor no formato de string:

```
SELECT      ?TituloDoFilme

WHERE       {
             ?director a onto:Director.
             ?director foaf:firstName "X_p"^^xsd:string.
             ?director foaf:familyName "X_f"^^xsd:string.
             ?movie onto:directedBy ?director.
             ?movie onto:hasTitle ?TituloDoFilme.
            }

ORDER BY    ?TituloDoFilme
```

Resultados:

Se buscarmos por $D = spike\_jonze$, ou se colocarmos $X_p = Spike$ e $X_f = Jonze$ teremos o seguinte resultado:

	TituloDoFilme
1.	"Her"
2.	"Torrance Rises"

3.2 Segunda Querie

Quais os primeiros nomes e últimos nomes dos atores que participaram do filme de título F_t , em ordem lexicográfica de nome e sobrenome, com seus respectivos personagens?

```
SELECT      ?PrimeiroNome ?UltimoNome ?Personagem

WHERE       {
             ?movie onto:hasTitle "Y_n"^^xsd : string.
             ?actor a onto:Actor.
             ?actor onto:appearsIn ?movie.
             ?actor foaf:firstName ?PrimeiroNome.
             ?actor foaf:familyName ?UltimoNome.
             ?actor onto:plays ?Personagem.
            }

ORDER BY    ?PrimeiroNome ?UltimoNome
```

Resultados:

Se buscarmos com $Y_n = Interstellar$, teremos o seguinte resultado para a querie:

	PrimeiroNome	UltimoNome	Personagem
1.	"Collete"	"Wolfe"	ms_hanley
2.	"David"	"Oyelowo"	school_principal
3.	"Ellen"	"Burstyn"	murph_older
4.	"John"	"Lithgow"	donald
5.	"Mackenzie"	"Foy"	murph_10
6.	"Matthew"	"McConaughey"	cooper
7.	"Timothée"	"Chalamet"	tom

Percebe-se que os resultados estão ordenados em relação ao nome e sobrenome.

3.3 Terceira Querie

Em quais filmes os atores X e Y atuaram juntos, com os respectivos diretores e anos de lançamento, do mais novo para o mais antigo?

```
SELECT      ?Filme ?Diretor ?Ano

WHERE       {
             onto:X onto:appearsIn ?Filme.
             onto:Y onto:appearsIn ?Filme.
             ?Filme onto:directedBy ?Diretor.
             ?Filme onto:wasReleasedIn ?Ano.
            }

ORDER BY    ?Ano
```

Também podemos fazer essa busca pelo nome dos atores como strings da seguinte forma:

```
SELECT      ?Filme (CONCAT(?directorFirstName, " ",
             ?directorFamilyName) AS ?Diretor) ?Ano

WHERE       {
             ?actorX foaf:firstName "X_p".
             ?actorX foaf:familyName "X_u".
             ?actorX onto:appearsIn ?movie.

             ?actorY onto:appearsIn ?movie.
             ?actorY foaf:firstName "X_p".
             ?actorY foaf:familyName "X_u".

             ?movie onto:directedBy ?director.
             ?director foaf:firstName ?directorFirstName.
             ?director foaf:familyName ?directorFamilyName.
             ?movie onto:hasTitle ?Filme.
             ?movie onto:wasReleasedIn ?Ano.
            }

ORDER BY    ?Ano
```

Resultados:

Se colocarmos $X = allen\_leech$ e $Y = charles\_dance$. Ou para o a segunda aplicação se definirmos $X_p = Allen$, $X_u = Leech$, $Y_p = Charles$ e $Y_u = Dance$, temos:

	Filme	Diretor	Ano
1.	the_imitation_game	morten_tyldum	2014

3.4 Quarta Querie

Quais filmes do diretor do filme F possuem X ou Y como atores, com as respectivas durações em ordem crescente?

```
SELECT      ?Filme ?Duracao

WHERE      {
            ?Filme onto:directedBy onto:D.

            {onto:X onto:appearsIn ?Filme}  UNION
            {onto:Y onto:appearsIn ?Filme}.

            ?Filme onto:hasDurationOf ?Duracao.
           }

ORDER BY    ?Duracao
```

