Projeto de MAC0444 - Sistemas Baseados em Conhecimento

Feito por:

Caio Lente (8941213)
Leonardo Padilha (9298295)
Lucas Santos (9345064)
Luís Felipe de Melo (9297961)

Esse projeto consiste em criar uma ontologia relacionada ao Cinema. Os conceitos utilizados baseiam-se nos agentes deste universo. Este relatório é uma síntese do que fizemos. O arquivo `ontologia.owl` que o acompanha não está com as inferências carregadas para que o conjunto entregue fique mais leve.

Conceitos

Os conceitos utilizados foram criados ou usados a partir da FOAF. Nossa hierarquia de classes é a seguinte:

- **Agent**
  - **Movie**
  - **Person**
    - **Actor**
    - **Director**

A primeira classe é subclasse de `owl:Thing`. Segue um pouco mais de informação sobre cada classe:

- **Agent**: representa algum agente do universo (pessoa, grupo, software ou um artefato, por exemplo).
- **Movie**: simboliza os filmes obtidos na base do IMDb. Possui as seguintes *Data Properties*:
  - `foaf:name`, que é o nome do filme.
  - `onto:year`, que é o ano em que o filme foi lançado.
- **Person:** definida em `foaf:Person`. Ela tem as *Data Properties* abaixo:
  - `foaf: firstName`, que é o primeiro nome da pessoa.
  - `foaf: familyName`, que é o sobrenome da pessoa.
- **Actor**: é um ator que fez pelo menos um filme da base de dados.
- **Director**: análogo a Actor, mas para diretores.

Tratamos todos os elementos midiáticos da base (filmes, séries, episódios individuais) como filmes, de acordo com a liberdade dada no enunciado.

Propriedades

Os objetos possuem apenas as propriedades, baseadas nos análogos `foaf`, que foram pedidas no enunciado, e são:

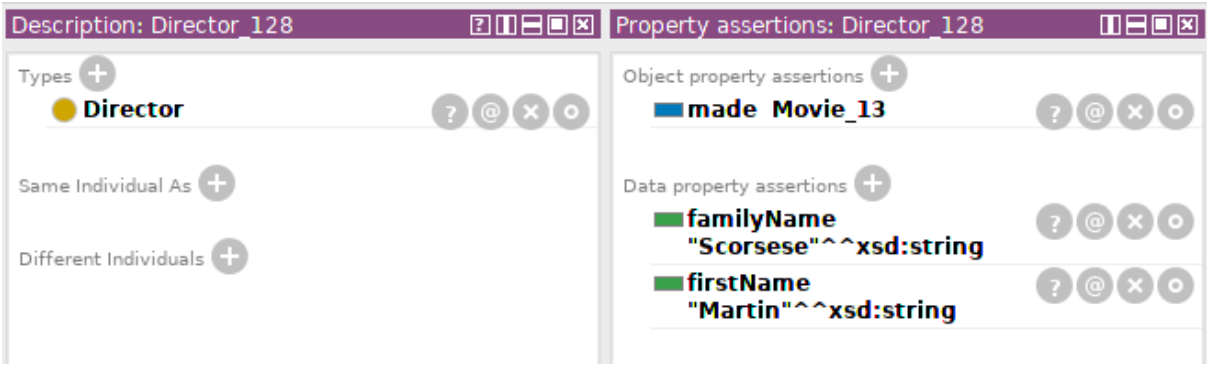
- `foaf:made`: significa *fez*, e é um predicado utilizado de pessoa para filme (`Person made Movie`).
- `maker`: significa *criador*, *feitor*, e é o inverso do predicado acima, utilizado de filme para pessoa (`Movie maker Person`).

Note que não diferenciamos a subclasse da pessoa que fez um filme (ator ou diretor). Tal diferenciação é levada em conta na hora das consultas no SPARQL, descritas mais abaixo.

Instâncias

Nossas instâncias possuem um nome interno, não relacionado ao nome real da entidade. O nome real está em `foaf:name`, no caso da classe **Movie** e em `foaf:firstName` e `foaf:familyName`, no caso de **Person**.

