!apt-get install openjdk-11-jdk-headless -qq > /dev/null

!wget -q https://archive.apache.org/dist/spark/spark-3.1.2/spark-3.1.2-bin-hadoop2.7.tgz

!tar xf spark-3.1.2-bin-hadoop2.7.tgz

import os

os.environ["JAVA\_HOME"] = "/usr/lib/jvm/java-11-openjdk-amd64"

os.environ["SPARK\_HOME"] = "/content/spark-3.1.2-bin-hadoop2.7"

!pip install -q findspark

import findspark

findspark.init()

from pyspark.sql import SparkSession

sc = SparkSession.builder.master("local[\*]").getOrCreate()

sc

!wget --verbose --show-progress --no-check-certificate https://raw.githubusercontent.com/jonates/opendata/refs/heads/master/netflix/netflix\_titles.csv -O netflix\_titles.csv

netflix = sc.read.csv(

path = "/content/netflix\_titles.csv",

inferSchema = True,

header = True,

sep = ',',

encoding = "UTF-8")

type(netflix)

netflix.count()

netflix.printSchema()

from pyspark.sql.functions import\*

from pyspark.sql.types import \*

from google.colab import drive

drive.mount('/content/drive')

separando\_atores = netflix.withColumn("lista\_atores", split(col("cast"), ", "))

atores = separando\_atores.withColumn("atores", explode(col("lista\_atores"))).drop("lista\_atores", "cast")

contagem\_atores = atores.groupBy("atores").count().orderBy(col("count").desc())

contagem\_atores.show(truncate = False)

contagem\_atores.coalesce(1).write.csv(

"/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/contagem\_atores.csv",

header=True,

mode="overwrite",

encoding="UTF-8",

nullValue="",

quoteAll=True

)

print("DataFrame 'contagem\_atores' salvo como CSV.")

output\_directory\_path = "/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/contagem\_atores.csv"

!rm -f {output\_directory\_path}/\*.crc {output\_directory\_path}/.\*.crc

!rm -rf {output\_directory\_path}/\_SUCCESS

print("Arquivos .crc e \_SUCCESS removidos.")

atores\_e\_titulos = atores.filter(col("atores").isNotNull() & col("title").isNotNull()).groupBy("atores").agg(collect\_set("title").alias("titulo\_alias"), count("\*").alias("count")).orderBy(col("count").desc())\

.withColumn("titulos", concat\_ws(", ", col("titulo\_alias"))).drop("titulo\_alias", "count")

atores\_e\_titulos.show(truncate=False)

atores\_e\_titulos.coalesce(1).write.csv(

"/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/atores\_e\_titulos.csv",

header=True,

mode="overwrite",

encoding="UTF-8",

nullValue="",

quoteAll=True

)

print("DataFrame 'atores\_e\_titulos' salvo como CSV.")

output\_directory\_path = "/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/atores\_e\_titulos.csv"

!rm -f {output\_directory\_path}/\*.crc {output\_directory\_path}/.\*.crc

!rm -rf {output\_directory\_path}/\_SUCCESS

print("Arquivos .crc e \_SUCCESS removidos.")

separando\_diretores = netflix.withColumn("lista\_diretores", split(col("director"), ", "))

diretores = separando\_diretores.withColumn("diretor", explode(col("lista\_diretores"))).drop("lista\_diretores", "director")

contagem\_diretores = diretores.groupBy("diretor").count().orderBy(col("count").desc()).filter(col("diretor").isNotNull())

contagem\_diretores.show(truncate = False)

contagem\_diretores.coalesce(1).write.csv(

"/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/contagem\_diretores.csv",

header=True,

mode="overwrite",

encoding="UTF-8",

nullValue="",

quoteAll=True

)

print("DataFrame 'contagem\_diretores' salvo como CSV.")

output\_directory\_path = "/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/contagem\_diretores.csv"

