**Laboratorio #2**  
  
**Presentado por**

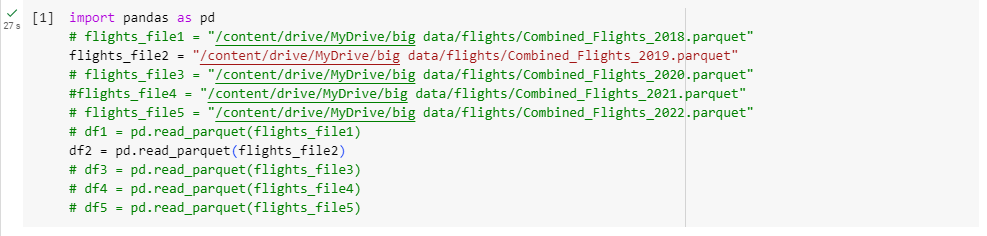
[Cristian Ballen Gamboa](mailto:cristian.balleng@ecci.edu.co)

[Breyner Andreit Rincon Quiroga](mailto:breynera.rinconq@ecci.edu.co)

**Documentación**

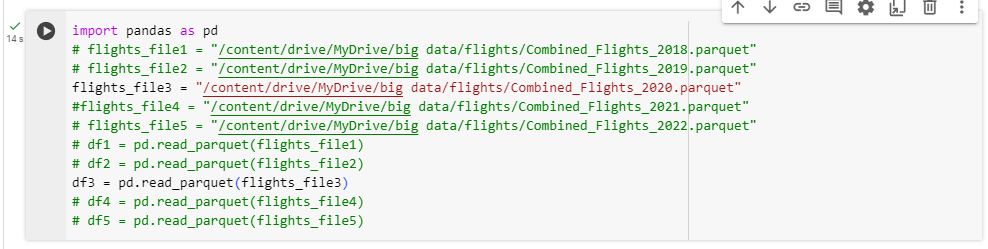
#1 Ejecución del archivo #2

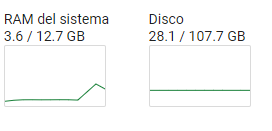
* Tiempo 27 segundos
* ram 4.7 gb



#2 Ejecución del archivo #3

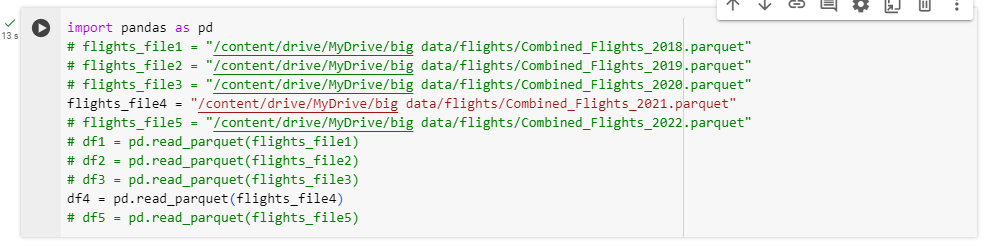
* Tiempo 14 segundos
* Ram 3.6 gb

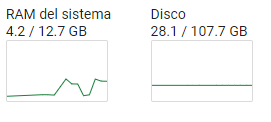




#3 Ejecución del archivo #4

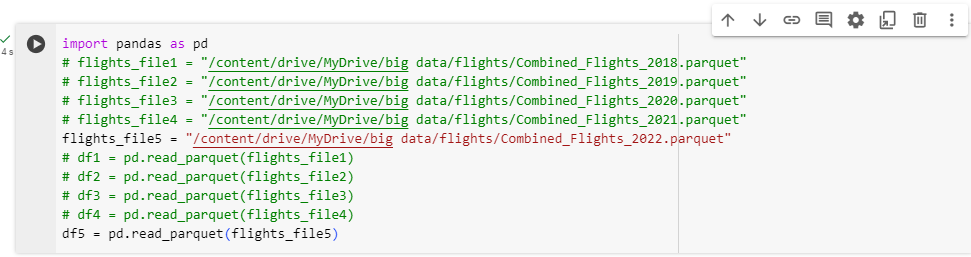
* Tiempo 13 segundos
* Ram 4.2 gb

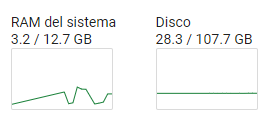




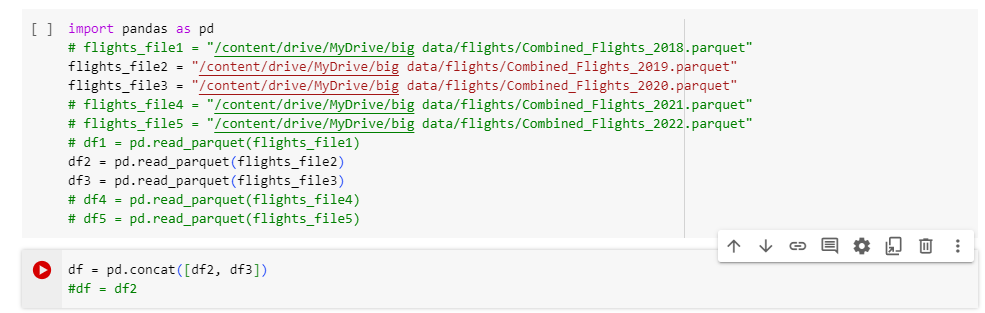
#4 Ejecución del archivo #5

* Tiempo 4 segundos
* Ram 3.2 gb



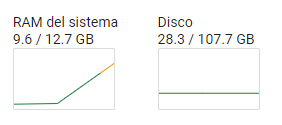


#5 concatenación del archivo 2 y 3 nos deja visualizar que la Ram no tiene suficiente espacio para realizar la ejecución y sólo se podrá ejecutar un archivo.



