IPCA INSTITUTO POLITÉCNICO DO CÁVADO E DO AVE

Trabalho Prático

Ana Paula Canuto da Silva, 24178

Integração de Sistemas de Informação

Prof. Luís Ferreira

Licenciatura em Engenharia em Sistemas Informáticos

(regime pós-laboral)

Escola Superior de Tecnologia

Instituto Politécnico do Cávado e do Ave

Data do documento: 19 de outubro de 2025

Índice

Introdução

A integração de sistemas de informação é essencial para transformar dados dispersos em conhecimento útil, permitindo que organizações tomem decisões baseadas em informação fiável e atualizada.

Neste contexto, o presente trabalho, intitulado "**Projeto ETL** com **Pentaho e Power BI – TMDB API**", tem como objetivo desenvolver um processo completo de **Extração, Transformação e Carga (ETL)** aplicado a dados provenientes de uma **API** pública de filmes (**The Movie Database – TMDB**).

A solução proposta utiliza o **Pentaho Data Integration (PDI)** para automatizar o pipeline de dados, aplicando expressões regulares, limpeza e validação, e carregando os resultados numa base **SQLite.** Por fim, os dados tratados foram visualizados no **Power BI,** permitindo uma análise clara das métricas de popularidade dos filmes.

O projeto demonstra, assim, a aplicação prática dos conceitos de integração de dados e a utilização de ferramentas profissionais para construção de pipelines **ETL** completos.

Parte I – Projeto ETL com Pentaho e Power BI – TMDB API

1- Ferramentas e Tecnologias Utilizadas

Nesse projeto fiz uso dos softwares e linguagem abaixo:

- Pentaho Data Integration (PDI – Kettle)
- SQLite
- Microsoft Power BI
- API REST (The Movie Database - TMDB)
- JSON, CSV e Excel
- Expressões Regulares (Regex)
- Scripts JavaScript no PDI

2- Enquadramento do problema

Este projeto foi desenvolvido no âmbito da Unidade Curricular de **Integração de Sistemas de Informação (ISI)**, com o objetivo de aplicar e demonstrar na prática os conceitos de **ETL (Extract, Transform and Load)** através de uma solução real de integração de dados.

O problema proposto consistiu na criação de um **pipeline de dados automatizado** capaz de:

- Extrair informações de filmes a partir de uma **API pública (TMDB)** em formato JSON;
- Realizar **limpeza, normalização e validação** dos dados, tratando valores nulos, formatos incorretos e inconsistências;
- Armazenar os resultados limpos e validados numa **base de dados SQLite**;
- Exportar os dados transformados para ficheiros **CSV e Excel**;
- Criar **dashboards interativos no Power BI**, permitindo uma análise visual das métricas de popularidade dos filmes.

O projeto visa simular um **processo real de integração de dados empresariais**, tal como ocorre em cenários de *business intelligence* e *data warehousing*, demonstrando a interoperabilidade entre sistemas distintos.

3- Estratégia Utilizada

A solução foi implementada com recurso ao Pentaho Data Integration (PDI), que orquestrou todo o fluxo ETL de forma modular.

O processo foi dividido em três fases principais: Extração, Transformação e Carga (Load).

3.1 – Extração (Extract)

A extração dos dados foi realizada a partir da API REST do The Movie Database (TMDB), utilizando o componente REST Client do Pentaho.

Cada chamada à API devolve um conjunto de filmes em formato JSON, contendo campos como title, release_date, original_language, overview, popularity, entre outros.

Os resultados foram guardados em múltiplos ficheiros JSON segmentados (movies_original.json_1, movies_original.json_2, ..., movies_original.json_19), permitindo melhor controlo sobre o volume e integridade dos dados.

