



Projeto de Engenharia e Análise de Dados

Aylton Neto - Squad 1
Compass.UOL



Resumo

- Projeto Divido em 4 Partes (Sprints)
 - Parte 1 - Carregar arquivos CSV
 - Parte 2 - Definição do Projeto e Ingestão de Dados
 - Parte 3 - Processamento e Modelagem
 - Parte 4 - Consumo e Apresentação
- Sobre o Programa de Bolsas
 - Aprendizado na Compass
 - Agregando aos Clientes

Projeto - Parte 1

Carregar Arquivos na RAW

- Script Python e Dockerfile
 - Carregar Dados para o Bucket

```
import boto3

user = boto3.client(
    service_name='s3',
    aws_access_key_id=[REDACTED],
    aws_secret_access_key=[REDACTED],
    region_name='us-east-1'
)

pasta_movies = f'Raw/Local/CSV/Movies/2023/05/29/movies.csv'
pasta_series = f'Raw/Local/CSV/Series/2023/05/29/series.csv'

user.upload_file('movies.csv', 'data-lake-do-aylton', pasta_movies)
user.upload_file('series.csv', 'data-lake-do-aylton', pasta_series)
```

Dockerfile ×

Compass.uol > Sprint07 > .DataAnalytics07 > DesafioETL > Dockerfile

```
1 FROM python:3.10-slim
2
3 COPY . .
4
5 WORKDIR /app
6
7 RUN pip install boto3
8
9 CMD python desafio_ETL.py
```

Projeto - Parte 2

Definição do Projeto

- Squad 1 - Animação/Comédia
- Tema: Avaliar o Cenário das Séries de Animação com Boa Avaliação
 - Em qual período foram lançadas mais séries boas?
 - Séries lançadas em períodos anteriores possuem relevância atualmente?
 - Existe predominância de algum país?
 - Existem padrões relacionados a duração?
 - Existem padrões relacionados ao nº de episódios?
 - Quais emissoras e estúdios se destacam?
 - Quais são as melhores animações?

Definição do Projeto

- Definir Amostra
 - Métricas

Resultados:

```
+-----+-----+
|summary|      numeroVotos|
+-----+-----+
|  count|              7764|
|  mean| 2047.3037094281299|
| stddev|14651.769885249427|
|   min|                5|
|   max|             518958|
+-----+-----+
```

```
+-----+-----+
|summary|      notaMedia|
+-----+-----+
|  count|              7764|
|  mean| 6.977807831014931|
| stddev|1.1028955069267738|
|   min|                1.0|
|   max|                9.7|
+-----+-----+
```

Ingestão de Dados do TMDb

- Explicando código

Trecho 1 - Bibliotecas e Chaves de Acesso

```
import json
```

```
import requests
```

```
import pandas as pd
```

```
import boto3
```

```
def lambda_handler(event, context):
```

```
    user = boto3.client(
```

```
        service_name='s3',
```

```
        aws_access_key_id= [REDACTED],
```

```
        aws_secret_access_key= [REDACTED],
```

```
        region_name='us-east-1'
```

```
    )
```

```
    api_key = [REDACTED]
```


Ingestão de Dados do TMDB

- Explicando código

Trecho 2 - Função para obter o TMDB(ID) pelo IMDB(ID) e extrair dados da API, caso encontre algum resultado.

```
def get_data_TMDB(id_IMDB, api_key):  
    url = f'https://api.themoviedb.org/3/find/{id_IMDB}?api_key={api_key}&external_source=imdb_id'  
    response1 = requests.get(url)  
    data1 = response1.json()  
  
    if 'tv_results' in data1 and len(data1['tv_results']) > 0:  
        id_TMDB = data1['tv_results'][0]['id']  
        url = f'https://api.themoviedb.org/3/tv/{id_TMDB}?api_key={api_key}'  
        response2 = requests.get(url)  
        data2 = response2.json()  
        return data2  
    return None
```

Ingestão de Dados do TMDB

- Explicando código

Trecho 3 - Realiza a busca do arquivo no bucket e faz a filtragem das animações acima da média

```
response = user.get_object(Bucket='data-lake-do-aylton', Key='Raw/Local/CSV/Series/2023/05/29/series.csv')  
data_CSV = pd.read_csv(response['Body'], sep='|')
```

```
filter_animations = data_CSV.loc[(data_CSV['genero'].str.contains('Animation')) &  
(data_CSV['notaMedia'] >= 7.0) &  
(data_CSV['numeroVotos'] >= 2050)]
```

```
df = filter_animations.drop_duplicates(subset='id').sort_values(by='numeroVotos', ascending=False)
```

Trecho 4 - Looping para fazer as requisições na API com base nos IDs e nos campos determinados

```
series_informations = []
```

```
count, num_json = 0, 0
```

```
for index, row in enumerate(df.iterrows()):
```

```
    id_IMDB = str(row[1]['id'])
```

```
    series_details = get_data_TMDB(id_IMDB, api_key)
```

```
    if series_details != None:
```

```
        title = series_details['name']
```

```
        origin = series_details['origin_country'][0] if series_details['origin_country'] else None
```

```
        company = series_details['production_companies'][0]['name'] if series_details['production_companies'] else None
```

```
        network = series_details['networks'][0]['name'] if series_details['networks'] else None
```

```
        production = series_details['in_production']
```

```
        episodes = series_details['number_of_episodes']
```

```
        seasons = series_details['number_of_seasons']
```

```
        dic_details = {
```

```
            'id': id_IMDB,
```

```
            'titulo': title,
```

```
            'pais': origin,
```

```
            'estudio': company,
```

```
            'emissora': network,
```

```
            'mediaEpisodios': episodes//seasons,
```

```
            'emProdução': production
```

```
        }
```

```
        series_informations.append(dic_details)
```

```
        count += 1
```

