DESAFIO DE ANÁLISE DE DADOS: FILMES E SÉRIES

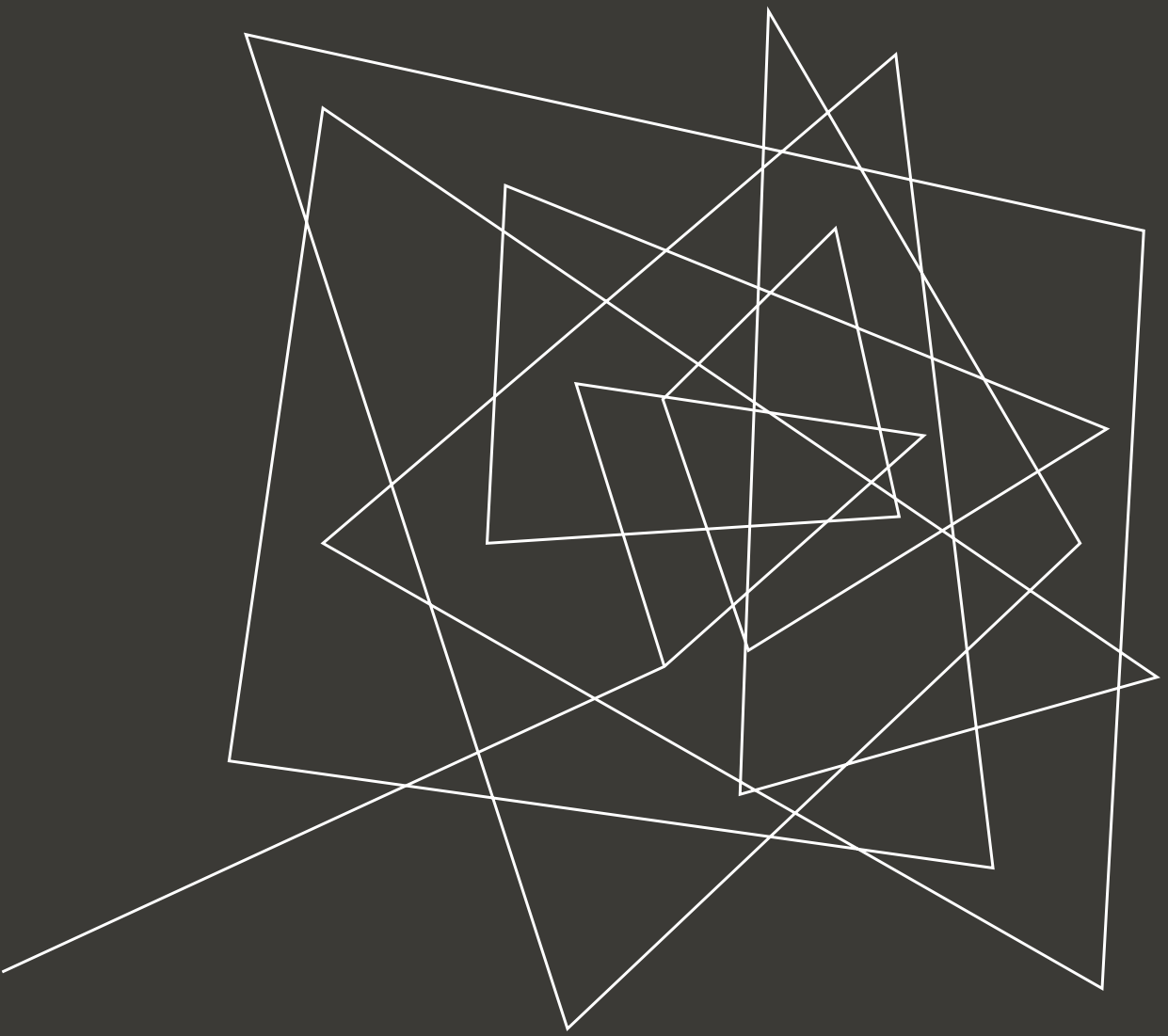
Jade Marinho Torres

ÍNDICE

INTRODUÇÃO

Olá a todos! Meu nome é Jade e estarei com vocês para apresentar minha análise.