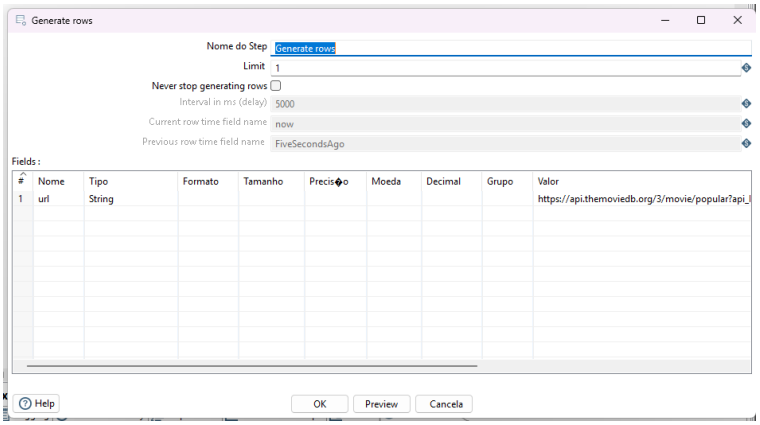


Figura 1 – Configuração do passo Generate Rows

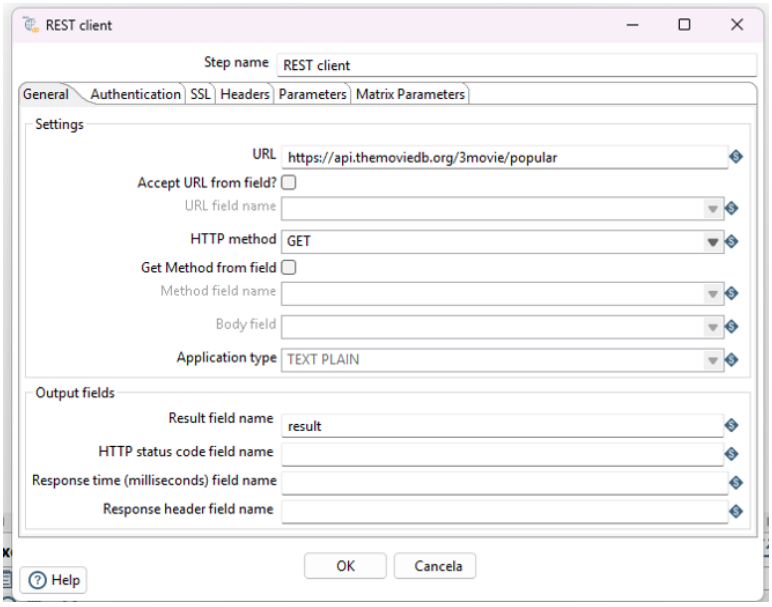
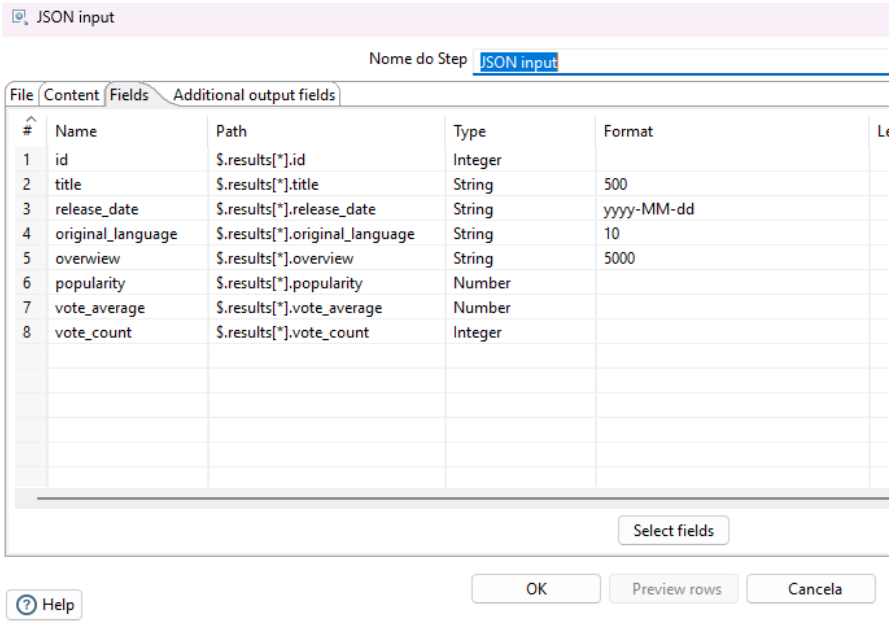


Figura 2 – Configuração REST Client no Pentaho, responsável por iniciar o processo e obter dados da API.

Após a extração, os ficheiros JSON foram processados pelo passo **JSON Input**, que converteu a estrutura hierárquica dos dados em colunas tabulares, adequadas para transformação posterior.



#	Name	Path	Type	Format
1	id	$.results[*].id	Integer	
2	title	$.results[*].title	String	500
3	release_date	$.results[*].release_date	String	yyyy-MM-dd
4	original_language	$.results[*].original_language	String	10
5	overwiew	$.results[*].overview	String	5000
6	popularity	$.results[*].popularity	Number	
7	vote_average	$.results[*].vote_average	Number	
8	vote_count	$.results[*].vote_count	Integer	

Figura 3 – Conversão de dados JSON em tabela no passo JSON Input.

3.2 - Transformação (Transform)

A fase de transformação foi implementada através do passo **Modified JavaScript Value**, responsável por aplicar lógica de limpeza, manipulação e validação dos dados.

Entre as operações realizadas destacam-se:

- **Normalização de datas** (release_date), garantindo formato padrão YYYY-MM-DD;
- **Substituição de valores nulos ou vazios** por indicadores padrão;
- **Validação de texto** através de **expressões regulares (Regex)** para eliminar caracteres inválidos;
- **Categorização da popularidade** em faixas (“Baixa”, “Média”, “Alta”);
- Criação de campos derivados (por exemplo, title_safe, release_date_safe).

