Joy Steven Castañeda Mancera – 63907

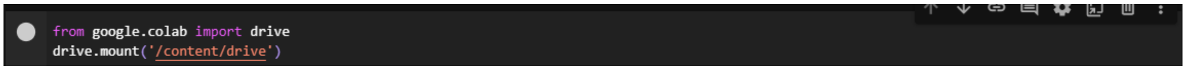
Andres Camilo Velasquez Contreras – 63111

Laboratorio Comprensión de los Datos

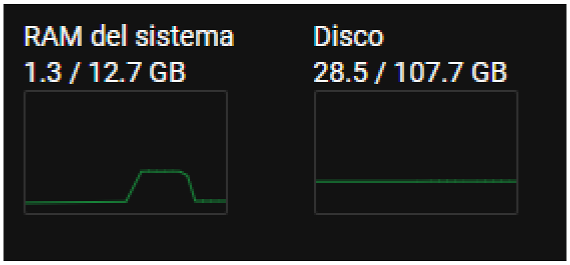
Objetivo Laboratorio

Comparar el desempeño de librerías de Python para carga y manipulación de datos tabulares.

Desarrollo Laboratorio

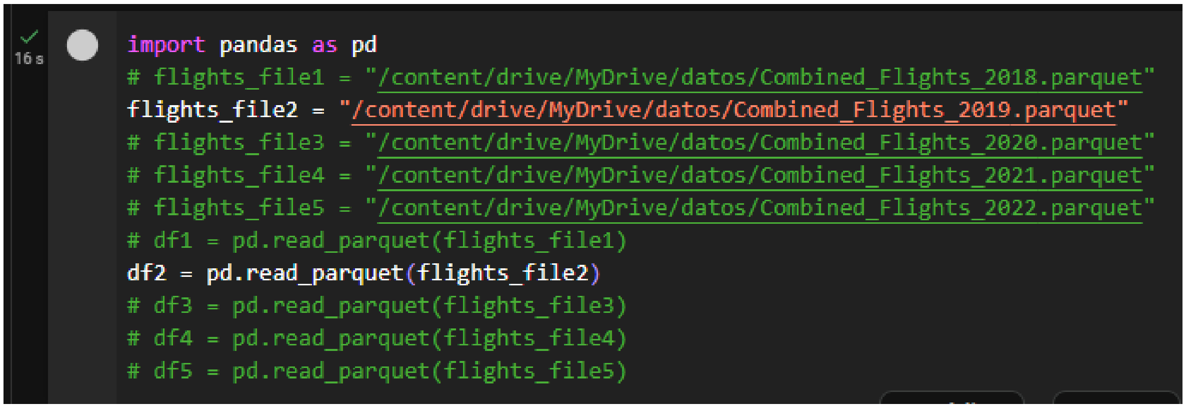


```
from google.colab import drive
drive.mount('/content/drive')
```

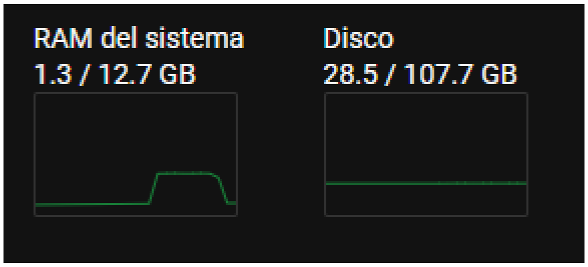


RAM del sistema
1.3 / 12.7 GB

Disco
28.5 / 107.7 GB

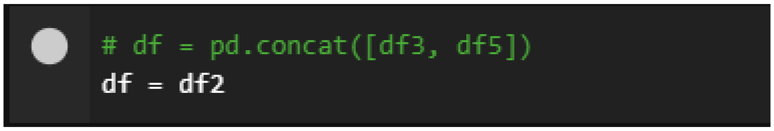


```
import pandas as pd
# flights_file1 = "/content/drive/MyDrive/datos/Combined_Flights_2018.parquet"
flights_file2 = "/content/drive/MyDrive/datos/Combined_Flights_2019.parquet"
# flights_file3 = "/content/drive/MyDrive/datos/Combined_Flights_2020.parquet"
# flights_file4 = "/content/drive/MyDrive/datos/Combined_Flights_2021.parquet"
# flights_file5 = "/content/drive/MyDrive/datos/Combined_Flights_2022.parquet"
# df1 = pd.read_parquet(flights_file1)
df2 = pd.read_parquet(flights_file2)
# df3 = pd.read_parquet(flights_file3)
# df4 = pd.read_parquet(flights_file4)
# df5 = pd.read_parquet(flights_file5)
```