Ingestão de Dados do TMDb

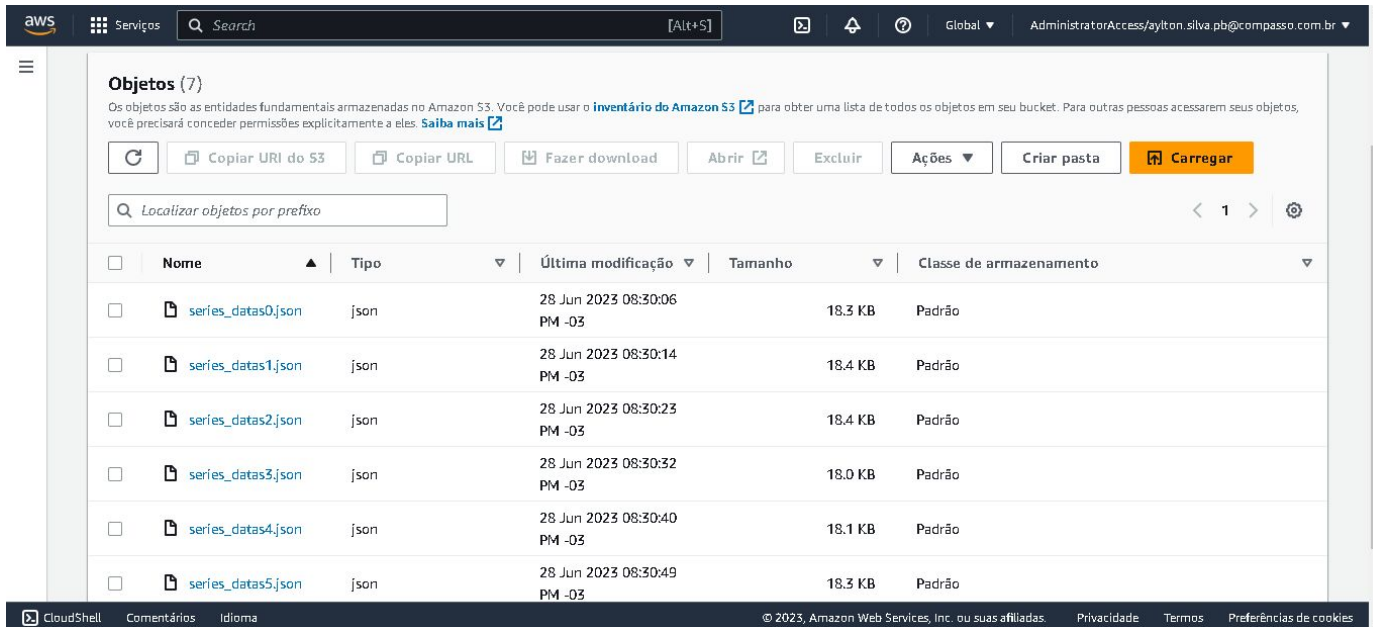
- Explicando código

```
# Trecho 5 - Condição para que um novo arquivo seja criado a cada 100 séries ou quando chegar ao fim da seleção
if count % 100 == 0 or index == len(df)-1:
    json_information = json.dumps(series_information)
    user.put_object(Body=json_information,
                    Bucket='data-lake-do-aylton',
                    Key=f'Raw/TMDb/JSON/2023/06/28/series_datas{num_json}.json')
    series_information = []
    num_json += 1
    continue







return {
    "statusCode": 200,
    "body": "Arquivo JSON criado com sucesso."
}
```

Ingestão de Dados do TMDb

- Arquivos JSON no Bucket



The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with the AWS logo, 'Serviços', a search bar, and a user profile. Below this, the main content area is titled 'Objetos (7)'. It includes a descriptive paragraph about S3 objects and a set of action buttons: 'Copiar URI do S3', 'Copiar URL', 'Fazer download', 'Abrir', 'Excluir', 'Ações', 'Criar pasta', and 'Carregar'. A search bar for object prefixes is also present. The main part of the interface is a table listing the objects.

<input type="checkbox"/>	Nome	Tipo	Última modificação	Tamanho	Classe de armazenamento
<input type="checkbox"/>	 series_dados0.json	json	28 Jun 2023 08:30:06 PM -03	18.3 KB	Padrão
<input type="checkbox"/>	 series_dados1.json	json	28 Jun 2023 08:30:14 PM -03	18.4 KB	Padrão
<input type="checkbox"/>	 series_dados2.json	json	28 Jun 2023 08:30:23 PM -03	18.4 KB	Padrão
<input type="checkbox"/>	 series_dados3.json	json	28 Jun 2023 08:30:32 PM -03	18.0 KB	Padrão
<input type="checkbox"/>	 series_dados4.json	json	28 Jun 2023 08:30:40 PM -03	18.1 KB	Padrão
<input type="checkbox"/>	 series_dados5.json	json	28 Jun 2023 08:30:49 PM -03	18.3 KB	Padrão

At the bottom of the console, there's a footer with 'CloudShell', 'Comentários', 'Idioma', and copyright information for Amazon Web Services, Inc. in 2023, along with links for 'Privacidade', 'Termos', and 'Preferências de cookies'.

Ingestão de Dados do TMDB

- Resultados
 - Após a filtragem pelo gênero “Animation”: 7764 Animações
 - Médias: numeroVotos = 2050 e notaMedia = 7.0
 - Após a extração no TMDB: 628 Animações
 - 7 Arquivos JSON no Bucket

Projeto - Parte 3

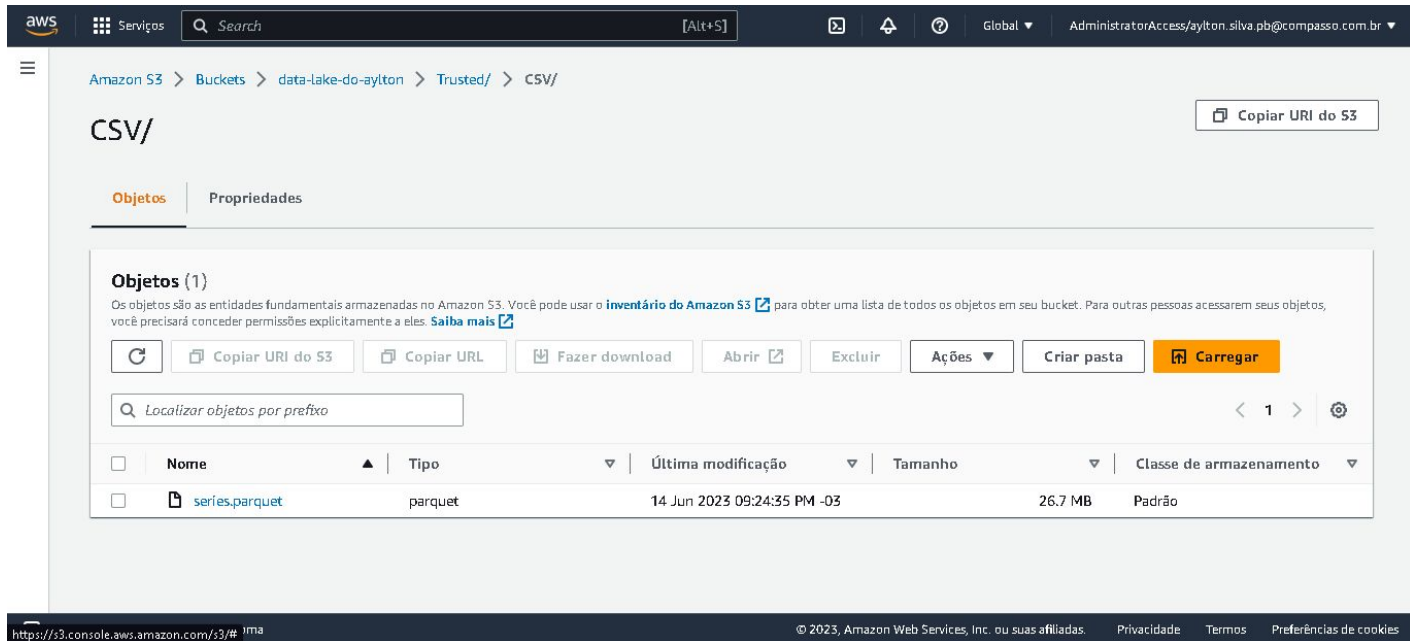
Processamento da Trusted

- 1ª JOB GLUE - Arquivos CSV

```
17 csv_path = "s3://data-lake-do-aylton/Raw/Local/CSV/Series/2023/05/29/series.csv"
18
19 csv_data = spark.read.format("csv").option("header", "true").option("delimiter", "|").load(csv_path)
20 csv_data = csv_data.repartition(1)
21 csv_data.write.mode("append").parquet("s3://data-lake-do-aylton/Trusted/CSV/")
22
23 job.commit()
```