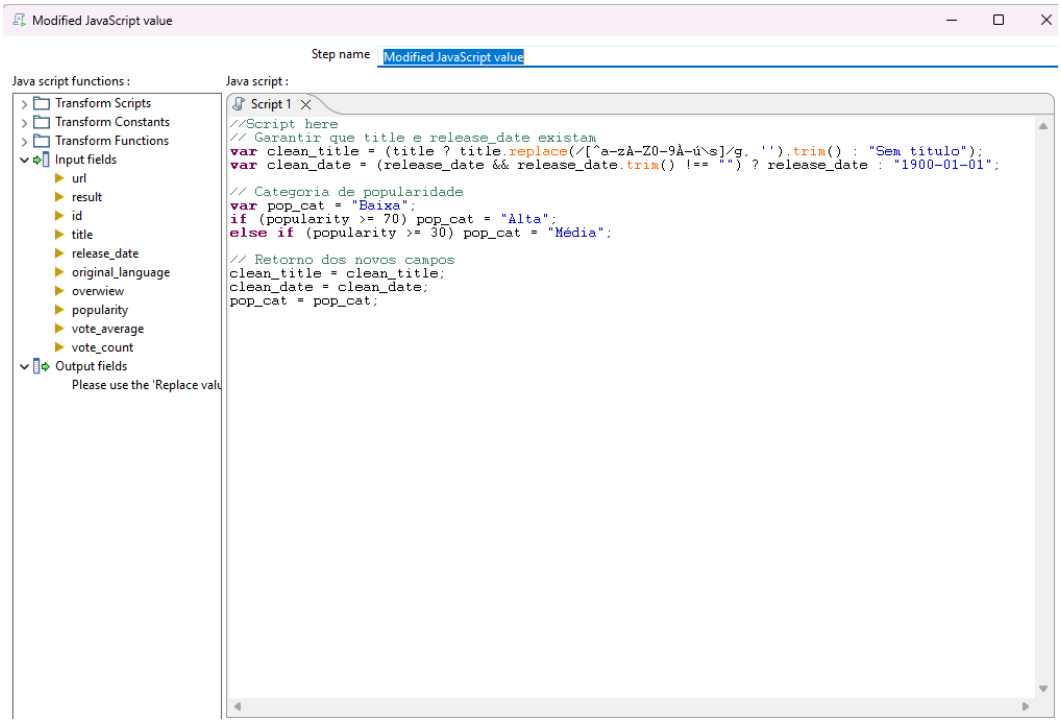


Figura 4 – Script JavaScript aplicado às transformações, incluindo expressões regulares e limpeza de dados.

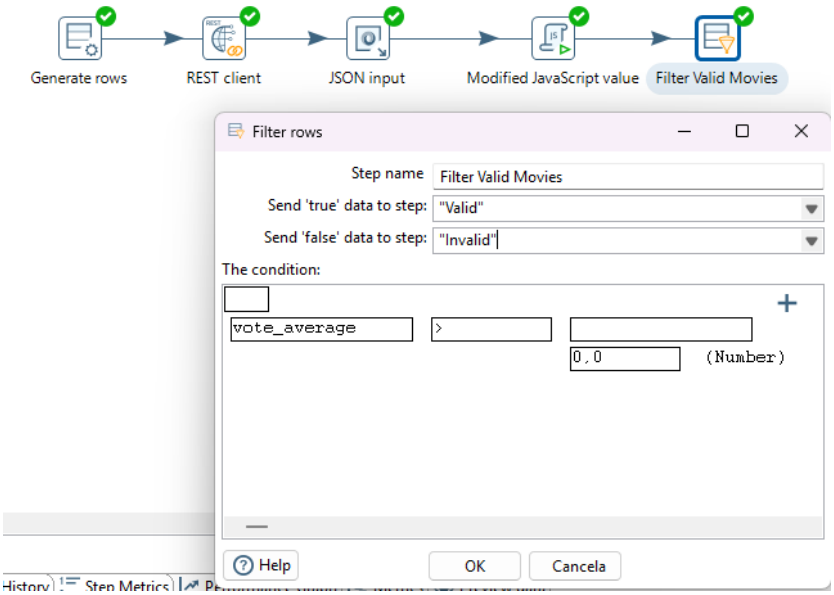


Figura 5 – Definindo parâmetros de validação para filmes "válidos".

3.3 - Carga (Load)

Os dados limpos e validados foram gravados numa **base de dados SQLite**, através do passo **Table Output**, que criou e populou a tabela movies_clean.

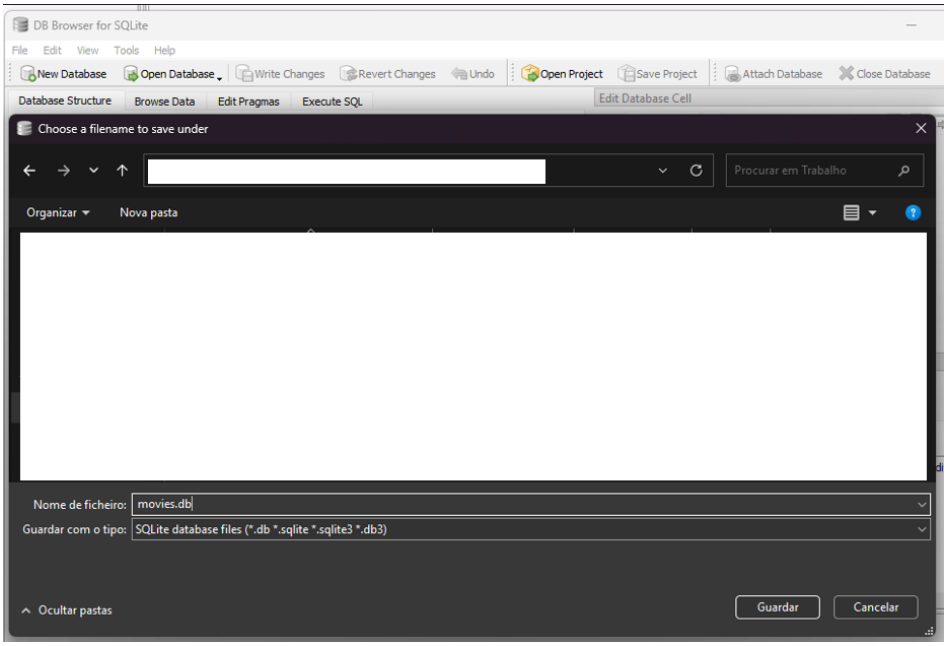


Figura 6 – Configuração da ligação à base de dados SQLite .

