Gabriel Gudiño / Yamila Orrego

1) En el container de Nifi, crear un .sh que permita descargar el archivo de https://data-engineer-edvai.s3.amazonaws.com/yellow_tripdata_2021-01.parquet y lo guarde en /home/nifi/ingest. Ejecutarlo

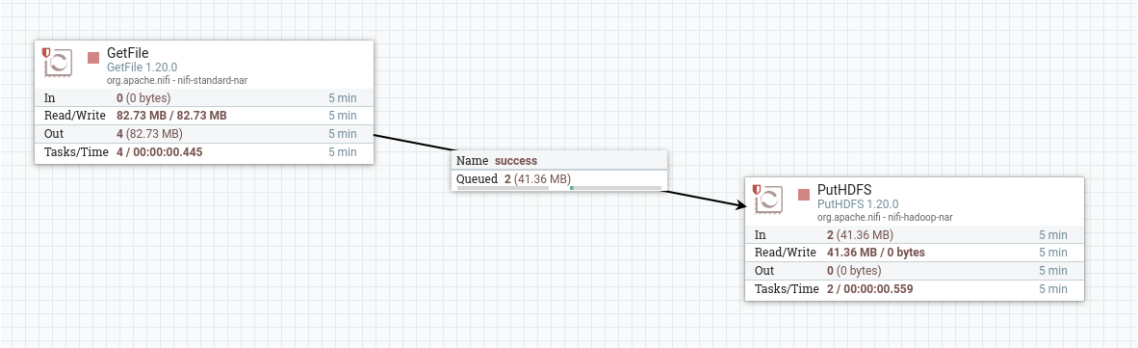
```
nifi@1b48bdded3ce:~/ingest/tripdata$ sh ingest.sh
--2023-04-11 20:31:37--  https://data-engineer-edvai.s3.amazonaws.com/yellow_tri
pdata_2021-01.parquet
Resolving data-engineer-edvai.s3.amazonaws.com (data-engineer-edvai.s3.amazonaws
.com)... 52.217.204.81, 52.216.143.68, 3.5.17.191, ...
Connecting to data-engineer-edvai.s3.amazonaws.com (data-engineer-edvai.s3.amazo
naws.com)|52.217.204.81|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 21686067 (21M) [application/x-www-form-urlencoded]
Saving to: '/home/nifi/ingest/tripdata/yellow_tripdata_2021-01.parquet'

yellow_tripdata_2021-01.parquet                     100%[===============
================================================================================
==========================================>]  20.68M  40.9MB/s    in 0.5s

2023-04-11 20:31:37 (40.9 MB/s) - '/home/nifi/ingest/tripdata/yellow_tripdata_20
21-01.parquet' saved [21686067/21686067]

nifi@1b48bdded3ce:~/ingest/tripdata$ ls
ingest.sh  yellow_tripdata_2021-01.parquet
nifi@1b48bdded3ce:~/ingest/tripdata$
```

2) Por medio de la interfaz gráfica de Nifi, crear un job que tenga dos procesos. a) GetFile para obtener el archivo del punto 1 (/home/nifi/ingest) b) putHDFS para ingestarlo a HDFS (directorio nifi)




```
hadoop@0485e86b7577:~$ hdfs dfs -ls /nifi
Found 2 items
-rw-r--r--   1 nifi supergroup     182386 2023-04-11 17:43 /nifi/starwars.csv
-rw-r--r--   1 nifi supergroup   21686067 2023-04-11 18:08 /nifi/yellow_tripdata_2021-01.parquet
hadoop@0485e86b7577:~$
```

3) Con el archivo ya ingestado en HDFS/nifi, escribir las consultas y agregar captura de pantalla del resultado. Para los ejercicios puedes usar SQL mediante la creación de una vista llamada yellow_tripdata. También debes chequear el diccionario de datos por cualquier duda que tengas respecto a las columnas del archivo https://www.nyc.gov/assets/tlc/downloads/pdf/data_dictionary_trip_records_yellow.pdf

3.1) Mostrar los resultados siguientes a. VendorId Integer b. Tpep_pickup_datetime date c. Total_amount double d. Donde el total (total_amount sea menor a 10 dólares)

```
>>> df2 = spark.sql("select VendorID, tpep_pickup_datetime, total_amount from tripdata WHERE total_amount < 10")
>>> df2.show()
+--------+--------------------+------------+
|VendorID|tpep_pickup_datetime|total_amount|
+--------+--------------------+------------+
|       1| 2020-12-31 21:51:20|         4.3|
|       2| 2020-12-31 21:42:11|         8.3|
|       2| 2020-12-31 21:04:21|        9.96|
|       2| 2020-12-31 21:43:41|         9.3|
|       2| 2020-12-31 21:36:08|         5.8|
|       1| 2020-12-31 21:03:13|         0.0|
|       1| 2020-12-31 21:30:32|         9.3|
|       2| 2020-12-31 21:16:19|         9.8|
|       2| 2020-12-31 21:57:26|         8.8|
|       2| 2020-12-31 21:33:33|        9.96|
|       2| 2020-12-31 21:39:08|         9.3|
|       2| 2020-12-31 21:45:29|         7.8|
|       1| 2020-12-31 21:43:00|        9.55|
|       2| 2020-12-31 21:52:28|         4.8|
|       1| 2020-12-31 21:39:50|         9.8|
|       2| 2020-12-31 21:17:39|         8.8|
|       1| 2020-12-31 21:49:10|         7.8|
|       2| 2020-12-31 21:47:51|        9.36|
|       2| 2020-12-31 21:11:53|         8.3|
|       1| 2020-12-31 21:34:38|         9.3|
+--------+--------------------+------------+
only showing top 20 rows

>>>
```

