Seminario vacacional
“Big data, Analítica de datos y gestión de la información”

Laboratorio #2.

Estudiante:
Joseph Oswald Quiroz Mejia.

Código:
69794.

Profesor:
Elias Buitrago Bolivar.

Universidad:
ECCI.

Fecha:
20/06/2024.

Desarrollo del laboratorio.

En primer lugar se realiza la vinculación del "Drive" con la aplicación Google Colab con el fin de que el archivo que se utilizara en las diferentes programaciones funcione de forma correcta, una vez realizada esta acción se procede a experimentar con cada una de las librerías que se hallan en el archivo y la programación, esto con el fin de cargar la mayor cantidad de datos sin reiniciar o saturar la RAM, se realiza el cargue de los datos en las librerías de Panda, Polars, Spark y Dask y una vez realizada estas pruebas se recolectan los datos de cada una de las librerías para poder realizar una comparación y determinar cual de esta librerías es la más eficiente en la carga de los datos y en la cantidad de datos que se pueden procesar.

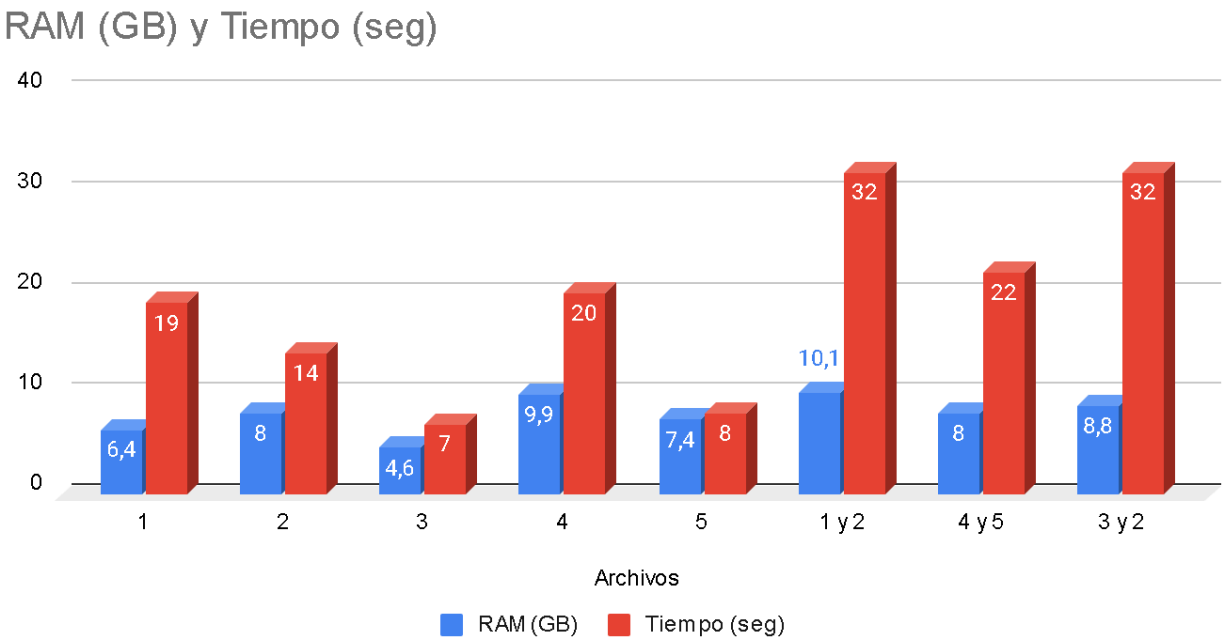
Comparación.

Se realiza una tabla de recolección con cada una de las librerías, con las variables de "Archivo, Ram GB y tiempo (Seg)" la variable de archivo nos define cual es el tipo de dato que se están ingresando y si cargamos uno o más al momento de ejecutar el programa, la variable Ram nos define cuánta es la cantidad de Ram que se utiliza para realizar dicho proceso, esta se mide en Gigabytes y el ordenador en la que se realizan la pruebas cuenta con una cantidad de 12.4 GB de Ram, finalmente la variable de tiempo es medida en segundos y nos muestra cuanto es el tiempo que demora la programación y la librería en cargar y procesar los datos ingresados, una vez son definidos los datos se procede a realizar la respectiva práctica de laboratorio en la que se definirá por comparativa que tipo de librería es la más eficiente.

