Projeto Final de Spark - Nível Básico

- ## Enviar os dados para o HDFS (através do namenode pelo terminal)

1) Pelo terminal do WSL2,como administrador, dentro da pasta spark, baixe o arquivo de dados .rar dentro da pasta spark:

In [ ]:

```
sudo curl -O https://mobileapps.saude.gov.br/esus-vepi/files/unAFkcaNDeXajurGB7LChj8SgQYS2p
```

Obs.: Caso o link venha a ficar inacessível, o arquivo está disponibilizado no diretório data do GitHub.
Para instalar unrar no Ubuntu você precisa executar os seguintes comandos no terminal:

In [ ]:

```
sudo apt-get update
sudo apt-get install unrar
```

In [ ]:

```
2) Descompacte o arquivo .rar com o comando: sudo unrar x 04bd3419b22b9cc5c6efac2c6528100d_
```

In [ ]:

```
3) Crie um diretório input dentro da pasta spark: mkdir input
```

In [ ]:

```
4) Envie os arquivos .csv para a pasta input: sudo mv *.csv /home/marco/projeto-final-spark
```

In [ ]:

```
### Verificando
marco@DESKTOP-G2455QH:~/projeto-final-spark/spark$ ls ./input/
HIST_PAINEL_COVIDBR_2020_Parte1_06jul2021.csv  HIST_PAINEL_COVIDBR_2021_Parte1_06jul2021.cs
HIST_PAINEL_COVIDBR_2020_Parte2_06jul2021.csv  HIST_PAINEL_COVIDBR_2021_Parte2_06jul2021.cs
```

In [ ]:

```
5)Acesse o namenode utilizando o comando docker exec -it namenode bash.
marco@DESKTOP-G2455QH:~/projeto-final-spark/spark$  docker exec -it namenode bash
```

In [ ]:

```
6) Crie a pasta projeto-final-spark no HDFS para salvar os arquivos de dados .CSV: hdfs dfs

root@namenode:/# hdfs dfs -mkdir -p /user/marco/projeto-final-spark
```

In [ ]:

```
7) Envie os arquivos de dados .CSV para a pasta projeto-final-spark no HDFS: hdfs dfs -put
root@namenode:/# hdfs dfs -put /input/*.csv /user/marco/projeto-final-spark
```

In [ ]:

```
8) Confirme se os arquivos foram enviados: hdfs dfs -ls /user/marco/projeto-final-spark
    root@namenode:/# hdfs dfs -ls /user/marco/projeto-final-spark
Found 4 items
-rw-r--r--   3 root supergroup   62492959 2022-04-26 02:39 /user/marco/projeto-final-spark/
-rw-r--r--   3 root supergroup   76520681 2022-04-26 02:39 /user/marco/projeto-final-spark/
-rw-r--r--   3 root supergroup   91120916 2022-04-26 02:39 /user/marco/projeto-final-spark/
-rw-r--r--   3 root supergroup    3046774 2022-04-26 02:39 /user/marco/projeto-final-spark/
r
```

No jupyter notbook....

Otimizar todos os dados do hdfs para uma tabela Hive particionada por município

In [1]:

```
# Criar Sessão Spark
'''
As vantagens de se utilizar a Classe SparkSession é que a mesma fornece acesso às funcional
•sql Executa as consultas Spark SQL
•catalog Gerenciar tabelas
•read função para ler dados de um arquivo ou outra fonte de dados
•conf objeto para gerenciar configurações de Spark
•sparkContext Core Spark API
'''
```

Out[1]:

```
'\nAs vantagens de se utilizar a Classe SparkSession é que a mesma fornece a
cesso às funcionalidades do spark:\n•sql Executa as consultas Spark SQL\n•ca
talog Gerenciar tabelas\n•read função para ler dados de um arquivo ou outra
fonte de dados\n•conf objeto para gerenciar configurações de Spark\n•sparkCo
ntext Core Spark API\n'
```

In [2]:

```
from pyspark.sql.functions import *

spark = SparkSession\
.builder\
.appName('Projeto final de Spark - Campanha Nacional de Vacinação contra Covid-19')\
.config('spark.some.config.option', 'some-value')\
.enableHiveSupport()\
.getOrCreate()
```