Processamento da Trusted

- 1ª JOB GLUE - Arquivos CSV



Amazon S3 > Buckets > data-lake-do-aylton > Trusted/ > CSV/

CSV/

[Copiar URI do S3](#)

Objetos | Propriedades

Objetos (1)

Os objetos são as entidades fundamentais armazenadas no Amazon S3. Você pode usar o [inventário do Amazon S3](#) para obter uma lista de todos os objetos em seu bucket. Para outras pessoas acessarem seus objetos, você precisará conceder permissões explicitamente a eles. [Saiba mais](#)

[Atualizar](#) [Copiar URI do S3](#) [Copiar URL](#) [Fazer download](#) [Abrir](#) [Excluir](#) [Ações](#) [Criar pasta](#) [Carregar](#)

<input type="checkbox"/>	Nome	Tipo	Última modificação	Tamanho	Classe de armazenamento
<input type="checkbox"/>	series.parquet	parquet	14 Jun 2023 09:24:35 PM -03	26.7 MB	Padrão

<http://s3.console.aws.amazon.com/s3/#> [Ima](#)

© 2023, Amazon Web Services, Inc. ou suas afiliadas. [Privacidade](#) [Termos](#) [Preferências de cookies](#)

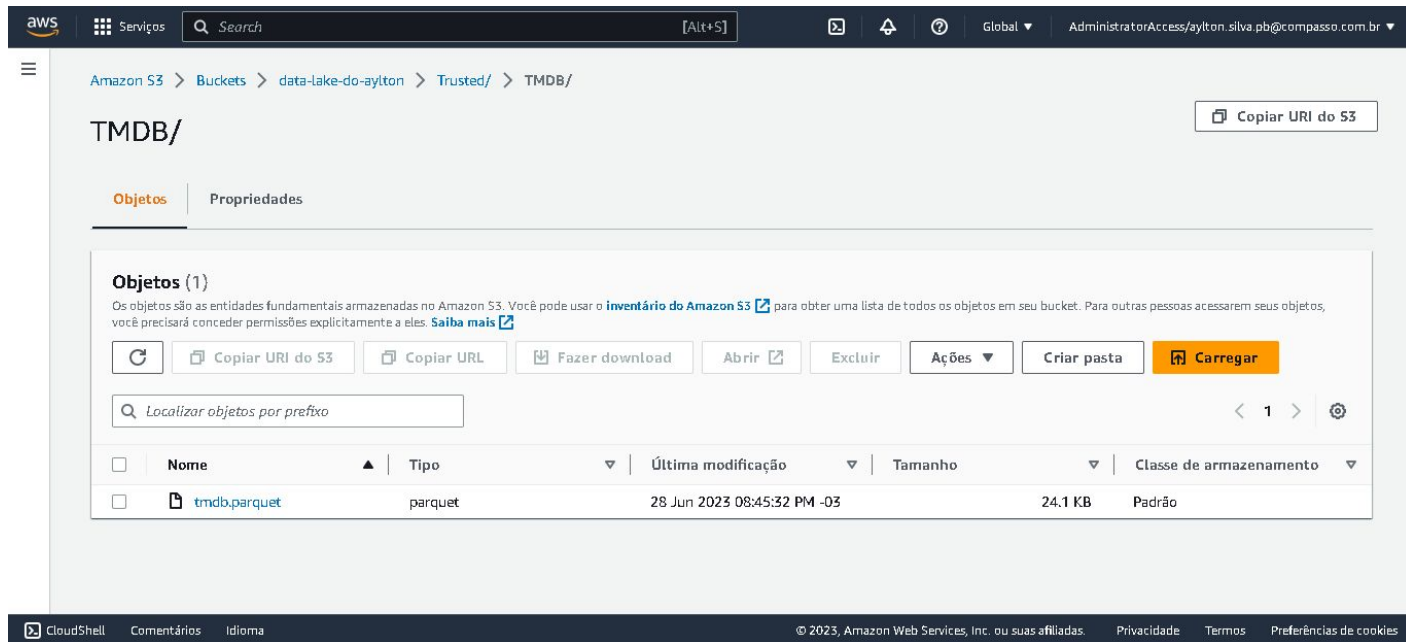
Processamento da Trusted

- 2ª JOB GLUE - Arquivos JSON

```
json_paths = [  
    "s3://data-lake-do-aylton/Raw/TMDB/JSON/2023/06/28/series_datas0.json",  
    "s3://data-lake-do-aylton/Raw/TMDB/JSON/2023/06/28/series_datas1.json",  
    "s3://data-lake-do-aylton/Raw/TMDB/JSON/2023/06/28/series_datas2.json",  
    "s3://data-lake-do-aylton/Raw/TMDB/JSON/2023/06/28/series_datas3.json",  
    "s3://data-lake-do-aylton/Raw/TMDB/JSON/2023/06/28/series_datas4.json",  
    "s3://data-lake-do-aylton/Raw/TMDB/JSON/2023/06/28/series_datas5.json",  
    "s3://data-lake-do-aylton/Raw/TMDB/JSON/2023/06/28/series_datas6.json"  
]  
  
json_data = spark.read.json(json_paths)  
json_data = json_data.repartition(1)  
json_data.write.mode("append").parquet("s3://data-lake-do-aylton/Trusted/TMDB/")  
  
job.commit()
```