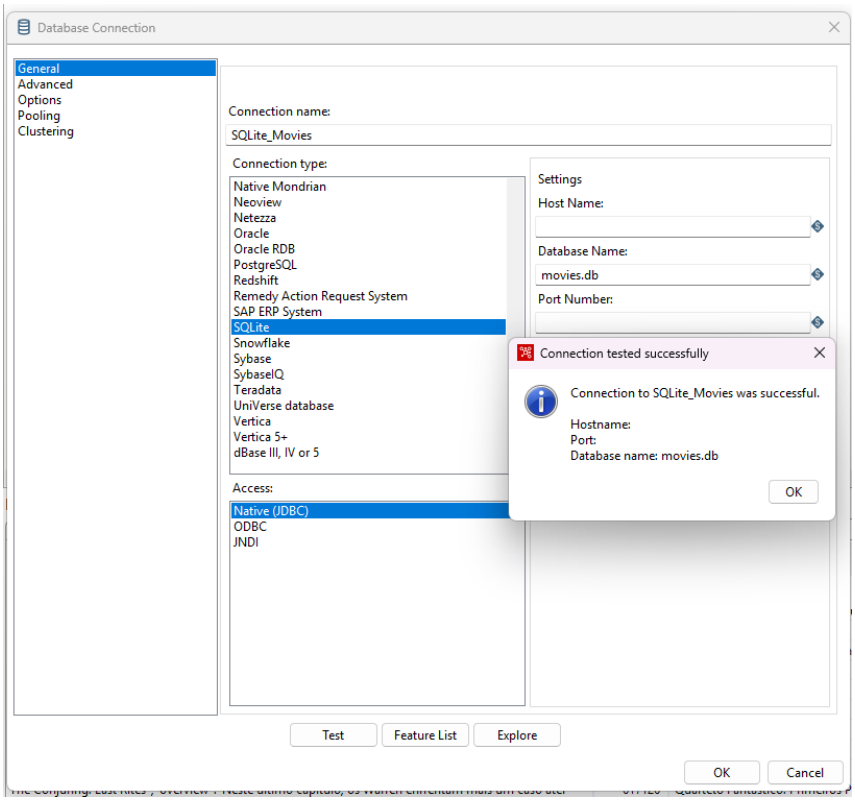


Figura 7 – Configurando conexão com a base de dados criada..

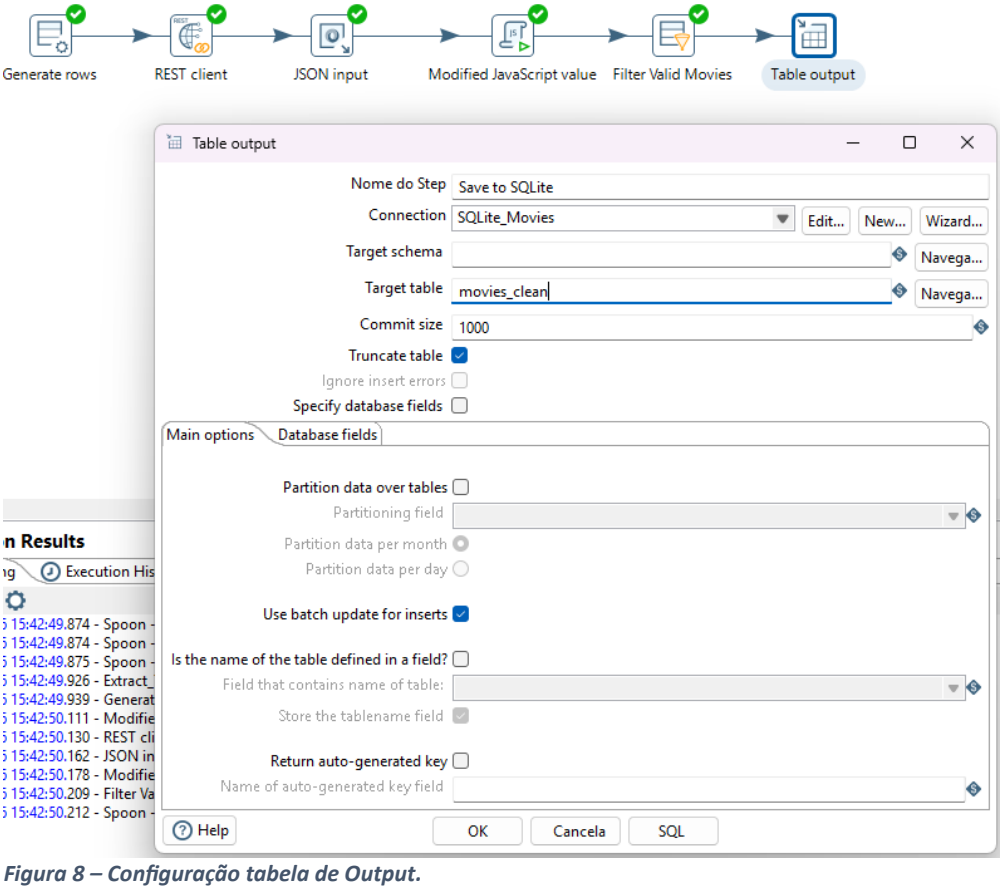


Figura 8 – Configuração tabela de Output.