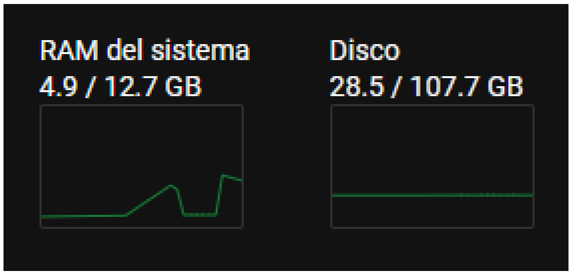


RAM del sistema
1.3 / 12.7 GB

Disco
28.5 / 107.7 GB

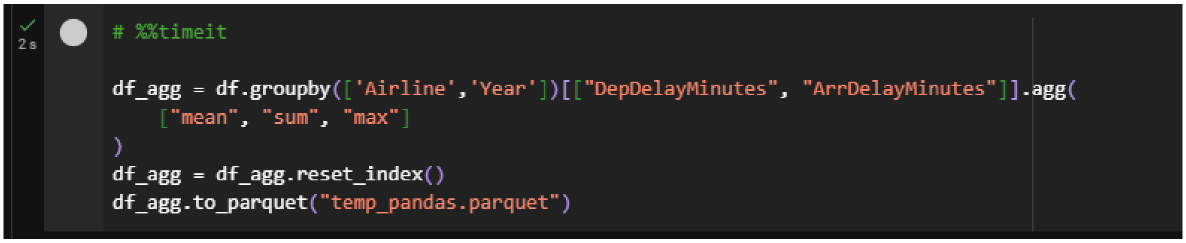


```python
# df = pd.concat([df3, df5])
df = df2
```

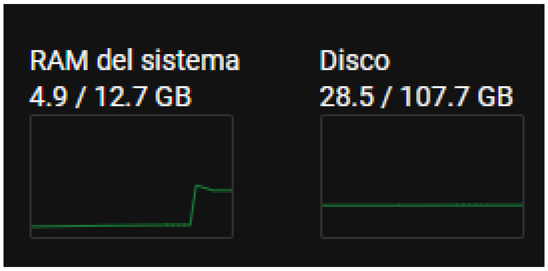


RAM del sistema
4.9 / 12.7 GB

Disco
28.5 / 107.7 GB

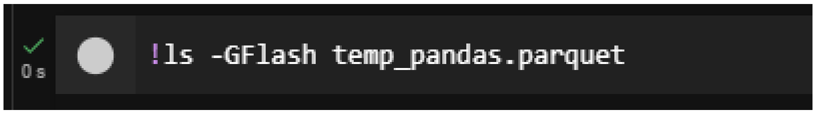


```python
# %%timeit

df_agg = df.groupby(['Airline','Year'])[["DepDelayMinutes", "ArrDelayMinutes"]].agg(
    ["mean", "sum", "max"]
)
df_agg = df_agg.reset_index()
df_agg.to_parquet("temp_pandas.parquet")
```

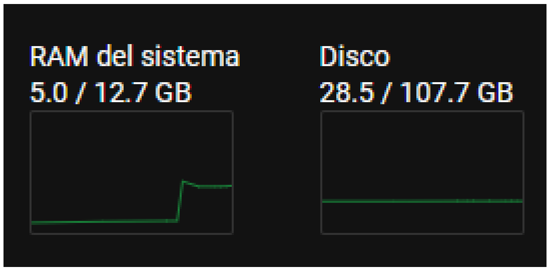


RAM del sistema
4.9 / 12.7 GB

Disco
28.5 / 107.7 GB

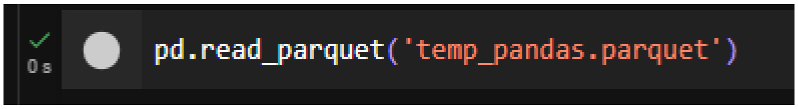


```
!ls -GFlash temp_pandas.parquet
```

