Projeto Semantix

Nível Básico

▼ 1. Enviar os dado para o hdfs

```
# entrando no conteiner
docker exec -it namenode bash

hdfs dfs -mkdir /user/dados_covid/brutos/

hdfs dfs -put /input/HIST_PAINEL_COVID /user/dados_covid/brutos/
```

Acessando o beeline

```
docker exec -it hive-server bash

beeline -u jdbc:hive2://localhost:10000
```

```
create database dados_covid location '/apps/painel_covid/warehouse/';
```

ingerindo dados havia tabala externa

```
create database dados_locais ;


create external table hist_covid_brutos(
regiao string,
estado string,
municipio string,
coduf int,
codmun int,
codRegiaoSaude int,
nomeRegiaoSaude string,
data date,
semanaEpi int,
populacaoTCU2019 int,
casosAcumulado int,
casosNovos int,
obitosAcumulado int,
obitosNovos int,
Recuperadosnovos int,
emAcompanhamentoNovos int,
interior_metropolitana boolean
)
row format delimited
fields terminated by ';'
lines terminated by '\n'
stored as textfile
location '/user/dados_covid/brutos/HIST_PAINEL_COVID/'
tblproperties('skip.header.line.count'='1');
```

Adequação e particionamento dos dados usando python

```
#adicionar formato parquet ao spark
curl -O https://repo1.maven.org/maven2/com/twitter/parquet-hadoop-
bundle/1.6.0/parquet-hadoop-bundle-1.6.0.jar

docker cp parquet-hadoop-bundle-1.6.0.jar jupyter-spark:/opt/spark/jars
```

```
hist_covid_bruto = spark.read.table("dados_locais.hist_covid_brutos")

hist_covid_bruto_null = hist_covid_bruto.na.drop(subset=('estado', 'municipio','codmun'))
hist_covid_saida = hist_covid_bruto_null.drop(
    'Recuperadosnovos',
    'emAcompanhamentoNovos',
    'interior_metropolitana')

hist_covid = hist_covid_saida.filter(hist_covid_saida.municipio!='')

hist_covid.write.partitionBy("municipio")\
    .saveAsTable("dados_covid.hist_covid_municipio")
```

O conjunto de conteines usados já vem com a `SparkSession` SparkSession pronta

▼ 2. Criar Visualização e salvar como tabela Hive

```
hist_covid = spark.read.table("dados_covid.hist_covid_municipio")

casos_acumulados_obitos_regiao = hist_covid\
    .filter(hist_covid.data=="2021-07-06")\
    .groupBy('regiao')\
    .sum('casosAcumulado','obitosacumulado')\
    .withColumnRenamed("sum(casosAcumulado)", "casosAcumulados")\
    .withColumnRenamed("sum(obitosacumulado)", "obitosAcumulado")\
    .withColumn('porcentagem',f.round((f.col("obitosAcumulado")/f.col('casosAcumulados'))*100, 2))

casos_acumulados_obitos_regiao.write\
    .saveAsTable("dados_covid.casos_acumulados_obitos_regiao")
```

▼ Salvar a terceira visualização em um tópico no Kafka

Criando topico

```
docker exec -it kafka bash

kafka-topics.sh --bootstrap-server localhost:9092 --create --topic dados-covid-kafka --partitions 2 --replication-factor 1

kafka-console-consumer.sh --bootstrap-server kafka:9092 --topic dados-covid-kafka
```

Criando visualização e enviando para kafka

```
casos_acumulados_obitos_estado = hist_covid\
    .filter(hist_covid.data=="2021-07-06")\
    .groupBy('estado')\
    .sum('casosAcumulado','obitosacumulado')\
    .withColumnRenamed("sum(casosAcumulado)", "casosAcumulados")\
    .withColumnRenamed("sum(obitosacumulado)", "obitosAcumulado")\
    .withColumn('porcentagem',f.round((f.col("obitosAcumulado")/f.col('casosAcumulados'))*100, 2))

colunas = [col for col in casos_acumulados_obitos_estado.columns]
```

```
envio_kafka = casos_acumulados_obitos_estado\
    .withColumn("value", f.to_json(f.struct([f.col(coluna) for coluna in colunas])))\
    .withColumnRenamed("estado","key")\
    .select("key","value")\
    .show()

envio_kafka.write.format("kafka")\
  .option("kafka.bootstrap.servers", "kafka:9092")\
  .option("topic", "dados-covid-kafka")\
  .save()
```

