Web Semântica 22/23 TP1 - PitchStats

Afonso Ribeiro Campos, 100055.

<u>Introdução</u>	2
Dados	3
Fonte e Descrição	3
Transformação em Triplos RDF	5
Reviews (Críticas)	5
Artist (Artista)	6
Label (Gravadora)	
Author (Autor)	
Operações Sobre os Dados	
get review count	7
get_author_types	7
get_reviews	8
get_artists	10
get_authors	11
get_genres	11
get labels	12
get_artist_detail	13
get author detail	13
get_label_detail	14
new_review	14
Aplicação Web	16
Home Page	16
Detalhe da Crítica	17
Detalhe do Autor	17
Detalhe do Artista	18
Detalhe da Gravadora	18
Géneros	19
Criar Crítica	19
Como Correr o Projeto	20

Introdução

A Pitchfork é um website com origem na América fundado em 1995 por Ryan Schriber e conhecido pela publicação diária de críticas musicais. A Pitchfork é até aos dias de hoje uma das mais influentes fontes de crítica de música.

O website aplica um sistema de crítica para os álbuns atribuindo uma avaliação numa escala de 0 a 10 (até uma casa decimal). Um álbum pode também ser classificado como "best new music" (melhor música nova) para o tempo de lançamento.

Até 2021 a Pitchfork reporta ter publicado pouco menos de 28 mil críticas, tendo coberto uma lista significativa de artistas musicais. Este projeto pretende fornecer aplicação web que permite sumarizar dados referentes às várias críticas publicadas pela Pitchfork desde 1999 até 2017 utilizando as seguintes tecnologias:

- RDF como formato dos dados em que a nossa aplicação se baseia;
- Triplestore GraphDB que toma a função de repositório dos dados;
- SPARQL para realizar pesquisas e alterações sobre os dados o que vai fornecer funcionalidade à aplicação;
- Django para a programação da aplicação web e integração com a lógica de acesso aos dados.

Dados

Fonte e Descrição

O *dataset* escolhido para o desenvolvimento da aplicação foi criado por Nolan Conaway que disponibilizou o mesmo no website <u>Kaggle</u>.

O *dataset* é disponibilizado como uma base de dados relacional sqlite e inclui mais de 18 mil críticas de álbuns do website da Pitchfork desde 5 de janeiro de 1999 até 8 de janeiro de 2017.

A base de dados possui 6 tabelas: artists, content, genres, labels, reviews e years. A maior parte da informação encontra-se na tabela reviews que mostra ser a tabela central da base de dados. Todas as outras tabelas estão relacionadas com a tabela reviews referenciando as suas linhas com foreign keys.

Segue-se um pequeno sumário das tabelas e os atributos explorados:

- Review contém as informações principais sobre as críticas, cada linha representando uma. Os seus atributos incluem:
 - o review id (identificador único de cada crítica);
 - o title (título da crítica/álbum);
 - o url (o url para o website da pitchfork onde a crítica pode ser lida);
 - o score (valor de 0 a 10 representante da avaliação do álbum);
 - best_new_music (valor 1 ou 0, 1 representando que o álbum foi nomeado como "best new music" na data de lançamento da crítica);
 - author e author_type (respetivamente, nome(s) do(s) autor(es) que redigiram a crítica e a sua função na empresa aquando a publicação da crítica));
 - o pub date (data da publicação da crítica no formato "YYYY-MM-DD").
- Labels contém duas colunas/atributos:
 - review_id (foreign key para a tabela reviews (relação many-to-one, i.e. uma crítica pode ter 0 ou mais gravadoras);
 - o label (nome da gravadora que publicou o álbum referente à crítica).
- Artists também com duas colunas/atributos:
 - review_id (foreign key para a tabela reviews (relação many-to-one, i.e. uma crítica pode ter 1 ou mais artistas));
 - o artist (nome do artista do álbum referente à crítica).
- Genres com atributos:
 - review_id (foreign key para a tabela reviews (relação many-to-one, i.e. uma crítica pode ter 0 ou mais géneros));
 - o genre (nome do género do albúm referente à crítica).