# 6 Al realizar la concatenación del archivo 3 y 5 podemos visualizar que la ram cuenta con espacio suficiente para concatenar estos dos archivos, , ( dependiendo el tamaño del archivo solo se podrá ejecutar si la ram tiene espacio suficiente ).

Debido a que el archivo 3 y 5 nos ayuda a evidenciar que podemos realizar la concatenación de los dos archivos, realizaremos el ejercicio con este ejemplo.

  
El código agrupa los datos de vuelo por aerolínea y año, calcula varias estadísticas sobre los retrasos en la salida y llegada, luego guarda los resultados agrupados y calculados en un archivo Parquet llamado "temp\_pandas.parquet".

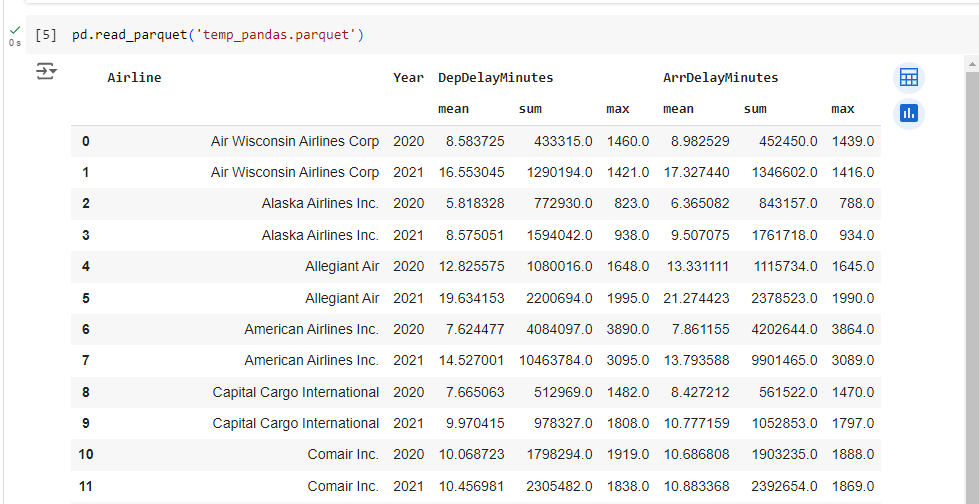


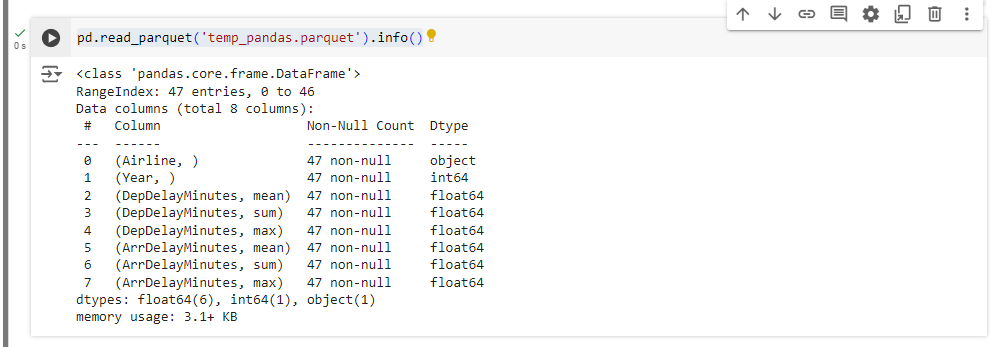
Se evidencia que la ejecución del código se demora 2 segundos.



lista detalladamente el archivo temp\_pandas.parquet en el directorio actual.