!rm -f {output\_directory\_path}/\*.crc {output\_directory\_path}/.\*.crc

!rm -rf {output\_directory\_path}/\_SUCCESS

print("Arquivos .crc e \_SUCCESS removidos.")

diretores\_e\_titulos = diretores.filter(col("diretor").isNotNull() & col("title").isNotNull()).groupBy("diretor").agg(collect\_set("title").alias("titulo\_alias"), count("\*").alias("count")).orderBy(col("count").desc())\

.withColumn("titulos", concat\_ws(", ", col("titulo\_alias"))).drop("titulo\_alias", "count")

diretores\_e\_titulos.show(truncate = False)

diretores\_e\_titulos.coalesce(1).write.csv(

"/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/diretores\_e\_titulos.csv",

header=True,

mode="overwrite",

encoding="UTF-8",

nullValue="",

quoteAll=True

)

print("DataFrame 'diretores\_e\_titulos' salvo como CSV.")

output\_directory\_path = "/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/diretores\_e\_titulos.csv"

!rm -f {output\_directory\_path}/\*.crc {output\_directory\_path}/.\*.crc

!rm -rf {output\_directory\_path}/\_SUCCESS

print("Arquivos .crc e \_SUCCESS removidos.")

ator\_join\_diretor = atores.join(diretores, on="title", how="inner")

ator\_e\_diretor = ator\_join\_diretor.groupBy("diretor", "atores").count().filter((col("count") >= 1) & (col("diretor").isNotNull())).orderBy(col("count").desc())

ator\_e\_diretor.show(truncate = False)

ator\_e\_diretor.coalesce(1).write.csv(

"/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/ator\_e\_diretor.csv",

header=True,

mode="overwrite",

encoding="UTF-8",

nullValue="",

quoteAll=True

)

print("DataFrame 'ator\_e\_diretor' salvo como CSV.")

output\_directory\_path = "/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/ator\_e\_diretor.csv"

!rm -f {output\_directory\_path}/\*.crc {output\_directory\_path}/.\*.crc

!rm -rf {output\_directory\_path}/\_SUCCESS

print("Arquivos .crc e \_SUCCESS removidos.")

ator\_e\_diretor\_com\_titulos = ator\_join\_diretor.filter(col("diretor").isNotNull() & col("atores").isNotNull() & col("title").isNotNull())\

.groupBy("diretor", "atores").agg(collect\_set("title").alias("titulo\_alias"), count("\*").alias("count"))\

.orderBy(col("count").desc()).withColumn("titulos", concat\_ws(", ", col("titulo\_alias"))).drop("titulo\_alias", "count")

ator\_e\_diretor\_com\_titulos.show(truncate=False)

ator\_e\_diretor\_com\_titulos.coalesce(1).write.csv(

"/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/ator\_e\_diretor\_com\_titulos.csv",

header=True,

mode="overwrite",

encoding="UTF-8",

nullValue="",

quoteAll=True

)

print("DataFrame 'ator\_e\_diretor\_com\_titulos' salvo como CSV.")

output\_directory\_path = "/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/ator\_e\_diretor\_com\_titulos.csv"

!rm -f {output\_directory\_path}/\*.crc {output\_directory\_path}/.\*.crc

!rm -rf {output\_directory\_path}/\_SUCCESS

print("Arquivos .crc e \_SUCCESS removidos.")

separando\_generos = netflix.withColumn("lista\_generos", split(col("listed\_in"), ", "))

generos\_temp = separando\_generos.withColumn("genero\_explodido", explode(col("lista\_generos")))

generos = generos\_temp.withColumn("genero", regexp\_replace(col("genero\_explodido"), "(Movies|TV Shows)$", "")).drop("lista\_generos", "listed\_in", "genero\_explodido")

cont\_genero = generos.groupBy("genero").count().orderBy(col("count").desc()).filter(col("genero").isNotNull())

cont\_genero.show(truncate = False)

cont\_genero.coalesce(1).write.csv(

"/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/cont\_genero.csv",

header=True,

mode="overwrite",

encoding="UTF-8",

nullValue="",

quoteAll=True

)

print("DataFrame 'cont\_genero' salvo como CSV.")