As seguintes tabelas/atributos foram ignorados:

- Reviews:
 - o artist (nome(s) do(s) artista(s)): redundante com a existência da tabela artists;
 - pub_weekday, pub_month, pub_day, pub_year (referentes à data de publicação): redundantes com a existência do atributo pub_date que sumariza todos estes.
- Years: atribui um ano a cada crítica, redundante com a existência do atributo pub date na tabela reviews;
- Content: relação one-to-one com a tabela reviews. O conteúdo, palavra por palavra, de cada crítica. Este atributo não foi utilizado porque se achou melhor apenas incluir o hyperlink para a crítica no website oficial da Pitchfork ao invés do conteúdo da crítica em si na app final.

Transformação em Triplos RDF

Com o auxílio da biblioteca Python sqlite3 foi criado o script "process_data.py" no diretório Dataset/ que transforma a base de dados relacional sqlite em dois ficheiros - pitchstats.nt (N-Triples) e pitchstats.n3 (formato N3) - ambos facilmente importados para um repositório de dados GraphDB.

Foram definidos dois prefixos, um para as entidades ("http://pitchstats/ent/") e outro para os predicados ("http://pitchstats/pred/").

Os triplos gerados envolvem 4 entidades:

Reviews (Críticas)

URI "http://pitchstats/ent/review/{identificador}".

Possíveis predicados/relações (com a entidade como sujeito):

Predicados	Objeto
pred:id	Identificador (literal, integer)
pred:album	Título da crítica/álbum (literal, string)
pred:score	Avaliação da crítica (literal, float)
pred:best_new	Classificação como "best new music" (literal, string, "1" para afirmativo)
pred:url	URL para a crítica oficial (literal, string)
pred:date	Data de publicação YYYY-MM-DD (literal, string)
pred:author	Autor da crítica (Entidade Author)
pred:author_type	Função do(s) autor(es) aquando publicação da crítica (literal, string)
pred:artist	Artista referente ao álbum da crítica (Entidade Artist)
pred:label	Gravadora referente ao albúm da crítica (Entidade Label)
pred:genre	Género (literal, string)

Artist (Artista)

URI "http://pitchstats/ent/artist/{identificador}".

Possíveis predicados/relações (com a entidade como sujeito):

Predicados	Objeto
pred:name	Nome do artista (literal, string)

Label (Gravadora)

URI "http://pitchstats/ent/label/{identificador}".

Possíveis predicados/relações (com a entidade como sujeito):

Predicados	Objeto
pred:name	Nome da gravadora (literal, string)

Author (Autor)

URI "http://pitchstats/ent/author/{identificador}".

Possíveis predicados/relações (com a entidade como sujeito):

Predicados	Objeto
pred:name	Nome do autor (literal, string)

Operações Sobre os Dados

Todas as operações foram compiladas em funções no script python "queries.py" presente no diretório Frontend/PitchStats/. Estas funções atuam sobre um repositório de dados GraphDB, nomeado "pitchstats", através da biblioteca Python s4api com a linguagem SPARQL.

Passamos a listar estas funções e as respectivas queries SPARQL.

get review count

Esta função não leva argumentos e retorna uma contagem do número de críticas distintas (?count) bem como o identificador mais alto de todas as críticas até à data (?higher_id).