É um prazer estar aqui hoje para compartilhar os resultados do desafio de análise de dados sobre filmes. Durante esse processo, explorei diversos aspectos dos gêneros Ação e Aventura e suas características, a fim de entregar um produto refinado e relevante para o Programa de Bolsas da Compasso.



PRINCIPAIS OBJETIVOS

Explorar a correlação entre a duração dos filmes e suas notas médias pelas décadas dos filmes mais populares.

GÊNERO DE FILMES E REFINAMENTO

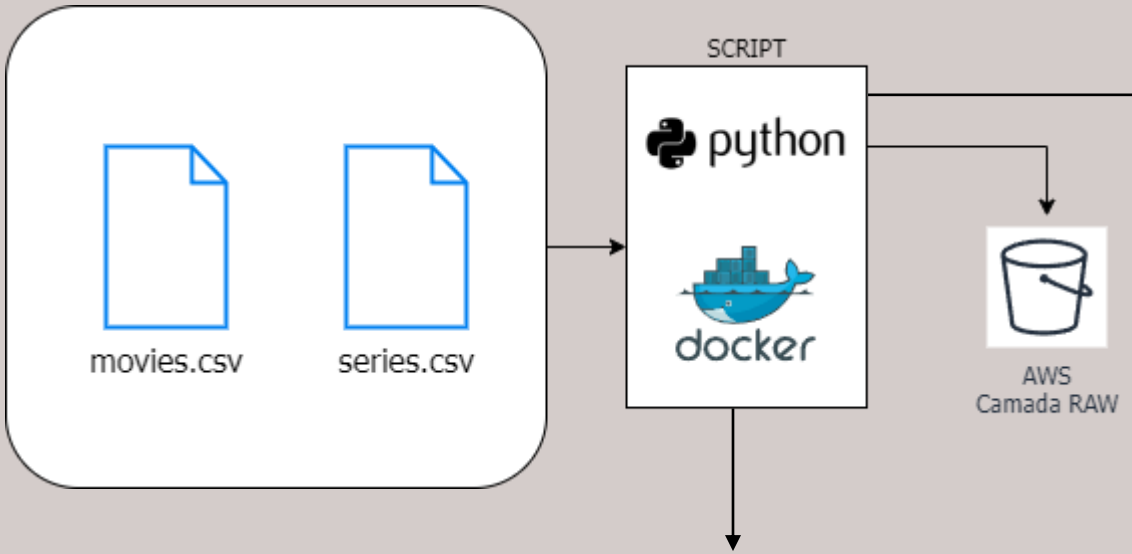
Em primeiro lugar, nossa Squad 5 – Five It – foi atribuído nossa análise nos gêneros de Ação e Aventura. Esses gêneros são amplamente apreciados pelo público.

Além disso, decidi por um refinamento no meu estudo, considerando apenas os filmes de Ação e Aventura que possuíam mais de 15 mil votos. Esse critério me permitiu trabalhar com filmes verdadeiramente populares.

PRIMEIRA ETAPA DO DESAFIO

Para alcançar nossos resultados, passamos por diferentes etapas. Primeiro, coletamos todos os dados sobre filmes e series do arquivo CSV. Em seguida, os aloquei dentro do Bucket AWS via Docker com Boto3.