Processamento da Trusted

- 2ª JOB GLUE - Arquivos JSON

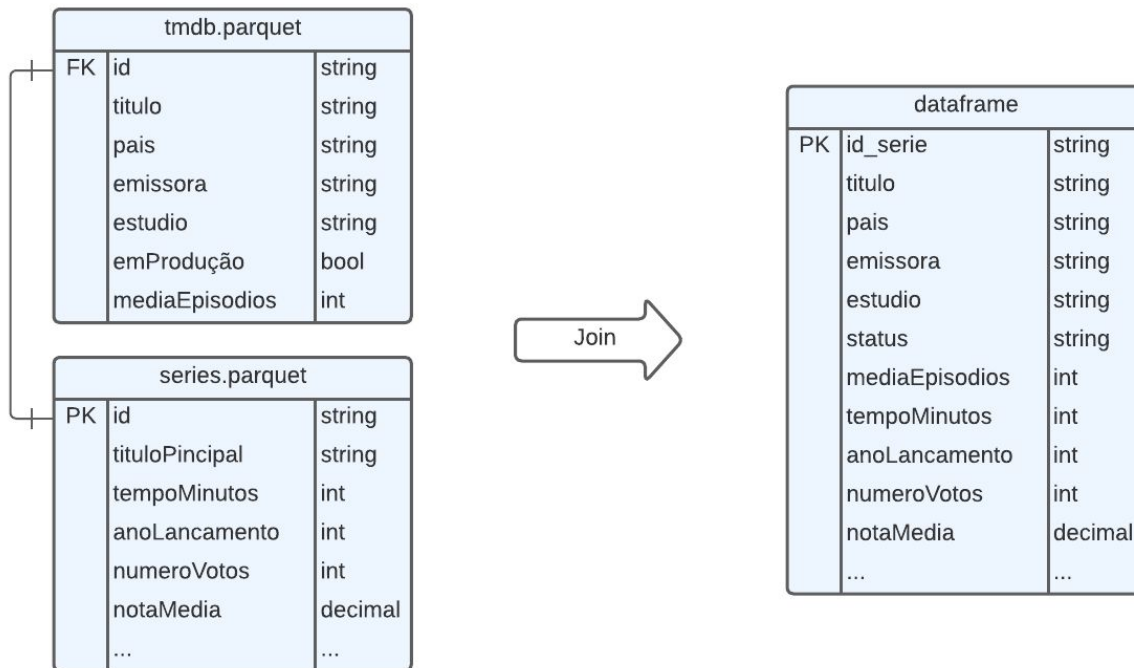


The screenshot shows the AWS S3 console interface. The breadcrumb navigation indicates the path: Amazon S3 > Buckets > data-lake-do-aylton > Trusted/ > TMDB/. The main heading is 'TMDB/'. There are two tabs: 'Objetos' (selected) and 'Propriedades'. Below the tabs, there is a section titled 'Objetos (1)' with a descriptive paragraph and a link to 'Saiba mais'. A row of action buttons includes 'Copiar URI do S3', 'Copiar URL', 'Fazer download', 'Abrir', 'Excluir', 'Ações', 'Criar pasta', and 'Carregar'. Below the buttons is a search bar with the placeholder text 'Localizar objetos por prefixo'. A table lists the objects with columns for selection, name, type, last modified, size, and storage class. One object is listed: 'tmdb.parquet' of type 'parquet', modified on '28 Jun 2023 08:45:32 PM -03', with a size of '24.1 KB' and storage class 'Padrão'.

	Nome	Tipo	Última modificação	Tamanho	Classe de armazenamento
<input type="checkbox"/>	tmdb.parquet	parquet	28 Jun 2023 08:45:32 PM -03	24.1 KB	Padrão

Modelagem de Dados

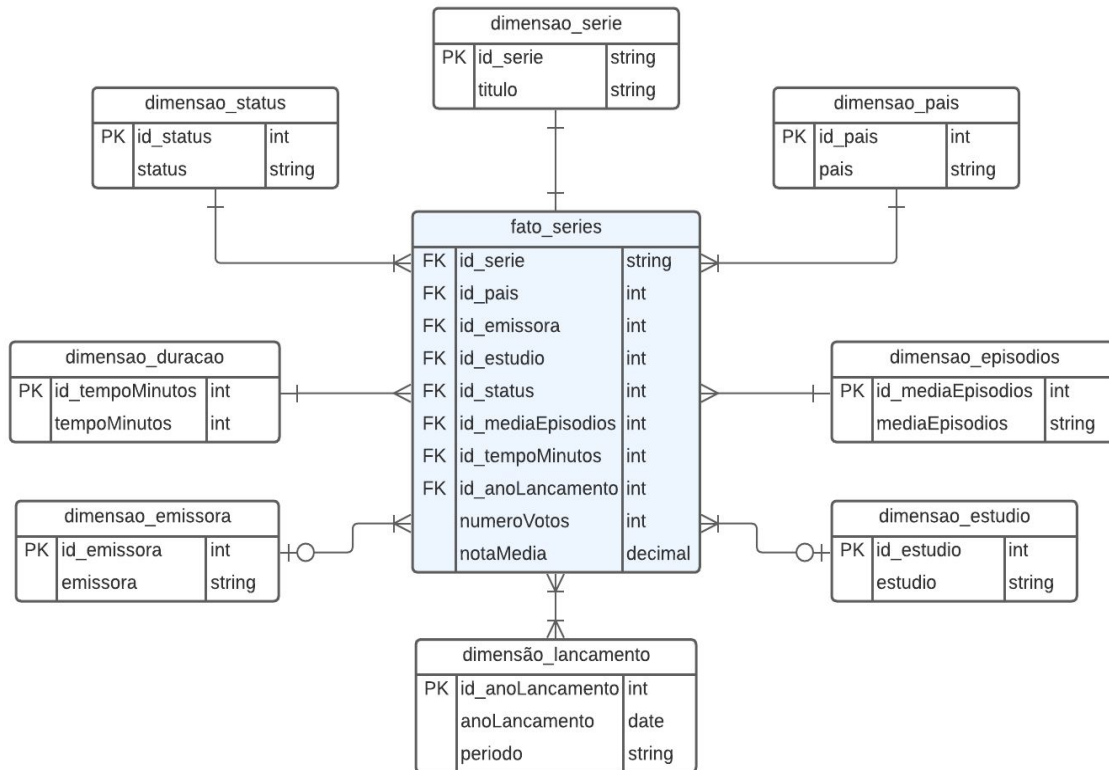
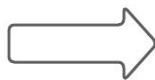
- Modelagem:



Modelagem de Dados

- Modelagem:

dataframe		
PK	id_serie	string
	titulo	string
	pais	string
	emissora	string
	estudio	string
	status	string
	mediaEpisodios	int
	tempoMinutos	int
	anoLancamento	int
	numeroVotos	int
	notaMedia	decimal



Processamento da Refined

- Explicando Código:

```
# 1ª Etapa | Bibliotecas e Trecho Padrão do GLUE
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.window import Window
import pyspark.sql.functions as func

## @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)
```

Processamento da Refined

- Explicando Código:

2ª Etapa | Carregar arquivos parquet

```
df_dataCSV = spark.read.parquet('s3://data-lake-do-aylton/Trusted/CSV/series.parquet')
```

```
df_tmdb = spark.read.parquet('s3://data-lake-do-aylton/Trusted/TMDB/tmdb.parquet')
```

3ª Etapa | Fazer a filtragem nos dados

```
df_dataCSV_filter = df_dataCSV.filter((df_dataCSV['genero'].contains('Animation')) &  
                                       (df_dataCSV['notaMedia'] >= 7.0) &  
                                       (df_dataCSV['numeroVotos'] >= 2050))
```

```
df_topSeries = df_dataCSV_filter.dropDuplicates(['id'])
```

4ª Etapa | Unir dataframes dataCSV e tmdb e modificar colunas

```
df_fato = df_topSeries.join(df_tmdb, on='id')
```

```
df_fato = df_fato.withColumnRenamed('id', 'id_serie')
```

```
df_fato = df_fato.withColumn('status', func.when(df_fato['emProdução'] == 'true', 'Em Produção').otherwise('Finalizado'))
```