In [3]:

```
from pyspark.sql  import *
from pyspark.sql.types import *
from pyspark.sql import SparkSession
```

In [4]:

```
#Utilizando o option("inferSchema","true") para que automaticamente o spark identifique os
#df_painel_covidbr = spark.read.option("sep",";").option("header","true").option("inferSche

df_painel_covidbr = spark.read.csv('hdfs://namenode/user/marco/projeto-final-spark/',
                                   sep=";", #sep(padrão ,): define o único caractere como s
                                   header=True, #header(padrão false): usa a primeira linha
                                   inferSchema=True, #inferSchema(padrão false): infere o e
                                   ignoreLeadingWhiteSpace=True, #ignoreLeadingWhiteSpace(p
                                   ignoreTrailingWhiteSpace=True) #ignoreTrailingWhiteSpace
```

In [7]:

```
#Visualizar o schema
#df_painel_covidbr.dtypes
df_painel_covidbr.printSchema()
```

```
root
 |-- regiao: string (nullable = true)
 |-- estado: string (nullable = true)
 |-- municipio: string (nullable = true)
 |-- coduf: integer (nullable = true)
 |-- codmun: integer (nullable = true)
 |-- codRegiaoSaude: integer (nullable = true)
 |-- nomeRegiaoSaude: string (nullable = true)
 |-- data: timestamp (nullable = true)
 |-- semanaEpi: integer (nullable = true)
 |-- populacaoTCU2019: integer (nullable = true)
 |-- casosAcumulado: decimal(10,0) (nullable = true)
 |-- casosNovos: integer (nullable = true)
 |-- obitosAcumulado: integer (nullable = true)
 |-- obitosNovos: integer (nullable = true)
 |-- Recuperadosnovos: integer (nullable = true)
 |-- emAcompanhamentoNovos: integer (nullable = true)
 |-- interior/metropolitana: integer (nullable = true)
```

In [8]:

```
df_painel_covidbr.show()
```

```
+------+------+---------+-----+------+-------------+---------------+-------
------------+---------+---------------+-------------+----------+---------
-----+-----------+---------------+--------------------+------------------
---+
|regiao|estado|municipio|coduf|codmun|codRegiaoSaude|nomeRegiaoSaude|
data|semanaEpi|populacaoTCU2019|casosAcumulado|casosNovos|obitosAcumulado|ob
itosNovos|Recuperadosnovos|emAcompanhamentoNovos|interior/metropolitana|
+------+------+---------+-----+------+-------------+---------------+-------
------------+---------+---------------+-------------+----------+---------
-----+-----------+---------------+--------------------+------------------
---+
|Brasil|  null|     null|   76|  null|         null|           null|2020-02
-25 00:00:00|        9|       210147125|             0|         0|
0|          0|           null|                null|                 null|
|Brasil|  null|     null|   76|  null|         null|           null|2020-02
-26 00:00:00|        9|       210147125|             1|         1|
0|          0|           null|                null|                 null|
|Brasil|  null|     null|   76|  null|         null|           null|2020-02
-27 00:00:00|        9|       210147125|             1|         0|
0|          0|           null|                null|                 null|
|Brasil|  null|     null|   76|  null|         null|           null|2020-02
-28 00:00:00|        9|       210147125|             1|         0|
0|          0|           null|                null|                 null|
|Brasil|  null|     null|   76|  null|         null|           null|2020-02
-29 00:00:00|        9|       210147125|             2|         1|
0|          0|           null|                null|                 null|
|Brasil|  null|     null|   76|  null|         null|           null|2020-03
-01 00:00:00|       10|       210147125|             2|         0|
0|          0|           null|                null|                 null|
|Brasil|  null|     null|   76|  null|         null|           null|2020-03
-02 00:00:00|       10|       210147125|             2|         0|
0|          0|           null|                null|                 null|
|Brasil|  null|     null|   76|  null|         null|           null|2020-03
-03 00:00:00|       10|       210147125|             2|         0|
0|          0|           null|                null|                 null|
|Brasil|  null|     null|   76|  null|         null|           null|2020-03
-04 00:00:00|       10|       210147125|             3|         1|
0|          0|           null|                null|                 null|
|Brasil|  null|     null|   76|  null|         null|           null|2020-03
-05 00:00:00|       10|       210147125|             7|         4|
0|          0|           null|                null|                 null|
|Brasil|  null|     null|   76|  null|         null|           null|2020-03
-06 00:00:00|       10|       210147125|            13|         6|
0|          0|           null|                null|                 null|
|Brasil|  null|     null|   76|  null|         null|           null|2020-03
-07 00:00:00|       10|       210147125|            19|         6|
0|          0|           null|                null|                 null|
|Brasil|  null|     null|   76|  null|         null|           null|2020-03
-08 00:00:00|       11|       210147125|            25|         6|
0|          0|           null|                null|                 null|
|Brasil|  null|     null|   76|  null|         null|           null|2020-03
-09 00:00:00|       11|       210147125|            25|         0|
0|          0|           null|                null|                 null|
|Brasil|  null|     null|   76|  null|         null|           null|2020-03
-10 00:00:00|       11|       210147125|            34|         9|
0|          0|           null|                null|                 null|
|Brasil|  null|     null|   76|  null|         null|           null|2020-03
```

```
-11 00:00:00|       11|       210147125|            52|        18|
0|          0|            null|                null|                null|
|Brasil|  null|     null|   76|  null|          null|           null|2020-03
-12 00:00:00|       11|       210147125|            77|        25|
0|          0|            null|                null|                null|
|Brasil|  null|     null|   76|  null|          null|           null|2020-03
-13 00:00:00|       11|       210147125|            98|        21|
0|          0|            null|                null|                null|
|Brasil|  null|     null|   76|  null|          null|           null|2020-03
-14 00:00:00|       11|       210147125|           121|        23|
0|          0|            null|                null|                null|
|Brasil|  null|     null|   76|  null|          null|           null|2020-03
-15 00:00:00|       12|       210147125|           200|        79|
0|          0|            null|                null|                null|
+------+------+---------+-----+------+--------------+---------------+-------
------------+---------+---------------+--------------+----------+----------
-----+-----------+----------------+---------------------+------------------
---+
only showing top 20 rows
```