DIAGRAMA E CÓDIGOS DA PRIMEIRA ETAPA



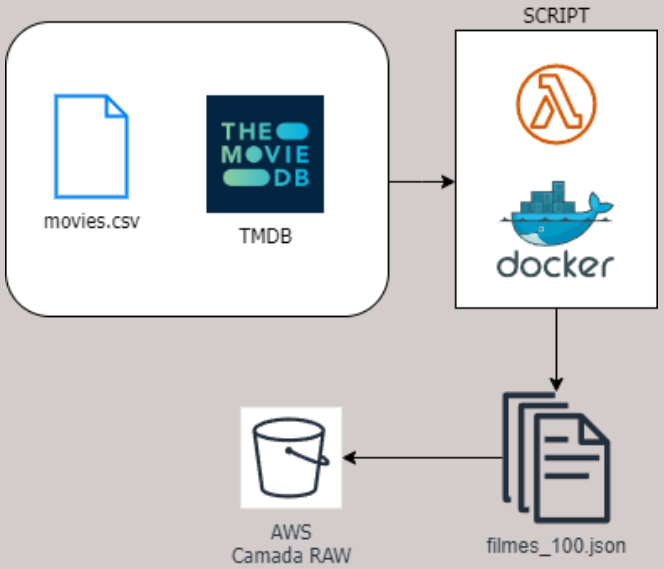

```
> Sprint7 > Dockerfile > ...
FROM python:3.6-alpine

WORKDIR /desafio

COPY ex3sprint7.py .

RUN pip3 install boto3

CMD ["python", "ex3sprint7.py"]
```

```python
import logging
import boto3
from botocore.exceptions import ClientError
import os

AccesskeyID = "XXXX"
Secretaccesskey = "XXXX"
region = "us-east-1"

def upload_file(file_name, bucket, folder_name=None):

    if folder_name is not None:
        object_name = os.path.join(folder_name, os.path.basename(file_name))
    else:
        object_name = os.path.basename(file_name)

    # Upload the file
    s3_client = boto3.client('s3', aws_access_key_id=AccesskeyID,aws_secret_access_key
    try:
        response = s3_client.upload_file(file_name, bucket, object_name)
    except ClientError as e:
        logging.error(e)
        return False
    return True

upload_file("movies.csv", "ex3sprint7bucket", "RAW/movies/2023/05/16/movies.csv")
upload_file("series.csv", "ex3sprint7bucket", "RAW/series/2023/05/16/series.csv")
```

SEGUNDA ETAPA DO DESAFIO

Após obtermos nosso CSV na camada RAW, avançamos para a importação dos dados JSON do TMDB. Focando especificamente em filmes de Ação e Aventura, procedi ao processamento desses dados a partir do CSV, aplicando um filtro para selecionar somente os filmes populares, com mais de 15 mil votos e que não tenha valores nulos no tempo em minutos. Em seguida, colocamos os dados JSON no Bucket da AWS.

DIAGRAMA E CÓDIGOS DA SEGUNDA ETAPA




```python
# Carregar o arquivo CSV 'movies.csv' em um DataFrame
df_movies = pd.read_csv('movies.csv', delimiter='|')

# Filtrar apenas os filmes de Ação/Aventura
df_filtered = df_movies[(df_movies['tempoMinutos'] != 'N') &
                        (df_movies['numeroVotos'] >= 15000) &
                        (df_movies['genero'].str.contains('action', case=False) &
                         df_movies['genero'].str.contains('adventure', case=False))]

df_filtered = df_filtered[['id', 'tituloPincipal', 'tempoMinutos', 'anoLancamento', 'genero', 'notaMedia', 'numeroVotos']]
df_filtered = df_filtered.groupby('tituloPincipal').first()
df_filtered = df_filtered.sort_values(by=['notaMedia', 'tempoMinutos'], ascending=[False, True])
```

