Laboratorio Numero 2

Luis Felipe Sanchez Sanchez - 90613
Sebastian Morales Devia - 73487

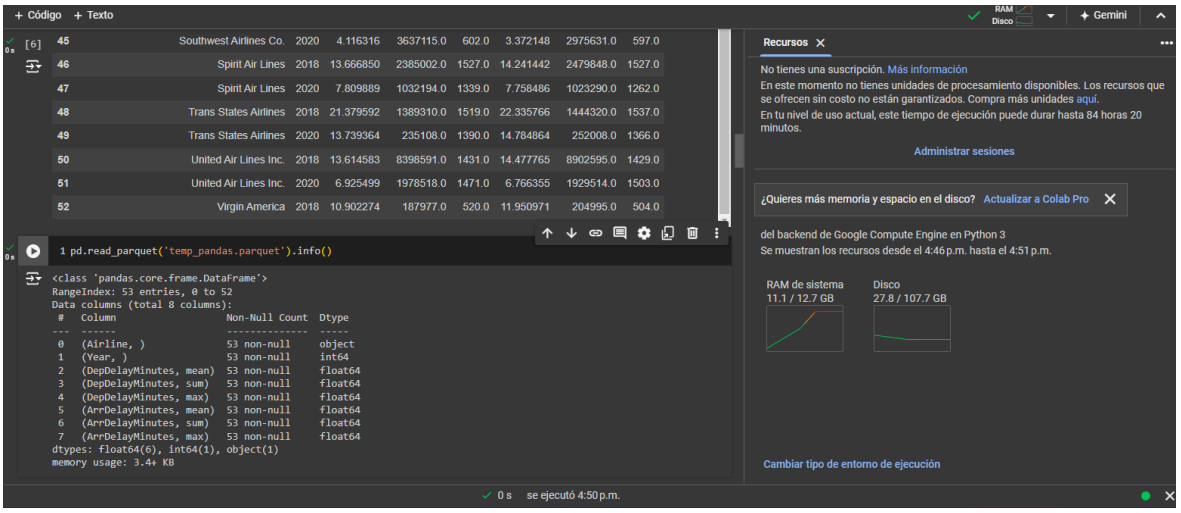
Universidad ECCI
Facultad de Ingeniería

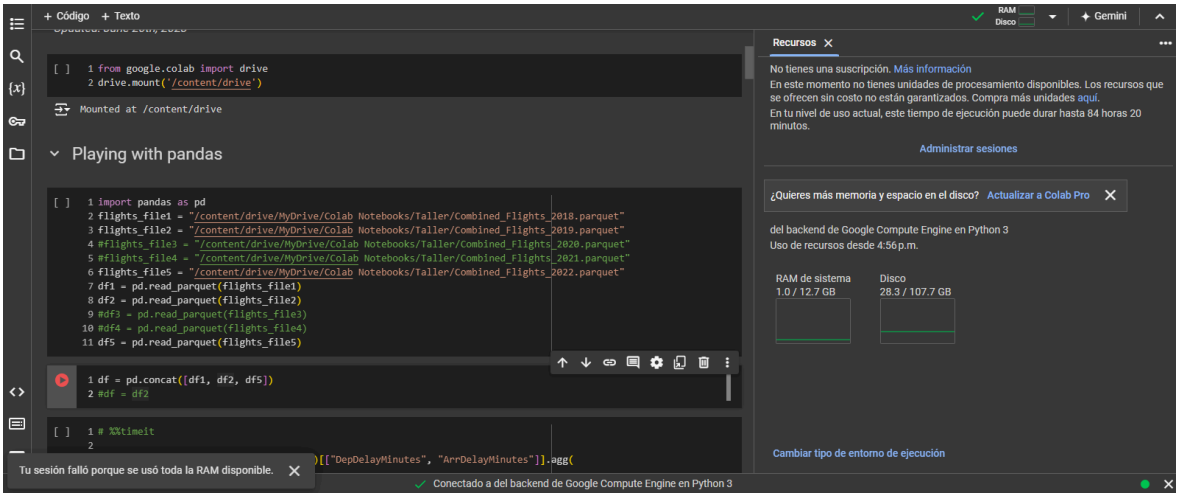
Elias Buitrago Bolivar

Bogotá
2024

Introducción

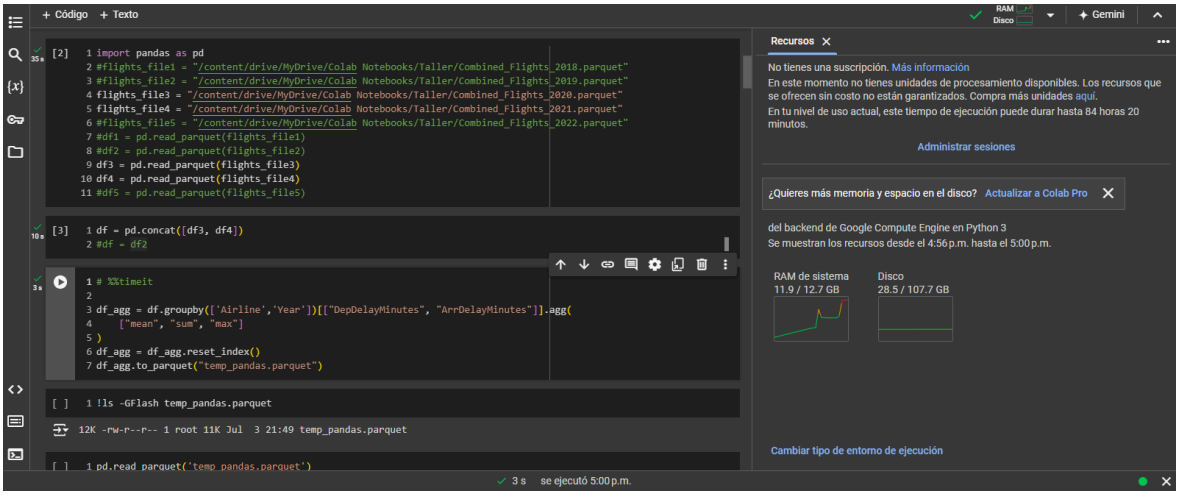
Al comparar diferentes bibliotecas en cuanto a su rendimiento y efectividad en el procesamiento de archivos .parquet, es necesario observar más de cerca con qué tipo de conjuntos de datos trabajan. En esta investigación, analizaremos las capacidades de varias bibliotecas conocidas para cargar y trabajar con archivos .parquet, considerando diferentes escenarios