In [9]:

```
#from pyspark.sql.functions import *
# Ajustando os dados e removendo a informação de hora, pois todas estão zeradas
df_painel_covidbr_limpo = df_painel_covidbr\
.withColumn('data',from_unixtime(unix_timestamp(df_painel_covidbr.data), 'yyyy-MM-dd'))
```

In [10]:

```
#Mostrando o novo dataframe com o campo data ajustado
df_painel_covidbr_limpo.show(3, vertical=True)
```

```
-RECORD 0---------------------------
 regiao                 | Brasil
 estado                 | null
 municipio              | null
 coduf                  | 76
 codmun                 | null
 codRegiaoSaude         | null
 nomeRegiaoSaude        | null
 data                   | 2020-02-25
 semanaEpi              | 9
 populacaoTCU2019       | 210147125
 casosAcumulado         | 0
 casosNovos             | 0
 obitosAcumulado        | 0
 obitosNovos            | 0
 Recuperadosnovos       | null
 emAcompanhamentoNovos  | null
 interior/metropolitana | null
-RECORD 1---------------------------
 regiao                 | Brasil
 estado                 | null
 municipio              | null
 coduf                  | 76
 codmun                 | null
 codRegiaoSaude         | null
 nomeRegiaoSaude        | null
 data                   | 2020-02-26
 semanaEpi              | 9
 populacaoTCU2019       | 210147125
 casosAcumulado         | 1
 casosNovos             | 1
 obitosAcumulado        | 0
 obitosNovos            | 0
 Recuperadosnovos       | null
 emAcompanhamentoNovos  | null
 interior/metropolitana | null
-RECORD 2---------------------------
 regiao                 | Brasil
 estado                 | null
 municipio              | null
 coduf                  | 76
 codmun                 | null
 codRegiaoSaude         | null
 nomeRegiaoSaude        | null
 data                   | 2020-02-27
 semanaEpi              | 9
 populacaoTCU2019       | 210147125
 casosAcumulado         | 1
 casosNovos             | 0
 obitosAcumulado        | 0
 obitosNovos            | 0
 Recuperadosnovos       | null
 emAcompanhamentoNovos  | null
 interior/metropolitana | null
only showing top 3 rows
```

