

1) Creacion del script en una carpeta nueva: home/nifi/scripts

```
nifi@468dca6ff6f6:~$ ls
hdfs ingest scripts
nifi@468dca6ff6f6:~$ cd scripts
nifi@468dca6ff6f6:~/scripts$ cat > descargar_datos.sh << 'EOF'
#!/bin/bash
# Script para descargar datos en el directorio de ingesta de NiFi

echo "Iniciando la descarga del archivo yellow_tripdata_2021-01.parquet con WGET..."

# 1. Asegura que el directorio de destino exista
mkdir -p /home/nifi/ingest

# 2. Comando de descarga: guarda el archivo directamente en /home/nifi/ingest/
wget -O /home/nifi/ingest/yellow_tripdata_2021-01.parquet https://data-engineer-edvai-public.s3.amazonaws.com/yellow_tripdata_2021-01.parquet

if [ $? -eq 0 ]; then
    echo "Descarga completada con éxito."
    echo "Archivo guardado en: /home/nifi/ingest/yellow_tripdata_2021-01.parquet"
else
    echo "Error durante la descarga."
fi
EOF
```

Agrego permisos de ejecución al script

```
nifi@468dca6ff6f6:~/scripts$ chmod +x descargar_datos.sh
```

Realizo la descarga:

```
nifi@468dca6ff6f6:~/scripts$ ./descargar_datos.sh
Iniciando la descarga del archivo yellow_tripdata_2021-01.parquet con WGET...
--2025-10-07 17:22:36-- https://data-engineer-edvai-public.s3.amazonaws.com/yellow_tripdata_2021-01.parquet
Resolving data-engineer-edvai-public.s3.amazonaws.com (data-engineer-edvai-public.s3.amazonaws.com)... 52.216.35.113, 52.217.86.164, 54.231.129.249
...
Connecting to data-engineer-edvai-public.s3.amazonaws.com (data-engineer-edvai-public.s3.amazonaws.com)|52.216.35.113|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 21686067 (21M) [application/vnd.apache.parquet]
Saving to: '/home/nifi/ingest/yellow_tripdata_2021-01.parquet'

/home/nifi/ingest/yellow_tripdata_20 100%[=====] 20.68M 1.12MB/s in 28s

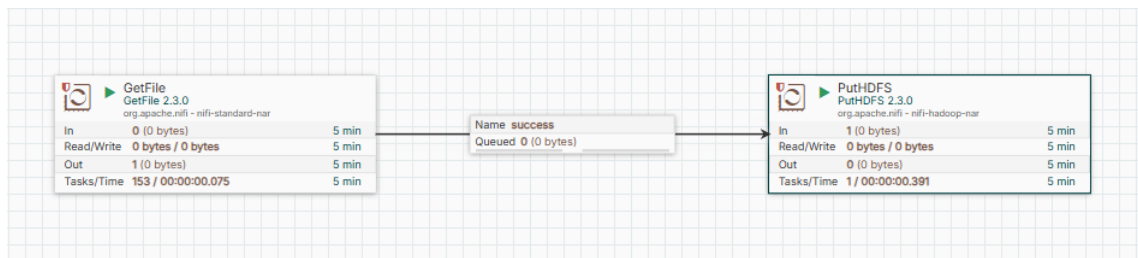
2025-10-07 17:23:05 (769 KB/s) - '/home/nifi/ingest/yellow_tripdata_2021-01.parquet' saved [21686067/21686067]

Descarga completada con éxito.
Archivo guardado en: /home/nifi/ingest/yellow_tripdata_2021-01.parquet
```

Compruebo que la descarga fue exitosa:

```
nifi@468dca6ff6f6:~$ cd ingest
nifi@468dca6ff6f6:~/ingest$ ls
yellow_tripdata_2021-01.parquet
nifi@468dca6ff6f6:~/ingest$ |
```

2) Creo el job en nifi:










Processor Details | GetFile 2.3.0

Settings

Scheduling

Properties

Required field

Property	Value
Input Directory	 /home/nifi/ingest
File Filter	 yellow_tripdata_2021-01.parquet
Path Filter	 No value set
Batch Size	 10
Keep Source File	 false
Recurse Subdirectories	 true
Polling Interval	 0 sec








Processor Details | PutHDFS 2.3.0

Settings

Scheduling

Properties

Required field

Property	Value
Hadoop Configuration Resources	 /home/nifi/hdfs/core-site.xml, /home/...
Kerberos User Service	 No value set
Additional Classpath Resources	 No value set
Directory	 /nifi
Conflict Resolution Strategy	 replace
Writing Strategy	 Write and rename
Block Size	 No value set

Podemos ver que se descargo correctamente:

Browse Directory

/nifi

Go!

