

Web Semântica- Trabalho prático 1

Relatório

Grupo 5

Bernardo Rodrigues - nº 88835 - Mestrado em Engenharia Informática

Diogo Andrade - nº 89265 - Mestrado em Engenharia Informática

Gonçalo Freixinho - nº 89251 - Mestrado em Engenharia Informática

6 de Maio de 2021

Índice

Introdução ao tema	3
Dados, suas fontes e sua transformação	4
Operações sobre os dados (SPARQL)	5
Funcionalidades da Aplicação (UI)	6
Conclusões	8
Configuração para executar a aplicação	9
Criar repositório na graphDB	9
Importar os dados na graphDB	9
Executar aplicação	9
Executar conversor de csv para nt	9

Introdução ao tema

Um Sistema de Informação caracteriza-se por um conjunto de componentes inter-relacionadas que são capazes de processar, armazenar, recuperar e distribuir informações, bem como permitir controlar uma determinada organização, de forma eficiente e organizada.

Assim, no âmbito deste tema e desta disciplina, foi proposto um problema que consiste em criar um Sistema de Informação baseado em Web em que fosse possível realizar processos de exposição e gestão de informação que se encontra na aplicação.

Para tal, foram adotadas várias tecnologias tais como: Python/Django para a programação da aplicação, dados em formato RDF (mais especificamente em NT), Triplestore GraphDB (utilizado como repositório de dados) e SPARQL (para a realização das pesquisas e alterações dos dados na triplestore).

Dados, suas fontes e sua transformação

Para o desenvolvimento do nosso sistema de informação, escolhemos um dataset acerca de shows da plataforma netflix, link do dataset:

- <https://www.kaggle.com/shivamb/netflix-shows>

Este dataset contém informação relativa aos vários filmes e séries aí contidos. A informação contida no dataset está no formato csv, deste modo é necessário proceder ao tratamento de dados para um formato rdf, ficheiros neste formato são caracterizados por cada entrada consistir num conjunto de 3 elementos, um sujeito, um predicado para o sujeito e o objeto.

Para realizar a conversão recorremos ao uso do script: conversor.py

Dados no formato CSV:

show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in
s1	TV Show	3%		João Miguel, Bianca	Brazil	August 14, 202	2020	TV-MA	4 Seasons	Internationa
s2	Movie	07:19	Jorge Michel Grau	Oscar Serrano, Azalia	Mexico	December 23,	2016	TV-MA	93 min	Dramas, Inte
s3	Movie	23:59	Gilbert Chan	Tedd Chan, Stella Ch	Singapore	December 20,	2011	R	78 min	Horror Movie
s4	Movie	9	Shane Acker	Elijah Wood, John C.	United Sta	November 16,	2009	PG-13	80 min	Action & Adv
s5	Movie	21	Robert Luketic	Jim Sturgess, Kevin S	United Sta	January 1, 202	2008	PG-13	123 min	Dramas
s6	TV Show	46	Serdar Akar	Yasemin Allen, Melis	Turkey	July 1, 2017	2016	TV-MA	1 Season	Internationa
s7	Movie	122	Yasir Al Yasiri	Amina Khalil, Ahmed	Egypt	June 1, 2020	2019	TV-MA	95 min	Horror Movie
s8	Movie	187	Kevin Reynolds	Samuel L. Jackson, Jc	United Sta	November 1, 2	1997	R	119 min	Dramas
s9	Movie	706	Shravan Kumar	Divya Dutta, Atul Kul	India	April 1, 2019	2019	TV-14	118 min	Horror Movie
s10	Movie	1920	Vikram Bhatt	Rajneesh Duggal, Ad	India	December 15,	2008	TV-MA	143 min	Horror Movie
s11	Movie	1922	Zak Hilditch	Thomas Jane, Molly	United Sta	October 20, 20	2017	TV-MA	103 min	Dramas, Thri
s12	TV Show	1983		Robert WiÅ™ckiewicz	Poland, Ur	November 30,	2018	TV-MA	1 Season	Crime TV Shc
s13	TV Show	1994	Diego Enrique Osorno		Mexico	May 17, 2019	2019	TV-MA	1 Season	Crime TV Shc
s14	Movie	2,215	Nottapon Boonpr	Artiwara Kongmalai	Thailand	March 1, 2019	2018	TV-MA	89 min	Documentari
s15	Movie	3022	John Suits	Omar Epps, Kate Wal	United Sta	March 19, 202	2019	R	91 min	Independent
s16	Movie	Oct-01	Kunle Afolayan	Sadiq Daba, David B	Nigeria	September 1, 2	2014	TV-14	149 min	Dramas, Inte