Um exemplo de instância em nossa ontologia é:



Podemos ver que ele possui um nome interno em nossa ontologia (`Director_128`), podemos ver que ele fez o filme `Movie_13` e que seu nome é Martin Scorsese, de acordo com as *Data Properties*. Para as classes **Actor** e **Director**, a leitura é semelhante.

Dados

Para termos os dados pedidos pelo enunciado em nossa ontologia, utilizamos uma API do IMDb. Com isso, requisitamos os atores e diretores pedidos. Depois disso, os filmes que fizeram. Então os atores e diretores presentes nesses filmes. Usamos a orientação de pegar só os primeiros 20 filmes que aparecem para cada ator e diretor indicado, para que a base de dados não ficasse tão grande.

Todo o processo foi automatizado, usando a linguagem R. Os scripts geraram arquivos no formato csv com os dados.

Para gerar a ontologia em OWL foi usado um programa em Ruby. Ele a monta usando os arquivos csv e o formato necessário. Com isso, o processo para gerar a ontologia é rápido.

Consultas e resultados

Seguem as consultas propostas e suas soluções em SPARQL. Para que as consultas funcionem, é necessária a seguinte lista de prefixos:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX onto: <http://www.semanticweb.org/padilha/ontologies/2017/10/untitled-ontology-11#>
PREFIX foaf: <http://www.ime.usp.br/~renata/FOAF-modified>
```

Para cada consulta, há um exemplo. Alguns foram encurtados.

1. Quais atores participaram do filme F?

```
SELECT ?actorFirstName ?actorFamilyName

WHERE { ?movie foaf:name "Filme F"^^xsd:string.

         ?actor a onto:Actor.
         ?actor foaf:made ?movie.
         ?actor foaf:firstName ?actorFirstName.
         ?actor foaf:familyName ?actorFamilyName
       }
```

```
SELECT ?actorFirstName ?actorFamilyName

WHERE { ?movie foaf:name "A Crime (2006)"^^xsd:string.

          ?actor a onto:Actor.
          ?actor foaf:made ?movie.
          ?actor foaf:firstName ?actorFirstName.
          ?actor foaf:familyName ?actorFamilyName
        }
```

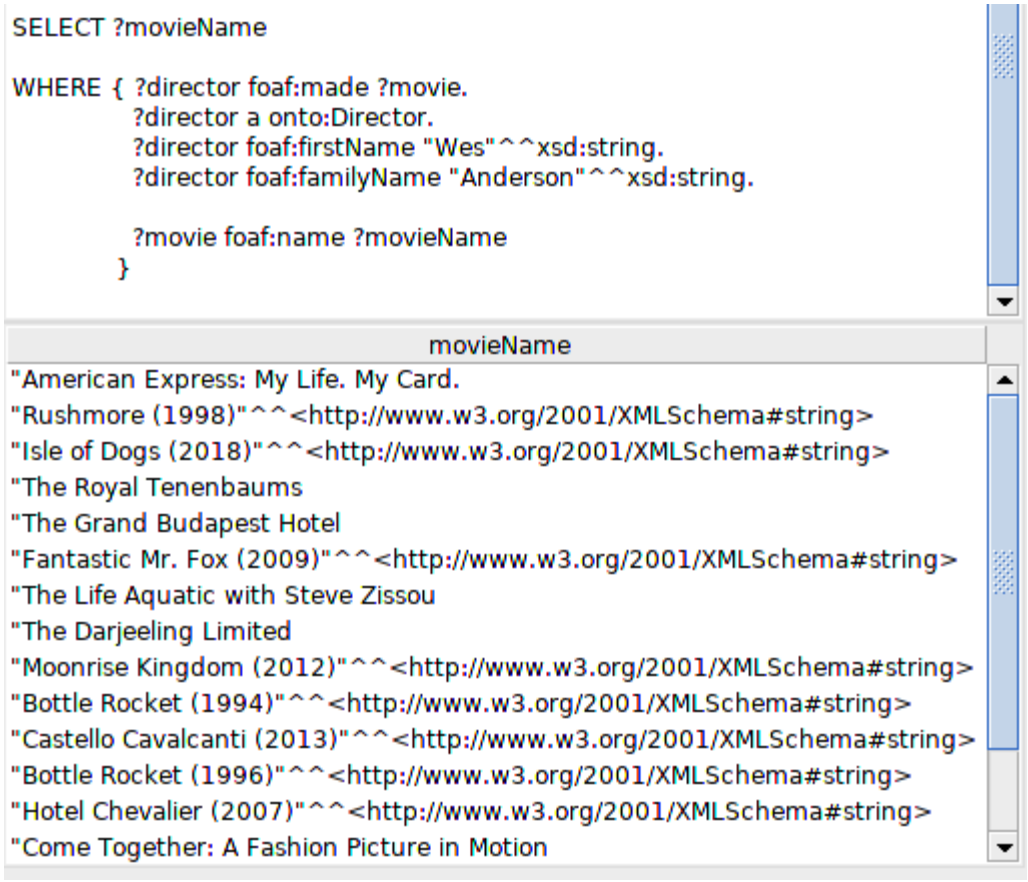
actorFirstName	actorFamilyName
"Norman"^^<http://www.w3.org/2001/	"Reedus"^^<http://www.w3.org/2001/
"Brian"^^<http://www.w3.org/2001/XM	"Tarantina"^^<http://www.w3.org/200
"Patrick"^^<http://www.w3.org/2001/>	"Collins"^^<http://www.w3.org/2001/X
"Lily"^^<http://www.w3.org/2001/XML	"Rabe"^^<http://www.w3.org/2001/XM
"Dennis Jay"^^<http://www.w3.org/20(	"Funny"^^<http://www.w3.org/2001/X
"Barnaby"^^<http://www.w3.org/2001	"Ruhe"^^<http://www.w3.org/2001/XM
"Kim"^^<http://www.w3.org/2001/XML	"Director"^^<http://www.w3.org/2001/
"Joe"^^<http://www.w3.org/2001/XML	"Grifasi"^^<http://www.w3.org/2001/X
"Memory"^^<http://www.w3.org/2001/	"Contento"^^<http://www.w3.org/200
"Tracy"^^<http://www.w3.org/2001/XM	"Westmoreland"^^<http://www.w3.org
"Stephen Michael"^^<http://www.w3.o	"Lee"^^<http://www.w3.org/2001/XML

2. Quais filmes foram dirigidos pelo diretor D?

