




UMA JANELA PARA O MUNDO: EXPLORANDO DIVERSIDADE E SUCESSO FINANCEIRO

Filmes/Animações Multiculturais
Por Marina Rodrigues Bueno

- 
- Gênero/Categoria escolhido: Animação/Filmes;
 - Tema: Uma Janela para o Mundo (Explorando Diversidade e Sucesso Financeiro);
 - Quantidade de filmes: 20;
 - Objetivo: Os filmes escolhidos abordam diversidade cultural, e fazem parte da minha vida agregando bastante conhecimento, posso afirmar que essa análise tem um valor sentimental para mim.

PONTOS PRINCIPAIS DA INGESTÃO DE DADOS DA TMDB

- Analisar o “movies”, pois a partir desses dados é feita a criação da análise final, e a compreensão de quais dados faltam.
- Criação do script que solicita os dados complementares para a análise na API TMDB.

Explicação do código a seguir:

1. Código executado no Lambda (AWS).
2. Filtragem necessária para evitar dados desnecessários, sobrecarga na API e aumento de custo na limpeza desses dados.
3. Dados com valores relevantes para a análise, não permitindo valores nulo ou muito baixos.
4. Salvamento dos dados em memória, no formato json e particionamento de acordo com as instruções da sprint.
5. Envio do arquivo em memória para o bucket data-lake-marina.

ANALISANDO O JSON:

1. Colunas organizadas;
2. Possui o id;
3. Valores relevantes para a pesquisa;
4. Manter os arquivos o mais limpo possível facilitou o processo de refinamento.

```
ers > User > Downloads > json_movies_info_O_R_P (5).json > ...  
[{"id": "tt0061852", "tituloPrincipal": "The Jungle Book", "orcamento": 4000000, "receita": 205843612, "popularidade": 51.007}, {"id": "tt0103639", "tituloPrincipal": "Aladdin", "orcamento": 28000000, "receita": 504050219, "popularidade": 60.885}, {"id": "tt0114148", "tituloPrincipal": "Pocahontas", "orcamento": 55000000, "receita": 346079773, "popularidade": 40.297}, {"id": "tt0116583", "tituloPrincipal": "The Hunchback of Notre Dame", "orcamento": 100000000, "receita": 325338851, "popularidade": 32.895}, {"id": "tt0118617", "tituloPrincipal": "Anastasia", "orcamento": 53000000, "receita": 139804348, "popularidade": 33.856}, {"id": "tt0120762", "tituloPrincipal": "Mulan", "orcamento": 90000000, "receita": 304320254, "popularidade": 76.997}, {"id": "tt0120917", "tituloPrincipal": "The Emperor's New Groove", "orcamento": 100000000, "receita": 169327687, "popularidade": 89.488}, {"id": "tt0138749", "tituloPrincipal": "The Road to El Dorado", "orcamento": 95000000, "receita": 76432727, "popularidade": 27.846}, {"id": "tt0230011", "tituloPrincipal": "Atlantis: The Lost Empire", "orcamento": 120000000, "receita": 186053725, "popularidade": 21.815}, {"id": "tt0245429", "tituloPrincipal": "Spirited Away", "orcamento": 19000000, "receita": 274925095, "popularidade": 68.389}, {"id": "tt0275847", "tituloPrincipal": "Lilo & Stitch", "orcamento": 80000000, "receita": 273144151, "popularidade": 49.142}, {"id": "tt0780521", "tituloPrincipal": "The Princess and the Frog", "orcamento": 105000000, "receita": 270997378, "popularidade": 72.343}, {"id": "tt1217209", "tituloPrincipal": "Brave", "orcamento": 185000000, "receita": 538983207, "popularidade": 59.238}, {"id": "tt2294629", "tituloPrincipal": "Frozen", "orcamento": 150000000, "receita": 1274219009, "popularidade": 67.971}, {"id": "tt2380307", "tituloPrincipal": "Coco", "orcamento": 175000000, "receita": 800526015, "popularidade": 78.064}, {"id": "tt2953050", "tituloPrincipal": "Encanto", "orcamento": 50000000, "receita": 253000000, "popularidade": 154.557}, {"id": "tt3521164", "tituloPrincipal": "Moana", "orcamento": 150000000, "receita": 690860472, "popularidade": 16.683}, {"id": "tt4633694", "tituloPrincipal": "Spider-Man: Into the Spider-Verse", "orcamento": 90000000, "receita": 375464627, "popularidade": 893.758}, {"id": "tt5109280", "tituloPrincipal": "Raya and the Last Dragon", "orcamento": 100000000, "receita": 130423032, "popularidade": 71.336}, {"id": "tt8097030", "tituloPrincipal": "Turning Red", "orcamento": 190000000, "receita": 18879922, "popularidade": 149.779}]
```