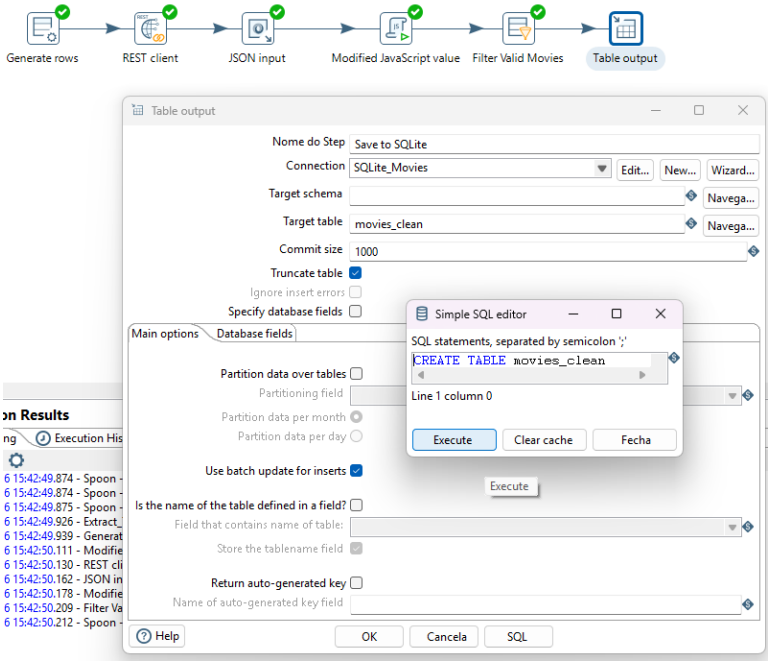


Figura 9 – Criação da tabela.

```sql
CREATE TABLE IF NOT EXISTS movies_clean (
    id INTEGER PRIMARY KEY,
    title TEXT,
    release_date TEXT,
    vote_average REAL,
    popularity REAL,
    overview TEXT
);
```

Figura 10 – SQL para criação da tabela no SQLite.

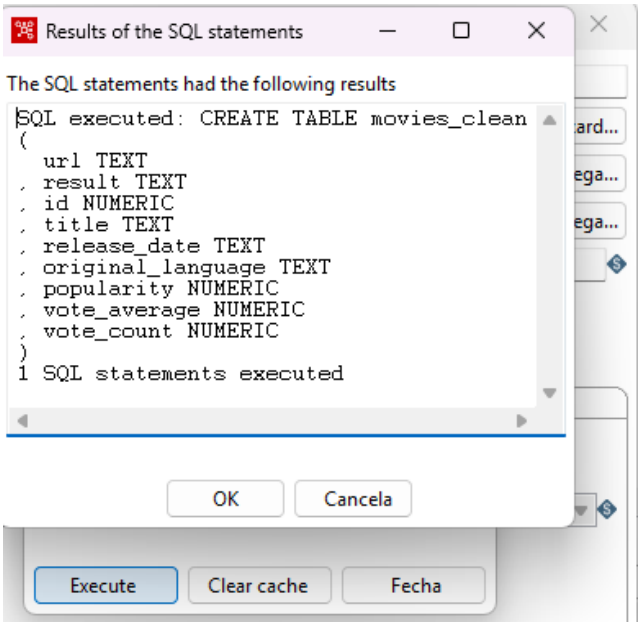


Figura 11 – Configuração SQL para executar.

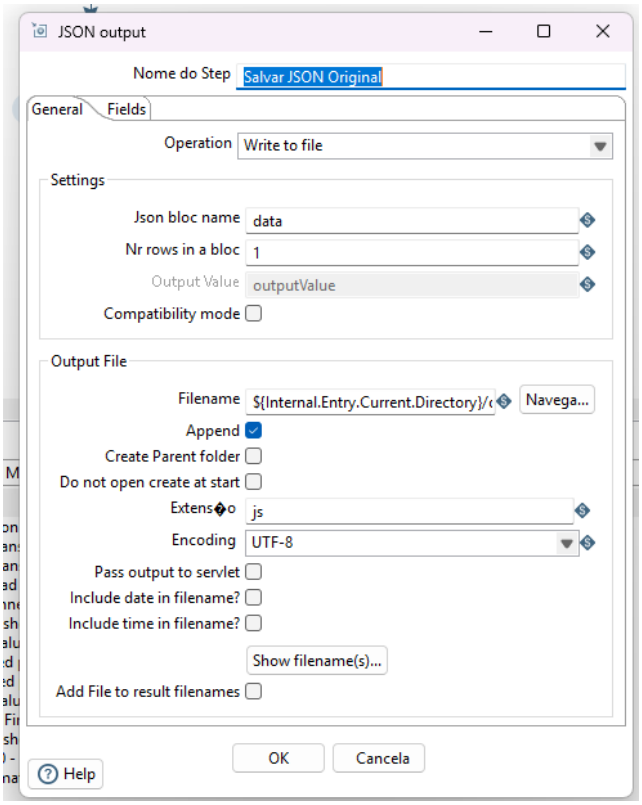


Figura 12 – Salvando dados originais em JSON.

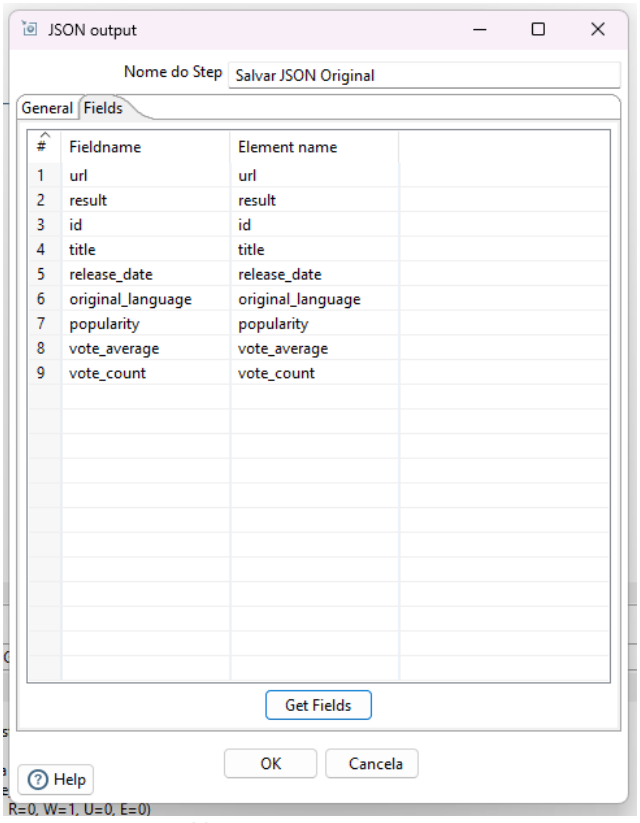


Figura 13 – Get Fields.