Pandas.

```
import pandas as pd
flights_file1 = "/content/drive/MyDrive/flights/Combined_Flights_2018.parquet"
# flights_file2 = "/content/drive/MyDrive/flights/Combined_Flights_2019.parquet"
# flights_file3 = "/content/drive/MyDrive/flights/Combined_Flights_2020.parquet"
# flights_file4 = "/content/drive/MyDrive/flights/Combined_Flights_2021.parquet"
# flights_file5 = "/content/drive/MyDrive/flights/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)
```

Resultados	Pandas	
Archivos	RAM (GB)	Tiempo (seg)
1	6,4	19
2	8	14
3	4,6	7
4	9,9	20
5	7,4	8
1 y 2	10,1	32
4 y 5	8	22
3 y 2	8,8	32



Conclusión:

Como podemos evidenciar con los datos de la tabla enteros la librería de pandas ejecuta de forma eficiente los datos suministrados su tiempo de duración es relativamente bueno ya que se cuantifica en un rango de 12 seg aprox. cuando se trata de un solo archivo, y de 32 seg aprox. cuando de dos archivos se trata, en el caso que se desee cargar más de 2 archivos en repetidas pruebas el sistema se satura por ello no se toma en cuenta esta variable de más de 2 archivos en cuestión de la memoria esta sobrepasa más de la mitad dando como resultado una gran saturación en la Ram.

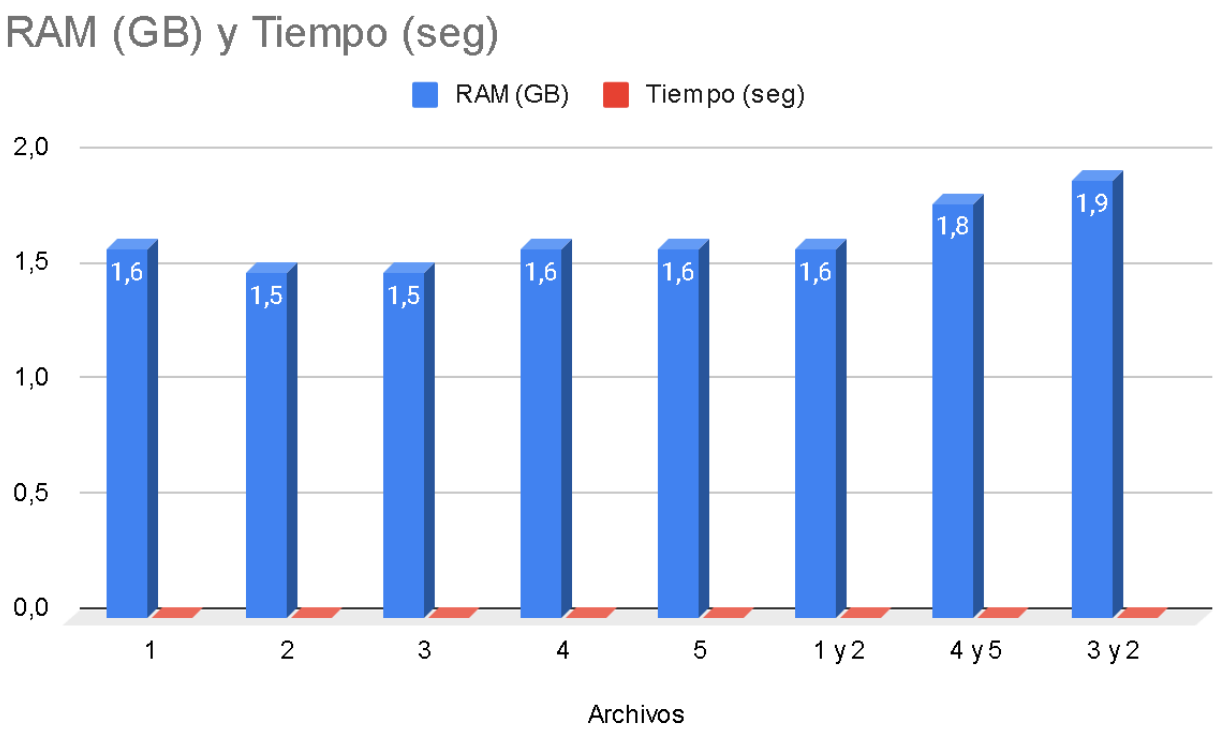
Polars.

```
[33] import polars as pl
```

```
flights_file1 = "/content/drive/MyDrive/flights/Combined_Flights_2018.parquet"
# flights_file2 = "/content/drive/MyDrive/flights/Combined_Flights_2019.parquet"
#flights_file3 = "/content/drive/MyDrive/flights/Combined_Flights_2020.parquet"
#flights_file4 = "/content/drive/MyDrive/flights/Combined_Flights_2021.parquet"
#flights_file5 = "/content/drive/MyDrive/flights/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)
```