TRUSTED

Pontos principais da etapa:

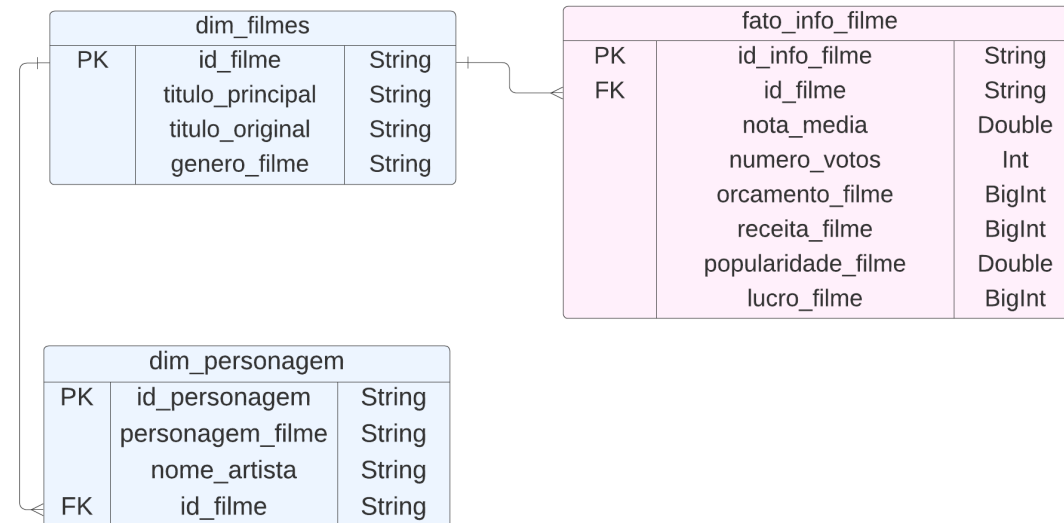
- Transformar os arquivos da Raw Zone em parquet, o “formato” mais confiante para a manipulação dos dados.
 1. Script padrão spark, rodado no job do glue.
 2. Leitura do arquivo json, armazenando-o em um DF.
 3. Transforma o arquivo em parquet enviando-o novamente ao bucket.

MODELAGEM DE DADOS

Modelo Snowflake:

Utilizado para permitir a conexão entre as dimensões dim_personagem e dim_filmes (exclusivamente), e dim_filmes com fato_info_filme, permitindo se necessário consultas relacionadas aos artistas.

Modelo Dimensional Análise Final





REFINED

- Fazer a limpeza dos dados, de acordo com o modelo dimensional proposto removendo toda informação irrelevante para a pesquisa.
- Transformar os dataframes limpos em tabelas no banco de dados pronto “projeto final marina”, feito no glue catalog.
- Script rodado no glue.

ANALISANDO O CÓDIGO DA TABELA "INFO_FATO_FILME"

```
Editar  Seleção  Ver  Acessar  ...  refined_fato_info_filme.py - compassuol_projeto - Visual Studio Code

refined_fato_info_filme.py 9+ X

9 > Refined > refined_fato_info_filme.py > ...

import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job
from pyspark.sql.functions import col, row_number, concat, lit
from pyspark.sql.window import Window

## @params: [JOB_NAME]
args = getResolvedOptions(sys.argv, ['JOB_NAME'])

sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['JOB_NAME'], args)

# Abre o parquet do movies da pasta Trusted
df_filmes_fixo = spark.read.option("header", "true").option("inferSchema", "true").parquet('s3://data-lake-marina/Trusted/Movies/part-00000-f58a830c-2e58-4f2f-84d1-4cba1d4643bc-c000.snappy.parquet')

# Abre o parquet dos dados da TMDb
df_analise = spark.read.option("header", "true").option("inferSchema", "true").parquet('s3://data-lake-marina/Trusted/TMDb/2023/06/12/part-00000-7c6d7e64-1782-413d-880b-92c22317f52a-c000.snappy.parquet')
```