Processamento da Refined

- Explicando Código:

```
# 5ª Etapa | Criação das Dimensões
```

```
# Dimensão Serie
```

```
df_serie = df_fato.select('id_serie', 'titulo')
```

```
# Dimensão Lançamento
```

```
df_lancamento = df_fato.select('anoLancamento').distinct().orderBy('anoLancamento')
```

```
df_lancamento = df_lancamento.withColumn('id_anoLancamento', func.row_number().over(Window.orderBy('anoLancamento')))
```

```
df_lancamento = df_lancamento.select('id_anoLancamento', 'anoLancamento')
```

```
df_lancamento = df_lancamento.withColumn('periodo', func.when(func.col("anoLancamento").between(1920, 1960), "Era de Ouro Americana")  
    .when(func.col("anoLancamento").between(1960, 1980), "Influência dos Animes")  
    .when(func.col("anoLancamento").between(1980, 2000), "TV por Assinatura")  
    .when(func.col("anoLancamento").between(2000, 2010), "Popularidade dos DVDs")  
    .when(func.col("anoLancamento").between(2010, 9999), "Serviços de Streaming"))
```

```
# Dimensão Duração
```

```
df_duracao = df_fato.withColumn('tempoMinutos', func.col('tempoMinutos').cast('integer'))
```

```
df_duracao = df_duracao.select('tempoMinutos').distinct().orderBy('tempoMinutos').filter(df_duracao['tempoMinutos'].isNotNull())
```

```
df_duracao = df_duracao.withColumn('id_tempoMinutos', func.row_number().over(Window.orderBy('tempoMinutos')))
```

```
df_duracao = df_duracao.select('id_tempoMinutos', 'tempoMinutos')
```


Dimensão Episódios

```
df_episodios = df_fato.select('mediaEpisodios').distinct().orderBy('mediaEpisodios')
df_episodios = df_episodios.withColumn('id_mediaEpisodios', func.row_number().over(Window.orderBy('mediaEpisodios'))))
df_episodios = df_episodios.select('id_mediaEpisodios', 'mediaEpisodios')
```

Dimensão País

```
df_pais = df_fato.select('pais').distinct().orderBy('pais').filter(df_fato['pais'].isNotNull())
df_pais = df_pais.withColumn('id_pais', func.row_number().over(Window.orderBy('pais'))))
df_pais = df_pais.select('id_pais', 'pais')
```

Dimensão Status

```
df_status = df_fato.select('status').distinct()
df_status = df_status.withColumn('id_status', func.row_number().over(Window.orderBy('status'))))
df_status = df_status.select('id_status', 'status')
```

Dimensão Estúdio

```
df_estudio = df_fato.select('estudio').distinct().orderBy('estudio').filter(df_fato['estudio'].isNotNull())
df_estudio = df_estudio.withColumn('id_estudio', func.row_number().over(Window.orderBy('estudio'))))
df_estudio = df_estudio.select('id_estudio', 'estudio')
```

Dimensão Emissora

```
df_emissora = df_fato.select('emissora').distinct().orderBy('emissora').filter(df_fato['emissora'].isNotNull())
df_emissora = df_emissora.withColumn('id_emissora', func.row_number().over(Window.orderBy('emissora'))))
df_emissora = df_emissora.select('id_emissora', 'emissora')
```

Processamento da Refined

- Explicando Código:

6ª Etapa | Criando Tabela Fato através de joins

```
df_fato = df_fato.join(df_lancamento, on='anoLancamento', how='left')
df_fato = df_fato.join(df_duracao, on='tempoMinutos', how='left')
df_fato = df_fato.join(df_episodios, on='mediaEpisodios', how='left')
df_fato = df_fato.join(df_pais, on='pais', how='left')
df_fato = df_fato.join(df_status, on='status', how='left')
df_fato = df_fato.join(df_estudio, on='estudio', how='left')
df_fato = df_fato.join(df_emissora, on='emissora', how='left')
```

Selecionando colunas da tabela Fato

```
df_fato = df_fato.select('id_serie', 'id_pais', 'id_emissora', 'id_estudio', 'id_status', 'id_mediaEpisodios',
                        'id_tempoMinutos', 'id_anoLancamento', 'numeroVotos', 'notaMedia')
```


Processamento da Refined

- Explicando Código:


7 Etapa | Salvar os dataframes como tabelas em formato Parquet na Refined

```
df_serie.repartition(1).write.saveAsTable(name='projeto_animacoes.dim_serie', mode='overwrite', path='s3://data-lake-do-aylton/Refined/Serie', format='parquet')
df_lancamento.repartition(1).write.saveAsTable(name='projeto_animacoes.dim_lancamento', mode='overwrite', path='s3://data-lake-do-aylton/Refined/Lancamento', format='parquet')
df_duracao.repartition(1).write.saveAsTable(name='projeto_animacoes.dim_duracao', mode='overwrite', path='s3://data-lake-do-aylton/Refined/Duracao', format='parquet')
df_episodios.repartition(1).write.saveAsTable(name='projeto_animacoes.dim_episodios', mode='overwrite', path='s3://data-lake-do-aylton/Refined/Episodios', format='parquet')
df_pais.repartition(1).write.saveAsTable(name='projeto_animacoes.dim_pais', mode='overwrite', path='s3://data-lake-do-aylton/Refined/Pais', format='parquet')
df_status.repartition(1).write.saveAsTable(name='projeto_animacoes.dim_status', mode='overwrite', path='s3://data-lake-do-aylton/Refined/Status', format='parquet')
df_emissora.repartition(1).write.saveAsTable(name='projeto_animacoes.dim_emissora', mode='overwrite', path='s3://data-lake-do-aylton/Refined/Emissora', format='parquet')
df_estudio.repartition(1).write.saveAsTable(name='projeto_animacoes.dim_estudio', mode='overwrite', path='s3://data-lake-do-aylton/Refined/Estudio', format='parquet')
df_fato.repartition(1).write.saveAsTable(name='projeto_animacoes.fato_series', mode='overwrite', path='s3://data-lake-do-aylton/Refined/Fato', format='parquet')

job.commit()
```