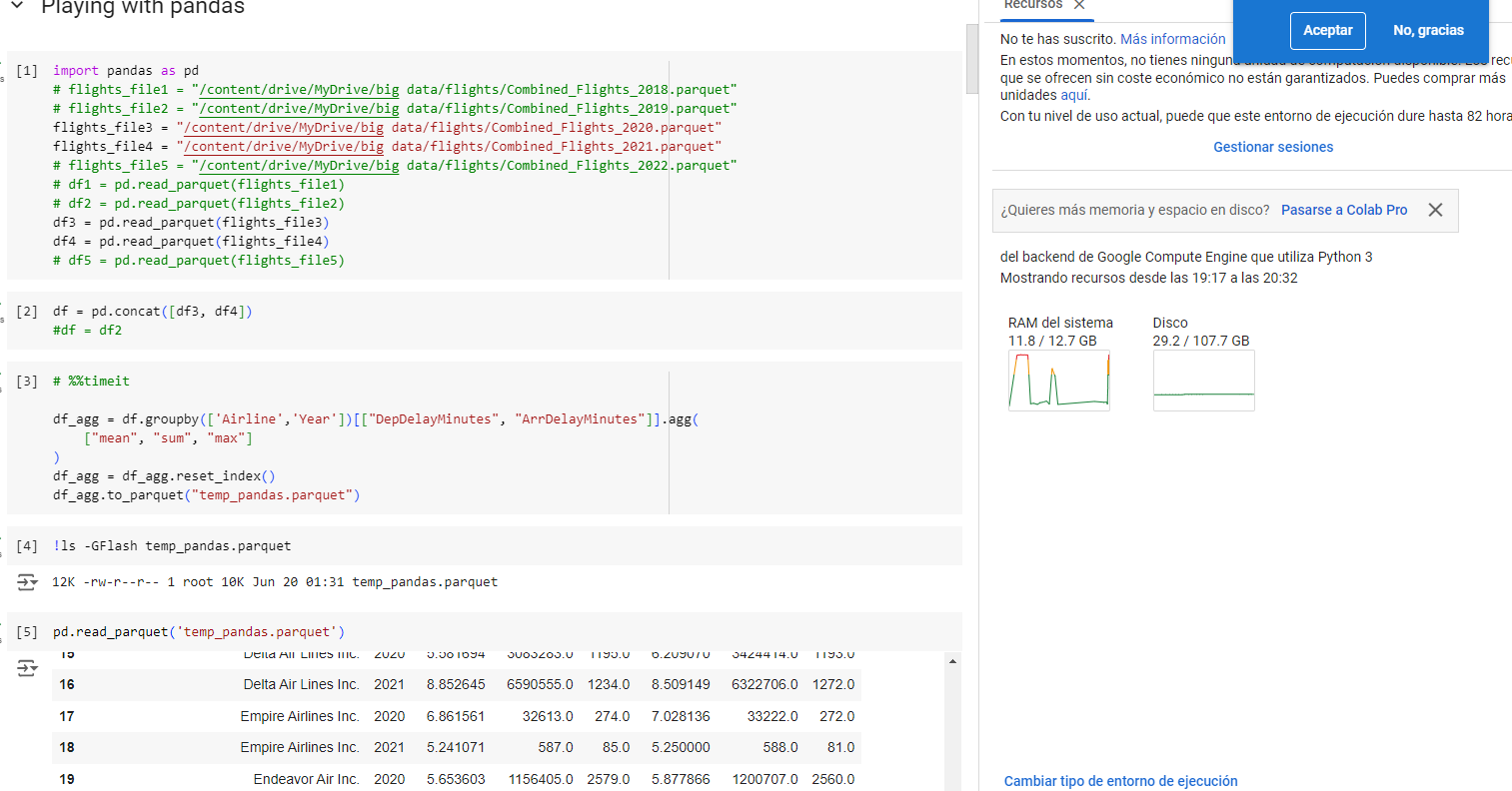
se visualiza la informacion

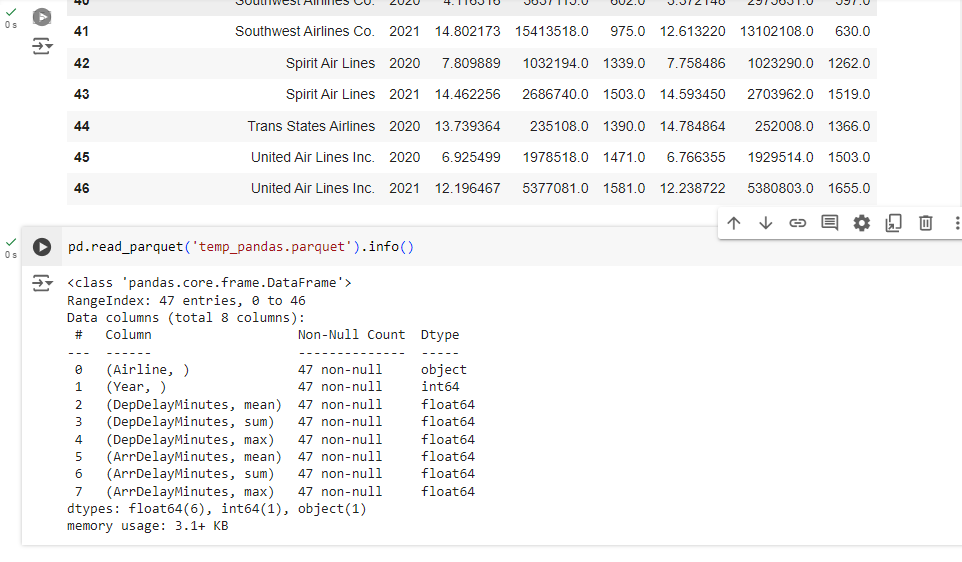


carga el archivo Parquet temp\_pandas.parquet en un DataFrame utilizando pd.read\_parquet(), y luego llama al método .info() en ese DataFrame para imprimir un resumen de la estructura y la información básica  
 

**Pandas**

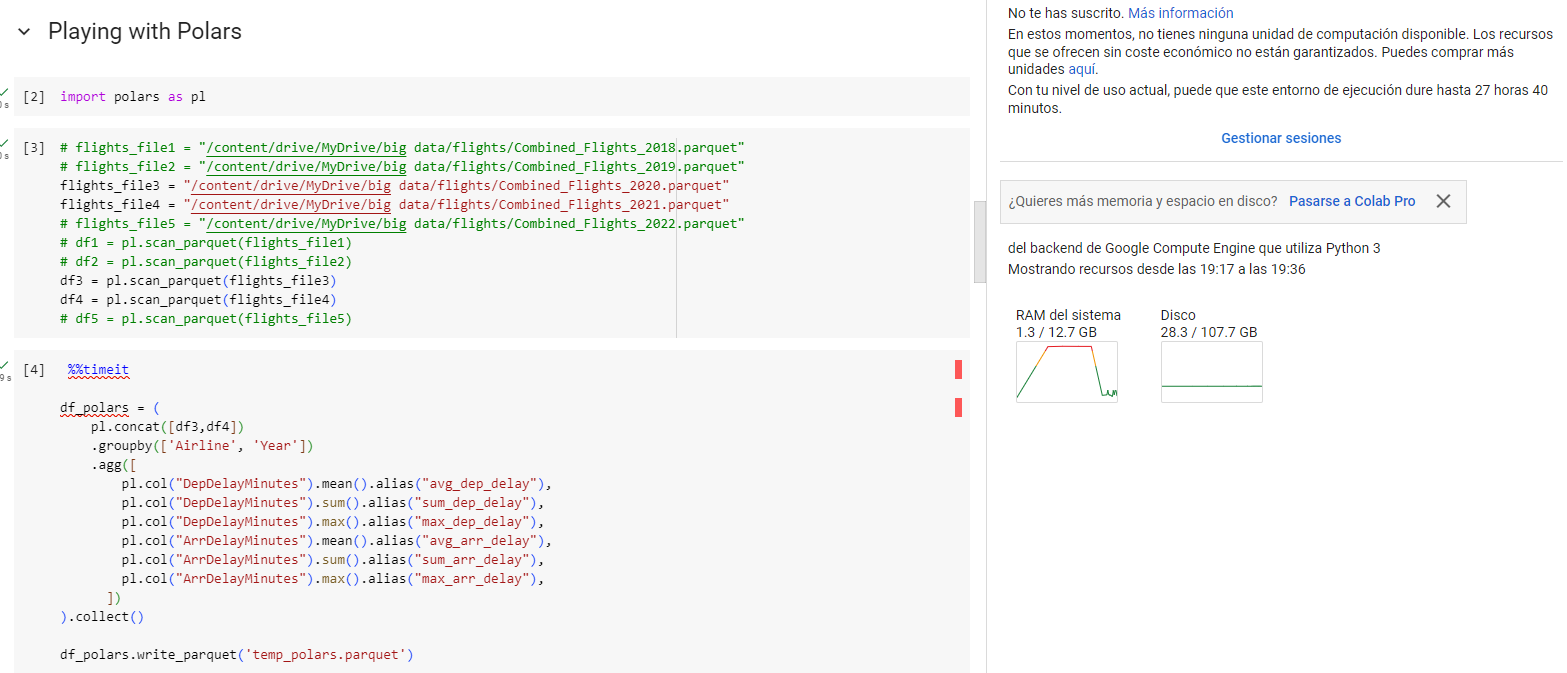
Con Pandas logramos evidenciar que el uso de Ram (11.8) y el tiempo ( 63 segundos).





