1. En el container de Nifi, crear un .sh que permita descargar el archivo yellow_tripdata_2021-01.parquet

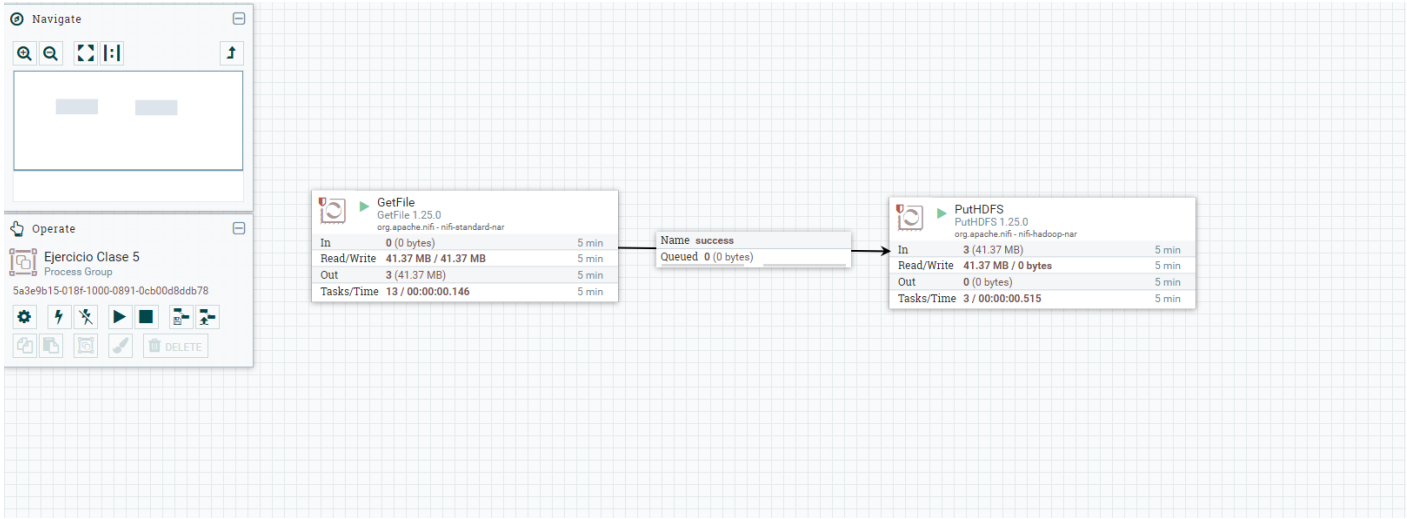
```
nifi@332d81692629: ~/ingest
nifi@332d81692629:~/ingest$ ls
ingest_Parquet.sh  ingest.sh  starwars.csv
nifi@332d81692629:~/ingest$ ./ingest_Parquet.sh
bash: ./ingest_Parquet.sh: Permission denied
nifi@332d81692629:~/ingest$ chmod 777 ingest_Parquet.sh
nifi@332d81692629:~/ingest$ ls -l
total 16
-rwxrwxrwx 1 nifi nifi  153 May  8 21:59 ingest_Parquet.sh
-rwxr--r-- 1 nifi nifi  100 Apr 26 02:07 ingest.sh
-rw-r--r-- 1 nifi nifi 5462 Apr 26 02:11 starwars.csv
nifi@332d81692629:~/ingest$ ./ingest_Parquet.sh
--2024-05-08 22:02:28--  https://dataengineerpublic.blob.core.windows.net/data-engineer/yellow_tripdata_2021-01.parquet
Resolving dataengineerpublic.blob.core.windows.net (dataengineerpublic.blob.core.windows.net)... 20.150.25.164
Connecting to dataengineerpublic.blob.core.windows.net (dataengineerpublic.blob.core.windows.net)|20.150.25.164|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 21686067 (21M) [application/octet-stream]
Saving to: '/home/nifi/ingest/yellow_tripdata_2021-01.parquet'

/home/nifi/ingest/yellow_tripdata_2021-0 100%[=========================================================================>]  20.68M  4.71MB/s    in 5.9s

2024-05-08 22:02:35 (3.49 MB/s) - '/home/nifi/ingest/yellow_tripdata_2021-01.parquet' saved [21686067/21686067]

nifi@332d81692629:~/ingest$ ls
ingest_Parquet.sh  ingest.sh  starwars.csv  yellow_tripdata_2021-01.parquet
nifi@332d81692629:~/ingest$
```