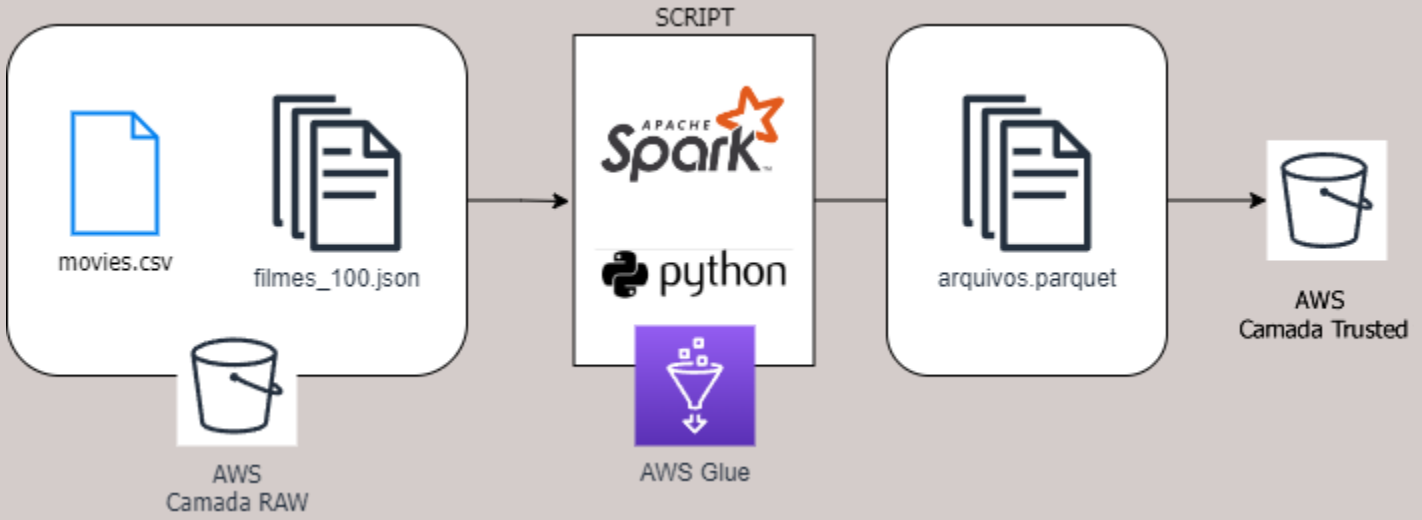
O código completo pode ser encontrado nesse link no meu github:
https://github.com/jademarinho/Sprint1/blob/main/Desafio/Sprint8/nb%20com%20informacoes%20de%20api%20final.ipynb

```python
# Iterar sobre os IDs dos filmes filtrados
for movie_id in df_filtered['id']:
    # Construir a URL da API para obter as informações do país de produção do filme
    url = f'{base_url}{movie_id}?api_key={api_key}&language=pt-BR'

    # Fazer solicitação para a URL e obter os dados do filme
    response = requests.get(url)

    if response.status_code == 200:
        filme_data = response.json()

        # Extrair o nome do país de produção do filme
        countries = filme_data.get('production_countries', [])
        production_country = countries[0]['name'] if countries else 'N/A'

        # Adicionar a informação do país de produção ao dataframe
        df_filtered.loc[df_filtered['id'] == movie_id, 'production_country'] = production_country

        # Adicionar os dados do filme à lista de resultados
        filmes.append(filme_data)

    # Atualizar a barra de progresso
    pbar.update(1)

    # Incrementar o contador de filmes
    contador += 1

    # Verificar se chegou a 100 filmes ou se atingiu o total de 754 filmes
    if contador % 100 == 0 or contador == len(df_filtered):
        # Salvar o JSON em um arquivo
        json_data = json.dumps(filmes, indent=4)

        # Gerar o caminho do arquivo no S3
        folder = "Raw/TMDB/JSON/2023/06/02/"
        nome_arquivo = f"{folder}filmes_{contador}.json"

        # Enviar o JSON para o S3
        s3_client.put_object(Body=json_data, Bucket=bucket, Key=nome_arquivo)

        # Limpar a lista de filmes
        filmes = []

# Fechar a barra de progresso
pbar.close()
```

TERCEIRA ETAPA DO DESAFIO – PARTE 1

Persistir dados da RAW na camada Trusted do data lake no formato PARQUET através de 2 Jobs em PySpark no AWS Glue, um para o CSV e outro para o JSON.

DIAGRAMA E CÓDIGOS DA TERCEIRA ETAPA



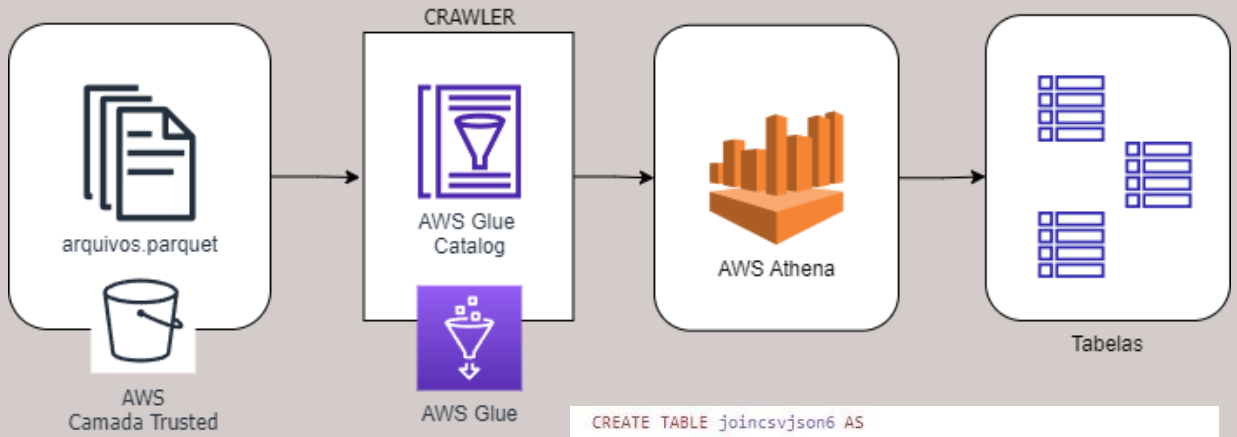

```
data_frame = spark.read.option("delimiter", "|").csv("s3://ex3sprint7bucket/upload/movies/2023/05/16/movies.csv/")

data_frame.write.parquet("s3://ex3sprint7bucket/Trusted/CSV/2023/06/12/")
```

```
df_movies = spark.read.option("multiline", "true").json("s3://ex3sprint7bucket/Raw/TMDB/JSON/2023/06/02/*.json")

df_movies.write.parquet("s3://ex3sprint7bucket/Trusted/JSON/2023/06/12")
```