Conversão de dados para NT:

```
<http://shows.org/type/tv_show> <http://shows.org/pred/type> "TV Show" .
<http://shows.org/shows/s1> <http://shows.org/pred/type> <http://shows.org/type/tv_show> .
<http://shows.org/shows/s1> <http://shows.org/pred/title> "3%" .
<http://shows.org/people/joão_miguel> <http://shows.org/pred/name> "João Miguel" .
<http://shows.org/shows/s1> <http://shows.org/pred/cast> <http://shows.org/people/joão_miguel> .
<http://shows.org/people/bianca_comparato> <http://shows.org/pred/name> "Bianca Comparato" .
<http://shows.org/shows/s1> <http://shows.org/pred/cast> <http://shows.org/people/bianca_comparato> .
<http://shows.org/people/michel_gomes> <http://shows.org/pred/name> "Michel Gomes" .
<http://shows.org/shows/s1> <http://shows.org/pred/cast> <http://shows.org/people/michel_gomes> .
<http://shows.org/people/rodolfo_valente> <http://shows.org/pred/name> "Rodolfo Valente" .
<http://shows.org/shows/s1> <http://shows.org/pred/cast> <http://shows.org/people/rodolfo_valente> .
<http://shows.org/people/vaneza_oliveira> <http://shows.org/pred/name> "Vaneza Oliveira" .
<http://shows.org/shows/s1> <http://shows.org/pred/cast> <http://shows.org/people/vaneza_oliveira> .
```

Operações sobre os dados (SPARQL)

Os dados foram então armazenados numa base do tipo *Triplestore*, neste caso a GraphDB. Para obter informações relativas aos dados, realizamos operações sobre eles, através do uso de queries na linguagem SPARQL.

Eis o exemplo de algumas queries SPARQL usadas e o seu propósito:

```
PREFIX pred: <http://shows.org/pred/>
SELECT ?actorname (GROUP_CONCAT(?title;SEPARATOR=";") AS ?titles)
WHERE {
    ?show pred:cast ?actor .
    ?actor pred:name ?actorname .
    FILTER regex(str(?actorname), 'John' , "i") .
    ?show pred:title ?title .
} GROUP BY ?actorname
```

Procura de actores que contenham ‘John’ no nome e a lista de filmes em que estes participaram. (Na aplicação o user tem a opção de definir o nome que quer procurar, assim como tendo também a opção de listar todos os actores).

```
PREFIX pred: <http://shows.org/pred/>

SELECT ?typename ?description ?date_added ?release_yearname ?durationname
(GROUP_CONCAT(DISTINCT ?countryname;SEPARATOR=";") AS ?countriesname)
(GROUP_CONCAT(DISTINCT ?listed_inname;SEPARATOR=";") AS ?listedname)
(GROUP_CONCAT(DISTINCT ?directorname;SEPARATOR=";") AS ?directorsname)
(GROUP_CONCAT(DISTINCT ?actorname;SEPARATOR=";") AS ?castname)
WHERE {
    ?show pred:title "" + "\"" + title + "\"" + "" .
    ?show pred:type ?type .
    ?type pred:type ?typename .
    ?show pred:country ?country .
    ?country pred:country ?countryname .
    ?show pred:description ?description .
    ?show pred:date_added ?date_added .
    ?show pred:release_year ?release_year .
    ?release_year pred:release_year ?release_yearname .
    ?show pred:duration ?duration .
    ?duration pred:duration ?durationname .
    ?show pred:listed_in ?listed_in .
    ?listed_in pred:listed_in ?listed_inname .
    OPTIONAL {
        ?show pred:director ?director .
        ?director pred:name ?directorname .
    }
    OPTIONAL {
        ?show pred:cast ?actor .
        ?actor pred:name ?actorname .
    }
} GROUP BY ?typename ?description ?date_added ?release_yearname ?durationname
```

