

# Web Semântica 22/23 TP1 - PitchStats

Afonso Ribeiro Campos, 100055.

<a href="#">Introdução</a>	<a href="#">2</a>
<a href="#">Dados</a>	<a href="#">3</a>
<a href="#">Fonte e Descrição</a>	<a href="#">3</a>
<a href="#">Transformação em Triplos RDF</a>	<a href="#">5</a>
<a href="#">Reviews (Críticas)</a>	<a href="#">5</a>
<a href="#">Artist (Artista)</a>	<a href="#">6</a>
<a href="#">Label (Gravadora)</a>	<a href="#">6</a>
<a href="#">Author (Autor)</a>	<a href="#">6</a>
<a href="#">Operações Sobre os Dados</a>	<a href="#">7</a>
<a href="#">get_review_count</a>	<a href="#">7</a>
<a href="#">get_author_types</a>	<a href="#">7</a>
<a href="#">get_reviews</a>	<a href="#">8</a>
<a href="#">get_artists</a>	<a href="#">10</a>
<a href="#">get_authors</a>	<a href="#">11</a>
<a href="#">get_genres</a>	<a href="#">11</a>
<a href="#">get_labels</a>	<a href="#">12</a>
<a href="#">get_artist_detail</a>	<a href="#">13</a>
<a href="#">get_author_detail</a>	<a href="#">13</a>
<a href="#">get_label_detail</a>	<a href="#">14</a>
<a href="#">new_review</a>	<a href="#">14</a>
<a href="#">Aplicação Web</a>	<a href="#">16</a>
<a href="#">Home Page</a>	<a href="#">16</a>
<a href="#">Detalhe da Crítica</a>	<a href="#">17</a>
<a href="#">Detalhe do Autor</a>	<a href="#">17</a>
<a href="#">Detalhe do Artista</a>	<a href="#">18</a>
<a href="#">Detalhe da Gravadora</a>	<a href="#">18</a>
<a href="#">Gêneros</a>	<a href="#">19</a>
<a href="#">Criar Crítica</a>	<a href="#">19</a>
<a href="#">Como Correr o Projeto</a>	<a href="#">20</a>

# 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 [Kaggle](#).

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:
  - review\_id (identificador único de cada crítica);
  - title (título da crítica/álbum);
  - url (o url para o website da pitchfork onde a crítica pode ser lida);
  - 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));
  - 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);
  - 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));
  - 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));
  - genre (nome do género do álbum referente à crítica).

As seguintes tabelas/atributos foram ignorados:

- *Reviews*:
  - 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 álbum 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.

```
PREFIX pred: <http://pitchstats/pred/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT DISTINCT (COUNT(?review) as ?count) (MAX(xsd:float(?id)) as
?higher_id) WHERE {
    ?review pred:id ?id .
}
```

## 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.

```
PREFIX pred: <http://pitchstats/pred/>

SELECT DISTINCT ?type WHERE {
    ?author pred:author_type ?type .
}
```

## 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 gênero 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;



```

filters = []
if filter_genre: filters.append(f"FILTER regex(?genre,
\"{filter_genre}\", \"i\")")
if filter_best_new_music is not None: filters.append(
    f"FILTER (xsd:integer(?best_new_music) = {filter_best_new_music})")
if filter_date_start: filters.append(f"FILTER (?date >=
\"{filter_date_start}\")")
if filter_date_end: filters.append(f"FILTER (?date <=
\"{filter_date_end}\")")
if filter_score_low: filters.append(f"FILTER (xsd:float(?score) >=
{filter_score_low})")
if filter_score_high: filters.append(f"FILTER (xsd:float(?score) <=
{filter_score_high})")
if filter_title: filters.append(f"FILTER regex(?title,
\"{filter_title}\", \"i\")")

```

- 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.

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX pred: <http://pitchstats/pred/>

SELECT DISTINCT ?review ?title ?score ?url ?best_new_music ?date WHERE
{{
  {{
    {filter_inner_string}
    ?review pred:album ?title .
    ?review pred:date ?date .
    ?review pred:score ?score .
    ?review pred:url ?url .
    OPTIONAL {{?review pred:genre ?genre}} .
    ?review pred:best_new ?best_new_music .
    {filter_string}
  }}
}} {sort_string}
OFFSET {offset}
{limit_string}
```

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.