A mesma informação foi exportada para ficheiros externos, com o objetivo de suportar análises posteriores no Power BI:

- Filmes_Validados.xlsx
- movies_transformation.csv
- movies_excluidos.csv

O processo também incluiu **armazenamento de logs e tratamento de exceções**, garantindo rastreabilidade e consistência.

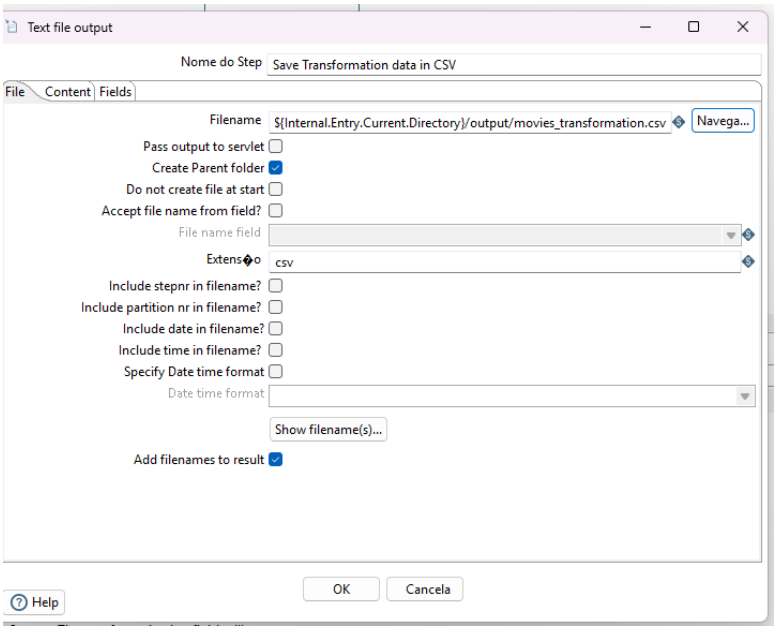


Figura 14 – File: Passos de exportação em formato Excel e CSV.

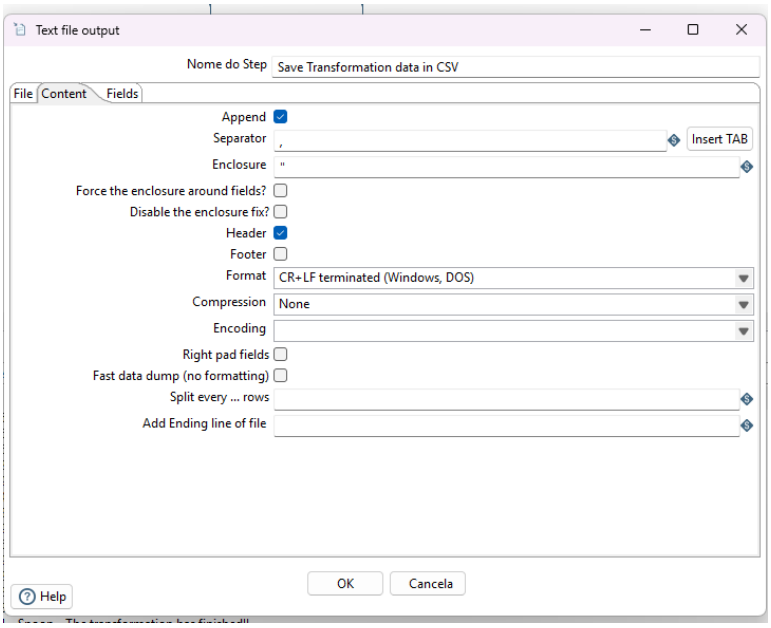


Figura 15 – Content: Passos de exportação em formato Excel e CSV.