Resultados	Polars	
Archivos	RAM (GB)	Tiempo (seg)
1	1,6	0
2	1,5	0
3	1,5	0
4	1,6	0

5	1,6	0
1 y 2	1,6	0
4 y 5	1,8	0
3 y 2	1,9	0



Conclusión:

Con la librería Polars se logra evidenciar un cambio muy favorable en el consumo de la memoria Ram ya que en el cargue de datos de uno y dos archivos no supera las 2GB de Ram en cuestión del tiempo podemos observar que es muy diferente a la de Pandas ya que está en un ciclo de 0 segundos independiente de cuántos archivos se procesan al mismo tiempo.

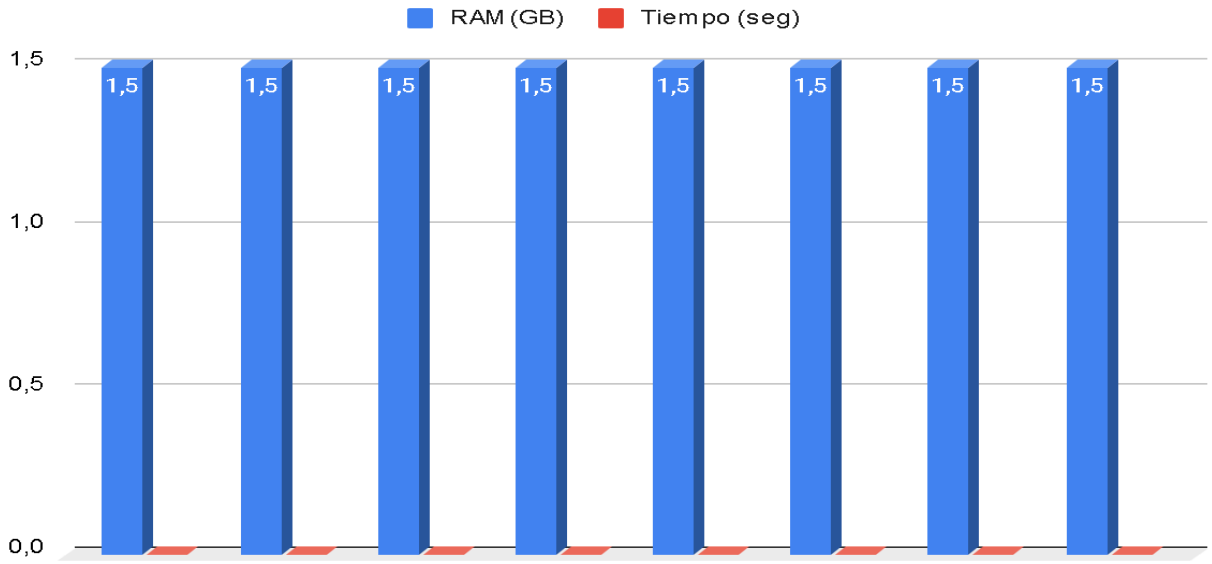
Spark.

```
flights_file1 = "/content/drive/MyDrive/flights/Combined_Flights_2018.parquet"
# flights_file2 = "/content/drive/MyDrive/flights/Combined_Flights_2019.parquet"
# flights_file3 = "/content/drive/MyDrive/flights/Combined_Flights_2020.parquet"
# flights_file4 = "/content/drive/MyDrive/flights/Combined_Flights_2021.parquet"
# flights_file5 = "/content/drive/MyDrive/flights/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)
```

```
[ ] # df_spark = df_spark1.union(df_spark2)
    # df_spark = df_spark.union(df_spark3)
    # df_spark = df_spark.union(df_spark4)
    # df_spark = df_spark.union(df_spark5)
```