```
PREFIX pred: <http://pitchstats/pred/>

SELECT DISTINCT ?artist ?name WHERE {{
    {review_str}
    {label_str}
    ?review pred:artist ?artist .
    ?artist pred:name ?name .
}} ORDER BY ?name
```

## 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.

```
PREFIX pred: <http://pitchstats/pred/>

SELECT DISTINCT ?author ?name WHERE {{
    {review_str}
    ?review pred:author ?author .
    ?author pred:name ?name .
}} ORDER BY ?name
```

## 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 respectivo 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.

```

PREFIX pred: <http://pitchstats/pred/>

SELECT DISTINCT ?genre (COUNT(?review) as ?review_count)
(AVG(xsd:float(?score)) as ?avg_score) (SUM(IF(?best_new = "1", 1, 0))
as ?best_new_count) WHERE {{
    {author_str}
    {artist_str}
    {label_str}
    {"?review pred:genre ?genre ." if not filter_review else
f"<{filter_review}> pred:genre ?genre ."}
    ?review pred:score ?score .
    ?review pred:best_new ?best_new .
}} GROUP BY ?genre

```

## 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.

```
PREFIX pred: <http://pitchstats/pred/>

SELECT DISTINCT ?label ?name WHERE {{
    {review_str}
    {artist_str}
    ?review pred:label ?label .
    ?label pred:name ?name .
}} ORDER BY ?name
```

## 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).

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX pred: <http://pitchstats/pred/>

SELECT DISTINCT ?name (AVG(xsd:float(?score)) as ?avg_score) WHERE {{
    <{artist_uri}> pred:name ?name .
    ?review pred:artist <{artist_uri}> .
    ?review pred:score ?score .
}} GROUP BY ?name
```

## 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).

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX pred: <http://pitchstats/pred/>

SELECT DISTINCT ?name (AVG(xsd:float(?score)) as ?avg_score) WHERE {{
    <{author_uri}> pred:name ?name .
    ?review pred:author <{author_uri}> .
    ?review pred:score ?score .
}} GROUP BY ?name
```

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: <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).

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
PREFIX pred: <http://pitchstats/pred/>
```

```
SELECT DISTINCT ?name (AVG(xsd:float(?score)) as ?avg_score) WHERE {{  
  <{label_uri}> pred:name ?name .  
  ?review pred:label <{label_uri}> .  
  ?review pred:score ?score .  
}} GROUP BY ?name
```

## new\_review

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

- título 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`.

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX pred: <http://pitchstats/pred/>
PREFIX rev: <http://pitchstats/ent/review/>