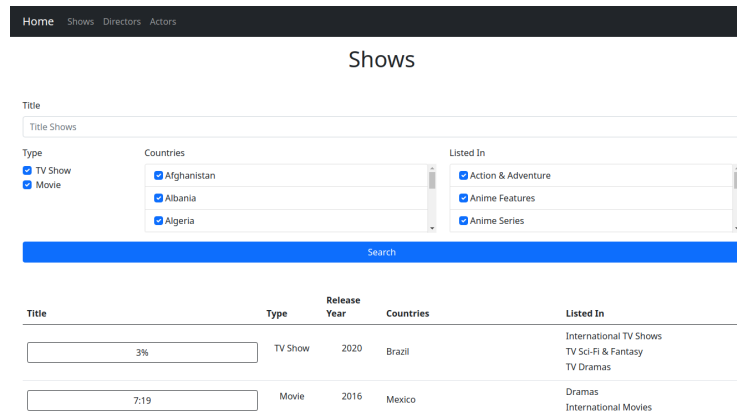
Informação de um filme/série pedido pelo utilizador, contém a descrição desta, país onde foi filmado, duração (em temporadas ou em minutos, dependendo se é série ou filme), elenco, entre outros.

Estas queries são processadas por um wrapper de python para SPARQL que permite à aplicação web Django obter a informação da query.

Funcionalidades da Aplicação (UI)

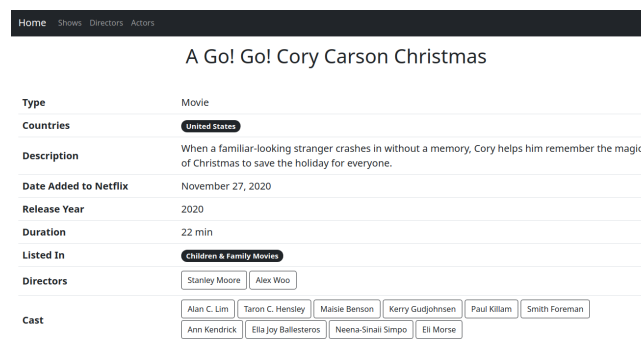
A app foi desenhada de modo a permitir aos utilizadores:

Observar e pesquisar todos os filmes/séries existentes, filtrar e procurar por séries/filmes dependendo da categoria, procurar por um filme específico e obter informação detalhada sobre este, tais como data de criação, o seu tipo e categoria.



Title	Type	Release Year	Countries	Listed In
3%	TV Show	2020	Brazil	International TV Shows TV Sci-Fi & Fantasy TV Dramas
7:19	Movie	2016	Mexico	Dramas International Movies

Visualizar informações detalhadas de cada filme/série (como por exemplo, o seu tipo, descrição, data que foi adicionado à plataforma, a sua duração, entre outras).



A Go! Go! Cory Carson Christmas

Type: Movie

Countries: United States

Description: When a familiar-looking stranger crashes in without a memory, Cory helps him remember the magic of Christmas to save the holiday for everyone.

Date Added to Netflix: November 27, 2020

Release Year: 2020

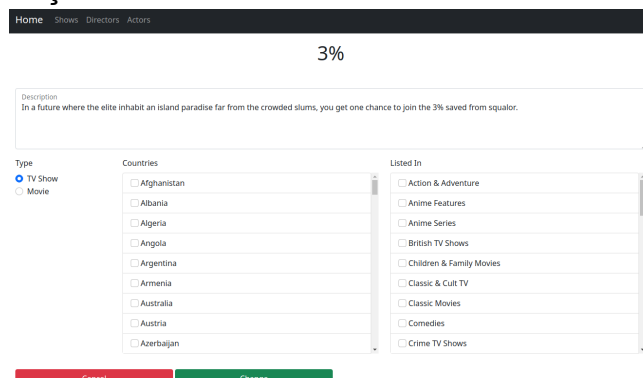
Duration: 22 min

Listed In: Children & Family Movies

Directors: Stanley Moore, Alex Woo

Cast: Alan C. Lim, Taron C. Hensley, Maisie Benson, Kerry Gudjohnsen, Paul Killam, Smith Foreman, Ann Kendrick, Ella Joy Ballesteros, Neena-Sinai Simpo, Eli Morse

Editar informações relativamente aos filmes/séries existentes na plataforma.