**Polars**

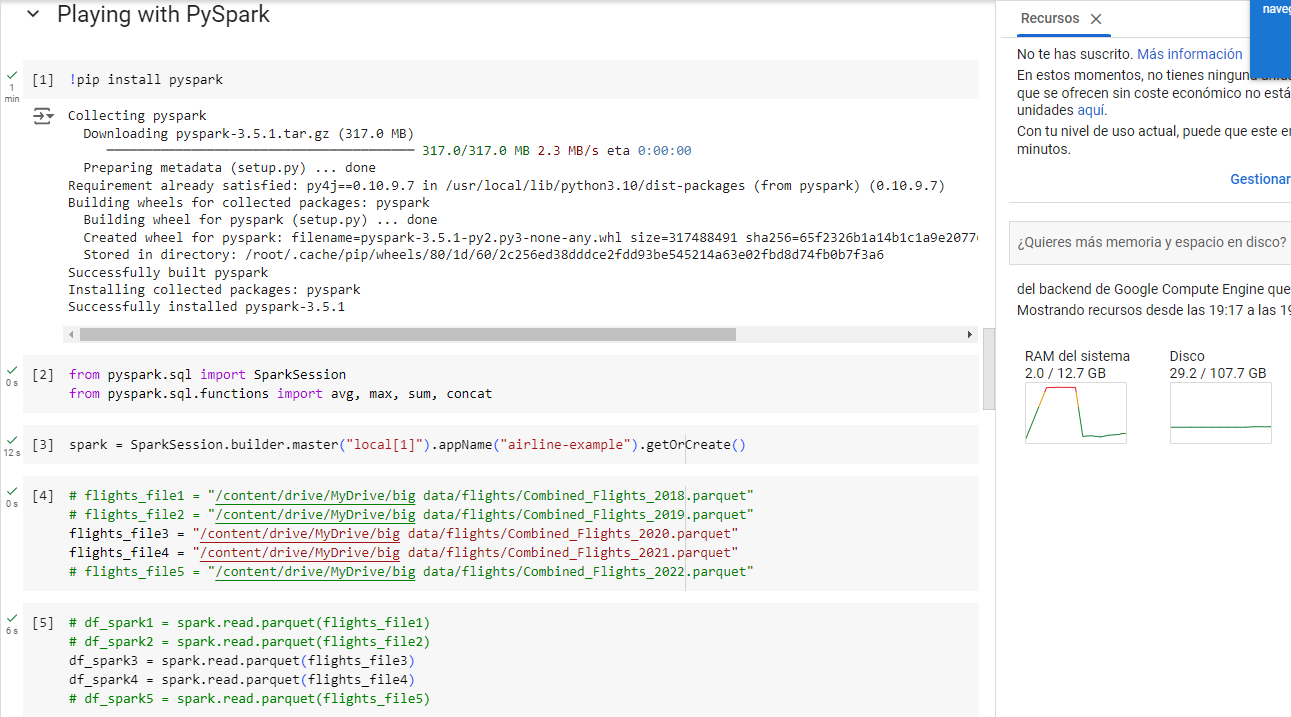
Con polars se evidencia que el uso de Ram (1.3gb) y de tiempo (9 segundos).