A contagem das reviews é utilizada para implementação de paginação e o identificador mais alto para inserção de novas reviews no repositório de modo a não existirem conflitos com identificadores.

get_author_types

Esta função não leva argumentos e retorna todos os tipos distintos de autores. Isto é utilizado para listar os tipos existentes de autores aquando a criação de uma crítica.

get reviews

Esta função retorna uma lista de reviews e leva vários argumentos opcionais de filtragem e ordenação:

- sort_by: Define o tipo de ordenação das críticas. Aceita 3 opções: "date" (ordenação por data), "alpha" (ordenação alfabética) ou "score" (ordenação por score de avaliação). Por default ordena por data;
- sort_order: Define a ordem de ordenação, 0 para descendente, 1 para ascendente. Por default toma o valor 0;

```
if sort_by == "date":
          sort_string = f"ORDER BY {'?date' if sort_order else

'DESC(?date)'}"
    elif sort_by == "alpha":
          sort_string = f"ORDER BY {'?title' if sort_order else

'DESC(?title)'}"
    elif sort_by == "score":
          sort_string = f"ORDER BY {'?score' if sort_order else

'DESC(?score)'}"
    else:
        sort_string = ''
```

- filter_genre: Filtra as críticas pelo género musical especificado, valor default "None" representa inexistência deste filtro;
- filter_best_new_music: Filtra as críticas com base em serem, ou não "best new music", valor default "None" representa inexistência deste filtro;
- filter_date_start: Filtra as críticas por data sendo todas as críticas resultantes posteriores à data especificada, valor default "None" representa inexistência deste filtro;
- filter_date_end: Filtra as críticas por data sendo todas as críticas resultantes posteriores à data especificada, valor default "None" representa inexistência deste filtro;
- filter_score_low: Filtra as críticas pelo score sendo o score de todas as críticas resultantes mais alto que o especificado, valor default "None" representa inexistência deste filtro;
- filter_score_high: Filtra as críticas pelo score sendo o score de todas as críticas resultantes mais baixo que o especificado, valor default "None" representa inexistência deste filtro;
- filter_title: Filtra as críticas pelas palavras chaves especificadas, case insensitive, valor default "None" representa inexistência deste filtro;

- filter_review: Filtra as críticas pelo seu uri, caso este filtro seja especificado apenas uma crítica será retornada (ou nenhuma caso não encontrado o URI), valor default "None" representa inexistência deste filtro;
- filter_artist: Filtra as críticas pelo artista, i.e. apenas críticas que incluem este artista são retornadas, valor default "None" representa inexistência deste filtro;
- filter_label: Filtra as críticas pela gravadora, i.e. apenas críticas que incluem esta gravadora são retornadas, valor default "None" representa inexistência deste filtro;
- filter_author: Filtra as críticas pelo autor, i.e. apenas críticas publicadas por este autor são retornadas, valor default "None" representa inexistência deste filtro;

```
filters_inner = []
if filter_review: filters_inner.append(f"<{filter_review}> pred:url
?url .")
if filter_artist: filters_inner.append(f"?review pred:artist
<{filter_artist}> .")
if filter_label: filters_inner.append(f"?review pred:label
<{filter_label}> .")
if filter_author: filters_inner.append(f"?review pred:author
<{filter_author}> .")
```

• page size: número de reviews a retornar com esta query, por default 10;

```
limit_string = ''
if page_size:
   limit_string = f"LIMIT {page_size}"
```

page: número da página a ser retornada, por default 0.

```
offset = page * page size if page size else 0
```

Esta função permite filtrar e ordenar reviews na app final pelos seus diversos parâmetros.

Adicionalmente, a função também retorna, para cada crítica, os seus artistas, autores, gravadoras e gêneros através de outras funções (abaixo explicadas).

get_artists

Esta função retorna uma lista de artistas e leva dois argumentos opcionais de filtragem:

- filter_review: Filtra os artistas por uma crítica, i.e os artistas retornados são os artistas a qual a crítica se refere, valor default "None" representa inexistência deste filtro;
- filter_label: Filtra os artistas pela gravadora, i.e. apenas artistas envolvidos com a gravadora são retornados, valor default "None" representa inexistência deste filtro.

```
review_str = f"<{filter_review}> pred:artist ?artist ." if
filter_review else ""
label_str = f"?review pred:label <{filter_label}> ." if filter_label
else ""
```