3%

Description: In a future where the elite inhabit an island paradise far from the crowded slums, you get one chance to join the 3% saved from squalor.

Type: ☒ TV Show ☐ Movie

Countries: ☐ Afghanistan ☐ Albania ☐ Algeria ☐ Angola ☐ Argentina ☐ Armenia ☐ Australia ☐ Austria ☐ Azerbaijan

Listed In: ☐ Action & Adventure ☐ Anime Features ☐ Anime Series ☐ British TV Shows ☐ Children & Family Movies ☐ Classic & Cult TV ☐ Classic Movies ☐ Comedies ☐ Crime TV Shows

Pesquisar pela lista de atores ou diretores e obter informação sobre quais os filmes onde estes participaram, quer seja a atuar quer seja a dirigir o filme/série.

Home Shows Directors Actors

Directors

Director Name

Name	Shows
Jorge Michel Grau	7:19
Gilbert Chan	23:59
Shane Acker	9
Robert Luketic	21 Killers Monster-in-Law The Ugly Truth

Home Shows Directors Actors

Actors

Actor Name

Name	Shows
Jolo Miguel	3%
Bianca Comparato	3% All The Reasons To Forget
Michel Gomes	3%
Rodolfo Valente	3%
Vaneza Oliveira	3%

Visualizar informação detalhada de cada pessoa (em que filmes participou como ator ou como diretor).

Home Shows Directors Actors

Stanley Moore

Director In

Title	Type	Others Directors
A Go! Go! Cory Carson Christmas	Movie	Alex Woo
Go! Go! Cory Carson: The Chrissy	Movie	Alex Woo

Actor In

Title	Type	Others Actors
Go! Go! Cory Carson: The Chrissy	Movie	<div>Alan C. Lim</div> <div>Maisie Benson</div> <div>Kerry Gudjohnsen</div> <div>Paul Killam</div> <div>Smith Foreman</div> <div>Ella Joy Ballesteros</div> <div>Terrance Smith</div> <div>Omari Weaver</div>

Conclusões

Com este trabalho, aprendemos novas competências e técnicas relacionadas com a área de web semântica que nos permitiram desenvolver com sucesso um sistema de informação baseado na web, usando Python/Django para a programação da aplicação, dados no formato RDF (NT) guardados numa base de dados Triplestore (GraphDB) e queries na linguagem SPARQL, assim sendo criámos uma aplicação onde os utilizadores têm acesso à informação relativa aos filmes e séries assim como a possibilidade de os poderem gerir.

Conseguimos assim, criar um Sistema de Informação baseado em Web constituído por vários “subsistemas”, que interagem entre si, em que é possível visualizar e gerir grandes quantidades de informação, mostrando sempre grande simplicidade quanto à sua compreensão e sobretudo eficácia e organização, de modo a ser possível responder a todos os critérios impostos para a realização deste trabalho e necessários ao seu bom funcionamento.

Configuração para executar a aplicação

Criar repositório na graphDB

- Setup > Repositories > Create new repository > GraphDB Free
 - Repository ID: shows
 - Ruleset: No inference

Importar os dados na graphDB

- Import > RDF > Import RDF files
 - data/shows.nt
 - Import

Executar aplicação

```
$ pip3 install -r requirements.txt  
$ python3 app/manage.py runserver
```

Nota:

A base de dados do graphDB foi guardada localmente, dependendo do url onde ela está guardada pode ser preciso alterar a variável *URL* (linha 4 app/shows/query.py) para o url local.

Executar conversor de csv para nt

```
$ python3 conversor/conversor.py -f data/shows.csv
```