In [ ]:

```
```

In [11]:

```
#Otimizar dados por tabelas particionadas por município
spark.sql("CREATE DATABASE IF NOT EXISTS covidbr")
df_painel_covidbr_limpo.write\
.mode('overwrite')\
.partitionBy('municipio')\
.format('csv')\
.saveAsTable('covidbr.municipio', path='hdfs://namenode:8020/user/hive/warehouse/covidbr/')
```

In [12]:

```
#Confirmar se os dados foram salvos no diretório
!hdfs dfs -ls 'hdfs://namenode:8020/user/hive/warehouse/covidbr/'
```

```
e:8020/user/hive/warehouse/covidbr/municipio=Adrianópolis
drwxr-xr-x   - root supergroup          0 2022-04-27 08:44 hdfs://namenod
e:8020/user/hive/warehouse/covidbr/municipio=Adustina
drwxr-xr-x   - root supergroup          0 2022-04-27 08:44 hdfs://namenod
e:8020/user/hive/warehouse/covidbr/municipio=Afogados da Ingazeira
drwxr-xr-x   - root supergroup          0 2022-04-27 08:44 hdfs://namenod
e:8020/user/hive/warehouse/covidbr/municipio=Afonso Bezerra
drwxr-xr-x   - root supergroup          0 2022-04-27 08:44 hdfs://namenod
e:8020/user/hive/warehouse/covidbr/municipio=Afonso Cláudio
drwxr-xr-x   - root supergroup          0 2022-04-27 08:44 hdfs://namenod
e:8020/user/hive/warehouse/covidbr/municipio=Afonso Cunha
drwxr-xr-x   - root supergroup          0 2022-04-27 08:44 hdfs://namenod
e:8020/user/hive/warehouse/covidbr/municipio=Afrânio
drwxr-xr-x   - root supergroup          0 2022-04-27 08:44 hdfs://namenod
e:8020/user/hive/warehouse/covidbr/municipio=Afuá
drwxr-xr-x   - root supergroup          0 2022-04-27 08:44 hdfs://namenod
e:8020/user/hive/warehouse/covidbr/municipio=Agrestina
drwxr-xr-x   - root supergroup          0 2022-04-27 08:44 hdfs://namenod
e:8020/user/hive/warehouse/covidbr/municipio=Agricolândia
drwxr-xr-x   - root supergroup          0 2022-04-27 08:44 hdfs://namenod
```

Criar as 3 visualizações pelo Spark com os dados enviados para o HDFS

In [13]:

```
# Acessar os dados da tabela municipio do Banco de dados covidbr pelo SparkSQL
spark.sql("SHOW DATABASES").show() #Mostrar os anco de dados
```

```
+------------+
|databaseName|
+------------+
|     covidbr|
|     default|
+------------+
```

In [14]:

```
# Selecionar o banco de dados covidbr
spark.sql("USE covidbr")
```

Out[14]:

```
DataFrame[]
```

In [15]:

```
# Acessar os dados da tabela municipio do Banco de dados covidbr pelo SparkSQL
spark.sql("SHOW TABLES").show() #Mostrar os anco de dados
```

```
+--------+---------+-----------+
|database|tableName|isTemporary|
+--------+---------+-----------+
| covidbr|municipio|      false|
+--------+---------+-----------+
```

Visualização 1 - Casos Recuperados | Em Acompanhamento

Obs.: Os resultados serão extraídos a partir da última data registrada que foi em 2021-07-06.

In [16]:

```
## 1.1 Casos Recuperados
df_casos_recuperados = spark.sql("SELECT Recuperadosnovos as Casos_Recuperados FROM municip
df_casos_recuperados.show()

## 1.2 Em Acompanhamento
df_em_acompanhamento = spark.sql("SELECT emAcompanhamentoNovos as Em_Acompanhamento FROM mu
df_em_acompanhamento.show()
```

```
+-----------------+
|Casos_Recuperados|
+-----------------+
|         17262646|
+-----------------+

+-----------------+
|Em_Acompanhamento|
+-----------------+
|          1317658|
+-----------------+
```