para cargas de archivos y distintos niveles de complejidad de datos. A partir de esta evaluación podemos identificar la mejor biblioteca en función de su velocidad, eficacia y capacidad para manejar grandes cantidades de datos en formato .parquet.

PANDAS
Prueba 1:    Archivos usados: 1 y 3 RAM: 11.1 / 12./ GB NOTAS: Ejecuta todas las líneas sin lio Ejecucción: 52 SEG
Prueba 2:



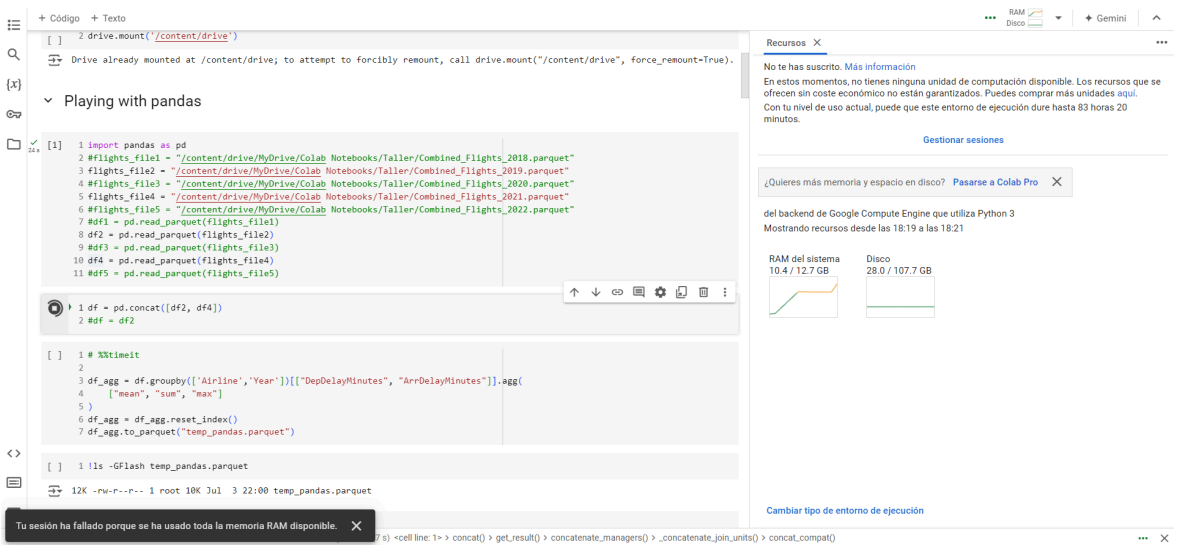
Archivos usados: 1, 2 y 5
RAM: -/ 12.7 GB
NOTAS: Desbordamiento en la sección de concatenación
Ejecucción: 39 SEG

Prueba 3:



Archivos usados: 3 y 4
RAM: 11.7/ 12.7 GB
NOTAS: Se ejecutaron los comandos sin desbordamiento
Ejecucción: 48 SEG