Temos a seguinte possibilidade para a busca por string:

```
SELECT      ?Filme ?Duracao

WHERE       {
            ?director foaf:firstName "D_p"^^xsd:string.
            ?director foaf:familyName "D_u"^^xsd:string.
            ?movie onto:directedBy ?director.

            ?actorX foaf:firstName "X_p"^^xsd:string.
            ?actorX foaf:familyName "X_u"^^xsd:string.

            ?actorY foaf:firstName "Y_p"^^xsd:string.
            ?actorY foaf:familyName "Y_u"^^xsd:string.

            {?actorX onto:appearsIn ?movie}
            UNION
            {?actorY onto:appearsIn ?movie}.

            ?movie onto:hasTitle ?Filme.
            ?movie onto:hasDurationOf ?Duracao.
            }

ORDER BY    ?Duracao
```

Resultados:

Se colocarmos $D = ethan\_coen$, $X = jon\_polito$ e $Y = bruce\_campbell$, temos o seguinte resultado:

	Filme	Duracao
1.	fargo	98
2.	intolerable_cruelty	100
3.	the_ladykillers	104
4.	the_hudsucker_proxy	111
5.	the_hudsucker_proxy	111
6.	miller_s_crossing	115
7.	barton_fink	116
8.	the_man_who_wasn_t_there	116

3.5 Quinta Querie

Quais pessoas atuaram em um filme e dirigiram algum filme (não necessariamente o mesmo filme, mas obrigatoriamente a mesma pessoa)?

```
SELECT    ?Pessoa ?FilmeAt ?FilmeDir

WHERE    {
          FILTER EXISTS {?movie onto:directedBy ?Pessoa}.
          FILTER EXISTS {?movie onto:featuring ?Pessoa}.
          ?FilmeAt onto:directedBy ?Pessoa.
          ?FilmeDir onto:featuring ?Pessoa.
         }
```

Se quisermos devolver o nome como string dos nomes da pessoas, temos:

```
SELECT    (CONCAT(?pessoaFirstName, " ",
          ?pessoaFamilyName) AS ?Pessoa) ?FilmeAt ?FilmeDir

WHERE    {
          ?person foaf:firstName ?pessoaFirstName.
          ?person foaf:familyName ?pessoaFamilyName.
          FILTER EXISTS ?movie onto:directedBy ?person.
          FILTER EXISTS ?movie onto:featuring ?person.
          ?FilmeAt onto:directedBy ?person.
          ?FilmeDir onto:featuring ?person.
         }
```

Resultados:

Realizando a busca acima temos uma série de resultados, segue alguns dos valores encontrados:

	Pessoa	FilmeAt	FilmeDir
1.	va_g_rdos	an_american_rhapsody	an_american_rhapsody
2.	sofia_coppola	the_virgin_suicides	the_godfather_part_iii
3.	sofia_coppola	the_virgin_suicides	cq
4.	sofia_coppola	the_virgin_suicides	the_godfather
5.	sofia_coppola	the_virgin_suicides	the_godfather_part_ii
6.	sofia_coppola	the_virgin_suicides	peggy_sue_got_married
7.	sofia_coppola	the_virgin_suicides	the_cotton_club
8.	sofia_coppola	the_virgin_suicides	torrance_rises

3.6 Sexta Querie

Quais diretores dirigiram algum filme em um ano entre os anos $N1$ e $N2$, em que os atores X e Y aparecem, do mais antigo para o mais novo?

```
SELECT      ?Ano ?Diretor ?Filme

WHERE       {
             {onto:X onto:appearsIn ?Filme}
             UNION
             {onto:Y onto:appearsIn ?Filme}.

             ?Filme onto:wasReleasedIn ?Ano.
             ?Filme onto:directedBy ?Diretor.

             FILTER (?Ano > N1).
             FILTER (?Ano < N2).
            }

ORDER BY    ?Ano
```

Podemos ter como resultado o nome do diretor como string se implementarmos da seguinte maneira:

```
SELECT      ?Ano (CONCAT(?directorFirstName, " ",
            ?directorFamilyName) AS ?Diretor) ?Filme

WHERE       {
            ?actorX foaf:firstName "X_p"^^xsd:string.
            ?actorX foaf:familyName "X_u"^^xsd:string.

            ?actorY foaf:firstName "Y_p"^^xsd:string.
            ?actorY foaf:familyName "Y_u"^^xsd:string.

            {?actorX onto:appearsIn ?movie}
            UNION {?actorY onto:appearsIn ?movie}.
            ?movie onto:hasTitle ?Filme.
            ?movie onto:wasReleasedIn ?Ano.
            ?movie onto:directedBy ?director.

            FILTER (?Ano > N_1).
            FILTER (?Ano < N_2).

            ?director foaf:firstName ?directorFirstName.
            ?director foaf:familyName ?directorFamilyName.
            }

ORDER BY    ?Ano
```