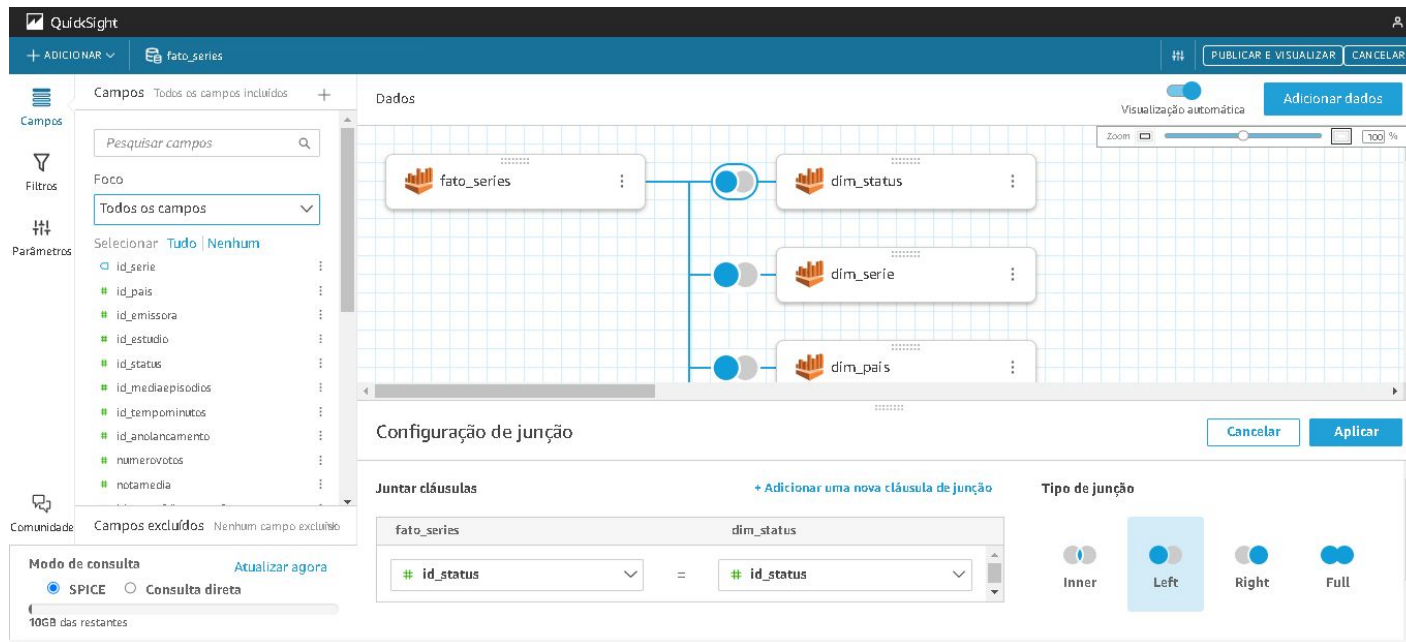


Projeto - Parte 4



Consumo e Apresentação

- Acessar dados no QuickSight



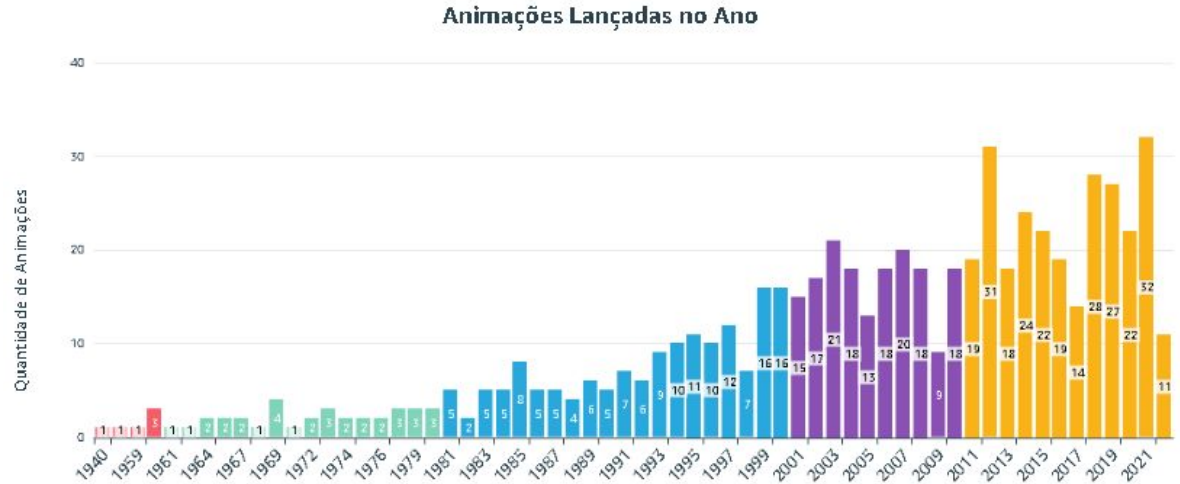
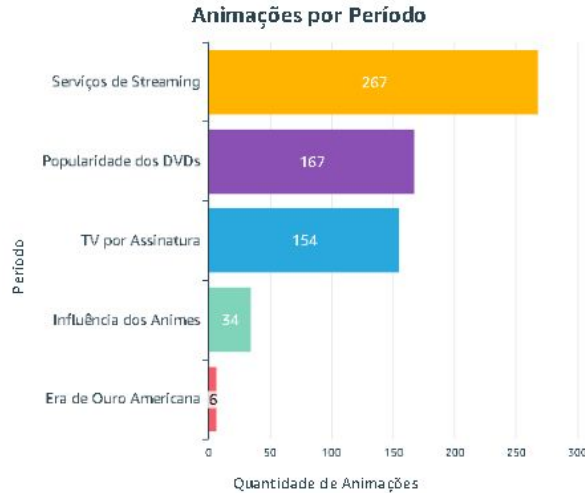
The screenshot displays the Amazon QuickSight console interface. At the top, the title bar shows 'QuickSight' and a search bar with 'fato_series' entered. Below the title bar, there are buttons for 'PUBLICAR E VISUALIZAR' and 'CANCELAR'. The main workspace is divided into several sections:

- Campos (Fields):** A sidebar on the left containing a search bar 'Pesquisar campos' and a list of fields under 'Todos os campos incluídos'. The fields include 'id_serie', 'id_pais', 'id_emissora', 'id_estudio', 'id_status', 'id_mediaepisodios', 'id_tempominutos', 'id_anlancamento', 'numerosovotos', and 'notamedia'.
- Dados (Data):** A central area showing a data model diagram. It features a fact table 'fato_series' connected to three dimension tables: 'dim_status', 'dim_serie', and 'dim_pais'. Each connection is represented by a blue line and a circular join symbol.
- Configuração de junção (Join Configuration):** A panel at the bottom showing the join configuration for the selected tables. It includes a 'Juntar cláusulas' (Join clauses) section with a dropdown for 'fato_series' and a dropdown for 'dim_status', both set to '# id_status'. A 'Tipo de junção' (Join type) section shows four options: 'Inner', 'Left' (selected), 'Right', and 'Full'.

At the bottom left, there is a 'Modo de consulta' (Query mode) section with 'SPICE' selected and 'Consulta direta' (Direct query) as an option. A status bar at the very bottom indicates '10GB das restantes'.

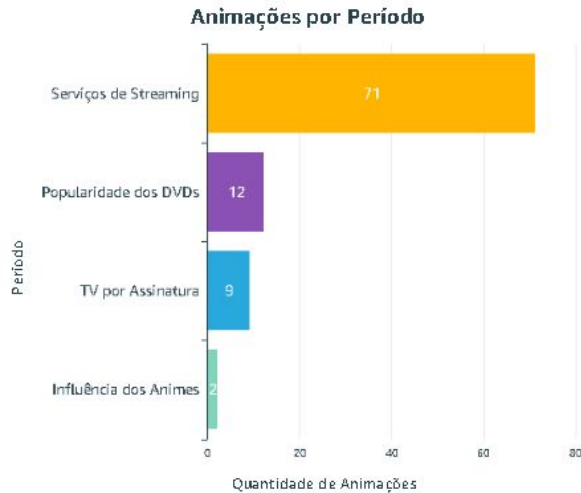
Consumo e Apresentação

- Qual foi o período em que foram lançadas mais séries boas?
 - Esperado: Maior quantidade de séries boas lançadas no último período