Visualização 2 - Casos Acumulado | Casos Novos | Incidência

Obs.: **Incidência:** Quantidade proporcional de casos confirmados para cada 1 milhão de habitantes. A população considerada no cálculo é a estimativa populacional para 2019 do Instituto Brasileiro de Geografia e Estatística (IBGE), salvo quando especificada referência distinta. **Método de cálculo: (casos confirmados ***

1.000.000) / população. Fonte: https://www.coronavirus.sc.gov.br/sobre-os-dados/ (https://www.coronavirus.sc.gov.br/sobre-os-dados/). No caso no painel será utilizado **Método de cálculo: (casos confirmados * 100.000) / população**

In [23]:

```
#2.1 Acumulado
df_casos_acumulados = spark.sql("SELECT casosAcumulado as Casos_Acumulados FROM municipio c
df_casos_acumulados.show()

#2.2 Casos novos
df_casos_novos = spark.sql("SELECT casosNovos as Casos_Novos FROM municipio order by 1 desc
df_casos_novos.show()

#2.3 Incidência (Casos acumulados)
df_incidencia = spark.sql("SELECT cast(((casosAcumulado*100000)/populacaoTCU2019) as decima
df_incidencia.show()
```

```
+----------------+
|Casos_Acumulados|
+----------------+
|        18855015|
+----------------+

+-----------+
|Casos_Novos|
+-----------+
|     115228|
+-----------+

+----------+
|Incidencia|
+----------+
|    9999.8|
+----------+
```

Visualização 3 - Óbtos Acumulados | Óbtos Novos | Letalidade | Mortalidade

Obs.1: **Letalidade:** Taxa percentual de casos confirmados que evoluíram a óbito. **Método de cálculo: (número de óbitos x 100) / número de casos confirmados**.

Obs.2: **Mortalidade:** Quantidade proporcional de óbitos para cada 1 milhão habitantes, seguindo a mesma metodologia de cálculo da incidência de casos confirmados. **Método de cálculo: (óbitos * 1.000.000) / população**. No caso no painel será utilizado **Método de cálculo: (óbitos * 100.000) / população** Fonte: https://www.coronavirus.sc.gov.br/sobre-os-dados/ (https://www.coronavirus.sc.gov.br/sobre-os-dados/)

In [17]:

```
#3.1 Óbitos acumulados
df_obitos_acumulados = spark.sql("SELECT obitosAcumulado as Obitos_Acumulados FROM municipi
df_obitos_acumulados.show()
```

```
+-----------------+
|Obitos_Acumulados|
+-----------------+
|           526892|
+-----------------+
```

In [18]:

```
#3.2 Obitos novos
df_obitos_novos = spark.sql("SELECT obitosNovos as Obitos_Novos FROM municipio order by 1 d
df_obitos_novos.show()

#3.3 Letalidade
df_letalidade = spark.sql("SELECT cast(((obitosAcumulado * 100) / casosAcumulado) as decima
df_letalidade.show()

#3.4 Mortalidade
df_mortalidade = spark.sql("SELECT cast(((obitosAcumulado * 100000) / populacaoTCU2019) as
df_mortalidade.show()
```

```
+------------+
|Obitos_Novos|
+------------+
|        4249|
+------------+

+----------+
|Letalidade|
+----------+
|    1500.0|
+----------+

+-----------+
|Mortalidade|
+-----------+
|      999.1|
+-----------+
```

Salvar a primeira visualização como tabela Hive

In [19]:

```
df_casos_recuperados.write.format('csv').saveAsTable('Recuperados')
```

In [20]:

```
df_em_acompanhamento.write.format('csv').saveAsTable('Acompanhamento')
```

In [21]:

```
#Visualizar as tabelas salvas
spark.sql('SHOW TABLES').show()
```

```
+--------+--------------+-----------+
|database|     tableName|isTemporary|
+--------+--------------+-----------+
| covidbr|acompanhamento|      false|
| covidbr|     municipio|      false|
| covidbr|   recuperados|      false|
+--------+--------------+-----------+
```

Salvar a segunda visualização com formato parquet e compressão snappy

In [24]:

```
df_casos_acumulados.write.option('compression', 'snappy').parquet('hdfs://namenode/user/mar
```