Resultados:

Se fizermos a busca com $X = jon\_polito$, $Y = bruce\_campbell$, $N_1 = 1993$ e $N_2 = 2001$ no primeiro exemplo temos como resultado:

	Ano	Diretor	Filme
1.	1994	ethan_coen	the_hudsucker_proxy
2.	1994	ethan_coen	the_hudsucker_proxy
3.	1994	joel_coen	the_hudsucker_proxy
4.	1994	joel_coen	the_hudsucker_proxy
5.	1996	ethan_coen	fargo
6.	1996	joel_coen	fargo
7.	1998	joel_coen	the_big_lebowski

3.7 Sétima Querie

Quais pares formados por uma uma atriz e um ator atuaram juntos em algum filme de duração entre $M1$ e $M2$?

```
SELECT   ?Atriz ?Ator ?Filme ?Duracao

WHERE    {
          ?Atriz a onto:Actor.
          ?Atriz foaf:gender "female"^^xsd:string.
          ?Ator a onto:Actor.
          ?Ator foaf:gender "male"^^xsd:string.

          ?Filme onto:featuring ?Atriz.
          ?Filme onto:featuring ?Ator.
          ?Filme onto:hasDurationOf ?Duracao.

          FILTER (?Duracao > M1).
          FILTER (?Duracao < M2).
         }
```

Se quisermos ter como saída a string com o nome do Ator e da Atriz, temos:

```
SELECT    (CONCAT(?actressFirstName, " ",
          ?actressFamilyName) as ?Atriz)
          (CONCAT(?actorFirstName, , ?actorFamilyName) as
          ?Ator) ?Filme ?Duracao

WHERE     {
          ?actress a onto:Actor.
          ?actress foaf:gender "female"^^xsd:string.
          ?actor a onto:Actor.
          ?actor foaf:gender "male"^^xsd:string.

          ?Filme onto:featuring ?actress.
          ?Filme onto:featuring ?actor.
          ?Filme onto:hasDurationOf ?Duracao.

          FILTER (?Duracao > M1).
          FILTER (?Duracao < M2).

          ?actress foaf:firstName ?actressFirstName.
          ?actress foaf:familyName ?actressFamilyName.
          ?actor foaf:firstName ?actorFirstName.
          ?actor foaf:familyName ?actorFamilyName.
          }
```

Resultados:

Realizando essa busca, com o segundo caso, com $M_1 = 180$ e $M_2 = 204$ temos:

	Atriz	Ator	Filme	Duracao
1.	"Talia Shire"	"Salvatore Po"	the_godfather_part_ii	202
2.	"Talia Shire"	"Richard Bright"	the_godfather_part_ii	202
3.	"Talia Shire"	"Father Medeglia"	the_godfather_part_ii	202
4.	"Talia Shire"	"Tom Dahlgren"	the_godfather_part_ii	202
5.	"Talia Shire"	"Roman Coppola"	the_godfather_part_ii	202
6.	"Talia Shire"	"Joe Grippo"	the_godfather_part_ii	202
7.	"Talia Shire"	"James Gounaris"	the_godfather_part_ii	202
8.	"Talia Shire"	"John Cazale"	the_godfather_part_ii	202

3.8 Oitava Querie

Qual o diretor que mais dirigiu filmes em que aparece o ator de primeiro nome X_p e último nome X_u?