output\_directory\_path = "/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/cont\_genero.csv"

!rm -f {output\_directory\_path}/\*.crc {output\_directory\_path}/.\*.crc

!rm -rf {output\_directory\_path}/\_SUCCESS

print("Arquivos .crc e \_SUCCESS removidos.")

genero\_e\_titulo = generos.filter(col("genero").isNotNull() & col("title").isNotNull()).groupBy("genero").agg(collect\_set("title").alias("titulo\_alias"), count("\*").alias("count")).orderBy(col("count").desc())\

.withColumn("titulos", concat\_ws(", ", col("titulo\_alias"))).drop("titulo\_alias", "count")

genero\_e\_titulo.show(truncate = False)

genero\_e\_titulo.coalesce(1).write.csv(

"/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/genero\_e\_titulo.csv",

header=True,

mode="overwrite",

encoding="UTF-8",

nullValue="",

quoteAll=True

)

print("DataFrame 'genero\_e\_titulo' salvo como CSV.")

output\_directory\_path = "/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/genero\_e\_titulo.csv"

!rm -f {output\_directory\_path}/\*.crc {output\_directory\_path}/.\*.crc

!rm -rf {output\_directory\_path}/\_SUCCESS

print("Arquivos .crc e \_SUCCESS removidos.")

pegando\_filme\_ou\_serie = netflix.select("type", "title")

filme\_ou\_serie = pegando\_filme\_ou\_serie.groupBy("type").count().orderBy(col("count").desc())\

.filter(col("type").isNotNull() & (col("type") != "William Wyler"))

filme\_ou\_serie.show()

filme\_ou\_serie.coalesce(1).write.csv(

"/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/filme\_ou\_serie.csv",

header=True,

mode="overwrite",

encoding="UTF-8",

nullValue="",

quoteAll=True

)

print("DataFrame 'filme\_ou\_serie' salvo como CSV.")

output\_directory\_path = "/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/filme\_ou\_serie.csv"

!rm -f {output\_directory\_path}/\*.crc {output\_directory\_path}/.\*.crc

!rm -rf {output\_directory\_path}/\_SUCCESS

print("Arquivos .crc e \_SUCCESS removidos.")

filme\_ou\_serie\_por\_titulo = filme\_ou\_serie

filme\_serie\_por\_titulo = pegando\_filme\_ou\_serie.filter(col("type").isNotNull() & col("title").isNotNull())\

.groupBy("type").agg(collect\_set("title").alias("titulo\_alias"), count("\*").alias("count")).orderBy(col("titulo\_alias").asc())\

.withColumn("titulos", concat\_ws(", ", col("titulo\_alias"))).drop("titulo\_alias", "count")

filme\_serie\_por\_titulo.show(truncate = False)

filme\_serie\_por\_titulo.coalesce(1).write.csv(

"/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/filme\_serie\_por\_titulo.csv",

header=True,

mode="overwrite",

encoding="UTF-8",

nullValue="",

quoteAll=True

)

print("DataFrame 'filme\_serie\_por\_titulo' salvo como CSV.")

output\_directory\_path = "/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/filme\_serie\_por\_titulo.csv"

!rm -f {output\_directory\_path}/\*.crc {output\_directory\_path}/.\*.crc

!rm -rf {output\_directory\_path}/\_SUCCESS

print("Arquivos .crc e \_SUCCESS removidos.")

separando\_paises = netflix.withColumn("lista\_paises", split(col("country"), ", "))

paises = separando\_paises.withColumn("pais", explode(col("lista\_paises"))).drop("lista\_paises", "country")

pais = paises.groupBy("pais").count().orderBy(col("count").desc()).filter(col("pais").isNotNull())

pais.show(truncate = False)

pais.coalesce(1).write.csv(

"/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/pais.csv",

header=True,

mode="overwrite",

encoding="UTF-8",

nullValue="",

quoteAll=True

)

print("DataFrame 'pais' salvo como CSV.")