In [27]:

```
df_casos_novos.write.option('compression', 'snappy').parquet('hdfs://namenode/user/marco/pr

## Ocorreu o erro pois já existia...
```

```
---------------------------------------------------------------------------
Py4JJavaError                             Traceback (most recent call last)
/opt/spark/python/pyspark/sql/utils.py in deco(*a, **kw)
     62         try:
---> 63             return f(*a, **kw)
     64         except py4j.protocol.Py4JJavaError as e:

/opt/spark/python/lib/py4j-0.10.7-src.zip/py4j/protocol.py in get_return_val
ue(answer, gateway_client, target_id, name)
    327                     "An error occurred while calling {0}{1}{2}.\n".
--> 328                     format(target_id, ".", name), value)
    329             else:

Py4JJavaError: An error occurred while calling o104.parquet.
: org.apache.spark.sql.AnalysisException: path hdfs://namenode/user/marco/pr
ojeto-final-spark/visualizacao/Casos_Novos already exists.;
        at org.apache.spark.sql.execution.datasources.InsertIntoHadoopFsRela
tionCommand.run(InsertIntoHadoopFsRelationCommand.scala:114)
        at org.apache.spark.sql.execution.command.DataWritingCommandExec.sid
eEffectResult$lzycompute(commands.scala:104)
        at org.apache.spark.sql.execution.command.DataWritingCommandExec.sid
eEffectResult(commands.scala:102)
        at org.apache.spark.sql.execution.command.DataWritingCommandExec.doE
xecute(commands.scala:122)
        at org.apache.spark.sql.execution.SparkPlan$$anonfun$execute$1.apply
(SparkPlan.scala:131)
        at org.apache.spark.sql.execution.SparkPlan$$anonfun$execute$1.apply
(SparkPlan.scala:127)
        at org.apache.spark.sql.execution.SparkPlan$$anonfun$executeQuery$1.
apply(SparkPlan.scala:155)
        at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationSco
pe.scala:151)
        at org.apache.spark.sql.execution.SparkPlan.executeQuery(SparkPlan.s
cala:152)
        at org.apache.spark.sql.execution.SparkPlan.execute(SparkPlan.scala:
127)
        at org.apache.spark.sql.execution.QueryExecution.toRdd$lzycompute(Qu
eryExecution.scala:80)
        at org.apache.spark.sql.execution.QueryExecution.toRdd(QueryExecutio
n.scala:80)
        at org.apache.spark.sql.DataFrameWriter$$anonfun$runCommand$1.apply
(DataFrameWriter.scala:668)
        at org.apache.spark.sql.DataFrameWriter$$anonfun$runCommand$1.apply
(DataFrameWriter.scala:668)
        at org.apache.spark.sql.execution.SQLExecution$$anonfun$withNewExecu
tionId$1.apply(SQLExecution.scala:78)
        at org.apache.spark.sql.execution.SQLExecution$.withSQLConfPropagate
d(SQLExecution.scala:125)
        at org.apache.spark.sql.execution.SQLExecution$.withNewExecutionId(S
QLExecution.scala:73)
        at org.apache.spark.sql.DataFrameWriter.runCommand(DataFrameWriter.s
cala:668)
        at org.apache.spark.sql.DataFrameWriter.saveToV1Source(DataFrameWrit
er.scala:276)
```

```
        at org.apache.spark.sql.DataFrameWriter.save(DataFrameWriter.scala:2
70)
        at org.apache.spark.sql.DataFrameWriter.save(DataFrameWriter.scala:2
28)
        at org.apache.spark.sql.DataFrameWriter.parquet(DataFrameWriter.scal
a:557)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorI
mpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodA
ccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at py4j.reflection.MethodInvoker.invoke(MethodInvoker.java:244)
        at py4j.reflection.ReflectionEngine.invoke(ReflectionEngine.java:35
7)
        at py4j.Gateway.invoke(Gateway.java:282)
        at py4j.commands.AbstractCommand.invokeMethod(AbstractCommand.java:1
32)
        at py4j.commands.CallCommand.execute(CallCommand.java:79)
        at py4j.GatewayConnection.run(GatewayConnection.java:238)
        at java.lang.Thread.run(Thread.java:748)


During handling of the above exception, another exception occurred:

AnalysisException                         Traceback (most recent call last)
<ipython-input-27-6c5df3bea6ed> in <module>()
----> 1 df_casos_novos.write.option('compression', 'snappy').parquet('hdf
s://namenode/user/marco/projeto-final-spark/visualizacao/Casos_Novos')

/opt/spark/python/pyspark/sql/readwriter.py in parquet(self, path, mode, par
titionBy, compression)
    839             self.partitionBy(partitionBy)
    840         self._set_opts(compression=compression)
--> 841         self._jwrite.parquet(path)
    842
    843     @since(1.6)

/opt/spark/python/lib/py4j-0.10.7-src.zip/py4j/java_gateway.py in __call__(s
elf, *args)
   1255         answer = self.gateway_client.send_command(command)
   1256         return_value = get_return_value(
-> 1257             answer, self.gateway_client, self.target_id, self.name)
   1258
   1259         for temp_arg in temp_args:

/opt/spark/python/pyspark/sql/utils.py in deco(*a, **kw)
     67                                             e.java_exception.getSta
ckTrace()))
     68             if s.startswith('org.apache.spark.sql.AnalysisException:
'):
---> 69                 raise AnalysisException(s.split(': ', 1)[1], stackTr
ace)
     70             if s.startswith('org.apache.spark.sql.catalyst.analysis'
):
     71                 raise AnalysisException(s.split(': ', 1)[1], stackTr
ace)

AnalysisException: 'path hdfs://namenode/user/marco/projeto-final-spark/visu
alizacao/Casos_Novos already exists.;'
```