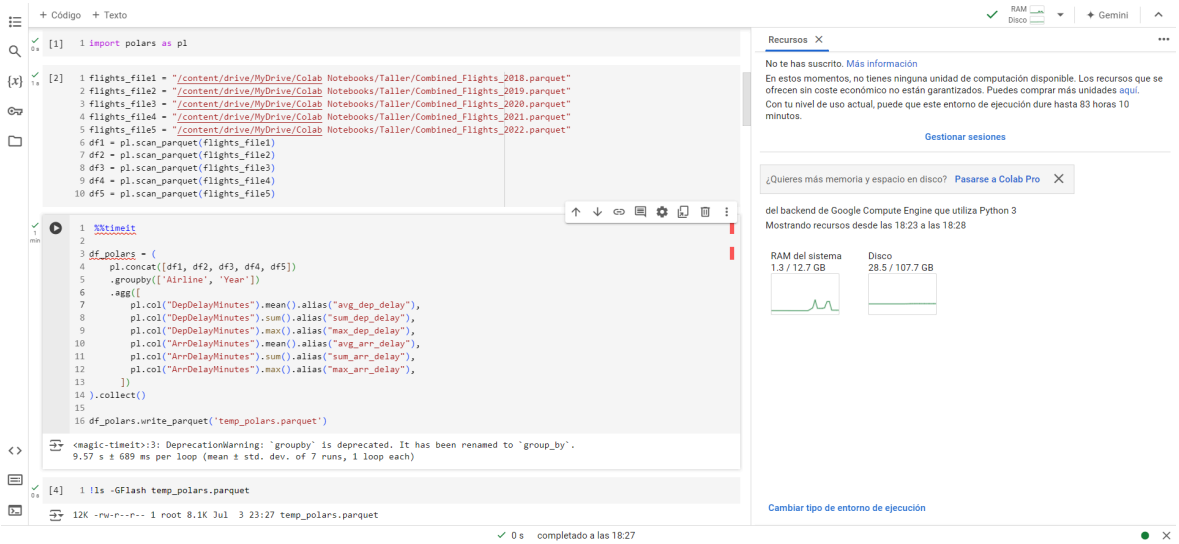
Prueba 4:



Archivos usados: 2 y 4
RAM: -/ 12.7 GB
NOTAS: en la carga de archivos se desbordo la RAM
Ejecucción: 28 SEG

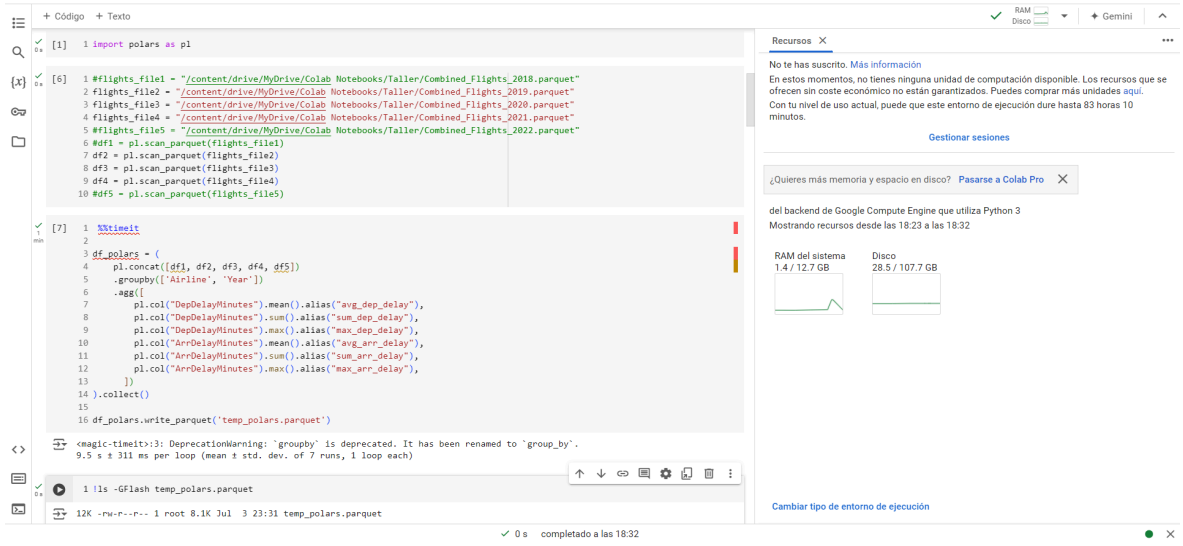
POLARS

Intento 1:



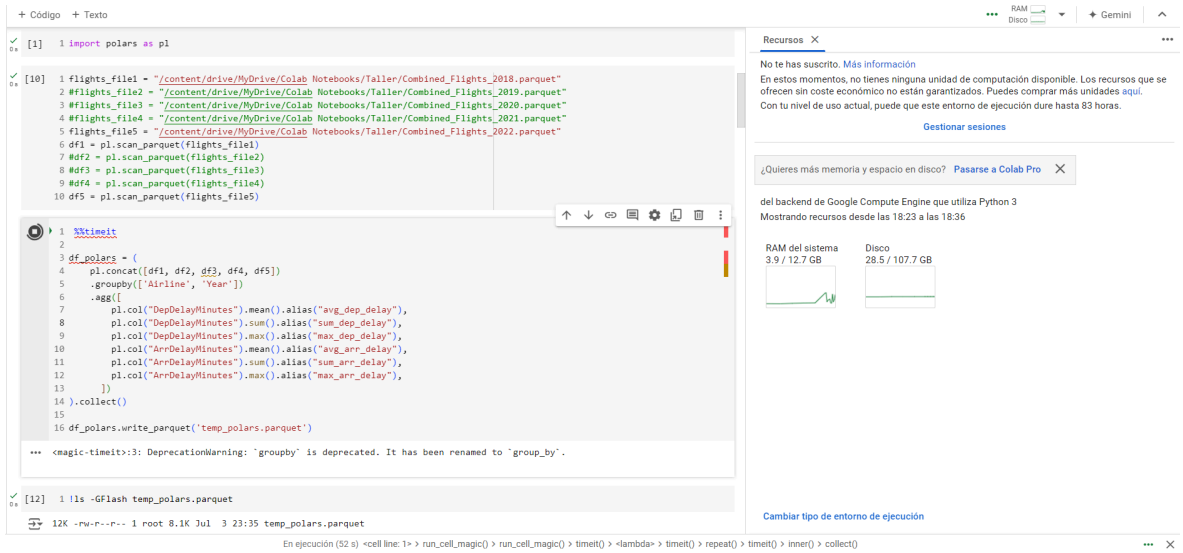
Archivos usados: Todos
RAM: 1.3 / 12.7 GB
NOTAS: Todo corrió sin problema
Ejecucción: 61 SEG

Intento 2:



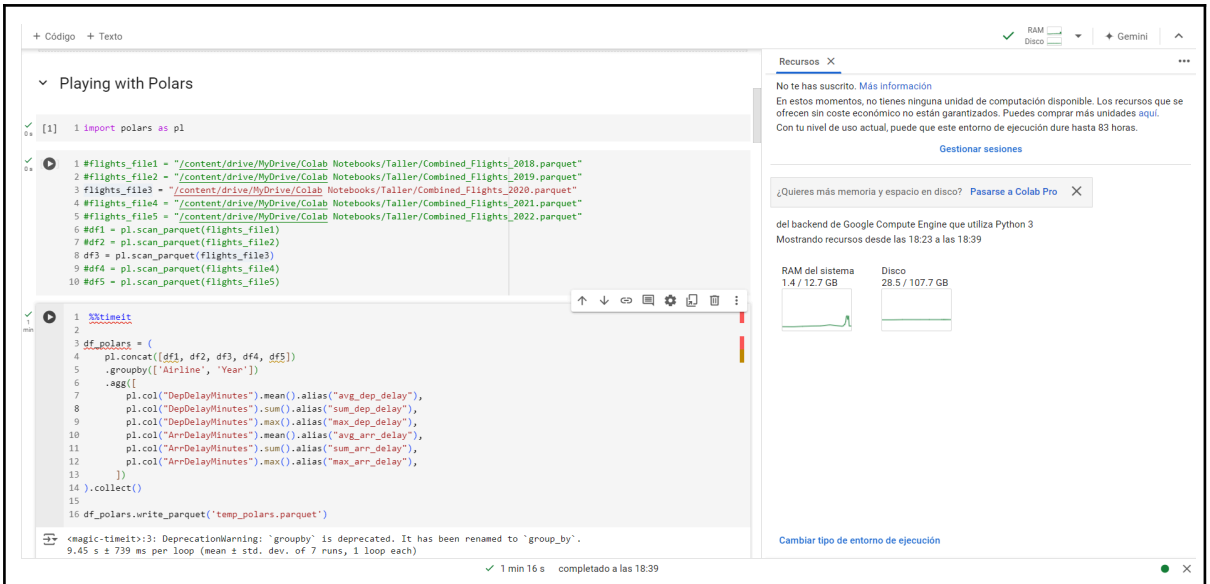
Archivos usados: 2, 3 y 4
RAM: 1.4 / 12.7 GB
NOTAS: Ejecutó sin problemas
Ejecucción: 60 SEG

Intento 3:



Archivos usados: 1 y 5
RAM: 3.9/ 12.7 GB
NOTAS: Se queda procesando dicha línea por más de 1:30 min
Ejecucción: +1:30 MIN