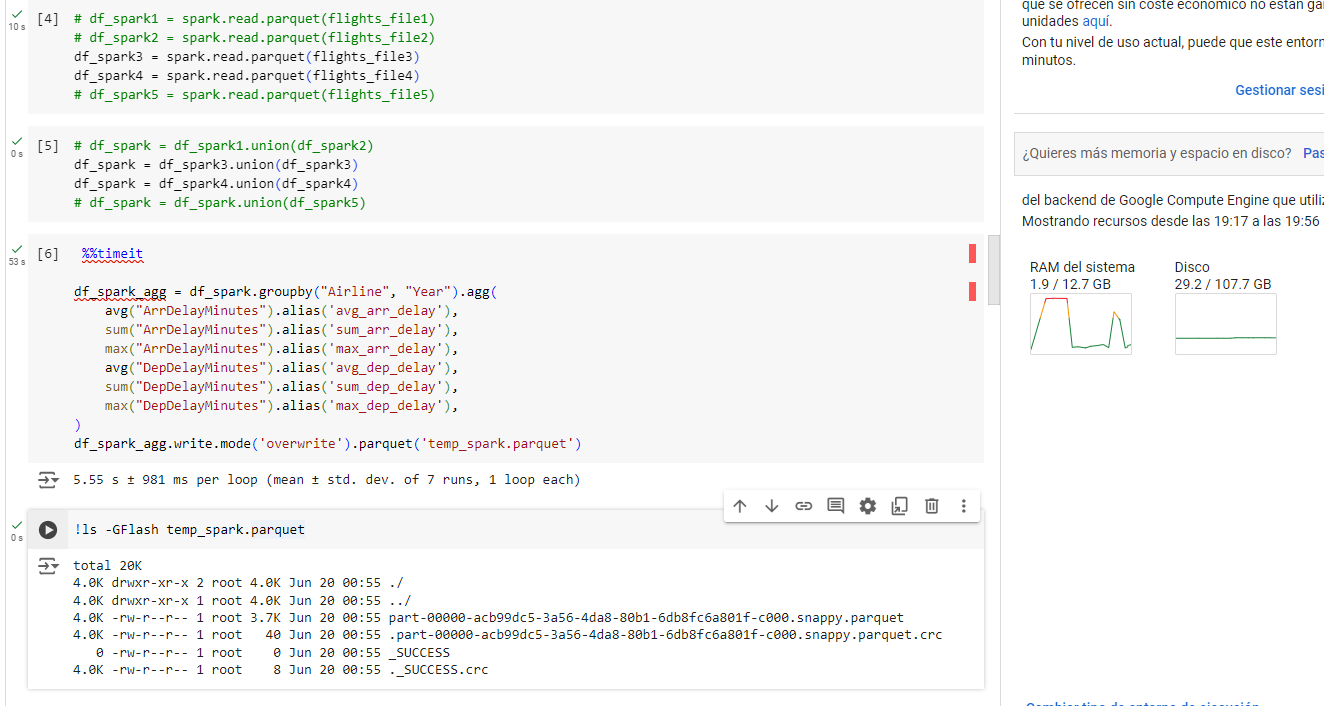


## 

## **PySpark**

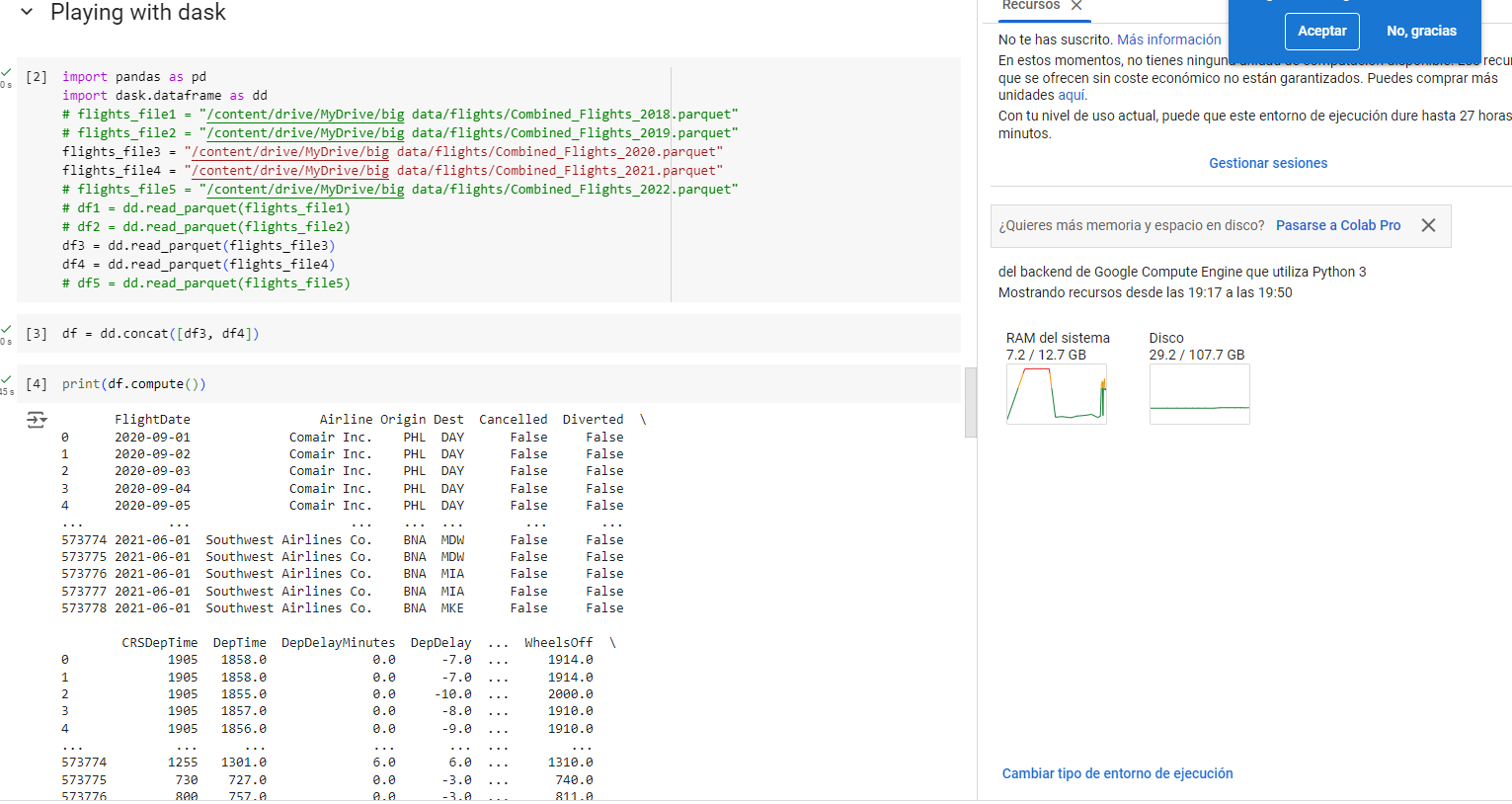
Con pyspark logramos evidenciar que el uso de Ram (2.0) y el tiempo ( 75 segundos)