In [28]:

```
df_incidencia.write.option('compression', 'snappy').parquet('hdfs://namenode/user/marco/pro
```

In [29]:

```
#Visualizar os dodos na pasta do HDFS
!hdfs dfs -ls 'hdfs://namenode/user/marco/projeto-final-spark/visualizacao'
```

```
Found 2 items
drwxr-xr-x   - root supergroup          0 2022-04-27 09:23 hdfs://namenode/u
ser/marco/projeto-final-spark/visualizacao/Casos_Novos
drwxr-xr-x   - root supergroup          0 2022-04-27 09:25 hdfs://namenode/u
ser/marco/projeto-final-spark/visualizacao/Incidencia
```

Salvar a terceira visualização em um tópico no Kafka

⚠ Diferente dos formatos nativos do spark para salvamento direto no HDFS,o Kafka permite apenas salvamento de dados únicos de tipo String em DataFrames de formato JSON. Por isso, iremos convertêlos antes de salvarmos em um tópico.

In [32]:

```
#3.1 Óbitos acumulados
obitos_acumulados_json = df_obitos_acumulados.select(to_json(struct('obitos_Acumulados')).a
obitos_acumulados_json.show(n=1, truncate=False, vertical=False)
```

```
+---------------------------+
|value                      |
+---------------------------+
|{"obitos_Acumulados":526892}|
+---------------------------+
```

In [33]:

```
#3.2 Obitos novos
obitos_novos_json = df_obitos_novos.select(to_json(struct('obitos_Novos')).alias('value'))
obitos_novos_json.show(n=1, truncate=False, vertical=False)
```

```
+--------------------+
|value               |
+--------------------+
|{"obitos_Novos":4249}|
+--------------------+
```

In [34]:

```
#3.3 Letalidade
letalidade_json = df_letalidade.select(to_json(struct('Letalidade')).alias('value'))
letalidade_json.show(n=1, truncate=False, vertical=False)
```

```
+---------------------+
|value                |
+---------------------+
|{"Letalidade":1500.0}|
+---------------------+
```

In [35]:

```
#3.4 Mortalidade
mortalidade_json = df_mortalidade.select(to_json(struct('Mortalidade')).alias('value'))
mortalidade_json.show(n=1, truncate=False, vertical=False)
```

```
+---------------------+
|value                |
+---------------------+
|{"Mortalidade":999.1}|
+---------------------+
```

Salvando os Dataframes convertidos em JSON em tópicos do Kafka

In [36]:

```
df_obitos_acumulados.selectExpr('to_json(struct(*)) As value')\
.write.format('kafka')\
.option('kafka.bootstrap.servers', 'kafka:9092')\
.option('topic', 'topic-obtidos_acumulados')\
.save()
```

In [37]:

```
df_obitos_novos.selectExpr('to_json(struct(*)) As value')\
.write\
.format("kafka")\
.option("kafka.bootstrap.servers","kafka:9092")\
.option("topic","topic-obitos_novos")\
.save()
```