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@d41c15beb563: ~/hadoop
hadoop@d41c15beb563:~/hadoop$ hdfs dfs -ls /nifi
Found 1 items
-rw-r--r--   1 nifi supergroup   21686067 2024-05-08 19:47 /nifi/yellow_tripdata_2021-01.parquet
hadoop@d41c15beb563:~/hadoop$
```

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)

```
>>> df_consultas = spark.sql("select vendorId, cast(tpep_pickup_datetime as date), total_amount from yellow_tripdata where total_amount < 10 ")
>>> df_consultas.show(10)
+--------+--------------------+------------+
|vendorId|tpep_pickup_datetime|total_amount|
+--------+--------------------+------------+
|       1|          2020-12-31|         4.3|
|       2|          2020-12-31|         8.3|
|       2|          2020-12-31|        9.96|
|       2|          2020-12-31|         9.3|
|       2|          2020-12-31|         5.8|
|       1|          2020-12-31|         0.0|
|       1|          2020-12-31|         9.3|
|       2|          2020-12-31|         9.8|
|       2|          2020-12-31|         8.8|
|       2|          2020-12-31|        9.96|
+--------+--------------------+------------+
only showing top 10 rows
```

3.2) Mostrar los 10 días que más se recaudó dinero (tpep_pickup_datetime, total amount)

```
>>> df_consultas = spark.sql("select cast(tpep_pickup_datetime as date), sum(total_amount) from yellow_tripdata group by tpep_pickup_datetime order by sum(total_amount) desc")
>>> df_consultas.show(10)
+--------------------+------------------+
|tpep_pickup_datetime| sum(total_amount)|
+--------------------+------------------+
|          2021-01-04|           7661.28|
|          2021-01-20|           2310.28|
|          2021-01-12|1164.6200000000001|
|          2021-01-03|           1118.43|
|          2021-01-10|            913.65|
|          2021-01-19|             894.2|
|          2021-01-06|            872.54|
|          2021-01-06|            872.05|
|          2021-01-03|            863.67|
|          2021-01-27|             836.3|
+--------------------+------------------+
only showing top 10 rows
```

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

```
>>> df_consultas = spark.sql("select trip_distance, total_amount as total from yellow_tripdata where trip_distance > 10 order by total_amount asc limit 10")
>>> df_consultas.show()
+-------------+-------+
|trip_distance|  total|
+-------------+-------+
|        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)

```
>>> df_consultas = spark.sql("select trip_distance, cast(tpep_pickup_datetime as date) from yellow_tripdata where payment_type = 1 and passenger_count > 2")
>>> df_consultas.show(10)
+-------------+--------------------+
|trip_distance|tpep_pickup_datetime|
+-------------+--------------------+
|         6.11|          2020-12-31|
|          1.7|          2020-12-31|
|         3.15|          2020-12-31|
|        10.74|          2020-12-31|
|         2.01|          2020-12-31|
|         2.85|          2020-12-31|
|         1.68|          2020-12-31|
|         0.77|          2020-12-31|
|          0.4|          2020-12-31|
|        16.54|          2020-12-31|
+-------------+--------------------+
only showing top 10 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)

```
>>> df_consultas = spark.sql("select trip_distance,cast(tpep_pickup_datetime as date), cast(passenger_count as integer), tip_amount from yellow_tripdata where trip_distance > 10 order by tip_amount desc")
>>> df_consultas.show(7)
+-------------+--------------------+---------------+----------+
|trip_distance|tpep_pickup_datetime|passenger_count|tip_amount|
+-------------+--------------------+---------------+----------+
|        427.7|          2021-01-20|              1|   1140.44|
|        267.7|          2021-01-03|              1|     369.4|
|        326.1|          2021-01-12|              0|    192.61|
|        260.5|          2021-01-19|              1|    149.03|
|         11.1|          2021-01-31|              0|     100.0|
|        14.86|          2021-01-01|              2|      99.0|
|         13.0|          2021-01-18|              0|      90.0|
+-------------+--------------------+---------------+----------+
only showing top 7 rows
```


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'

```
>>> df_consultas = spark.sql("select RateCodeID, sum(total_amount), avg(total_amount) from yellow_tripdata where RateCodeID != 6 group by RateCodeID")
>>> df_consultas.show(10)
+----------+--------------------+------------------+
|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|
+----------+--------------------+------------------+
```