output\_directory\_path = "/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/pais.csv"

!rm -f {output\_directory\_path}/\*.crc {output\_directory\_path}/.\*.crc

!rm -rf {output\_directory\_path}/\_SUCCESS

print("Arquivos .crc e \_SUCCESS removidos.")

paises\_e\_titulo = paises.filter(col("pais").isNotNull() & col("title").isNotNull()).groupBy("pais").agg(collect\_set("title").alias("titulo\_alias"), count("\*").alias("count")).orderBy(col("count").desc())\

.withColumn("titulos", concat\_ws(", ", col("titulo\_alias"))).drop("titulo\_alias", "count")

paises\_e\_titulo.show(truncate = False)

paises\_e\_titulo.coalesce(1).write.csv(

"/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/paises\_e\_titulo.csv",

header=True,

mode="overwrite",

encoding="UTF-8",

nullValue="",

quoteAll=True

)

print("DataFrame 'paises\_e\_titulo' salvo como CSV.")

output\_directory\_path = "/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/paises\_e\_titulo.csv"

!rm -f {output\_directory\_path}/\*.crc {output\_directory\_path}/.\*.crc

!rm -rf {output\_directory\_path}/\_SUCCESS

print("Arquivos .crc e \_SUCCESS removidos.")

mes\_add = netflix.withColumn("mes", date\_format(to\_date(col("date\_added"), "MMMM d, yyyy"), "MMMM"))

ano\_add = netflix.withColumn("ano", year(to\_date(col("date\_added"), "MMMM d, yyyy")).cast(IntegerType()))

mes\_de\_add = mes\_add.groupBy("mes").count().orderBy(col("count").desc()).filter(col("mes").isNotNull())

mes\_de\_add.show(truncate = False)

mes\_de\_add.coalesce(1).write.csv(

"/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/mes\_de\_add.csv",

header=True,

mode="overwrite",

encoding="UTF-8",

nullValue="",

quoteAll=True

)

print("DataFrame 'mes\_de\_add' salvo como CSV.")

output\_directory\_path = "/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/mes\_de\_add.csv"

!rm -f {output\_directory\_path}/\*.crc {output\_directory\_path}/.\*.crc

!rm -rf {output\_directory\_path}/\_SUCCESS

print("Arquivos .crc e \_SUCCESS removidos.")

mes\_ano = netflix.withColumn("mes\_ano", date\_format(to\_date(col("date\_added"), "MMMM d, yyyy"), "MMMM-yyyy"))

mes\_e\_ano = mes\_ano.groupBy("mes\_ano").count().orderBy(col("count").desc()).filter(col("mes\_ano").isNotNull())

mes\_e\_ano.show(truncate = False)

mes\_e\_ano.coalesce(1).write.csv(

"/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/mes\_e\_ano.csv",

header=True,

mode="overwrite",

encoding="UTF-8",

nullValue="",

quoteAll=True

)

print("DataFrame 'mes\_e\_ano' salvo como CSV.")

output\_directory\_path = "/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/mes\_e\_ano.csv"

!rm -f {output\_directory\_path}/\*.crc {output\_directory\_path}/.\*.crc

!rm -rf {output\_directory\_path}/\_SUCCESS

print("Arquivos .crc e \_SUCCESS removidos.")

ano\_de\_add = ano\_add.groupBy("ano").count().orderBy(col("ano").asc()).filter(col("ano").isNotNull())

ano\_de\_add.show()

ano\_de\_add.coalesce(1).write.csv(

"/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/ano\_de\_add.csv",

header=True,

mode="overwrite",

encoding="UTF-8",

nullValue="",

quoteAll=True

)

print("DataFrame 'ano\_de\_add' salvo como CSV.")