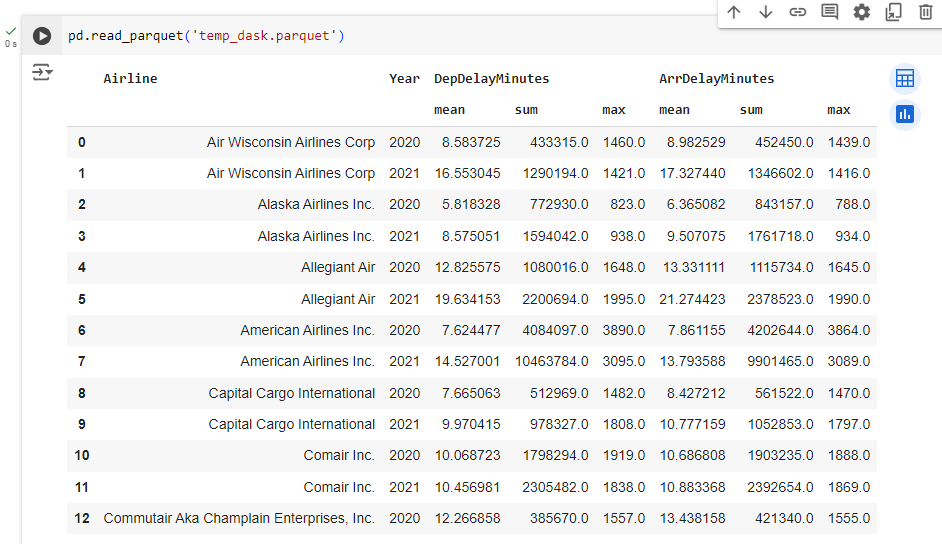


## **dask**

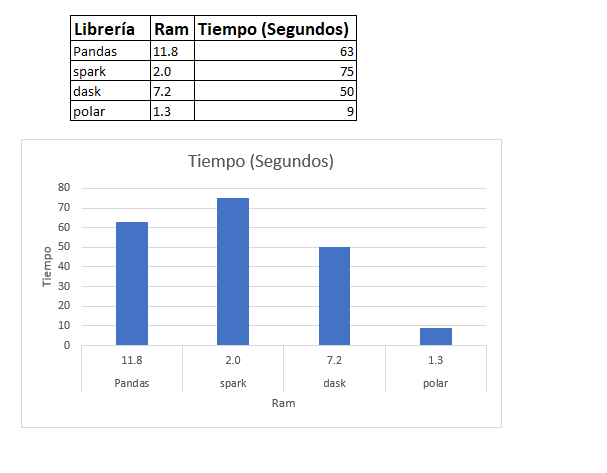
Con dask logramos evidenciar que el uso de Ram (7.2) y el tiempo ( 50 segundos)