In [38]:

```
df_letalidade.selectExpr('to_json(struct(*)) As value')\
.write\
.format("kafka")\
.option("kafka.bootstrap.servers","kafka:9092")\
.option("topic","topic-letalidade")\
.save()
```

In [39]:

```
df_mortalidade.selectExpr('to_json(struct(*)) As value')\
.write\
.format("kafka")\
.option("kafka.bootstrap.servers","kafka:9092")\
.option("topic","topic-mortalidade")\
.save()
```

Criar a visualização pelo Spark com os dados enviados para o HDFS

In [68]:

```
df_spark = spark.sql('''SELECT regiao as Regiao,
                    MAX(casosAcumulado) as Casos,
                    MAX(obitosAcumulado) as Obitos,
                    MAX(data) as Atualizacao
                    FROM municipio
                    GROUP BY regiao
                    ORDER BY regiao''')

df_spark.show()

df_spark.printSchema()
```

```
+------------+--------+------+-----------+
|      Regiao|   Casos|Obitos|Atualizacao|
+------------+--------+------+-----------+
|      Brasil|18855015|526892| 2021-07-06|
|Centro-Oeste|  686433| 19485| 2021-07-06|
|    Nordeste| 1141612| 24428| 2021-07-06|
|       Norte|  557708| 15624| 2021-07-06|
|     Sudeste| 3809222|130389| 2021-07-06|
|         Sul| 1308643| 31867| 2021-07-06|
+------------+--------+------+-----------+

root
 |-- Regiao: string (nullable = true)
 |-- Casos: decimal(10,0) (nullable = true)
 |-- Obitos: integer (nullable = true)
 |-- Atualizacao: string (nullable = true)
```

In [48]:

```
df_spark.limit(10).toPandas()
```

Out[48]:

	Regiao	Casos	Obitos	Atualizacao
0	Brasil	18855015	526892	2021-07-06
1	Centro-Oeste	686433	19485	2021-07-06
2	Nordeste	1141612	24428	2021-07-06
3	Norte	557708	15624	2021-07-06
4	Sudeste	3809222	130389	2021-07-06
5	Sul	1308643	31867	2021-07-06

In [ ]:

```

```

Fontes de consultas: pyspark.sql module: https://spark.apache.org/docs/2.0.2/api/python/pyspark.sql.html#module-pyspark.sql.functions (https://spark.apache.org/docs/2.0.2/api/python/pyspark.sql.html#module-pyspark.sql.functions) Spark SQL, DataFrames and Datasets Guide: https://spark.apache.org/docs/2.0.2/sql-programming-guide.html#sql (https://spark.apache.org/docs/2.0.2/sql-programming-guide.html#sql) Spark SQL Guide: https://spark.apache.org/docs/latest/sql-getting-started.html (https://spark.apache.org/docs/latest/sql-getting-started.html) Como obter no SQL o último registro inserido de acordo com sua data: https://pt.stackoverflow.com/questions/13125/como-obter-no-sql-o-%C3%BAltimo-registro-inserido-de-acordo-com-sua-data (https://pt.stackoverflow.com/questions/13125/como-obter-no-sql-o-%C3%BAltimo-registro-inserido-de-acordo-com-sua-data) Sobre os dados do Coronavirus: https://www.coronavirus.sc.gov.br/sobre-os-dados/ (https://www.coronavirus.sc.gov.br/sobre-os-dados/) Projeção da população do Brasil: https://www.ibge.gov.br/apps/populacao/projecao/index.html (https://www.ibge.gov.br/apps/populacao/projecao/index.html) CAST para DECIMAL no MySQL: https://stackoverflow.com/questions/11830509/cast-to-decimal-in-mysql (https://stackoverflow.com/questions/11830509/cast-to-decimal-in-mysql) Tutorial: Usar o fluxo estruturado do Apache Spark com o Apache Kafka no HDInsight: https://docs.microsoft.com/pt-br/azure/hdinsight/hdinsight-apache-kafka-spark-structured-streaming (https://docs.microsoft.com/pt-br/azure/hdinsight/hdinsight-apache-kafka-spark-structured-streaming)

In [ ]:

```

```