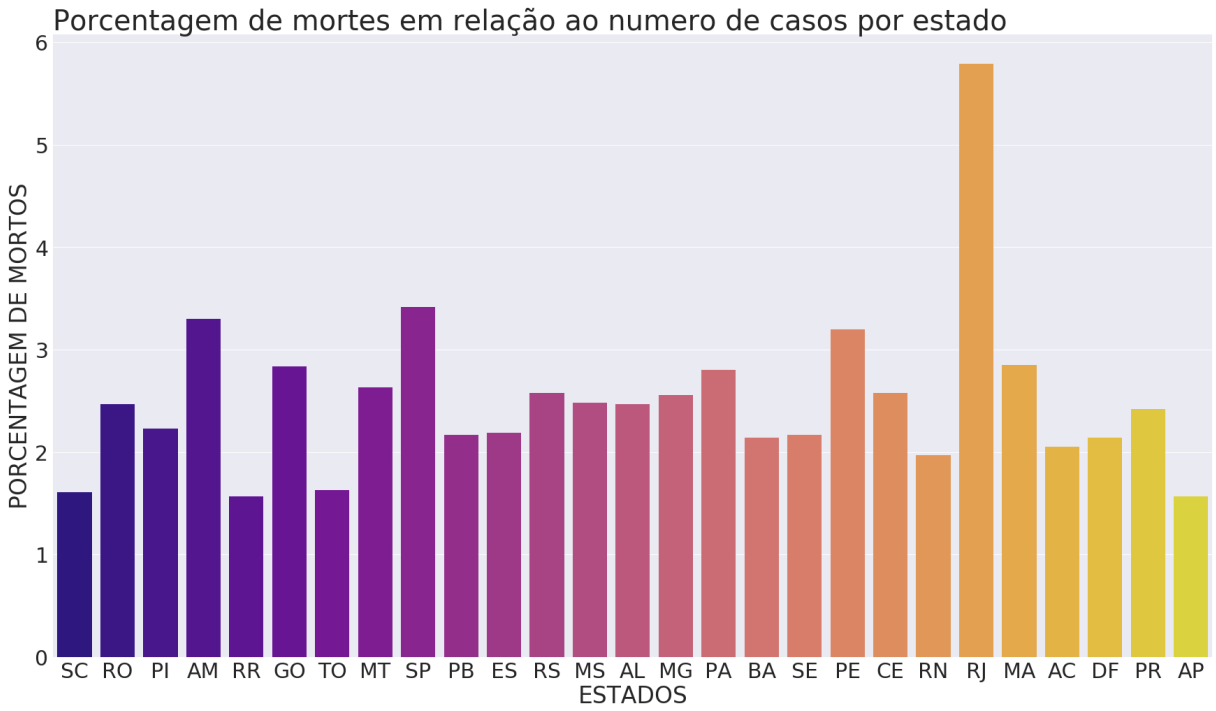
▼ Criar a visualização pelo Spark

```
import matplotlib.pyplot as plt
import seaborn as sns

para_grafico = casos_acumulados_obitos_estado.toPandas()

plt.figure(figsize=(25,15))
plt.title('Porcentagem de mortes em relação ao numero de casos por estado', loc='left', fontsize=40)
sns.set(font_scale=3)
sns.barplot(y='porcentagem', x='estado',  data =para_grafico, palette='plasma');
plt.ylabel('PORCENTAGEM DE MORTOS')
plt.xlabel('ESTADOS')

plt.tight_layout()
```



▼ Salvar a visualização do exercício 6 em um tópico no Elastic e criação de dashboard

```
hist_covid.write\
    .option("header",True)\
    .csv('/user/dados_covid/output/casos_acumulados_obitos_estado/')
```

```
hdfs dfs -get /user/dados_covid/output/casos_acumulados_obitos_estado input/
```