```
SELECT ?movieName

WHERE { ?director foaf:made ?movie.
        ?director a onto:Director.
        ?director foaf:firstName "Diretor"^^xsd:string.
        ?director foaf:familyName "D"^^xsd:string.

        ?movie foaf:name ?movieName
       }
```



```
SELECT ?movieName

WHERE { ?director foaf:made ?movie.
          ?director a onto:Director.
          ?director foaf:firstName "Wes"^^xsd:string.
          ?director foaf:familyName "Anderson"^^xsd:string.

          ?movie foaf:name ?movieName
        }
```

movieName
"American Express: My Life. My Card.
"Rushmore (1998)"^^<http://www.w3.org/2001/XMLSchema#string>
"Isle of Dogs (2018)"^^<http://www.w3.org/2001/XMLSchema#string>
"The Royal Tenenbaums
"The Grand Budapest Hotel
"Fantastic Mr. Fox (2009)"^^<http://www.w3.org/2001/XMLSchema#string>
"The Life Aquatic with Steve Zissou
"The Darjeeling Limited
"Moonrise Kingdom (2012)"^^<http://www.w3.org/2001/XMLSchema#string>
"Bottle Rocket (1994)"^^<http://www.w3.org/2001/XMLSchema#string>
"Castello Cavalcanti (2013)"^^<http://www.w3.org/2001/XMLSchema#string>
"Bottle Rocket (1996)"^^<http://www.w3.org/2001/XMLSchema#string>
"Hotel Chevalier (2007)"^^<http://www.w3.org/2001/XMLSchema#string>
"Come Together: A Fashion Picture in Motion

3. Em quais filmes o ator X atuou?

```
SELECT ?movieName

WHERE { ?actor foaf:made ?movie.
        ?actor a onto:Actor.
        ?actor foaf:firstName "Ator"^^xsd:string.
        ?actor foaf:familyName "X"^^xsd:string.

        ?movie foaf:name ?movieName
      }
```

```
SELECT ?movieName

WHERE { ?actor foaf:made ?movie.
          ?actor a onto:Actor.
          ?actor foaf:firstName "George"^^xsd:string.
          ?actor foaf:familyName "Clooney"^^xsd:string.