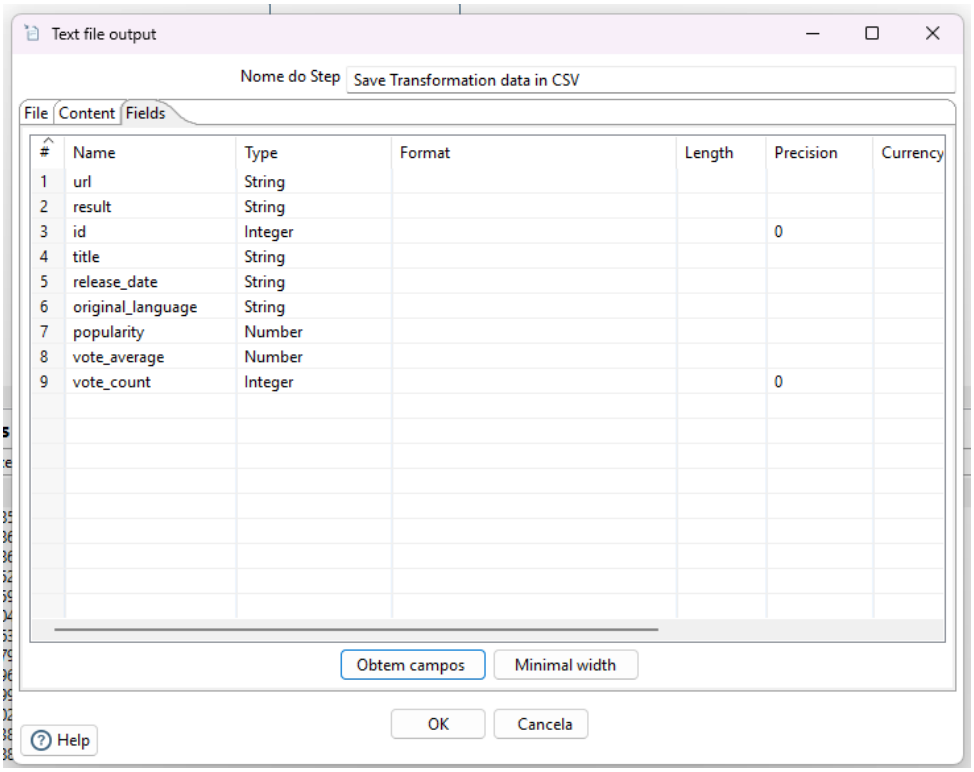


Figura 16 – Fields: Passos de exportação em formato Excel e CSV.

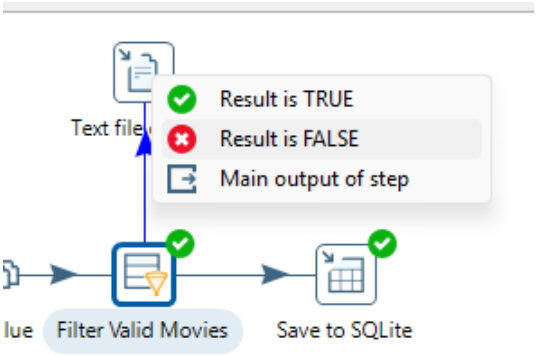


Figura 17 – Escolha de qual tipo de dados quer enviar para salvar, TRUE (validados) ou FALSE(excluídos).

4- Estrutura da Transformação ETL

O diagrama global do fluxo desenvolvido no Pentaho está representado na **Figura18**, que ilustra a sequência lógica das operações.

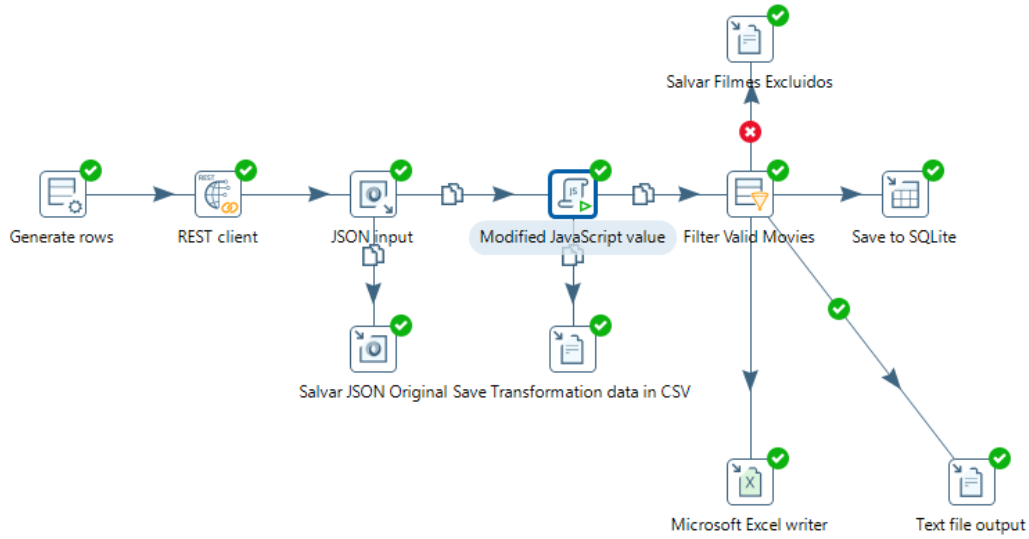


Figura 18 – Estrutura geral da transformação ETL no Pentaho (Generate Rows → REST Client → JSON Input → JavaScript → Filter → Outputs).

4.1 - Descrição do Fluxo Principal:

1. **Generate Rows:** inicializa parâmetros de execução.
2. **REST Client:** consulta a API TMDB.
3. **JSON Input:** interpreta o JSON e converte-o em registos tabulares.
4. **Modified JavaScript Value:** aplica transformações e expressões regulares.
5. **Filter Valid Movies:** separa registos válidos e inválidos.
6. **Table Output (SQLite):** grava dados no repositório local.
7. **Excel e CSV Output:** exporta resultados processados.
8. **JSON Save Original:** guarda a versão bruta dos dados extraídos.

5- Job de Controlo e Execução

O processo foi orquestrado num **Job principal do Pentaho**, garantindo o controlo de execução e modularidade.

Funcionalidades incluídas:

- Execução sequencial de transformações (Extract, Transform, Load);
- Geração automática de **logs de execução**;
- Tratamento de exceções e reconexão automática em falhas de rede;

- Criação de datasets distintos para **dados válidos e inválidos**;
- Exportação simultânea para múltiplos formatos (Excel, CSV e SQLite).