output\_directory\_path = "/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/ano\_de\_add.csv"

!rm -f {output\_directory\_path}/\*.crc {output\_directory\_path}/.\*.crc

!rm -rf {output\_directory\_path}/\_SUCCESS

print("Arquivos .crc e \_SUCCESS removidos.")

ano\_add\_titulo = ano\_add.filter(col("ano").isNotNull() & col("title").isNotNull()).groupBy("ano").agg(collect\_set("title").alias("titulo\_alias"), count("\*").alias("count")).orderBy(col("ano").asc())\

.withColumn("titulos", concat\_ws(", ", col("titulo\_alias"))).drop("titulo\_alias", "count")

ano\_add\_titulo.show(truncate=False)

ano\_add\_titulo.coalesce(1).write.csv(

"/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/ano\_add\_titulo.csv",

header=True,

mode="overwrite",

encoding="UTF-8",

nullValue="",

quoteAll=True

)

print("DataFrame 'ano\_add\_titulo' salvo como CSV.")

output\_directory\_path = "/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/ano\_add\_titulo.csv"

!rm -f {output\_directory\_path}/\*.crc {output\_directory\_path}/.\*.crc

!rm -rf {output\_directory\_path}/\_SUCCESS

print("Arquivos .crc e \_SUCCESS removidos.")

ano\_add\_selecionado = ano\_add.select("date\_added", "ano")

generos\_selecionado = generos.select("date\_added", "genero", "title")

genero\_join\_ano = ano\_add\_selecionado.join(generos\_selecionado, on="date\_added", how="inner")

genero\_por\_ano = genero\_join\_ano.groupBy("ano", "genero").count().orderBy(col("ano").desc(),col("count").desc()).filter(col("ano").isNotNull())

genero\_por\_ano.show(truncate = False)

genero\_por\_ano.coalesce(1).write.csv(

"/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/genero\_por\_ano.csv",

header=True,

mode="overwrite",

encoding="UTF-8",

nullValue="",

quoteAll=True

)

print("DataFrame 'genero\_por\_ano' salvo como CSV.")

output\_directory\_path = "/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/genero\_por\_ano.csv"

!rm -f {output\_directory\_path}/\*.crc {output\_directory\_path}/.\*.crc

!rm -rf {output\_directory\_path}/\_SUCCESS

print("Arquivos .crc e \_SUCCESS removidos.")

genero\_por\_ano\_e\_titulo = genero\_join\_ano.filter(col("ano").isNotNull() & col("genero").isNotNull() & col("title").isNotNull())\

.groupBy("ano", "genero").agg(collect\_set("title").alias("titulo\_alias"), count("\*").alias("count"))\

.orderBy(col("count").desc()).withColumn("titulos", concat\_ws(", ", col("titulo\_alias"))).drop("titulo\_alias", "count")

genero\_por\_ano\_e\_titulo.show(truncate = False)

genero\_por\_ano\_e\_titulo.coalesce(1).write.csv(

"/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/genero\_por\_ano\_e\_titulo.csv",

header=True,

mode="overwrite",

encoding="UTF-8",

nullValue="",

quoteAll=True

)

print("DataFrame 'genero\_por\_ano\_e\_titulo' salvo como CSV.")

output\_directory\_path = "/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/genero\_por\_ano\_e\_titulo.csv"

!rm -f {output\_directory\_path}/\*.crc {output\_directory\_path}/.\*.crc

!rm -rf {output\_directory\_path}/\_SUCCESS

print("Arquivos .crc e \_SUCCESS removidos.")

genero\_tipo = filme\_ou\_serie.join(generos, on="type", how="inner")

genero\_por\_tipo = genero\_tipo.groupBy("type", "genero").count().orderBy(col("count").desc()).filter(col("type").isNotNull())

genero\_por\_tipo.show(truncate = False)

genero\_por\_tipo.coalesce(1).write.csv(

"/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/genero\_por\_tipo.csv",

header=True,

mode="overwrite",

encoding="UTF-8",

nullValue="",

quoteAll=True

)

print("DataFrame 'genero\_por\_tipo' salvo como CSV.")