          ?movie foaf:name ?movieName
        }
```

movieName
"Beyond Batman: Bigger, Bolder, Brighter - The Production Design of 'Batman &
"Batman: The Motion Picture Anthology 1989-1997 - Beyond Batman: Dressed
"Fantastic Mr. Fox (2009)"^^<http://www.w3.org/2001/XMLSchema#string>
"Burn After Reading (2008)"^^<http://www.w3.org/2001/XMLSchema#string
"Batman & Robin (1997)"^^<http://www.w3.org/2001/XMLSchema#string>
"A Very Murray Christmas

4. Em quais filmes o ator X atuou junto com Y?

```
SELECT ?movieName

WHERE { ?actorX foaf:made ?movie.
        ?actorX foaf:firstName "Ator"^^xsd:string.
        ?actorX foaf:familyName "X"^^xsd:string.

        ?actorY foaf:made ?movie.
        ?actorY foaf:firstName "Ator"^^xsd:string.
        ?actorY foaf:familyName "Y"^^xsd:string.

        ?movie foaf:name ?movieName
      }
```

```
SELECT ?movieName

WHERE { ?actorX foaf:made ?movie.
        ?actorX foaf:firstName "Cameron"^^xsd:string.
        ?actorX foaf:familyName "Diaz"^^xsd:string.

        ?actorY foaf:made ?movie.
        ?actorY foaf:firstName "Drew"^^xsd:string.
        ?actorY foaf:familyName "Barrymore"^^xsd:string.

        ?movie foaf:name ?movieName
    }
```

movieName
"101 Most Unforgettable SNL Moments
"Charlie's Angels (2000)"^^<http://www.w3.org/2001/XMLSchema#string>

5. Quem foram os diretores dos filmes nos quais os atores X e Y atuam juntos?

```
SELECT ?directorFirstName ?directorFamilyName

WHERE { ?director foaf:made ?movie.
        ?director a onto:Director.
        ?director foaf:firstName ?directorFirstName.
        ?director foaf:familyName ?directorFamilyName.

        ?actorX foaf:made ?movie.
        ?actorX foaf:firstName "Ator"^^xsd:string.
        ?actorX foaf:familyName "X"^^xsd:string.

        ?actorY foaf:made ?movie.
        ?actorY foaf:firstName "Ator"^^xsd:string.
        ?actorY foaf:familyName "Y"^^xsd:string
      }
```

```
SELECT ?directorFirstName ?directorFamilyName

WHERE { ?director foaf:made ?movie.
        ?director a onto:Director.
        ?director foaf:firstName ?directorFirstName.
        ?director foaf:familyName ?directorFamilyName.

        ?actorX foaf:made ?movie.
        ?actorX foaf:firstName "Uma"^^xsd:string.
        ?actorX foaf:familyName "Thurman"^^xsd:string.

        ?actorY foaf:made ?movie.
        ?actorY foaf:firstName "Harvey"^^xsd:string.
        ?actorY foaf:familyName "Keitel"^^xsd:string
      }
```

directorFirstName	directorFamilyName
"Quentin"^^<http://www.w3.org/2001/X	"Tarantino"^^<http://www.w3.org/2
"F. Gary"^^<http://www.w3.org/2001/XI	"Gray"^^<http://www.w3.org/2001/

6. Qual o diretor que mais dirigiu filmes do ator X?

```
SELECT ?directorFirstName ?directorFamilyName

WHERE { SELECT ?director (COUNT(?director) as ?countDirectors)
?directorFirstName ?directorFamilyName
       WHERE { ?director foaf:made ?movie.
               ?director a onto:Director.
               ?director foaf:firstName ?directorFirstName.
               ?director foaf:familyName ?directorFamilyName.