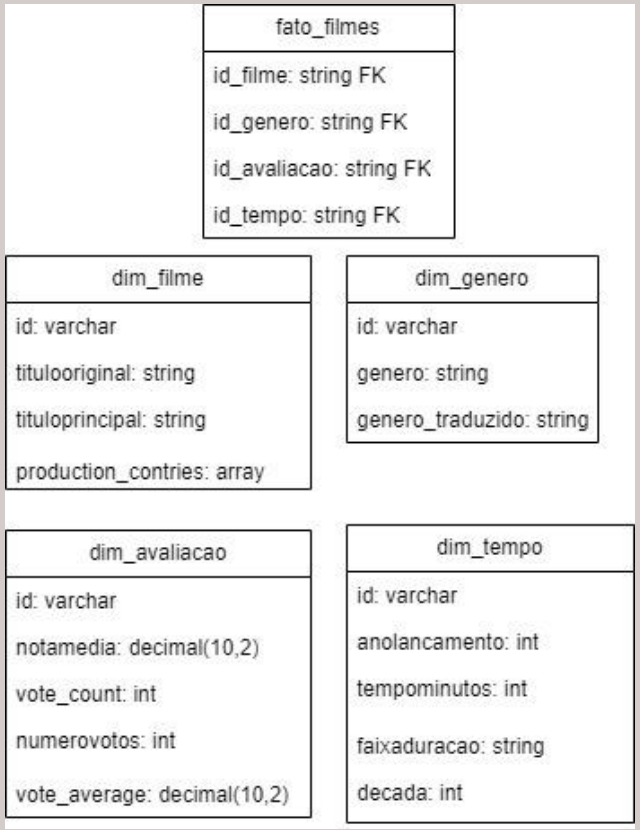
TERCEIRA ETAPA DO DESAFIO – PARTE 2

Nesta etapa crucial do desafio, iremos extrair os arquivos Parquet da camada Trusted utilizando o Glue Data Catalog. Em seguida, criaremos tabelas e aplicaremos manipulações no AWS Athena, visando a modelagem dos dados no formato dimensional. No meu caso específico, realizarei um JOIN entre o CSV e o JSON, selecionando os dados relevantes para a criação do modelo dimensional.

DIAGRAMA E CÓDIGOS DA TERCEIRA ETAPA




```
CREATE TABLE joincsvjson6 AS
SELECT DISTINCT
    CAST(csv.id AS VARCHAR) AS id,
    CAST(csv.tituloPrincipal AS VARCHAR) AS tituloPrincipal,
    CAST(csv.tituloOriginal AS VARCHAR) AS tituloOriginal,
    CAST(csv.tempoMinutos AS INT) AS tempoMinutos,
    CAST(csv.anoLancamento AS INT) AS anoLancamento,
    CAST(csv.genero AS VARCHAR) AS genero,
    CAST(csv.notaMedia AS DECIMAL(10, 2)) AS notaMedia,
    CAST(csv.numeroVotos AS INT) AS numeroVotos,
    json.imdb_id,
    json.production_countries,
    CAST(json.vote_average AS DECIMAL(10, 2)) AS vote_average,
    CAST(json.vote_count AS INT) AS vote_count
FROM csvrefined2 AS csv
JOIN "AwsDataCatalog"."jsonrefined"."12" AS json
ON csv.id = json.imdb_id;
```