Com isto vamos obter uma lista completa dos artistas (não especificando filtros) ou os artistas de uma certa crítica/gravadora.

get_authors

Esta função retorna uma lista de autores e leva um argumento opcional de filtragem:

• filter_review: Filtra os autores por uma crítica, i.e os autores retornados são os autores envolvidos na crítica, valor default "None" representa inexistência deste filtro.

```
review_str = f"<{filter_review}> pred:author ?author ." if
filter_review else ""
```

Com isto vamos obter uma lista completa dos autores (não especificando filtros) ou os autores de uma crítica.

get_genres

Esta função retorna uma lista de géneros bem como os seus scores médios, uma contagem de críticas com o respetivo género e o número de críticas desse género classificadas como "best_new_music", tomando 4 argumentos opcionais:

- filter_artist: Filtra os géneros por um artista, i.e. as contagens e as médias serão apenas referentes às reviews do respectivo artista, valor default "None" representa inexistência deste filtro;
- filter_label: Filtra os géneros por uma gravadora, i.e. as contagens e as médias serão apenas referentes às reviews da respectiva gravadora, valor default "None" representa inexistência deste filtro;
- filter_author: Filtra os géneros por um autor, i.e. as contagens e as médias serão apenas referentes às reviews do respetivo autor, valor default "None" representa inexistência deste filtro;

```
artist_str = f"?review pred:artist <{filter_artist}> ." if
filter_artist else ""
author_str = f"?review pred:author <{filter_author}> ." if
filter_author else ""
label_str = f"?review pred:label <{filter_label}> ." if filter_label
else ""
```

• filter_review: Filtra os géneros por uma crítica retornando apenas os géneros da respectiva crítica, valor default "None" representa inexistência deste filtro.

Com isto vamos obter várias estatísticas sobre os géneros. Por exemplo: os géneros avaliados por um autor e como varia o score atribuído às críticas entre géneros para esse autor.

get_labels

Esta função retorna uma lista de gravadoras e leva dois argumentos opcionais de filtragem:

- filter_review: Filtra as gravadoras por uma crítica, i.e as gravadoras retornadas são aquelas a qual a crítica diz respeito, valor default "None" representa inexistência deste filtro:
- filter_artist: Filtra as gravadoras pelo artista, i.e. apenas gravadoras envolvidas com o artista são retornadas, valor default "None" representa inexistência deste filtro.

```
artist_str = f"?review pred:artist <{filter_artist}> ." if
filter_artist else ""
review_str = f"<{filter_review}> pred:label ?label ." if filter_review
else ""
```

Com isto vamos obter uma lista completa das gravadoras (não especificando filtros) ou as gravadoras de uma crítica/artista.

get artist detail

Esta função retorna todos os detalhes sobre um determinado artista:

- o seu nome;
- o score médio das suas avaliações;
- as suas avaliações (get_reviews);
- as suas gravadoras (get_labels);
- os seus géneros e respectivos argumentos (get genres).

get_author_detail

Esta função retorna todos os detalhes sobre um determinado autor:

- o seu nome;
- o score médio das suas críticas:
- as suas críticas (get reviews);
- os seus géneros e respectivos argumentos (get_genres).