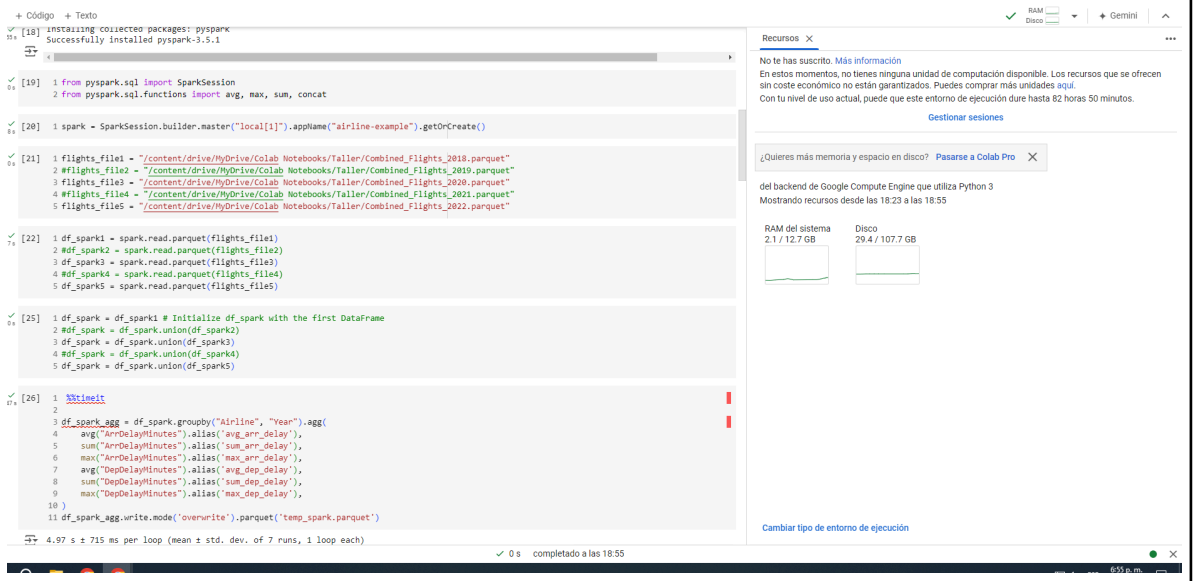
Intento 4:



Archivos usados: 3
RAM: 1.4/ 12.7 GB
NOTAS: Ejecuta sin problemas
Ejecucción: 1.16 MIN

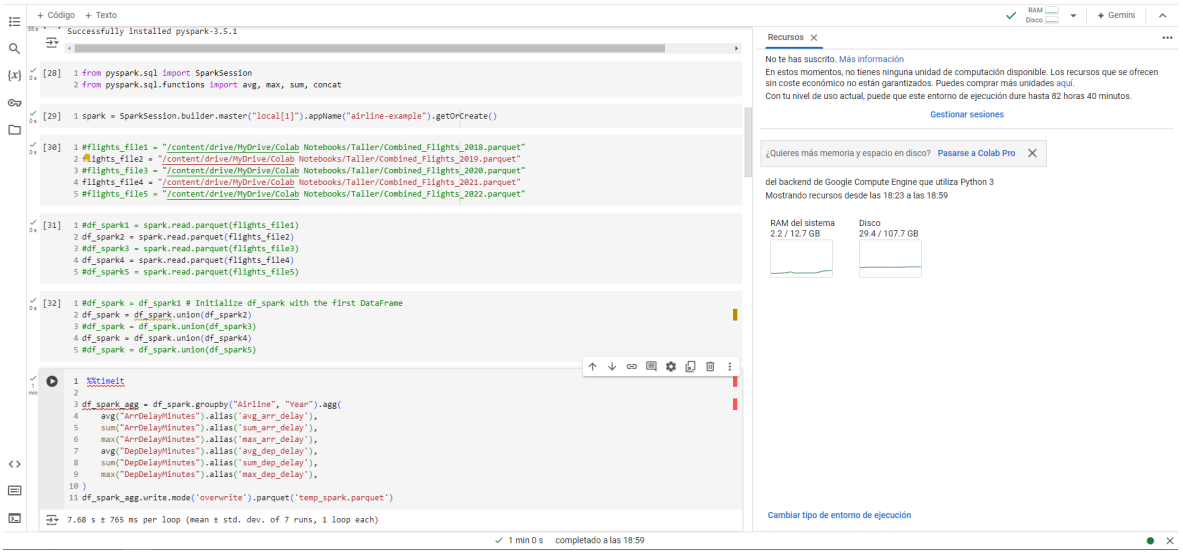
PYSPARK

Intento 1:



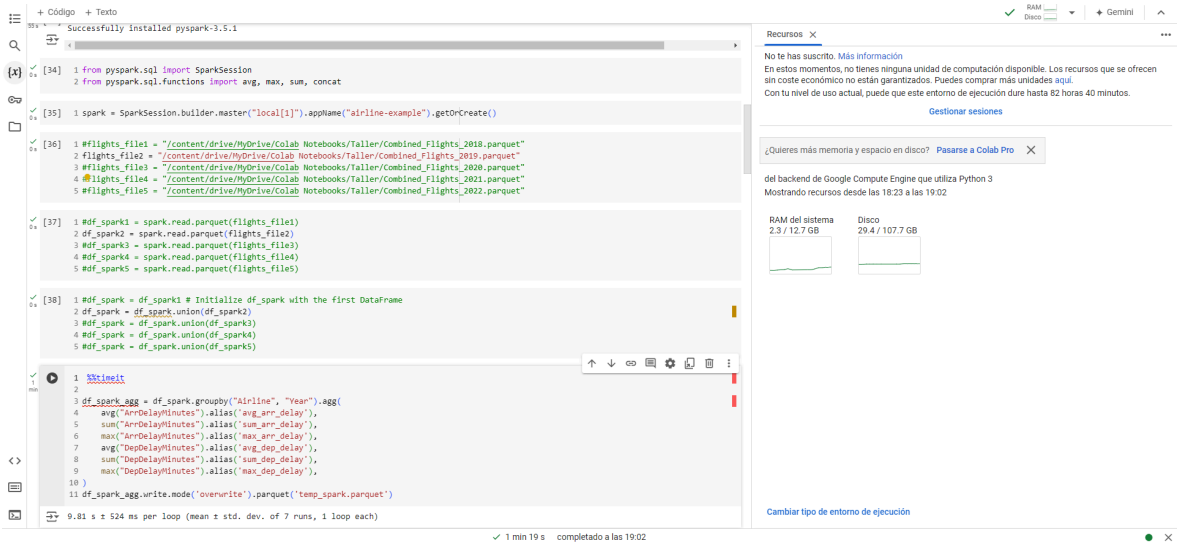
Archivos usados: 1, 3 y 5
RAM: 2.1 / 12.7 GB
NOTAS: Ejecución exitosa
Ejecución: 1.81 MIN

Intento 2:



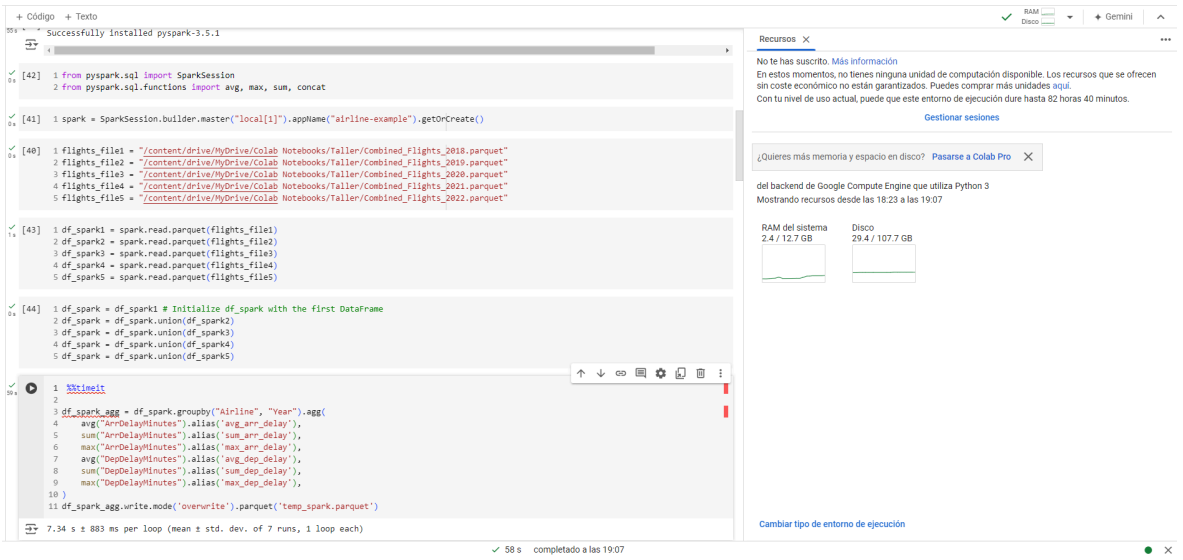
Archivos usados: 2 y 4
RAM: 2.2/ 12.7 GB
NOTAS: Compilo sin ningún error
Ejecución: 1 MIN

Intento 3:



Archivos usados: 2
RAM: 2.4/ 12.7 GB
NOTAS: Ejecución sin problemas
Ejecución: 1.10 MIN