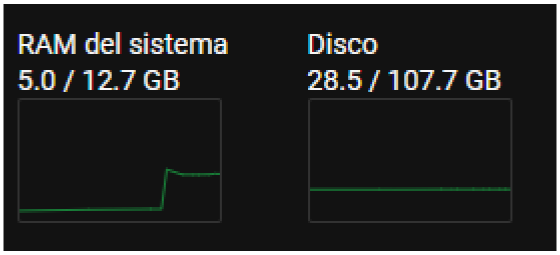


RAM del sistema
5.0 / 12.7 GB

Disco
28.5 / 107.7 GB

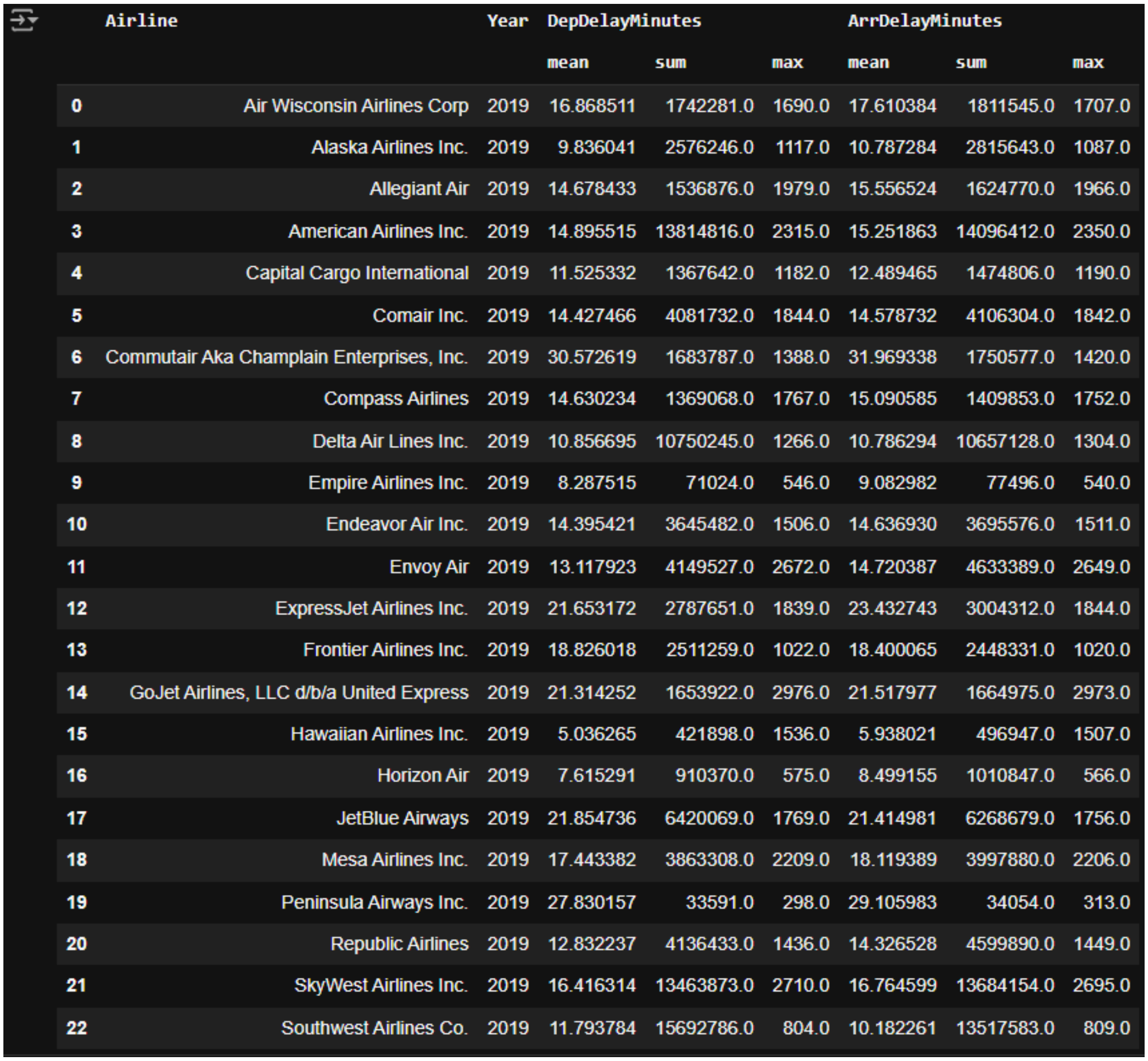


```
pd.read_parquet('temp_pandas.parquet')
```