Resultados	Spark	
Archivos	RAM (GB)	Tiempo (seg)
1	1,5	0
2	1,5	0
3	1,5	0
4	1,5	0
5	1,5	0
1 y 2	1,5	0
4 y 5	1,5	0
3 y 2	1,5	0



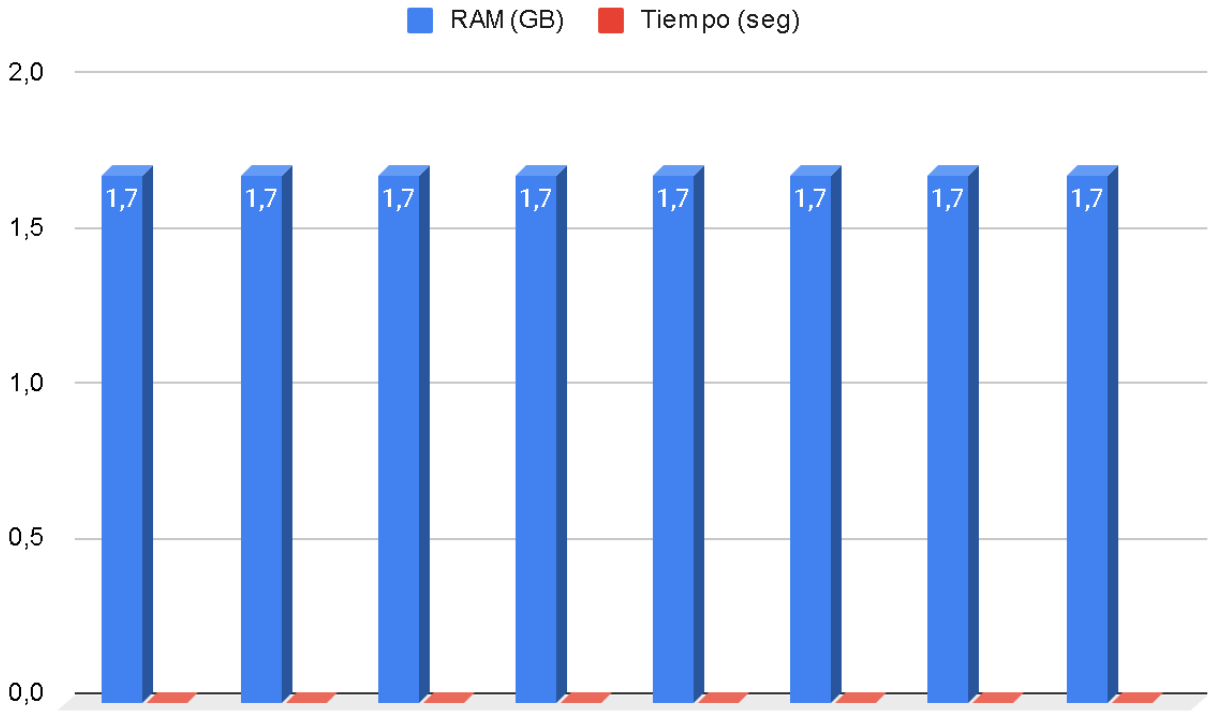
Conclusión:

En el caso de la librería Spark podemos ver como la compilación de los datos por unidad son muy buenos a comparación de las dos librerías anteriores no solo no consume nada de Ram si no que además es más rápido, ahora bien según las pruebas realizadas podemos inferir que con más de dos archivos cargados, este tipo de librería podría cargarlos de forma eficiente, ahora bien podemos observar que la mayor parte del tiempo la utilizo para llamar esta data “temp spark.parquet”.

Dask.

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

Resultados	Dask	
Archivos	RAM (GB)	Tiempo (seg)
1	1,7	0
2	1,7	0
3	1,7	0
4	1,7	0
5	1,7	0
1 y 2	1,7	0
4 y 5	1,7	0
3 y 2	1,7	0



Conclusión:

Al igual que ocurre en la librería de Spark la velocidad de cargue de los datos es muy rápida y a su vez el consumo de la Ram es menor que las dos primeras librerías ya sea esto con uno o dos archivos, por lo cual se logra determinar que en el cargue de más de dos datos lo puede hacer de forma eficiente, pero no es mejor que la librería de Spark ya que por 0.2 GB Spark es mas rapida.