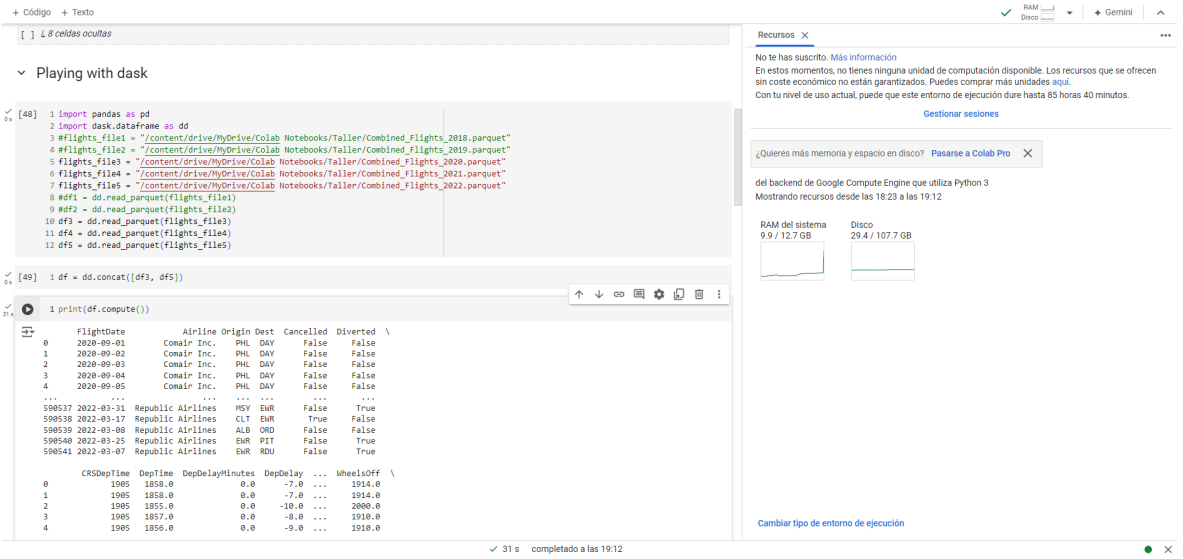
Intento 4:



Archivos usados: 1, 2, 3, 4 y 5
RAM: 2.4/ 12.7 GB
NOTAS: Exitosa la ejecución
Ejecución: 59 SEG

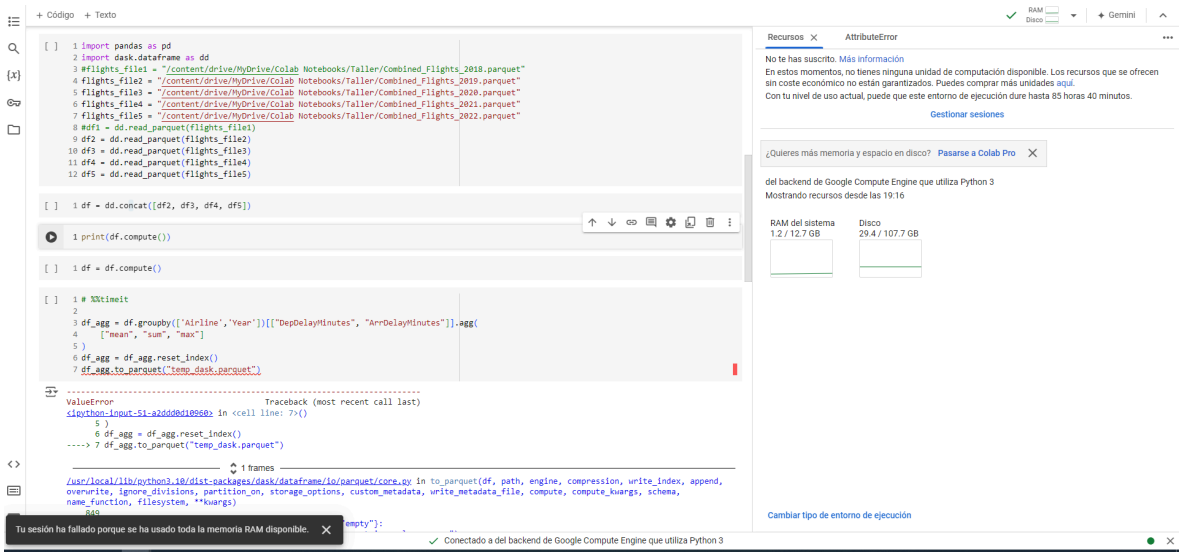
DASK

Intento 1:



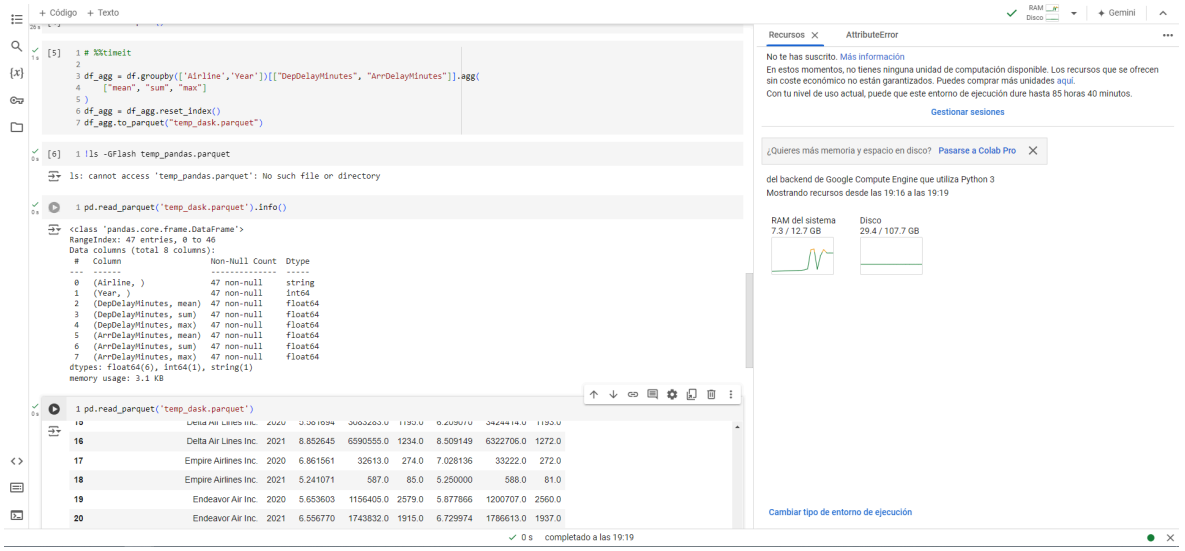
Archivos usados: 3, 4 y 5
RAM: 9.9/ 12.7 GB
NOTAS: Se corta el la linea timeit
Ejecución: 31 SEG

Intento 2:



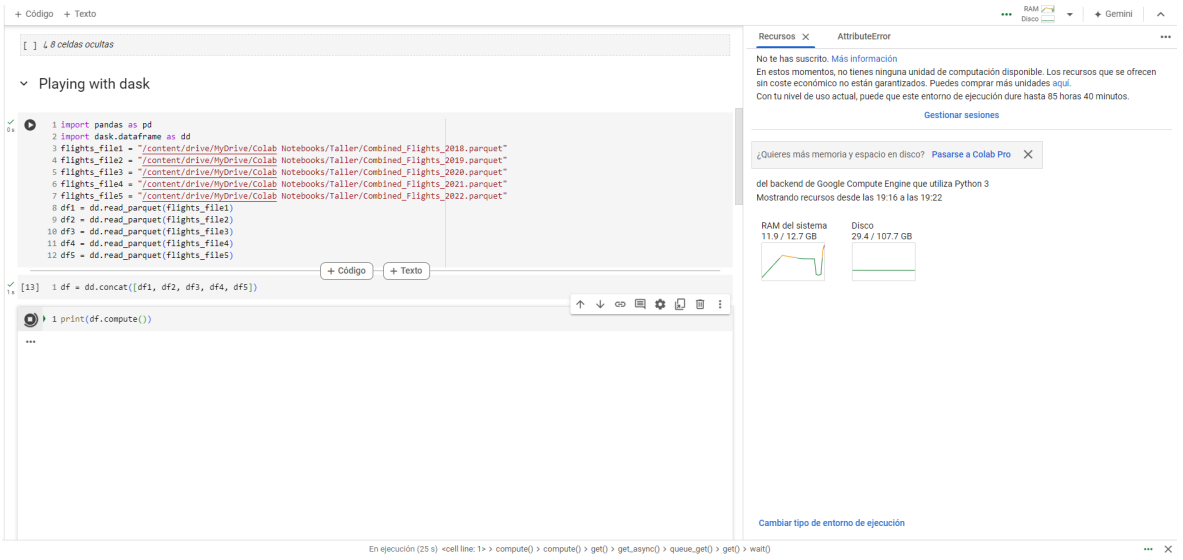
Archivos usados: 2, 3, 4 y 5
RAM: -/ 12.7 GB
NOTAS: Se desbordó el código con esa cantidad
Ejecución: N/A

Intento 3:



Archivos usados: 3 y 4
RAM: 7.3/ 12.7 GB
NOTAS: Ejecución exitosa
Ejecución: 57 SEG

Intento 4:



Archivos usados: 1, 2, 3, 4 y 5
RAM: 11.9/ 12.7 GB
NOTAS: Desbordamiento con todos los archivos
Ejecución: 40 SEG

Conclusiones

- La librería pandas la mayoría de veces que se corrían varios datos si no colapsaba la RAM, quedaba al límite
- El tiempo de ejecución con pandas es promedio a todas las demás librerías, sin embargo, con pandas se rompe
- El correr 2 conjuntos de datos podría provocar que se caiga COLAB
- El correr con DASK varios datos puede provocar en promedio más de 10 GB de consumo por ejecucción
- A pesar de las dificultades en la carga de datos y las limitaciones, pandas sigue siendo la opción usada para estas tareas
- Si no queremos exceder la RAM, es es factible usar cualquiera que no sea pandas, ya que disminuye la cantidad recursos consumidos operativos