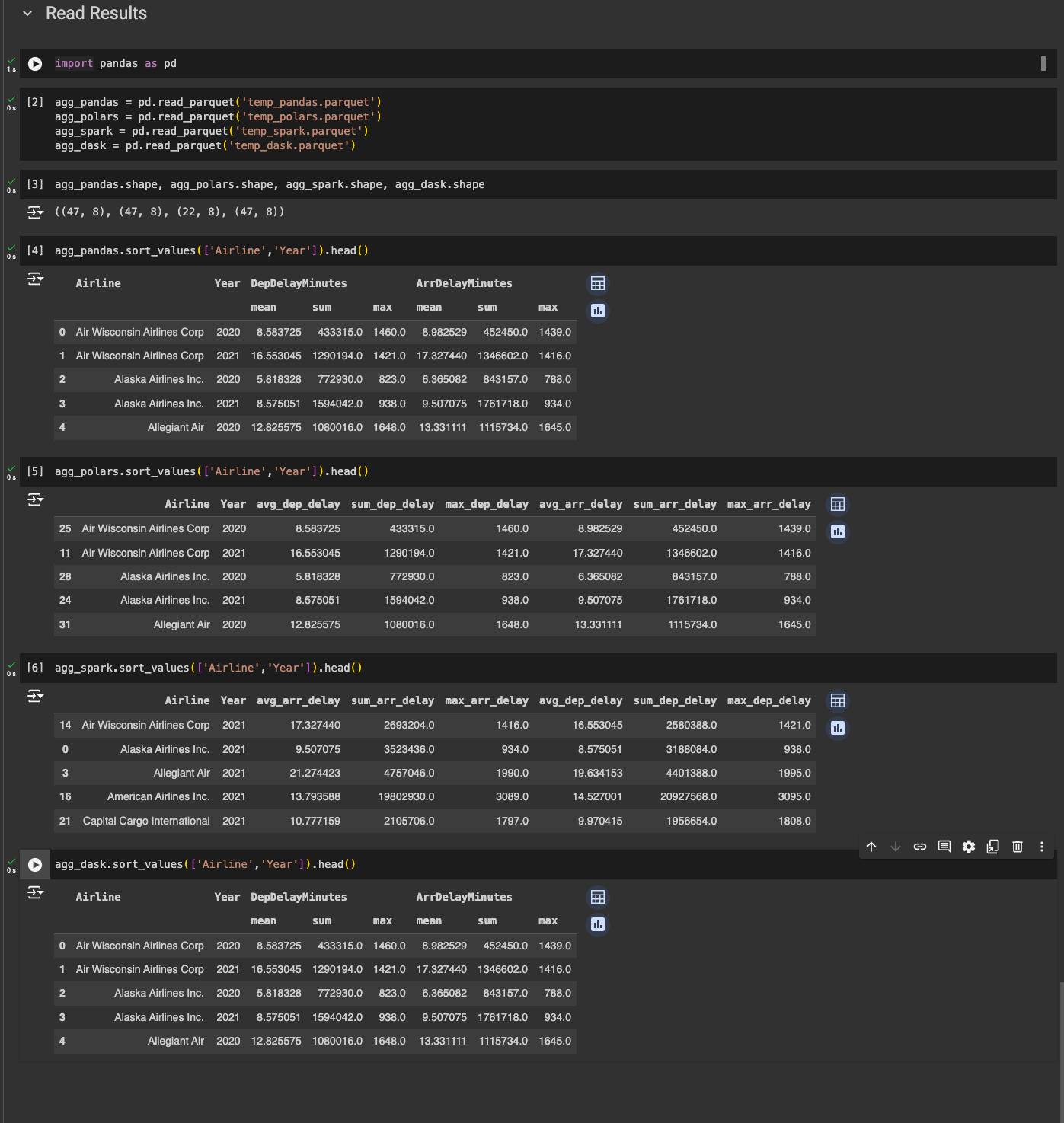




**Resultados**



Logramos evidenciar que polars es significativamente mejor para la lectura de los datos dando mejores resultados en tiempo y uso de ram.

Al visualizar la lectura de los archivos podemos observar que el resultado de pandas, polars y dask son iguales, el único que difiere de resultados es spark, de igual manera los encabezados suelen cambiar según la librería  
  


**Conclusión**

Según la ejecución de todos los archivos y códigos logramos evidenciar que polars es el que tiene mejor rendimiento, ocupando menos ram y menor tiempo en ejecución.