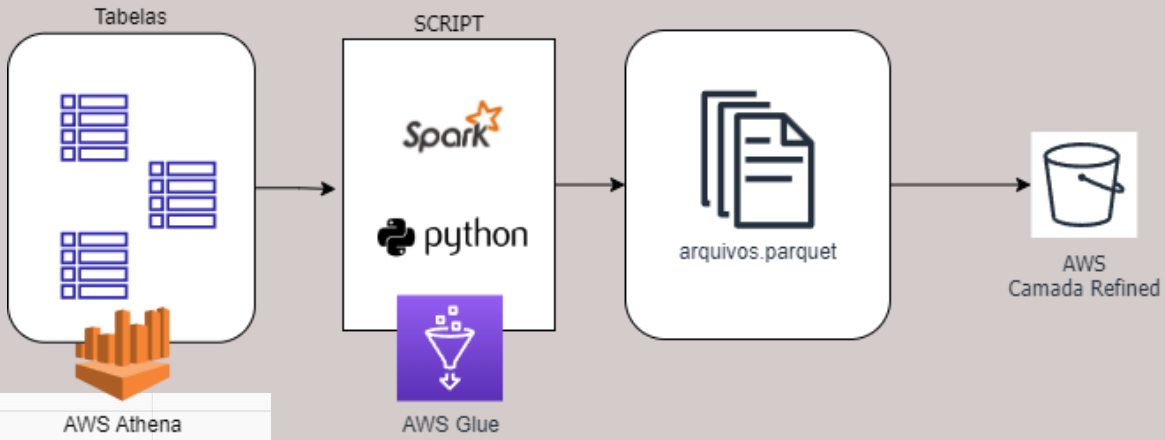
Para saber mais sobre: https://github.com/jademarinho/Sprint1/blob/main/Sprint9/readme.md

TERCEIRA ETAPA DO DESAFIO – PARTE 3

Persistir dados da AWS Athena na camada Refined do data lake no formato PARQUET através de Jobs em PySpark no AWS Glue.

Observação importante: Durante o criação do meu dash eu tive que adicionar mais campos para atender minhas necessidades, tal como faixaduracao e genero_traducao e tirar o production_countries.

DIAGRAMA E CÓDIGOS DA TERCEIRA ETAPA




```sql
CREATE OR REPLACE VIEW "traducao_genero" AS
SELECT
    id,
    CASE
        WHEN genero = 'Action,Adventure,Western' THEN 'Ação, Aventura, Western'
        WHEN genero = 'Action,Adventure,Thriller' THEN 'Ação, Aventura, Suspense'
        WHEN genero = 'Action,Adventure,History' THEN 'Ação, Aventura, História'
        WHEN genero = 'Action,Adventure,Horror' THEN 'Ação, Aventura, Terror'
        WHEN genero = 'Action,Adventure,Romance' THEN 'Ação, Aventura, Romance'
        WHEN genero = 'Action,Adventure,Fantasy' THEN 'Ação, Aventura, Fantasia'
        WHEN genero = 'Action,Adventure,Crime' THEN 'Ação, Aventura, Crime'
        WHEN genero = 'Action,Adventure,War' THEN 'Ação, Aventura, Guerra'
        WHEN genero = 'Action,Adventure,Comedy' THEN 'Ação, Aventura, Comédia'
        WHEN genero = 'Action,Adventure,Mystery' THEN 'Ação, Aventura, Mistério'
        WHEN genero = 'Action,Adventure,Sci-Fi' THEN 'Ação, Aventura, Ficção Científica'
        WHEN genero = 'Action,Adventure,Animation' THEN 'Ação, Aventura, Animação'
        WHEN genero = 'Action,Adventure,Drama' THEN 'Ação, Aventura, Drama'
        WHEN genero = 'Action,Adventure,Family' THEN 'Ação, Aventura, Família'
        WHEN genero = 'Action,Adventure,Biography' THEN 'Ação, Aventura, Biografia'
        WHEN genero = 'Action,Adventure' THEN 'Ação, Aventura'
        ELSE genero
    END AS generos
FROM "csvrefined"."joincsvjson6"
```

```sql
CREATE OR REPLACE VIEW "filmes_por_duracao_em_horas" AS
SELECT
  id
, faixa_duracao
, "count"(*) quantidade_filmes
FROM
  (
   SELECT
     id
   , (CASE WHEN (tempominutos < 90) THEN 'Menos de 1h e 30min' WHEN (tempominutos BETWEEN 90 AND 119) THEN 'Entre 1h e 30min e 2h' WHEN (tempominutos BETWEEN 120 AND
       149) THEN 'Entre 2h e 2h e 30min' WHEN (tempominutos BETWEEN 150 AND 179) THEN 'Entre 2h e 30min e 3h' WHEN (tempominutos BETWEEN 180 AND 209) THEN 'Entre 3h e
       3h e 30min' ELSE 'Mais de 3h e 30min' END) faixa_duracao
   FROM
     "csvrefined"."joincsvjson6"
)  subquery
GROUP BY id, faixa_duracao
```