Adicionalmente esta função também retorna uma lista com todos os "author_types" do autor discriminados por data para mais tarde apresentarmos uma "timeline" das funções do autor na Pitchfork.

```
PREFIX pred: <a href="http://pitchstats/pred/">
SELECT DISTINCT ?type ?date WHERE {{
    ?review pred:author <{author_uri}> .
    ?review pred:author_type ?type .
    ?review pred:date ?date .
}} ORDER BY ?date
```

get_label_detail

Esta função retorna todos os detalhes sobre uma determinada gravadora:

- o seu nome;
- o score médio das suas avaliações;
- as suas avaliações (get_reviews);
- os seus artistas (get_artists);
- os seus géneros e respectivos argumentos (get genres).

new_review

Esta função permite a inserção de uma nova crítica no repositório. Leva os seguintes argumentos:

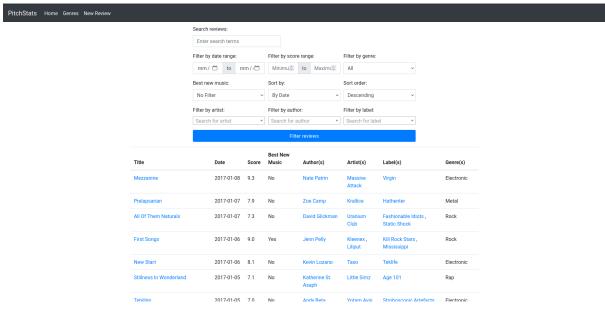
- titulo da crítica/album;
- avaliação/score;
- best new music ("1" para sim);
- url;
- data de publicação;
- URI do autor;
- URI do artista;
- URI da gravadora;
- género.

O identificador da nova crítica é selecionado através do valor de retorno da função get_review_count.

Aplicação Web

Home Page

URL: {URL_BASE}/



Esta página lista todas as críticas presentes no repositório de dados. O utilizador tem também a opção de filtrar as reviews e alterar a sua ordenação default com os vários filtros disponíveis no topo da página.

Esta página encontra-se ligada à view Django "home" que recorre a parâmetros de URL para determinar os valores dos vários filtros/ordenações. A view recorre às funções definidas no script queries.py para popular a página, como:

- get reviews (para a lista das críticas com as várias filtragens);
- get_genres (para popular o filtro de género com as opções possíveis);
- get_artists (para popular o filtro de artista com as opções possíveis);
- get authors (para popular o filtro de autor com as opções possíveis);
- get_labels (para popular o filtro de gravadora com as opções possíveis).

Detalhe da Crítica

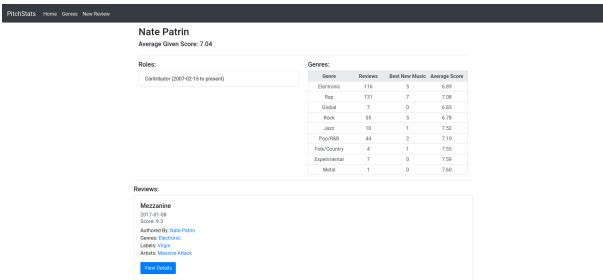
URL: {URL_BASE}/reviews/{identificador_da_crítica}/



Esta página, ligada à view "review_detail", tira partido da função get_reviews com um filtro para uma única crítica para pôr em foco os vários detalhes da crítica como o título, o score, url, data de publicação, os seus autores, géneros, gravadoras e artistas.

Detalhe do Autor

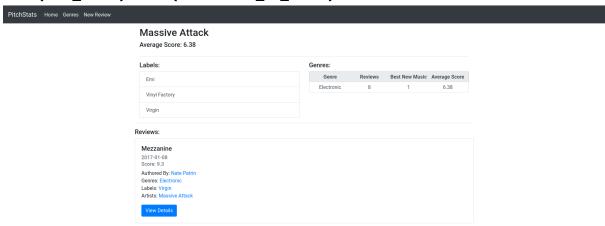
URL: {URL_BASE}/authors/{identificador_do_autor}/



Esta página, ligada à view "author_detail", tira partido da função get_author_detail para pôr em foco os vários detalhes de um autor como o seu nome, o seu score médio dado às críticas, uma timeline das suas funções na Pitchfork, uma lista dos seus géneros avaliados com as respectivas contagens e finalmente a lista das suas críticas.