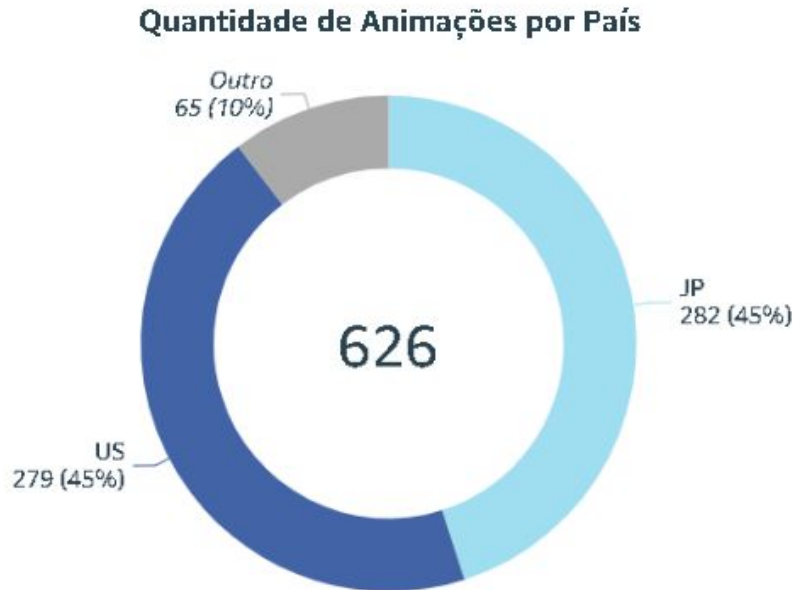
Consumo e Apresentação

- Séries lançadas em períodos anteriores possuem relevância atualmente?
 - Esperado: Quantidade pequena de séries antigas (pouca relevância)



Consumo e Apresentação

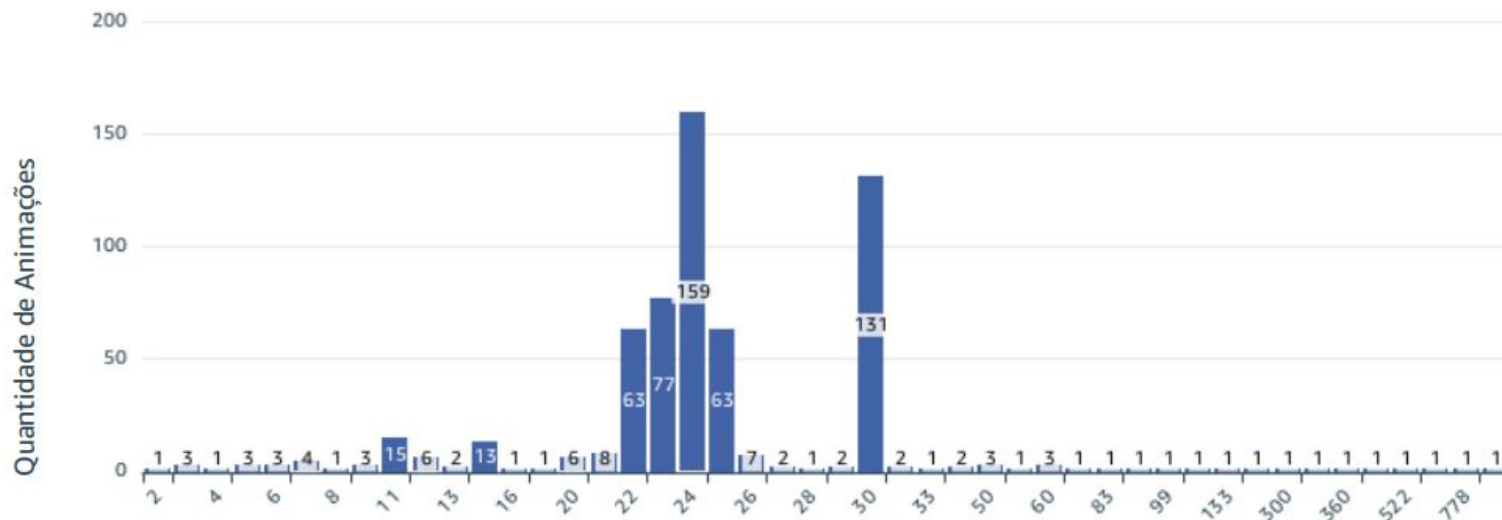
- Existe predominância de algum país?
 - Esperado: Animações Japonesas devido a popularidade dos Animes



Consumo e Apresentação

- Existem padrões relacionados a duração?
 - Esperado: 24min de duração (Baseado nos Animes)

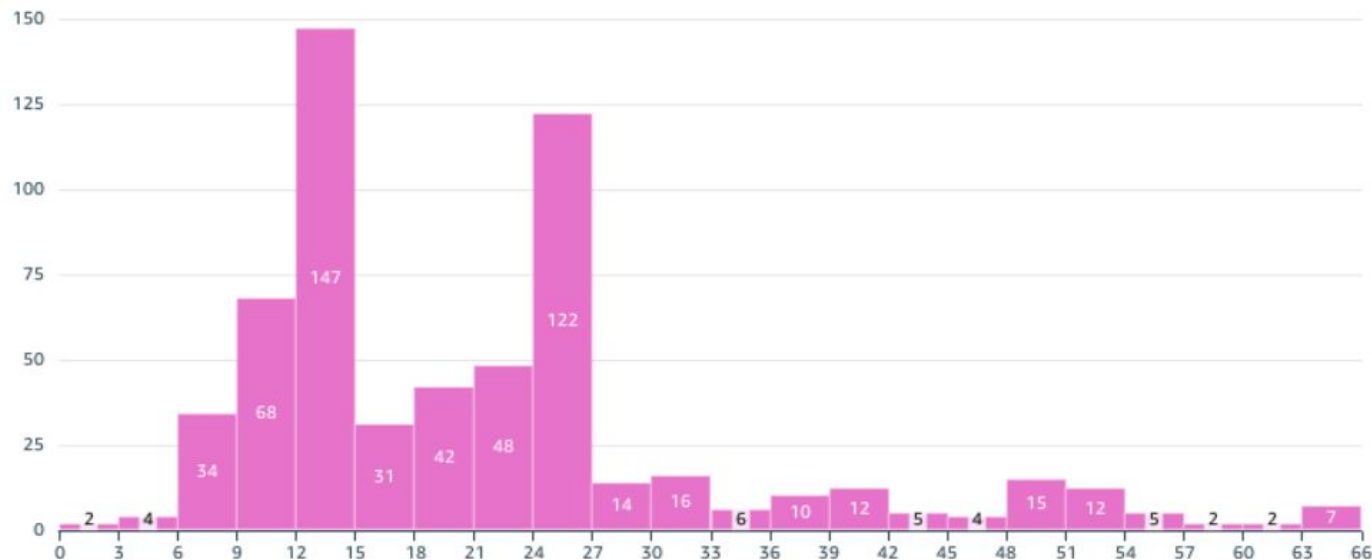
Frequencia de Duração das Animações



Consumo e Apresentação

- Existem padrões relacionados ao nº de episódios?
 - Esperado: 12 episódios por temporada (Baseado nos Animes)