Para saber mais sobre: https://github.com/jademarinho/Sprint1/blob/main/Sprint9/readme.md

QUARTA ETAPA DO DESAFIO

A partir do modelo de dados criado na etapa anterior, criei um dashboard interativo para visualização dos resultados. Nesse dashboard, apresentamos gráficos e métricas que nos permitiram obter insights. Por exemplo, as tendencias de duração de filme no passar das décadas e padrões interessantes que nos ajudaram a compreender melhor a evolução desses gêneros cinematográficos.

A DURAÇÃO DE UM FILME DEVE ESTAR DIRETAMENTE RELACIONADA À RESISTÊNCIA DA BEXIGA HUMANA.

Alfred Hitchcock

VISUALIZAÇÃO DA DASHBOARD

A inspiração para a paleta de cores da dashboard foi extraída do filme que conquistou o Primeiro lugar em termos de Maior Nota e também foi testado o contraste para daltônicos.

Protan(red), Deutan(green) e Tritan(blue), respectivamente.

Senhor dos Anéis: O Retorno do Rei. Fonte: @colorpalette.cinema

ÁREAS DE CONHECIMENTO

EFICIÊNCIA NO PROCESSAMENTO DE DADOS

Aprender sobre o processo de coleta, armazenamento e processamento de dados desde a camada RAW até a camada Trusted permite otimizar e automatizar essas etapas. Isso resulta em uma maior eficiência na manipulação e transformação dos dados, acelerando a disponibilidade de informações relevantes para os clientes da Compass.

ESCALABILIDADE E FLEXIBILIDADE

O aprendizado sobre o uso de serviços como AWS S3, AWS Lambda e AWS Glue proporciona uma base sólida para lidar com grandes volumes de dados e ambientes de análise em constante evolução.

GERAÇÃO DE VALOR

ANÁLISE E VISUALIZAÇÃO DE DADOS

Com o conhecimento adquirido no uso de ferramentas como AWS Glue, AWS Athena e AWS QuickSight, é possível realizar análises avançadas e criar visualizações impactantes dos dados. Isso permite aos clientes da Compass obter insights valiosos de maneira mais eficiente e concisa.

TOMADA DE DECISÕES EMBASADAS EM DADOS

Os conhecimentos adquiridos no Programa de Bolsas permitem uma compreensão mais profunda sobre como coletar, transformar e analisar dados relevantes para o negócio da Compass. Isso capacita os clientes a tomar decisões embasadas em dados sólidos, aumentando a precisão e melhorando os resultados de suas estratégias e ações.

FERRAMENTAS VALIOSAS

Além dos aprendizados da AWS, ao longo do Programa de Bolsas aprendemos sobre Segurança da Informação, Metodologias Ágeis, versionamento com Git, uso de terminal no Linux, SQL para Análise de Dados, Fundamentos de Big Data, Estatística e Docker.

RESUMO

Toda a trilha de conhecimentos fornecidos pelo Programa de Bolsas – da coleta, processamento, refinamento e análise - capacitam os estagiários da Compass a obter insights valiosos e repassar para os clientes tomarem decisões informadas e desenvolver estratégias eficazes com base em análises de dados confiáveis e visualizações intuitivas. Essa capacidade resulta em um maior valor agregado para os clientes, impulsionando o crescimento e o sucesso de seus negócios.

OBRIGADA

Jade Marinho Torres

jademarinho@hotmail.com

github.com/jademarinho

https://www.linkedin.com/in/jade-marinho/