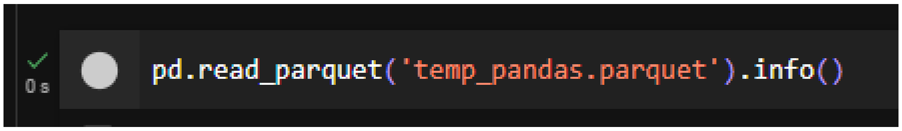


RAM del sistema
5.0 / 12.7 GB

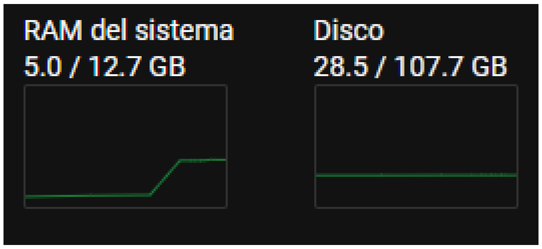
Disco
28.5 / 107.7 GB



	Airline	Year	DepDelayMinutes			ArrDelayMinutes		
			mean	sum	max	mean	sum	max
0	Air Wisconsin Airlines Corp	2019	16.868511	1742281.0	1690.0	17.610384	1811545.0	1707.0
1	Alaska Airlines Inc.	2019	9.836041	2576246.0	1117.0	10.787284	2815643.0	1087.0
2	Allegiant Air	2019	14.678433	1536876.0	1979.0	15.556524	1624770.0	1966.0
3	American Airlines Inc.	2019	14.895515	13814816.0	2315.0	15.251863	14096412.0	2350.0
4	Capital Cargo International	2019	11.525332	1367642.0	1182.0	12.489465	1474806.0	1190.0
5	Comair Inc.	2019	14.427466	4081732.0	1844.0	14.578732	4106304.0	1842.0
6	Commutair Aka Champlain Enterprises, Inc.	2019	30.572619	1683787.0	1388.0	31.969338	1750577.0	1420.0
7	Compass Airlines	2019	14.630234	1369068.0	1767.0	15.090585	1409853.0	1752.0
8	Delta Air Lines Inc.	2019	10.856695	10750245.0	1266.0	10.786294	10657128.0	1304.0
9	Empire Airlines Inc.	2019	8.287515	71024.0	546.0	9.082982	77496.0	540.0
10	Endeavor Air Inc.	2019	14.395421	3645482.0	1506.0	14.636930	3695576.0	1511.0
11	Envoy Air	2019	13.117923	4149527.0	2672.0	14.720387	4633389.0	2649.0
12	ExpressJet Airlines Inc.	2019	21.653172	2787651.0	1839.0	23.432743	3004312.0	1844.0
13	Frontier Airlines Inc.	2019	18.826018	2511259.0	1022.0	18.400065	2448331.0	1020.0
14	GoJet Airlines, LLC d/b/a United Express	2019	21.314252	1653922.0	2976.0	21.517977	1664975.0	2973.0
15	Hawaiian Airlines Inc.	2019	5.036265	421898.0	1536.0	5.938021	496947.0	1507.0
16	Horizon Air	2019	7.615291	910370.0	575.0	8.499155	1010847.0	566.0
17	JetBlue Airways	2019	21.854736	6420069.0	1769.0	21.414981	6268679.0	1756.0
18	Mesa Airlines Inc.	2019	17.443382	3863308.0	2209.0	18.119389	3997880.0	2206.0
19	Peninsula Airways Inc.	2019	27.830157	33591.0	298.0	29.105983	34054.0	313.0
20	Republic Airlines	2019	12.832237	4136433.0	1436.0	14.326528	4599890.0	1449.0
21	SkyWest Airlines Inc.	2019	16.416314	13463873.0	2710.0	16.764599	13684154.0	2695.0
22	Southwest Airlines Co.	2019	11.793784	15692786.0	804.0	10.182261	13517583.0	809.0