               ?actor foaf:made ?movie.
               ?actor foaf:firstName "Ator"^^xsd:string.
               ?actor foaf:familyName "X"^^xsd:string.
             }

             GROUP BY ?director ?directorFirstName ?directorFamilyName
             ORDER BY DESC (?countDirectors)
             LIMIT 1
     }
```

```
SELECT ?directorFirstName ?directorFamilyName
WHERE { SELECT ?director (COUNT(?director) as ?countDirectors) ?directorFirstName
?directorFamilyName
         WHERE { ?director foaf:made ?movie.
                 ?director a onto:Director.
                 ?director foaf:firstName ?directorFirstName.
                 ?director foaf:familyName ?directorFamilyName.
                 ?actor foaf:made ?movie.
                 ?actor foaf:firstName "Uma"^^xsd:string.
                 ?actor foaf:familyName "Thurman"^^xsd:string.
               }
         GROUP BY ?director ?directorFirstName ?directorFamilyName
         ORDER BY DESC (?countDirectors)
         LIMIT 1
       }
```

directorFirstName	directorFamilyName
"Quentin"^^<http://www.w3.org/2001/XMLS	"Tarantino"^^<http://www.w3.org/2001/

7. Qual o ator que mais aparece nos filmes do diretor D?

```
SELECT ?actorFirstName ?actorFamilyName

WHERE { SELECT ?actor (COUNT(?actor) as ?countActors) ?actorFirstName
?actorFamilyName
       WHERE { ?actor foaf:made ?movie.
               ?actor a onto:Actor.
               ?actor foaf:firstName ?actorFirstName.
               ?actor foaf:familyName ?actorFamilyName.

               ?director foaf:made ?movie.
               ?director foaf:firstName "Diretor"^^xsd:string.
               ?director foaf:familyName "X"^^xsd:string.
             }

             GROUP BY ?actor ?actorFirstName ?actorFamilyName
             ORDER BY DESC (?countActors)
             LIMIT 1
     }
```

```
SELECT ?actorFirstName ?actorFamilyName
WHERE { SELECT ?actor (COUNT(?actor) as ?countActors) ?actorFirstName
?actorFamilyName
         WHERE { ?actor foaf:made ?movie.
                 ?actor a onto:Actor.
                 ?actor foaf:firstName ?actorFirstName.
                 ?actor foaf:familyName ?actorFamilyName.
                 ?director foaf:made ?movie.
                 ?director foaf:firstName "Wes"^^xsd:string.
                 ?director foaf:familyName "Anderson"^^xsd:string.
               }
         GROUP BY ?actor ?actorFirstName ?actorFamilyName
         ORDER BY DESC (?countActors)
         LIMIT 1
       }
```

actorFirstName	actorFamilyName
"Bill"^^<http://www.w3.org/2001/XMLSchen	"Murray"^^<http://www.w3.org/2001/XM

8. Entre os anos N1 e N2, quais diretores dirigiram filmes onde X e Y aparecem?

```
SELECT ?directorFirstName ?directorFamilyName

WHERE { SELECT DISTINCT ?director ?directorFirstName ?directorFamilyName

        WHERE { ?director foaf:made ?movie.
                ?director a onto:Director.
                ?director foaf:firstName ?directorFirstName.
                ?director foaf:familyName ?directorFamilyName.

                ?actorX foaf:made ?movie.
                ?actorX foaf:firstName "Ator"^^xsd:string.
                ?actorX foaf:familyName "X"^^xsd:string.

                ?actorY foaf:made ?movie.
                ?actorY foaf:firstName "Ator"^^xsd:string.
                ?actorY foaf:familyName "Y"^^xsd:string.

                ?movie onto:year ?movieYear
                FILTER ( ?movieYear > N1 && ?movieYear < N2 )
              }
        }
```

```
SELECT ?directorFirstName ?directorFamilyName
WHERE { SELECT DISTINCT ?director ?directorFirstName ?directorFamilyName
        WHERE { ?director foaf:made ?movie.
                ?director a onto:Director.
                ?director foaf:firstName ?directorFirstName.
                ?director foaf:familyName ?directorFamilyName.
                ?actorX foaf:made ?movie.
                ?actorX foaf:firstName "Harvey"^^xsd:string.
                ?actorX foaf:familyName "Keitel"^^xsd:string.
                ?actorY foaf:made ?movie.
                ?actorY foaf:firstName "Frances"^^xsd:string.
                ?actorY foaf:familyName "McDormand"^^xsd:string.
                ?movie onto:year ?movieYear
                FILTER ( ?movieYear > 2010 && ?movieYear < 2017 )
              }
      }
```

directorFirstName	directorFamilyName
"Wes"^^<http://www.w3.org/2001/XMLSche	"Anderson"^^<http://www.w3.org/2001/

9. Entre os anos N1 e N2, quais atores atuaram juntos nos filmes onde X e Y aparecem?