Nesse casos, assim como nos subsequentes, foi utilizado o recurso de subqueries como estratégia para realizar a filtragem necessária.

```
SELECT   (CONCAT(?directorFirstName, " ",
         ?directorFamilyName) AS ?Diretor)

WHERE   {
          {
         SELECT ?director (COUNT(?director) as ?countDirector)
               ?directorFirstName ?directorFamilyName

         WHERE {
               ?Ator a onto:Actor.
                ?Ator foaf:firstName "X_p"^^xsd : string.
                ?Ator foaf:familyName "X_u"8sd : string.
                ?movie onto:featuring ?Ator.
                ?movie onto:directedBy ?director.
                ?director foaf:firstName ?directorFirstName.
                ?director foaf:familyName ?directorFamilyName.
         }

         GROUP BY ?director ?directorFirstName
               ?directorFamilyName

         ORDER BY DESC (?countDirector)

         LIMIT 1
          }
        }
```

Resultados:
Realizando a busca com os valores $X_p = John$ e $X_u = McConnell$ temos:

	Diretor
1.	"Ethan Coen"

3.9 Nona Querie

Qual o filme mais antigo em que o ator X atuou, em que ano foi lançado e qual era seu personagem? Havendo empate, devolver todos do mesmo ano.

```
SELECT    ?Filme ?Ano ?Personagem

WHERE    {
          {
          SELECT ?Ano

          WHERE {
                ?Filme onto:featuring onto:X.
                ?Filme onto:wasReleasedIn ?Ano.
          }

          ORDER BY (?Ano)

          LIMIT 1
          }
          ?Filme onto:featuring onto:X.
          ?Filme onto:wasReleasedIn ?Ano.
          onto:X onto:plays ?Personagem.
         }
```

Também é possível realizar a busca por string com o nome do ator ou da atriz:

```
SELECT    ?Filme ?Ano ?Personagem

WHERE    {
          {
          SELECT ?Filme ?Ator ?Ano

          WHERE {
                ?Ator a onto:Actor.
                ?Ator foaf:firstName "X_p"^^xsd : string.
                ?Ator foaf:familyName "X_u"^^xsd : string.
                ?Ator onto:appearsIn ?movie.
                ?movie onto:wasReleasedIn ?Ano.
          }

          ORDER BY (?Ano)

          LIMIT 1
          }
          ?movie onto:featuring ?Ator.
          ?movie onto:wasReleasedIn ?Ano.
          ?Ator onto:plays ?Personagem.
         }
```

Resultados:

Se realizarmos a busca com $X = chris\_cooper$ teremos o seguinte resultado:

	Filme	Ano	Personagem
1.	the_horse_whisperer	1998	col_frank_fitts_usmc
2.	the_horse_whisperer	1998	frank_booker

3.10 Décima Querie

Qual o filme mais longo dirigido por D, e qual a sua duração? Havendo empate, devolver todos da mesma duração.

```
SELECT    ?Filme ?Duracao

WHERE    {
          {
          SELECT ?Duracao

          WHERE {
                ?movie onto:directedBy onto:D.
                ?movie onto:hasDurationOf ?Duracao.
          }

          ORDER BY DESC (?Duracao)

          LIMIT 1
          }
          ?Filme onto:directedBy onto:D.
          ?Filme onto:hasDurationOf ?Duracao.
         }
```

E se quisermos buscar pela string do nome do diretor:

```
SELECT    ?Filme ?Duracao

WHERE    {
          {
          SELECT ?director ?Duracao

          WHERE {
                ?director a onto:Director.
                ?director foaf:firstName "D_p"^^xsd : string.
                ?director foaf:familyName "D_u"^^xsd : string.
                ?movie onto:directedBy ?director.
                ?movie onto:hasDurationOf ?Duracao.
          }

          ORDER BY DESC (?Duracao)

          LIMIT 1
          }
          ?Filme onto:directedBy ?director.
          ?Filme onto:hasDurationOf ?Duracao.
         }
```

Resultados:
Se colocarmos $D = sofia\_coppola$, o resultado será:

	Filme	Duracao
1.	marie_antoinette	123

4 Conclusão

Durante o desenvolvimento do projeto, pudemos perceber que existem diferentes formas de criar conceitos e propriedades em uma ontologia para obter os mesmos resultados às consultas em SPARQL. Também percebemos que, algumas vezes, durante a etapa de consultas, é necessário retornar a ontologia e modificá-la de forma que ela se adéque melhor às buscas realizadas. Dessa forma, criamos apenas classes e propriedades necessárias para responder às questões de competência.

Com isso conseguimos criar um sistema baseado em conhecimento que permite realizar buscas avançadas de conceitos com as queries e, com isso, compreender todas as etapas que envolvem a produção de uma ontologia, desde a produção, povoamento e uso como ferramenta de busca.