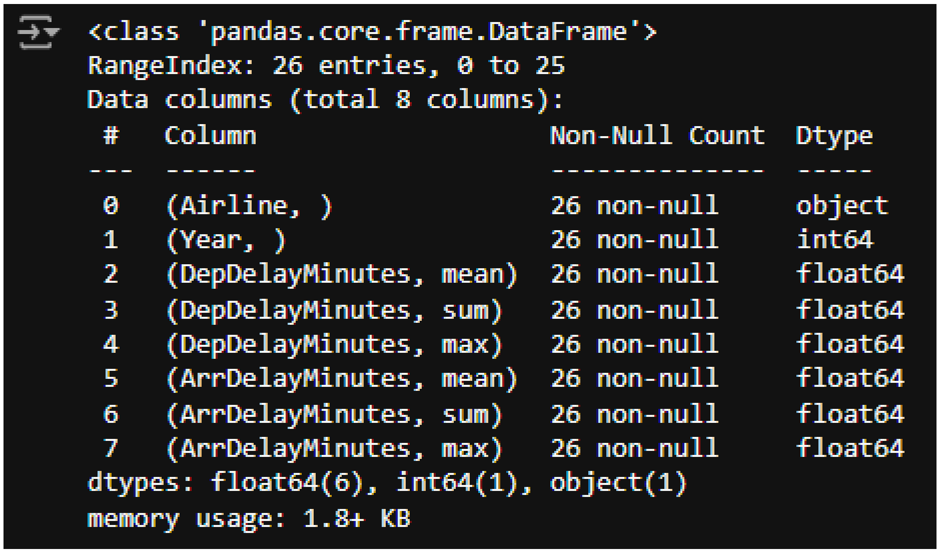


```python
pd.read_parquet('temp_pandas.parquet').info()
```

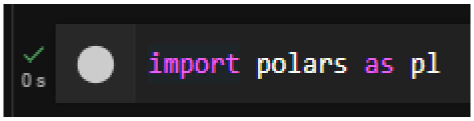


RAM del sistema
5.0 / 12.7 GB

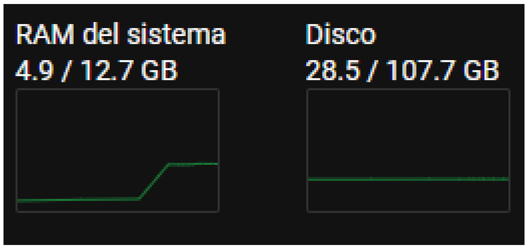
Disco
28.5 / 107.7 GB



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26 entries, 0 to 25
Data columns (total 8 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   (Airline, )             26 non-null     object 
 1   (Year, )                26 non-null     int64  
 2   (DepDelayMinutes, mean) 26 non-null     float64
 3   (DepDelayMinutes, sum)  26 non-null     float64
 4   (DepDelayMinutes, max)  26 non-null     float64
 5   (ArrDelayMinutes, mean) 26 non-null     float64
 6   (ArrDelayMinutes, sum)  26 non-null     float64
 7   (ArrDelayMinutes, max)  26 non-null     float64
dtypes: float64(6), int64(1), object(1)
memory usage: 1.8+ KB
```

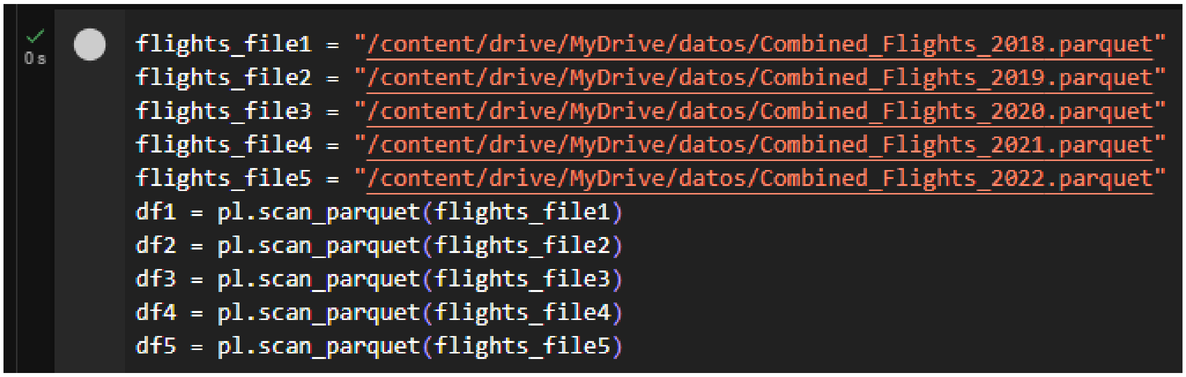