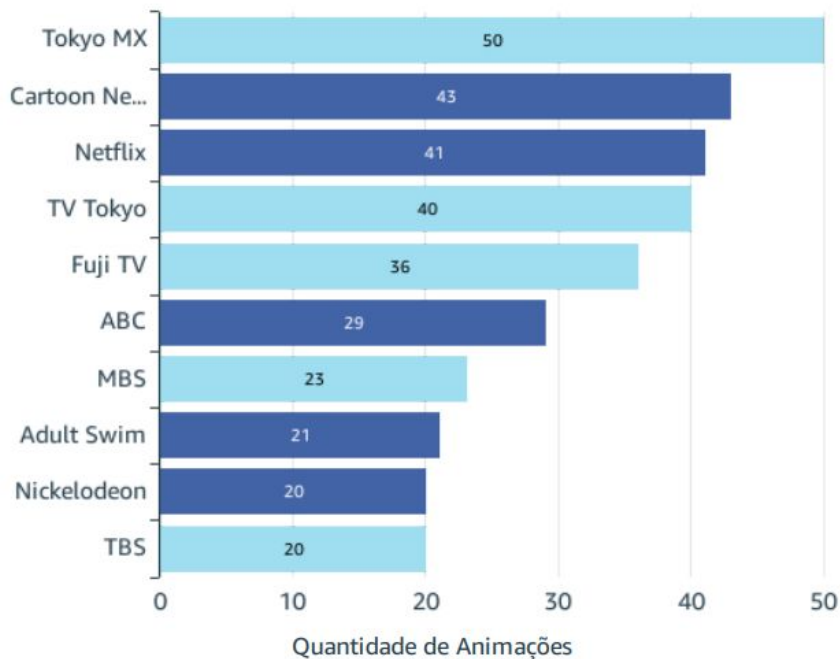
Média de Episódios por Temporada



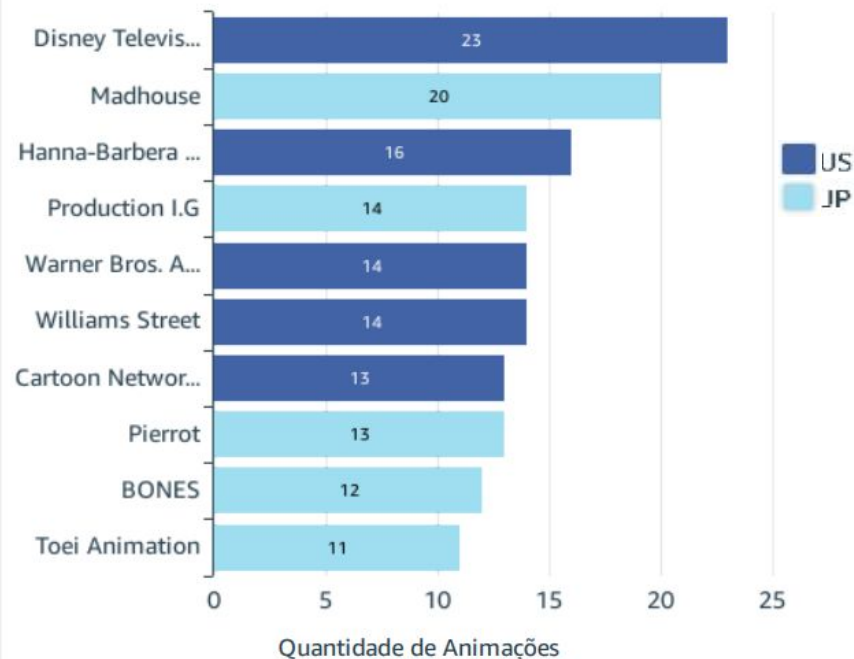
Consumo e Apresentação

- Quais emissoras e estúdios se destacam?

Top Emissoras



Top Estúdios



Consumo e Apresentação

- Quais são as melhores animações de acordo com a análise?

Top Séries

País	Título	Emissora	Estúdio	Status	Ano	Nota	Numero de Votos	Duração
US	Arcane	Netflix	Fortiche ...	Em Produção	2021	9,0	205,071	41
	Avatar: The Last ...	Nickelodeon	Nickelodeon ...	Finalizado	2005	9,3	314,371	23
	Batman: The ...	FOX	DC Comics	Finalizado	1992	9,0	104,459	23
	Rick and Morty	Adult Swim	Williams Street	Em Produção	2013	9,1	518,958	23
JP	Attack on Titan	MBS	Production I.G	Em Produção	2013	9,0	371,218	24
	Death Note	Nippon TV	Madhouse	Finalizado	2006	9,0	323,143	24
	Fullmetal ...	MBS	BONES	Finalizado	2009	9,1	167,814	24
	Hunter x Hunter	Nippon TV	Madhouse	Finalizado	2011	9,0	100,138	24
IR	The Heroes	null	null	Em Produção	2008	9,4	165,984	30

Sobre o Programa de Bolsas

Aprendizado na Compass

- Tecnologias Aprendidas

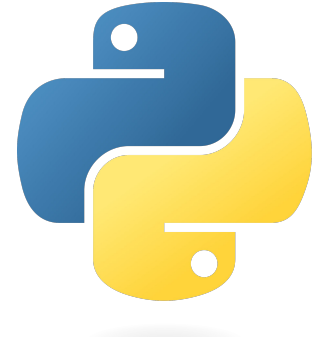
Aprendizado na Compass

- Tecnologias Aprendidas



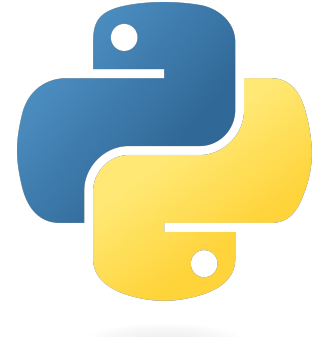
Aprendizado na Compass

- Tecnologias Aprendidas



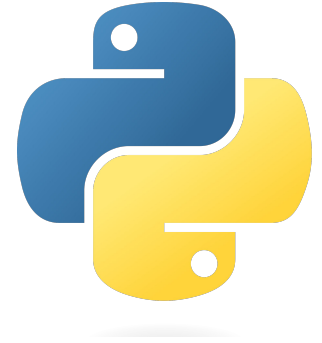
Aprendizado na Compass

- Tecnologias Aprendidas



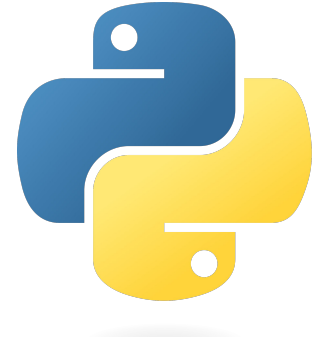
Aprendizado na Compass

- Tecnologias Aprendidas



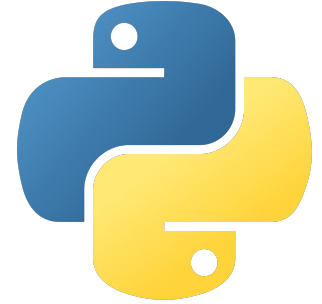
Aprendizado na Compass

- Tecnologias Aprendidas



Aprendizado na Compass

- Tecnologias Aprendidas



Aprendizado na Compass

- Não somente Hardskills!
 - Trabalho em Equipe
 - Equilíbrio Emocional
 - Pontualidade
 - Organização



Agregando aos Clientes

- Como agregar?
 - Análises
 - Arquiteturas
 - Soluções
 - Insights





Muito Obrigado!