Detalhe do Artista

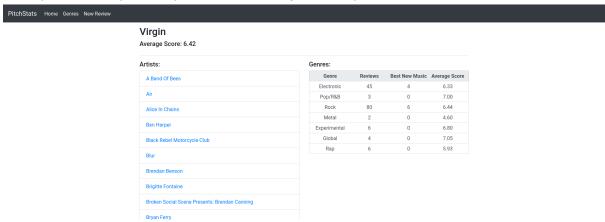
URL: {URL_BASE}/artists/{identificador_do_artista}/



Esta página, ligada à view "artist_detail", tira partido da função get_artist_detail para pôr em foco os vários detalhes de um artista como o seu nome, o seu score médio, uma lista das suas labels, uma lista dos seus géneros com as respectivas contagens e finalmente a lista das suas críticas.

Detalhe da Gravadora

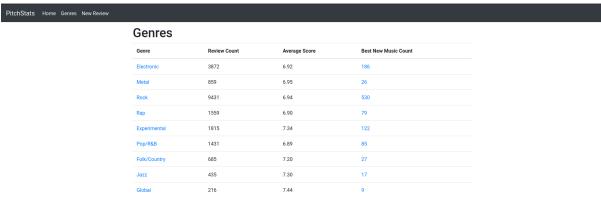
URL: {URL_BASE}/labels/{identificador_da_gravadora}/



Esta página, ligada à view "label_detail", tira partido da função get_label_detail para pôr em foco os vários detalhes de uma gravadora como o seu nome, o seu score médio, uma lista dos seus artistas, uma lista dos seus géneros com as respectivas contagens e finalmente a lista das suas críticas.

Géneros

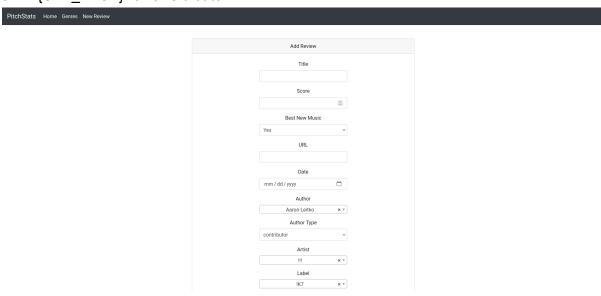
URL: {URL_BASE}/genres/



Esta página, ligada à view "genre_list", tira partido da função get_genres para listar todos os géneros musicais e as suas várias contagens. Aqui podemos ver números de críticas discriminadas por género, score médio de avaliação por género e número de álbuns considerados "best new music" por género.

Criar Crítica

URL: {URL_BASE}/reviews/create/



Esta página, ligada à view "create_review" que recorre às funções definidas no script queries.py para popular a página, como:

- get_genres (para popular o select de género com as opções possíveis);
- get artists (para popular o select de artista com as opções possíveis);
- get_authors (para popular o select de autor com as opções possíveis);
- get_labels (para popular o select de gravadora com as opções possíveis);
- get_author_types (para popular o select de author type).

O formulário presente na página cria um pedido HTTP POST para o endereço {URL_BASE}/reviews/. Este URL está ligado à view "reviews" que tira partido da função new_review para inserir uma nova crítica com os dados fornecidos no formulário.

Como Correr o Projeto

A raiz do projeto contém um ficheiro requirements.txt que lista todos os requisitos Python para correr o projeto (aconselha-se inclusive a criação de um virtual environment para correr o projeto).

Para gerar os ficheiros nt e n3 a partir do dataset sqlite:

\$ cd Dataset/

\$ python3 process_data.py

Com os ficheiros RDF criados, deve-se correr uma instância da GraphDB em "localhost:7200" e ter um repositório "pitchstats" criado e com um dos ficheiros "pitchstats.n3" ou "pitchstats.nt" importados no repositório.

Finalmente para correr o projeto Django a partir da raiz do projeto:

\$ cd Frontend/

\$ python3 manage.py runserver