output\_directory\_path = "/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/genero\_por\_tipo.csv"

!rm -f {output\_directory\_path}/\*.crc {output\_directory\_path}/.\*.crc

!rm -rf {output\_directory\_path}/\_SUCCESS

print("Arquivos .crc e \_SUCCESS removidos.")

genero\_por\_tipo\_com\_titulo = genero\_tipo.filter(col("type").isNotNull() & col("genero").isNotNull() & col("title").isNotNull())\

.groupBy("type", "genero").agg(collect\_set("title").alias("titulo\_alias"), count("\*").alias("count"))\

.orderBy(col("count").desc()).withColumn("titulos", concat\_ws(", ", col("titulo\_alias"))).drop("titulo\_alias", "count")

genero\_por\_tipo\_com\_titulo.show(truncate = False)

genero\_por\_tipo\_com\_titulo.coalesce(1).write.csv(

"/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/genero\_por\_tipo\_com\_titulo.csv",

header=True,

mode="overwrite",

encoding="UTF-8",

nullValue="",

quoteAll=True

)

print("DataFrame 'genero\_por\_tipo\_com\_titulo' salvo como CSV.")

output\_directory\_path = "/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/genero\_por\_tipo\_com\_titulo.csv"

!rm -f {output\_directory\_path}/\*.crc {output\_directory\_path}/.\*.crc

!rm -rf {output\_directory\_path}/\_SUCCESS

print("Arquivos .crc e \_SUCCESS removidos.")

genero\_e\_pais\_add = paises.join(generos, on="title", how="inner")

genero\_por\_pais = genero\_e\_pais\_add.groupBy("pais", "genero").count().orderBy(col("count").desc()).filter(col("pais").isNotNull() & col("genero").isNotNull())

genero\_por\_pais.show(truncate = False)

genero\_por\_pais.coalesce(1).write.csv(

"/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/genero\_por\_pais.csv",

header=True,

mode="overwrite",

encoding="UTF-8",

nullValue="",

quoteAll=True

)

print("DataFrame 'genero\_por\_pais' salvo como CSV.")

output\_directory\_path = "/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/genero\_por\_pais.csv"

!rm -f {output\_directory\_path}/\*.crc {output\_directory\_path}/.\*.crc

!rm -rf {output\_directory\_path}/\_SUCCESS

print("Arquivos .crc e \_SUCCESS removidos.")

genero\_por\_pais\_com\_titulo = genero\_e\_pais\_add.filter(col("pais").isNotNull() & col("genero").isNotNull() & col("title").isNotNull())\

.groupBy("pais", "genero").agg(collect\_set("title").alias("titulo\_alias"), count("\*").alias("count"))\

.orderBy(col("count").desc()).withColumn("titulos", concat\_ws(", ", col("titulo\_alias"))).drop("titulo\_alias", "count")

genero\_por\_pais\_com\_titulo.show(truncate = False)

genero\_por\_pais\_com\_titulo.coalesce(1).write.csv(

"/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/genero\_por\_pais\_com\_titulo.csv",

header=True,

mode="overwrite",

encoding="UTF-8",

nullValue="",

quoteAll=True

)

print("DataFrame 'genero\_por\_pais\_com\_titulo' salvo como CSV.")

output\_directory\_path = "/content/drive/MyDrive/projeto\_netflix\_big\_data\_powerBi/genero\_por\_pais\_com\_titulo.csv"

!rm -f {output\_directory\_path}/\*.crc {output\_directory\_path}/.\*.crc

!rm -rf {output\_directory\_path}/\_SUCCESS

print("Arquivos .crc e \_SUCCESS removidos.")