```python
import polars as pl
```

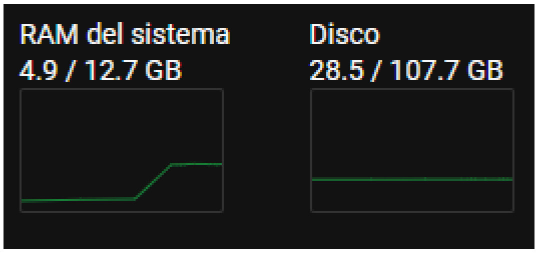


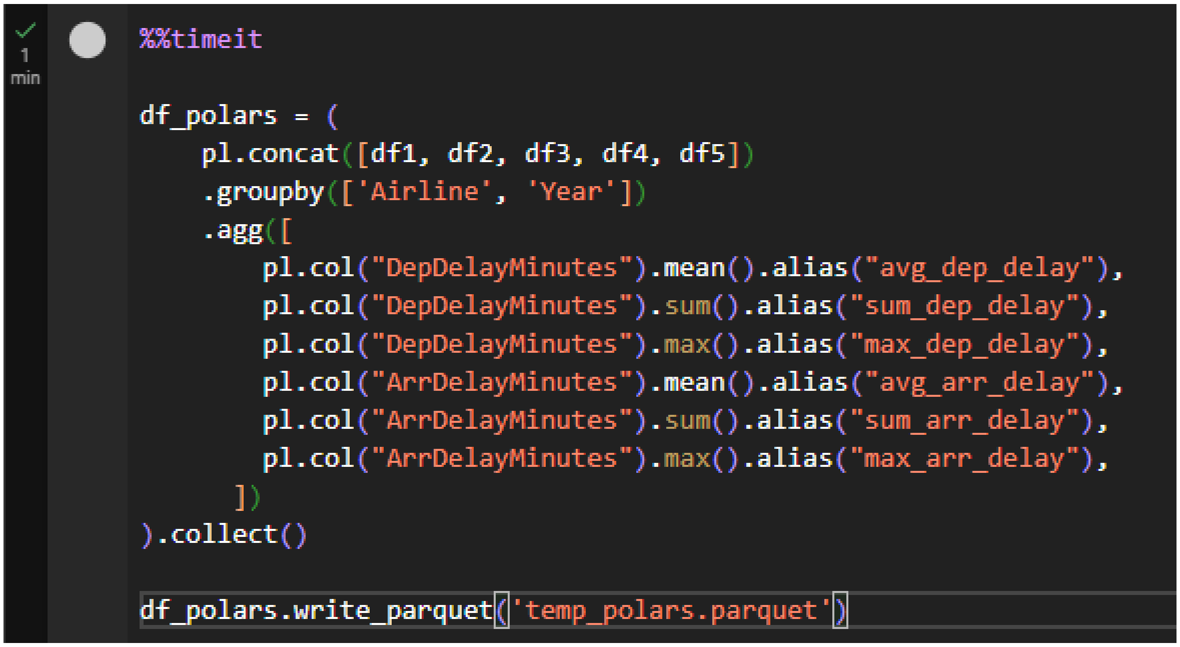
RAM del sistema
4.9 / 12.7 GB

Disco
28.5 / 107.7 GB

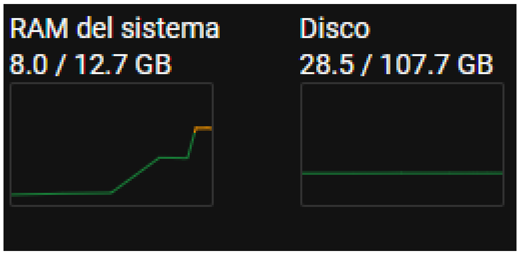


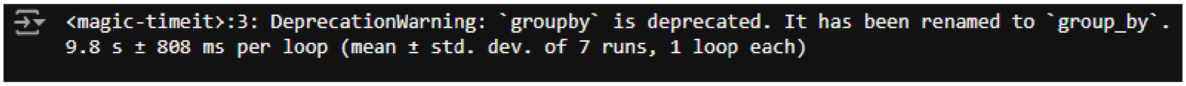
```python
flights_file1 = "/content/drive/MyDrive/datos/Combined_Flights_2018.parquet"
flights_file2 = "/content/drive/MyDrive/datos/Combined_Flights_2019.parquet"
flights_file3 = "/content/drive/MyDrive/datos/Combined_Flights_2020.parquet"
flights_file4 = "/content/drive/MyDrive/datos/Combined_Flights_2021.parquet"
flights_file5 = "/content/drive/MyDrive/datos/Combined_Flights_2022.parquet"
df1 = pl.scan_parquet(flights_file1)
df2 = pl.scan_parquet(flights_file2)
df3 = pl.scan_parquet(flights_file3)
df4 = pl.scan_parquet(flights_file4)
df5 = pl.scan_parquet(flights_file5)
```



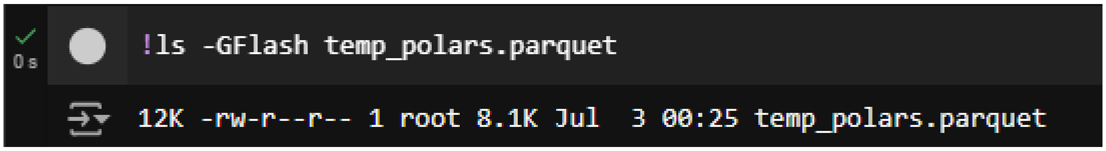