```
SELECT ?actorFirstName ?actorFamilyName

WHERE { SELECT DISTINCT ?actor ?actorFirstName ?actorFamilyName

        WHERE { ?actor foaf:made ?movie.
                ?actor a onto:Actor.
                ?actor foaf:firstName ?actorFirstName.
                ?actor foaf:familyName ?actorFamilyName.

                ?actorX foaf:made ?movie.
                ?actorX foaf:firstName "Ator"^^xsd:string.
                ?actorX foaf:familyName "X"^^xsd:string.

                ?actorY foaf:made ?movie.
                ?actorY foaf:firstName "Ator"^^xsd:string.
                ?actorY foaf:familyName "Y"^^xsd:string.

                ?movie onto:year ?movieYear
                FILTER ( ?movieYear > N1 && ?movieYear < N2 )
              }
        }
```

10. Quais filmes do diretor do filme F possuem X ou Y como atores?

```
SELECT DISTINCT ?movieName

WHERE { ?movie foaf:name "Filme F"^^xsd:string.
        ?director foaf:made ?movie.
        ?director a onto:Director.

        ?director foaf:made ?otherMovie.

        { ?actor foaf:made ?otherMovie.
          ?actor foaf:firstName "Ator"^^xsd:string.
          ?actor foaf:familyName "X"^^xsd:string. }
        UNION
        { ?actorX foaf:made ?otherMovie.
          ?actorX foaf:firstName "Ator"^^xsd:string.
          ?actorX foaf:familyName "Y"^^xsd:string. }

        ?otherMovie foaf:name ?movieName
      }
```

```
SELECT DISTINCT ?movieName
WHERE { ?movie foaf:name "Kill Bill: Vol. 1 (2003)"^^xsd:string.
          ?director foaf:made ?movie.
          ?director a onto:Director.
          ?director foaf:made ?otherMovie.
          {?actor foaf:made ?otherMovie.
           ?actor foaf:firstName "Uma"^^xsd:string.
           ?actor foaf:familyName "Thurman"^^xsd:string. }
          UNION
          {?actor foaf:made ?otherMovie.
           ?actor foaf:firstName "Samuel L."^^xsd:string.
           ?actor foaf:familyName "Jackson"^^xsd:string. }
         ?otherMovie foaf:name ?movieName
        }
```

movieName
"Kill Bill: The Whole Bloody Affair
"Kill Bill: Vol. 2 (2004)"^^<http://www.w3.org/2001/XMLSchema#string>
"Kill Bill: Vol. 1 (2003)"^^<http://www.w3.org/2001/XMLSchema#string>
"Pulp Fiction (1994)"^^<http://www.w3.org/2001/XMLSchema#string>
"Inglourious Basterds (2009)"^^<http://www.w3.org/2001/XMLSchema#string>

Observações:

- Note que sempre que fazemos uma consulta, o formato utilizado é **Person** made **Movie**. Usamos assim para que ficasse mais semântico, mas o mesmo poderia ser feito na ontologia inferida no formado **Movie** maker **Person**.
- As consultas aqui exibidas refletem o caso geral. No caso ***Ator***/***Diretor*** representa o primeiro nome de uma pessoa, e ***X***/***Y***/***D*** representam o seu sobrenome.
- "**Filme F**" é um filme com aspas ao redor.
- ***N1*** e ***N2*** são valores inteiros. Basta substituí-los por 2017, por exemplo.
- Nas consultas 6 e 7 foi utilizado um `SELECT` dentro do outro. Isso foi feito para que apenas as colunas desejadas sejam exibidas. Como estamos usando o `GROUP BY`, é necessário que a função aritmética exiba seu resultado na consulta. O `SELECT` de fora tem a função de eliminá-la.
- Isso também foi feito nas consultas 8 e 9. Existem atores e diretores homônimos, por isso, usamos o `DISTINCT` para diferenciá-los pelo URI.