INSERT DATA {{
    rev:{review_id} pred:id "{review_id}" ;
                pred:album "{title}" ;
                pred:score "{score}" ;
                pred:best_new "{best_new}" ;
                pred:url "{url}" ;
                pred:date "{date}" ;
                pred:author <{author_uri}> ;
                pred:author_type "{author_type}" ;
                pred:artist <{artist_uri}> ;
                pred:label <{label_uri}> ;
                pred:genre "{genre}" .
}}
```



# Aplicação Web



## Home Page


URL: {URL\_BASE}/


[PitchStats](#) [Home](#) [Genres](#) [New Review](#)


Search reviews:


Filter by date range:  
mm /  to mm / 

Filter by score range:  
Minimum  to Maximum 

Filter by genre:  
All 

Best new music:  
No Filter 

Sort by:  
By Date 

Sort order:  
Descending 

Filter by artist:

Filter by author:

Filter by label:

[Filter reviews](#)

Title	Date	Score	Best New Music	Author(s)	Artist(s)	Label(s)	Genre(s)
<a href="#">Mezzanine</a>	2017-01-08	9.3	No	<a href="#">Nate Patrin</a>	<a href="#">Massive Attack</a>	<a href="#">Virgin</a>	Electronic
<a href="#">Prelapsarian</a>	2017-01-07	7.9	No	<a href="#">Zoe Camp</a>	<a href="#">Krallice</a>	<a href="#">Hathenter</a>	Metal
<a href="#">All Of Them Natural</a>	2017-01-07	7.3	No	<a href="#">David Glickman</a>	<a href="#">Uranium Club</a>	<a href="#">Fashionable Idiots</a> , <a href="#">Static Shock</a>	Rock
<a href="#">First Songs</a>	2017-01-06	9.0	Yes	<a href="#">Jenn Pelly</a>	<a href="#">Kleenex</a> , <a href="#">Liliput</a>	<a href="#">Kill Rock Stars</a> , <a href="#">Mississippi</a>	Rock
<a href="#">New Start</a>	2017-01-06	8.1	No	<a href="#">Kevin Lozano</a>	<a href="#">Taso</a>	<a href="#">Teklife</a>	Electronic
<a href="#">Stillness In Wonderland</a>	2017-01-05	7.1	No	<a href="#">Katherine St. Asaph</a>	<a href="#">Little Simz</a>	<a href="#">Age 101</a>	Rap
<a href="#">Tahillim</a>	2017-01-05	7.0	No	<a href="#">Aniv Beta</a>	<a href="#">Yotam Avni</a>	<a href="#">Stroboscopic Artefacts</a>	Electronic

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}/

PitchStats Home Genres New Review

**Mezzanine**  
9.3  
<http://pitchfork.com/reviews/albums/22703-mezzanine/>

Date: 2017-01-08  
Authors: [Nate Patrin](#)  
Genres: [Electronic](#)  
Labels: [Virgin](#)  
Artists: [Massive Attack](#)

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}/

PitchStats Home Genres New Review

**Nate Patrin**  
Average Given Score: 7.04

**Roles:**  
Contributor (2007-02-15 to present)

**Genres:**

Genre	Reviews	Best New Music	Average Score
Electronic	116	5	6.89
Rap	131	7	7.08
Global	7	0	6.83
Rock	55	5	6.78
Jazz	10	1	7.52
Pop/R&B	44	2	7.10
Folk/Country	4	1	7.55
Experimental	7	0	7.59
Metal	1	0	7.60

**Reviews:**

**Mezzanine**  
2017-01-08  
Score: 9.3  
Authored By: [Nate Patrin](#)  
Genres: [Electronic](#)  
Labels: [Virgin](#)  
Artists: [Massive Attack](#)  
[View Details](#)

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}/

PitchStats Home Genres New Review

**Massive Attack**  
Average Score: 6.38

**Labels:**

Emi
Vinyl Factory
Virgin

**Genres:**

Genre	Reviews	Best New Music	Average Score
Electronic	8	1	6.38

**Reviews:**

**Mezzanine**  
2017-01-08  
Score: 9.3  
Authored By: [Nate Patrin](#)  
Genres: [Electronic](#)  
Labels: [Virgin](#)  
Artists: [Massive Attack](#)  
[View Details](#)

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}/

PitchStats Home Genres New Review

**Virgin**  
Average Score: 6.42

**Artists:**

<a href="#">A Band Of Bees</a>
<a href="#">Air</a>
<a href="#">Alice In Chains</a>
<a href="#">Ben Harper</a>
<a href="#">Black Rebel Motorcycle Club</a>
<a href="#">Blur</a>
<a href="#">Brendan Benson</a>
<a href="#">Brigitte Fontaine</a>
<a href="#">Broken Social Scene Presents: Brendan Canning</a>
<a href="#">Bryan Ferry</a>

**Genres:**

Genre	Reviews	Best New Music	Average Score
Electronic	45	4	6.33
Pop/R&B	3	0	7.00
Rock	80	6	6.44
Metal	2	0	4.60
Experimental	6	0	6.80
Global	4	0	7.05
Rap	6	0	5.93

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.

## Gêneros

URL: {URL\_BASE}/genres/

PitchStats Home Genres New Review			
Genres			
Genre	Review Count	Average Score	Best New Music Count
<a href="#">Electronic</a>	3872	6.92	<a href="#">186</a>
<a href="#">Metal</a>	859	6.95	<a href="#">26</a>
<a href="#">Rock</a>	9431	6.94	<a href="#">530</a>
<a href="#">Rap</a>	1559	6.90	<a href="#">79</a>
<a href="#">Experimental</a>	1815	7.34	<a href="#">122</a>
<a href="#">Pop/R&amp;B</a>	1431	6.89	<a href="#">85</a>
<a href="#">Folk/Country</a>	685	7.20	<a href="#">27</a>
<a href="#">Jazz</a>	435	7.30	<a href="#">17</a>
<a href="#">Global</a>	216	7.44	<a href="#">9</a>

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/

PitchStats Home Genres New Review	
Add Review	
Title	<input type="text"/>
Score	<input type="text"/>
Best New Music	<input type="text"/>
URL	<input type="text"/>
Date	<input type="text"/>
Author	<input type="text"/>
Author Type	<input type="text"/>
Artist	<input type="text"/>
Label	<input type="text"/>

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
```