```
%%timeit

df_polars = (
    pl.concat([df1, df2, df3, df4, df5])
    .groupby(['Airline', 'Year'])
    .agg([
        pl.col("DepDelayMinutes").mean().alias("avg_dep_delay"),
        pl.col("DepDelayMinutes").sum().alias("sum_dep_delay"),
        pl.col("DepDelayMinutes").max().alias("max_dep_delay"),
        pl.col("ArrDelayMinutes").mean().alias("avg_arr_delay"),
        pl.col("ArrDelayMinutes").sum().alias("sum_arr_delay"),
        pl.col("ArrDelayMinutes").max().alias("max_arr_delay"),
      ])
).collect()

df_polars.write_parquet('temp_polars.parquet')
```



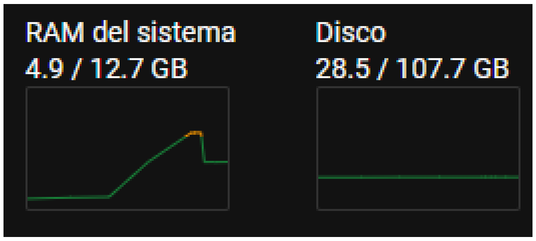


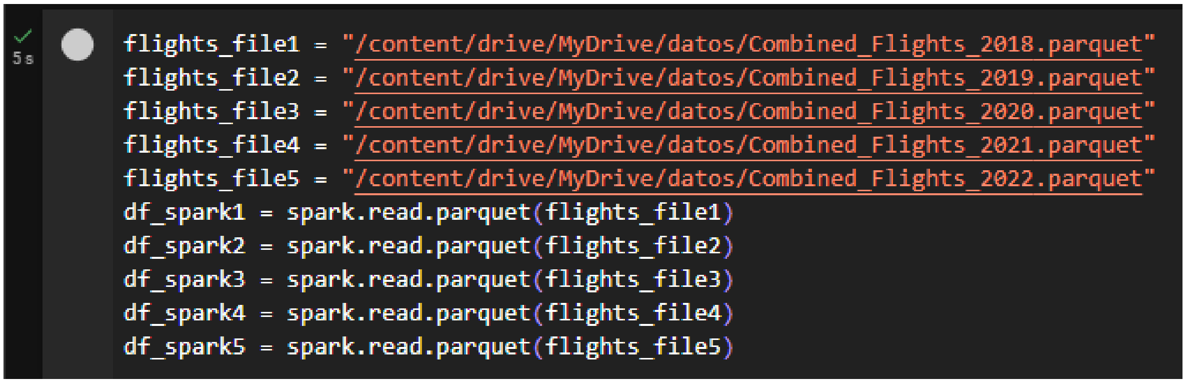
```
<magic-timeit>:3: DeprecationWarning: `groupby` is deprecated. It has been renamed to `group_by`.
9.8 s ± 808 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
```



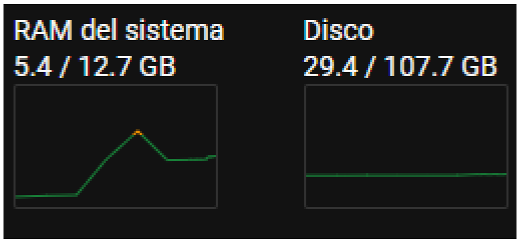
```
!ls -GFlash temp_polars.parquet
```

