# **Projeto de Bases de Dados (CC2005) - parte 2**

## 1. Elementos do grupo

**Grupo nº** **28**

|  |  |
| --- | --- |
| **Nº mecanográfico** | **Nome** |
| 202208926 | Mariana Serralheiro |
| 202207213 | Ana Amorim |

## 2. Ajustes ao modelo da BD

Descrição

O universo que consideramos engloba o mundo das séries presentes na fonte de dados do IMDB (https://datasets.imdbws.com), do TMDB (<https://www.themoviedb.org>) e do Kaggle ([https://www.kaggle.com](https://www.kaggle.com/datasets/harshitshankhdhar/imdb-dataset-of-top-1000-movies-and-tv-shows)). Filtramos a informação aqui recolhida para criar uma base de dados mais concisa que representa todas as séries que saíram no ano de 2021 na plataforma Netflix (<https://www.netflix.com>).

Ajustes nas classes e as suas associações:

* Criação da classe Género:
  + Descrição: género de cada série

*Atributos:* tipo

*Associações:* 1:N (para a classe Série)

* Classe Série:

*Associações:* 1:N (para a classe Personagem), N:1 (para a classe Género), N:M (para a classe Escritor)

* Classe Personagem:

*Associações:* N:1 (para a classe Série e Ator)

* Classe Escritor: manteu-se igual
* Classe Ator:

*Associações:* 1:N (para a classe Personagem)

UML

A screenshot of a computer

Description automatically generated

Modelo relacional

Criamos uma nova tabela Escritores\_Serie para que a ligação N:M entre as classes Escritor e Serie ficasse corrigida e modificamos as chaves externas que estavam em falta.

A diagram of a computer

Description automatically generated

3. Povoamento de tabelas

Fontes de dados usadas para o povoamento da BD:

<https://www.themoviedb.org>

<https://datasets.imdbws.com>

<https://kaggle.com>

<https://www.netflix.com>

Primeiramente, recorremos á plataforma Kaggle (<https://kaggle.com>) e selecionamos todas as series desejadas.

Seguidamente, recorremos aos restantes sites para obter a informação que precisávamos.

Depois de passarmos esta informação para várias tabelas no Excel, exportamos cada uma para um ficheiro CSV para facilitar a importação para a consola SQLite.

Finalmente, na consola SQLite editámos a estrutura de cada tabela para que ela cumprisse todos os requerimentos que lhe impusemos no modelo relacional.

|  |  |
| --- | --- |
| **Nome da tabela** | **Nº de entradas** |
| Series | 1832 |
| Generos | 34 |
| Personagens | 5096 |
| Atores | 3238 |
| Escritores | 617 |
| Escritores\_Serie | 562 |

4. Aplicação Python

|  |  |
| --- | --- |
| **“Endpoint”** | **“Funcionalidade”** |
| / | Página de entrada, que dá acesso às outras funcionalidades. |
| /series/ | Ligação a uma tabela com todos os atributos da classe Serie. |
| /series/<int:id>/ | Ligação a uma página com a informação da serie com o id pesquisado pelo user. |
| /series/search/<expr>/ | Ligação a uma página com todos os resultados da pesquisa de uma expressão pelo user associada ao nome de qualquer serie. |
| /atores/ | Ligação a uma tabela com todos os atributos da classe Ator. |
| /atores/<int:id>/ | Ligação a uma página com a informação do ator com o id pesquisado pelo user. |
| /atores/search/<expr>/ | Ligação a uma página com a informação de um ator por qualquer expressão pesquisada pelo user. |
| /generos/ | Ligação a uma tabela com todos os atributos da classe Genero. |
| /generos/<int:id>/ | Ligação a uma página com a informação do género com o id pesquisado pelo user. |
| /personagens/ | Ligação a uma tabela com todos os atributos da classe Personagem. |
| /personagens/<int:id>/ | Ligação a uma página com a informação da personagem com o id pesquisado pelo user. |
| /personagens/search/<expr>/ | Ligação a uma página com a informação de uma personagem por qualquer expressão pesquisada pelo user. |
| /escritores/ | Ligação a uma tabela com todos os atributos da classe Escritor. |
| /escritores/<int:id>/ | Ligação a uma página com a informação do escritor com o id pesquisado pelo user. |
| /escritores/search/<expr>/ | Ligação a uma página com a informação de um escritor por qualquer expressão pesquisada pelo user. |

Agregação agrupada de informação:

Endpoint: /personagens/<int:id>/

Código:

A black background with green text

Description automatically generated

Explicação:

Este endpoint leva a uma página que apresenta o id e nome de uma personagem pesquisada pelo user tal como o nome do ator que o interpreta e da serie onde aparece.

Utilizamos este group by de modo a não haver repetição do nome dos atores na junção das três tabelas.

Junção de informação envolvendo 3 ou mais tabelas

Endpoint: /series/<int:id>/

Código:

A screenshot of a computer program

Description automatically generated

Explicação:

Este endpoint leva a uma página que apresenta todos os atributos da classe Serie relativos a uma serie concreta pesquisada através do id pelo user. Contém também ligações para todas as outras classes. Estas ligações dirigem-nos para páginas com a informação dos atores, personagens, escritores e tipo de género presentes nesta série.

Pesquisa de informação

Endpoint: /series/search/<expr>/

Código:

A black background with green text

Description automatically generated

Explicação:

Este endpoint leva a uma página com uma lista de séries que contenham uma qualquer expressão dada pelo user.

O nosso código apresenta vários exemplos de consultas mencionadas no ponto três. Porém como estes são muito semelhantes aos apresentados acima, não achámos necessário explicar todos.