3.2) Mostrar los 10 días que más se recaudó dinero (tpep_pickup_date time, total_amount)

```
>>> df3 = spark.sql("select CAST(tpep_pickup_datetime AS date) tpep_pickup_date,SUM
( total_amount) total from tripdata GROUP BY tpep_pickup_date ORDER BY total DESC L
IMIT 10")
>>> df3.show()
+----------------+-----------------+
|tpep_pickup_date|            total|
+----------------+-----------------+
|      2021-01-28|961322.5600002451|
|      2021-01-22|942205.9300002148|
|      2021-01-29|937373.5100002222|
|      2021-01-21|932444.4500002082|
|      2021-01-15|931628.1900002063|
|      2021-01-14|926664.0400001821|
|      2021-01-27|  895259.87000017|
|      2021-01-19|890581.4500001629|
|      2021-01-07|887670.1600001527|
|      2021-01-08| 878002.730000146|
+----------------+-----------------+
```

3.3) Mostrar los 10 viajes que menos dinero recaudó en viajes mayores a 10 millas (trip_distance, total_amount)

```
>>> df4 = spark.sql("select trip_distance, total_amount from tripdata WHERE trip_distance > 10 ORDER BY total_amount limit 10")
>>> df4.show()
+-------------+------------+
|trip_distance|total_amount|
+-------------+------------+
|        12.68|      -252.3|
|        34.35|     -176.42|
|        14.75|      -152.8|
|        33.96|     -127.92|
|         29.1|      -119.3|
|        26.94|      -111.3|
|        20.08|      -107.8|
|        19.55|      -102.8|
|        19.16|      -90.55|
|        25.83|      -88.54|
+-------------+------------+
```

3.4) Mostrar los viajes de más de dos pasajeros que hayan pagado con tarjeta de crédito (mostrar solo las columnas trip_distance y tpep_pickup_datetime)

```
>>> df5 = spark.sql("select trip_distance,CAST(tpep_pickup_datetime as DATE) from tripda
ta WHERE passenger_count >= 2")
>>> df5.show()
+-------------+--------------------+
|trip_distance|tpep_pickup_datetime|
+-------------+--------------------+
|          2.7|          2020-12-31|
|         6.11|          2020-12-31|
|         1.21|          2020-12-31|
|          7.4|          2020-12-31|
|          1.7|          2020-12-31|
|         0.81|          2020-12-31|
|         1.66|          2020-12-31|
|         0.93|          2020-12-31|
|         1.16|          2020-12-31|
|         19.1|          2020-12-31|
|         3.15|          2020-12-31|
|         0.64|          2020-12-31|
|          3.0|          2020-12-31|
|        10.74|          2020-12-31|
|          1.0|          2020-12-31|
|         2.01|          2020-12-31|
|         3.45|          2020-12-31|
|         2.85|          2020-12-31|
|         1.68|          2020-12-31|
|         0.77|          2020-12-31|
+-------------+--------------------+
only showing top 20 rows
```

3.5) Mostrar los 7 viajes con mayor propina en distancias mayores a 10 millas (mostrar campos tpep_pickup_datetime, trip_distance, passenger_count, tip_amount)

```
>>> df6 = spark.sql("select trip_distance, CAST(tpep_pickup_datetime as DATE), passenger
_count, tip_amount from tripdata WHERE trip_distance > 10 ORDER BY tip_amount DESC LIMIT
 7")
>>> df6.show()
+-------------+--------------------+---------------+----------+
|trip_distance|tpep_pickup_datetime|passenger_count|tip_amount|
+-------------+--------------------+---------------+----------+
|        427.7|          2021-01-20|            1.0|   1140.44|
|        267.7|          2021-01-03|            1.0|     369.4|
|        326.1|          2021-01-12|            0.0|    192.61|
|        260.5|          2021-01-19|            1.0|    149.03|
|         11.1|          2021-01-31|            0.0|     100.0|
|        14.86|          2021-01-01|            2.0|      99.0|
|         13.0|          2021-01-18|            0.0|      90.0|
+-------------+--------------------+---------------+----------+
```

3.6) Mostrar para cada uno de los valores de RateCodeID, el monto total y el monto promedio. Excluir los viajes en donde RateCodeID es ‘Group Ride’

```
>>> df7 = spark.sql("select RateCodeID,SUM(total_amount), AVG(total_amount) from tripdat
a WHERE RateCodeID != 6 GROUP BY RateCodeID")
>>> df7.show()
+----------+--------------------+------------------+
|RateCodeID|   sum(total_amount)| avg(total_amount)|
+----------+--------------------+------------------+
|       1.0|1.9496468430212937E7|15.606626116946773|
|       4.0|   90039.93000000082| 74.90842762063296|
|       3.0|   67363.26000000043| 78.69539719626219|
|       2.0|   973635.4700000732| 65.52937609369182|
|      99.0|  1748.0699999999997| 48.55749999999999|
|       5.0|  255075.08999999086|48.939963545662096|
+----------+--------------------+------------------+
```