ANALISANDO O CÓDIGO "INFO_FATO_FILME"

```
Editar  Seleção  Ver  Acessar  ...  refined_fato_info_filme.py - compassuol_projeto - Visual Studio Code

refined_fato_info_filme.py 9+ X

9 > Refined > refined_fato_info_filme.py > ...
Trusted/TMDB/2023/06/12/part-00000-7c6d7e64-1782-413d-880b-92c22317f52a-c000.snappy.parquet')

# Filtra a coluna ID dos dados TMDB
df_analise_ids = df_analise.select('id')

# Faz o Join com o movies procurando os filmes pelo ID
df = df_filmes_fixo.join(df_analise_ids, df_filmes_fixo['id'] == df_analise_ids['id'], 'inner').dropDuplicates(['id'])

# Seleciona as colunas específicas de acordo com o join
df_fato_analise_final = df.select(df_filmes_fixo['id'], 'notaMedia', 'numeroVotos')

# Junta o DF novo com algumas colunas do DF que contém dados da API
df_fato_analise_final = df_fato_analise_final.join(df_analise.select('id', 'orcamento', 'receita', 'popularidade'), 'id', 'inner')

# Renomeia as colunas que serão visualizadas no ATHENA
df_fato_analise_final = df_fato_analise_final.withColumnRenamed('id', 'id filme').withColumnRenamed('notaMedia', 'nota_media').withColumnRenamed('numeroVotos', 'numero_votos').withColumnRenamed('orcamento', 'orcamento_filme').withColumnRenamed('receita', 'receita_filme').withColumnRenamed('popularidade', 'popularidade_filme')

# Adiciona uma coluna com o lucro dos filmes
df_fato_analise_final = df_fato_analise_final.withColumn('lucro_filme', col('receita_filme') - col('orcamento_filme'))
```

ANALISANDO "INFO_FATO_FILME"

```
window = Window.orderBy("id_filme")
df_fato_analise_final = df_fato_analise_final.withColumn("row_number", row_number().over(window))

# Cria uma nova coluna de "id_info_filme" concatenando a string "IF0" com o/
# valor da coluna "row_number" e exclui a mesma.
df_fato_analise_final = df_fato_analise_final.withColumn("id_info_filme", concat(lit("IF0"), col("row_number"))).drop("row_number")

# Seleciona as colunas na ordem necessária
df_fato_fixa = df_fato_analise_final.select("id_info_filme", "id_filme", "nota_media", "numero_votos",
"orcamento_filme", "receita_filme", "popularidade_filme", "lucro_filme")

# Usa o banco de dados criado
spark.sql("use projetofinalmarina")

# particiona o df em 1, salva o arquivo parquet e cria a tabela no BD
df_fato_fixa.coalesce(1).write.saveAsTable(name="fato_info_filme", mode="overwrite", path='s3://data-lake-marina/
Refined/FatoFilmesAnalise/', format="parquet")

job.commit()
```



DIFICULDADES

- Tempo: As sprints duram apenas 10 dias, aprender uma linguagem ou ferramenta nova e realizar as atividades propostas dentro do prazo foi desafiador.
- Aprender slq, python e o spark em um curto período de tempo dificultou a construção dos códigos por conta das sintaxes.
- Eu possuo facilidade em entender e criar a lógica, ou seja, sei como fazer mas como não domino ainda a sintaxe, demoro um pouco na construção dos códigos.

TIPOS DE ANÁLISE E SEUS OBJETIVOS:

- **Comparação entre Receita e Orçamento:** O objetivo é determinar se os filmes foram lucrativos, levando em consideração a diferença entre a receita obtida e o orçamento estabelecido.
- **Relação entre Nota Média e Lucro:** Há filmes com nota média abaixo de 7.5 que ainda obtiveram lucro.? Isso fornecerá insights sobre a possível desconexão entre a avaliação crítica e o desempenho financeiro, especialmente quando se trata de filmes com aspectos culturais.
- **Elaboração de Ranking de Popularidade:** Esse ranking será responsável para analisar quais filmes foram mais populares e quais foram menos populares.

COMO TODO CONHECIMENTO OBTIDO IRÁ CONTRIBUIR PARA AGREGAR VALOR AOS CLIENTES DA COMPASS?

Engenharia de Dados: compreende a extração, transformação em formato confiável e limpeza dos dados, garantindo sua integridade. Além disso, envolve o armazenamento seguro dos dados e a construção de um banco de dados, facilitando a extração de valor pelos analistas.

Análise de Dados: uma vez que os dados tenham sido extraídos, limpos, organizados e verificados, é o momento em que os analistas podem extrair insights relevantes. Essas informações transmitidas por meio de um Dashboard (por exemplo) são essenciais para que o cliente e sua equipe possam tomar decisões que ajudem nos negócios confiando em todo o processo do pipeline de dados. Elas precisam ser organizadas para a visualização dos dados e pela organização do painel (dashboard), de forma a facilitar a compreensão e interpretação dos resultados.

Design: desempenha um papel fundamental na acessibilidade e na criação de uma harmonia visual para a apresentação dos dados.



FIM