6- Resultados Obtidos

Após a execução do processo ETL, foram gerados os seguintes resultados:

- **Base de dados SQLite:** movies.db, contendo a tabela movies_clean.
- **Ficheiros exportados:**
    - Filmes_Validados.xlsx
    - movies_excluidos.csv
    - movies_transformation.csv

Os dados foram posteriormente importados para o **Power BI**, onde foram criadas visualizações interativas.

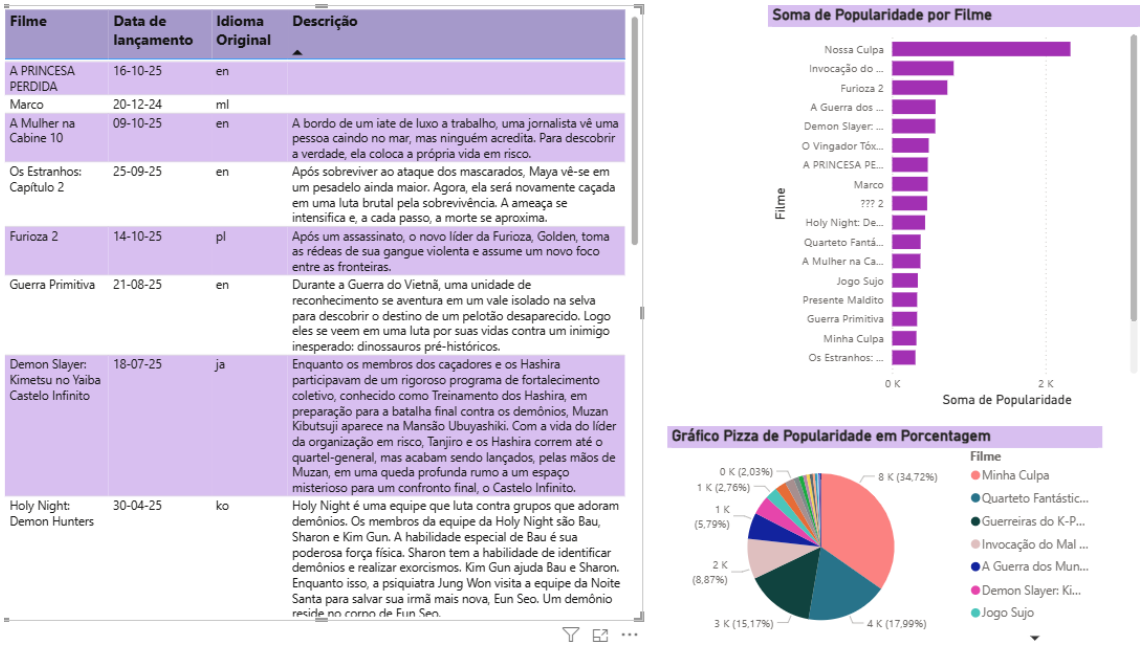


Figura 19 – Dashboard do Power BI com gráficos de barras e pizza baseados nos dados transformados.

7- Análise dos Resultados

Os gráficos produzidos no Power BI permitiram uma leitura clara sobre a **popularidade dos filmes** obtidos via API.

- O **Gráfico de Barras** "Soma de Popularidade por Filme" mostrou que *Nossa Culpa* lidera em popularidade, seguida por *Invocação do Mal* e *Furiosa 2*.
- O **Gráfico de Pizza** destacou a distribuição percentual da popularidade total, evidenciando a predominância de poucos títulos em relação ao total de filmes.
- A **tabela interativa** permitiu explorar detalhes de cada filme, como data de lançamento, idioma original e descrição.

Estas análises demonstram a eficácia do pipeline desenvolvido, que converteu dados brutos da API em **informação estruturada e visualmente útil** para apoio à decisão.

8- Conclusão e Trabalhos Futuros

Este trabalho permitiu aplicar de forma prática os conceitos de **Integração de Sistemas de Informação** e **processos ETL**, utilizando o **Pentaho Data Integration** como ferramenta central de orquestração.

8.1 - A solução construída demonstrou:

- A capacidade de integração de múltiplos formatos (JSON, CSV, Excel, SQL);
- O uso eficaz de expressões regulares para **validação e normalização** de dados;
- A implementação de uma arquitetura modular com **logs e controlo de erros**;
- A interligação entre **camadas de dados, transformação e visualização**.

8.2 - Problemas encontrados:

Apesar do eficiente tratamento nos dados, verificou-se que nem todos os filmes da API foram analisados, pois os dados importados, após o primeiro tratamento, foram subdivididos em vários **JSONs** , mas como o projeto salva os dados a cada passo, é possível em uma fase futura, unir todos eles, tendo uma maior quantidade de filmes e dados a serem analisados.

8.3 - Trabalhos futuros propostos:

- Integração de novas APIs (ex.: séries, atores, géneros);
- Agendamento automático de execução com **Pentaho Scheduler**;

- Envio automático de relatórios via e-mail;
- Integração com bases de dados em nuvem (BigQuery, Azure SQL, etc.);
- Aplicação de **técnicas de data mining ou machine learning** para previsão de tendências de popularidade.

9- Referências Bibliográficas

- Pentaho Data Integration – *Official Documentation*, Hitachi Vantara.
- SQLite – *Official Documentation*. Disponível em: https://www.sqlite.org/
- Microsoft Power BI – *Official Documentation*. Disponível em: https://learn.microsoft.com/power-bi/
- JSON.org – *The JSON Data Interchange Standard*. https://www.json.org/
- TMDB API Documentation – *The Movie Database Developers*. https://developer.themoviedb.org/

Stack Overflow – *Discussões sobre ETL, Pentaho e SQLite*

10- Apêndice