Show

25

entries

Search:

<input type="checkbox"/>	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
<input type="checkbox"/>	-rw-r--r--	nifi	supergroup	0 B	Oct 07 14:56	1	128 MB	yellow_tripdata_2021-01.parquet	

3)

```
>>> df = spark.read.option("header", "true").parquet("/nifi/yellow_tripdata_2021-01.parquet")
>>> df.p
df.persist(      df.printSchema(
>>> df.printSchema()
root
 |-- VendorID: long (nullable = true)
 |-- tpep_pickup_datetime: timestamp (nullable = true)
 |-- tpep_dropoff_datetime: timestamp (nullable = true)
 |-- passenger_count: double (nullable = true)
 |-- trip_distance: double (nullable = true)
 |-- RatecodeID: double (nullable = true)
 |-- store_and_fwd_flag: string (nullable = true)
 |-- PULocationID: long (nullable = true)
 |-- DOLocationID: long (nullable = true)
 |-- payment_type: long (nullable = true)
 |-- fare_amount: double (nullable = true)
 |-- extra: double (nullable = true)
 |-- mta_tax: double (nullable = true)
 |-- tip_amount: double (nullable = true)
 |-- tolls_amount: double (nullable = true)
 |-- improvement_surcharge: double (nullable = true)
 |-- total_amount: double (nullable = true)
 |-- congestion_surcharge: double (nullable = true)
 |-- airport_fee: double (nullable = true)

>>> df.createOrReplaceTempView("v_df")
```

3.1 A

```
>>> df_select = spark.sql("select VendorID,tpep_pickup_datetime,total_amount, case when total_amount < 10 then total_amount else null end as total_
amount_menor10 from v_df")
>>> df_select.printSchema()
root
 |-- VendorID: long (nullable = true)
 |-- tpep_pickup_datetime: timestamp (nullable = true)
 |-- total_amount: double (nullable = true)
 |-- total_amount_menor10: double (nullable = true)

>>> df_select.show(5)
+-----+-----+-----+-----+
|VendorID|tpep_pickup_datetime|total_amount|total_amount_menor10|
+-----+-----+-----+-----+
|1|2020-12-31 21:30:10|11.8|null|
|1|2020-12-31 21:51:20|4.3|4.3|
|1|2020-12-31 21:43:30|51.95|null|
|1|2020-12-31 21:15:48|36.35|null|
|2|2020-12-31 21:31:49|24.36|null|
+-----+-----+-----+-----+
only showing top 5 rows
```

B

```
>>> df.createOrReplaceTempView("v_df")
>>> df_select = spark.sql("select VendorID,CAST(tpep_pickup_datetime as DATE),total_amount from v_df where total_amount < 10")
>>> df_select.show(5)
+-----+-----+-----+
|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|
+-----+-----+-----+
only showing top 5 rows
```

3.2

```
>>> df_top10 = spark.sql("select CAST(tpep_pickup_datetime as DATE),round(SUM(total_amount),2) as total_day from v_df group by CAST(tpep_pickup_dat
etime as DATE) order by total_day DESC limit 10")
>>> df_top10.show()
+-----+-----+
|tpep_pickup_datetime|total_day|
+-----+-----+
|2021-01-28|961322.56|
|2021-01-22|942205.93|
|2021-01-29|937373.51|
|2021-01-21|932444.45|
|2021-01-15|931628.19|
|2021-01-14|926664.04|
|2021-01-27|895259.87|
|2021-01-19|890581.45|
|2021-01-07|887670.16|
|2021-01-08|878002.73|
+-----+-----+
```

3.3

```
>>> df_10viajes = spark.sql("select trip_distance, total_amount from v_df where trip_distance > 10 order by total_amount ASC limit 10")
>>> df_10viajes.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

```
>>> df_2pass = spark.sql("select trip_distance, CAST(tpep_pickup_datetime as DATE) from v_df where passenger_count > 2 and payment_type = 1")
>>> df_2pass.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

```
>>> df_7viajes = spark.sql("select trip_distance, CAST(tpep_pickup_datetime as DATE), passenger_count, tip_amount from v_df where trip_distance > 10 order by tip_amount DESC limit 7")
>>> df_7viajes.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

```
>>> df_rate = spark.sql("select RateCodeID, round(SUM(total_amount),2) as monto_total, round(avg(total_amount),2) as monto_promedio from v_df where RateCodeID != 6 group by RateCodeID")
>>> df_rate.show()
```

RateCodeID	monto_total	monto_promedio
1.0	1.949646843E7	15.61
4.0	90039.93	74.91
3.0	67363.26	78.7
2.0	973635.47	65.53
99.0	1748.07	48.56
5.0	255075.09	48.94