```
12K -rw-r--r-- 1 root 8.1K Jul  3 00:25 temp_polars.parquet
```



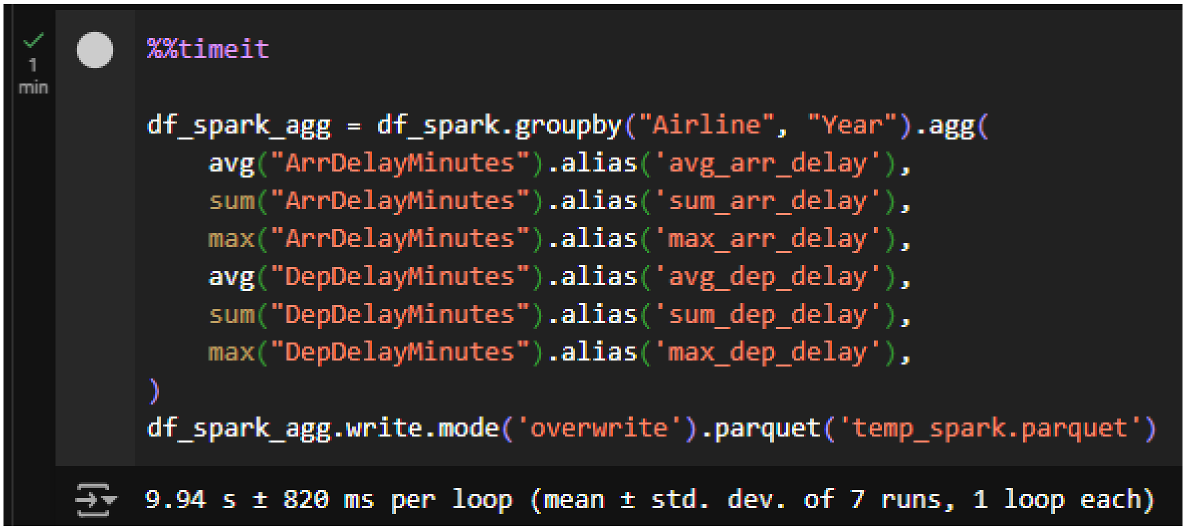


```
flights_file1 = "/content/drive/MyDrive/datos/Combined_Flights_2018.parquet"
flights_file2 = "/content/drive/MyDrive/datos/Combined_Flights_2019.parquet"
flights_file3 = "/content/drive/MyDrive/datos/Combined_Flights_2020.parquet"
flights_file4 = "/content/drive/MyDrive/datos/Combined_Flights_2021.parquet"
flights_file5 = "/content/drive/MyDrive/datos/Combined_Flights_2022.parquet"
df_spark1 = spark.read.parquet(flights_file1)
df_spark2 = spark.read.parquet(flights_file2)
df_spark3 = spark.read.parquet(flights_file3)
df_spark4 = spark.read.parquet(flights_file4)
df_spark5 = spark.read.parquet(flights_file5)
```



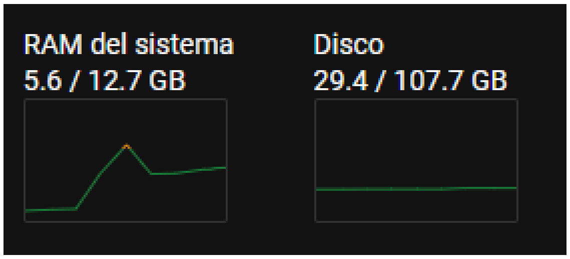
RAM del sistema
5.4 / 12.7 GB

Disco
29.4 / 107.7 GB



```
%%timeit

df_spark_agg = df_spark.groupby("Airline", "Year").agg(
    avg("ArrDelayMinutes").alias('avg_arr_delay'),
    sum("ArrDelayMinutes").alias('sum_arr_delay'),
    max("ArrDelayMinutes").alias('max_arr_delay'),
    avg("DepDelayMinutes").alias('avg_dep_delay'),
    sum("DepDelayMinutes").alias('sum_dep_delay'),
    max("DepDelayMinutes").alias('max_dep_delay'),
)
df_spark_agg.write.mode('overwrite').parquet('temp_spark.parquet')
```

```
9.94 s ± 820 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
```



RAM del sistema
5.6 / 12.7 GB

Disco
29.4 / 107.7 GB