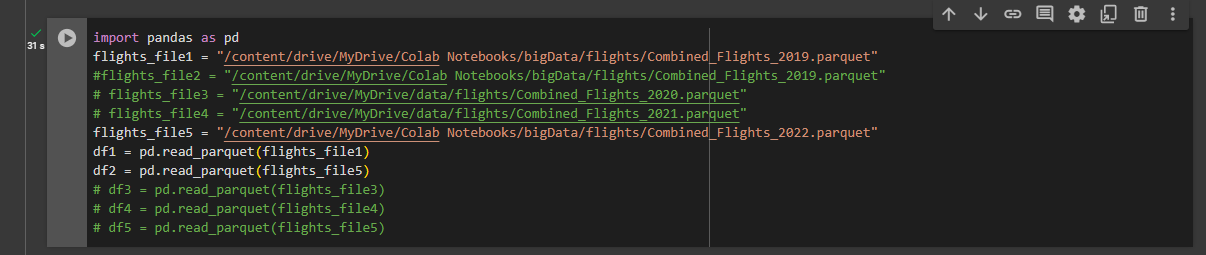
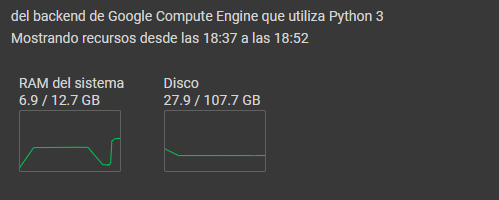
Roger Alberto Arias Triana - Taller 2

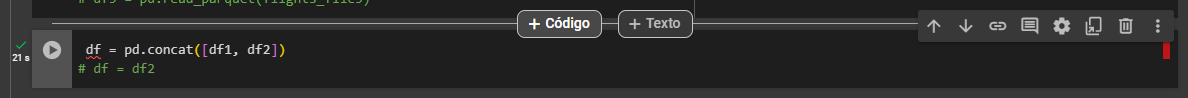
**Utilizando la librería Pandas**

Se cargan 2 archivos, tiempo de ejecución de carga: 31s



Consumo de recursos de máquina con 2 archivos:

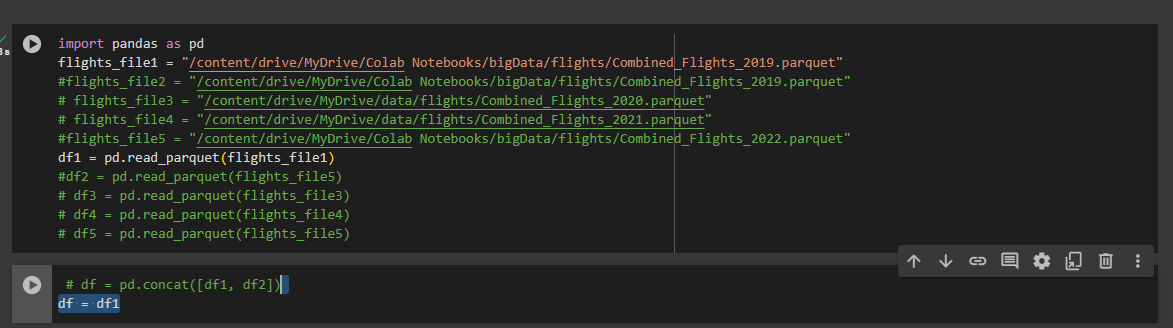


Tiempo de ejecución: 21s  


**Resultado en prueba:**  
Al concatenar los dos archivos con Pandas se consume toda la memoria ram posible, por tal motivo se usa solo un archivo sin concatenar

cargue de archivo

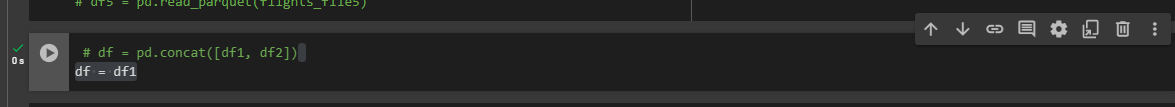
tiempo de ejecución: 28s desde las 18:37 hasta las 18:55



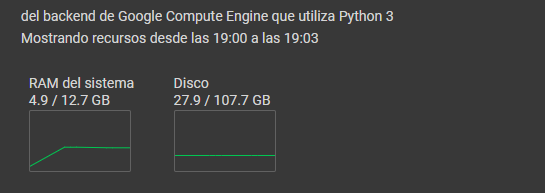
Consumo de recursos de máquina:



Concatenación: tiempo de ejecución: 0s

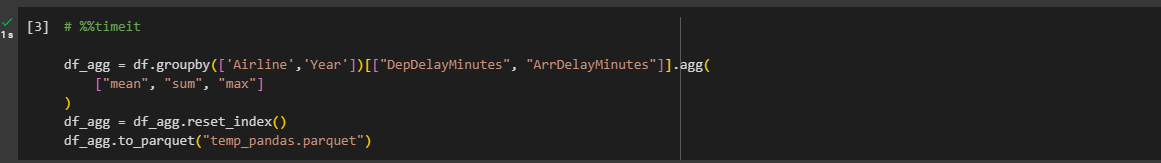


Consumo de recursos de máquina:

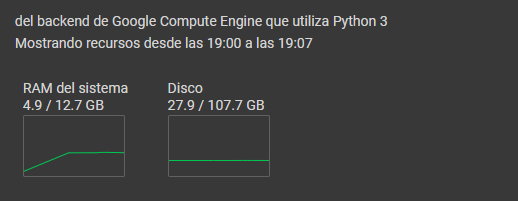


agrupación de la información carga

tiempo de ejecución: 1s

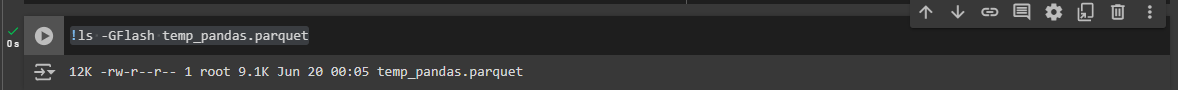


Consumo de recursos de máquina:

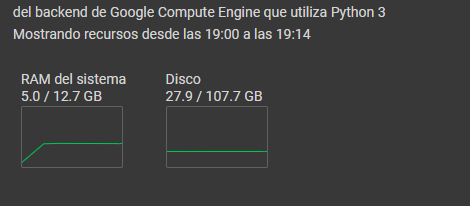


Enumeración de archivos cargados y guarde del dataFrame

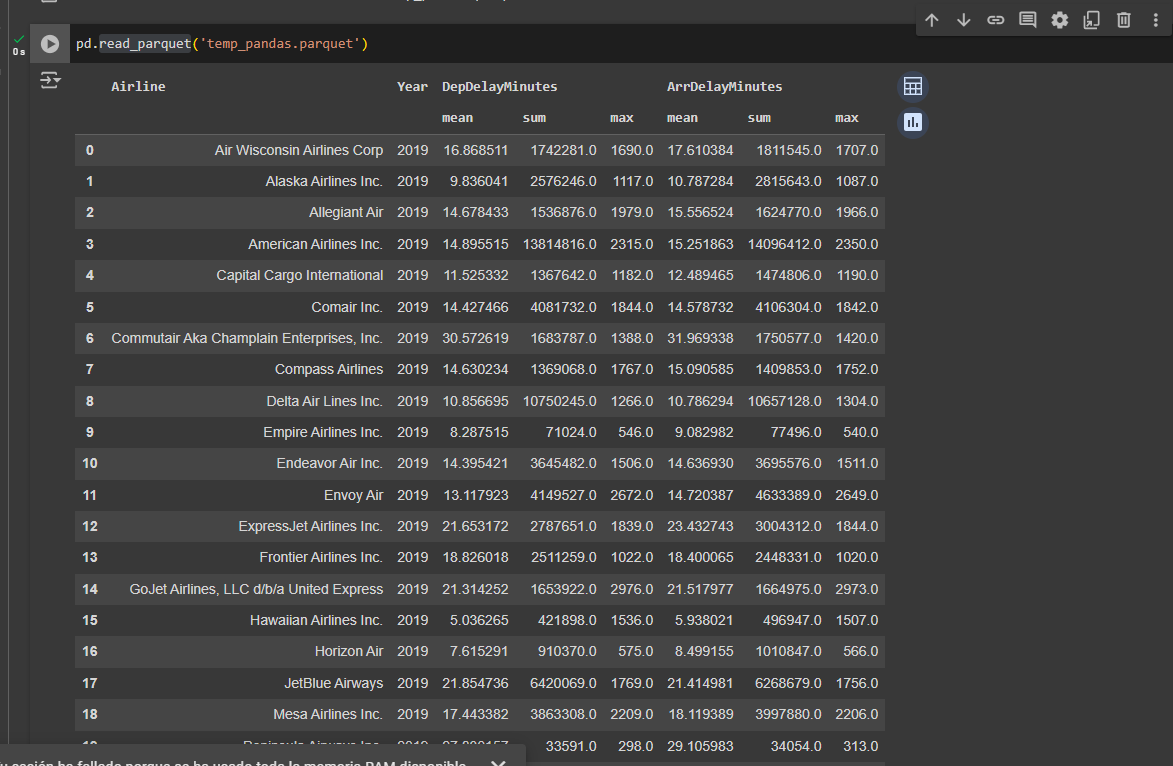
tiempo de ejecución: 0s

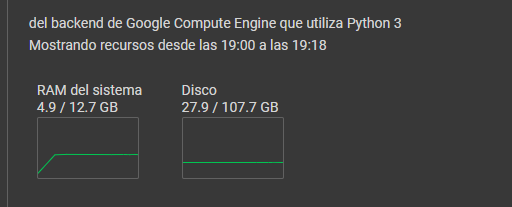
****

Consumo de recursos de máquina:



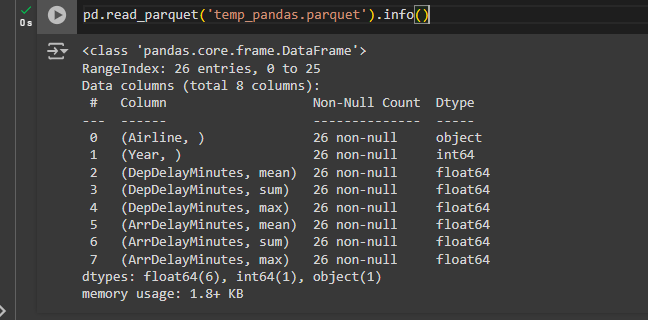
Lectura del archivo creado con Pandas

tiempo de ejecución: 0s ****

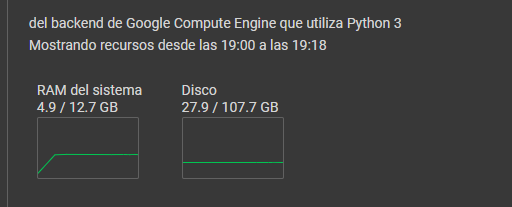
Consumo de recursos de máquina:

se lee el archivo y se saca un resumen simple del archivo

tiempo de ejecución: 0s

****

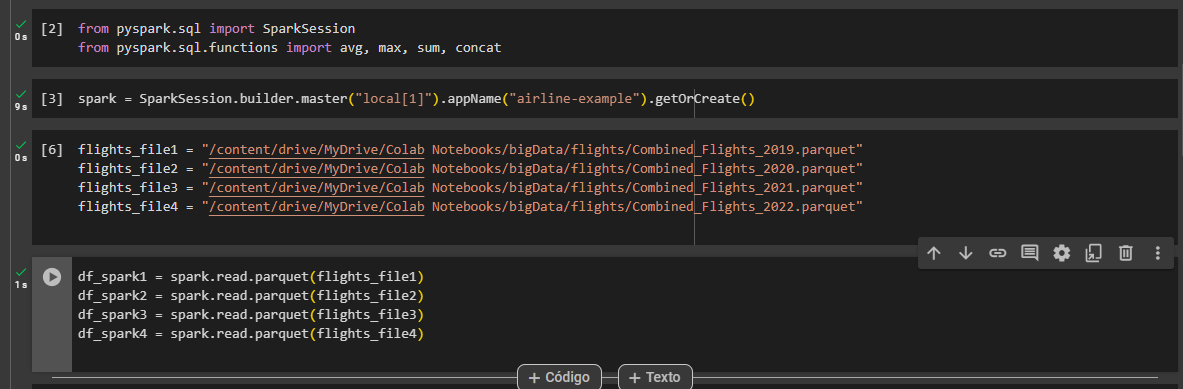
Consumo de recursos de máquina:



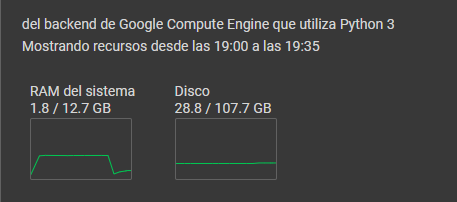
Uso de Spark:

cargue de los 4 archivos

tiempo de ejecución: 10s

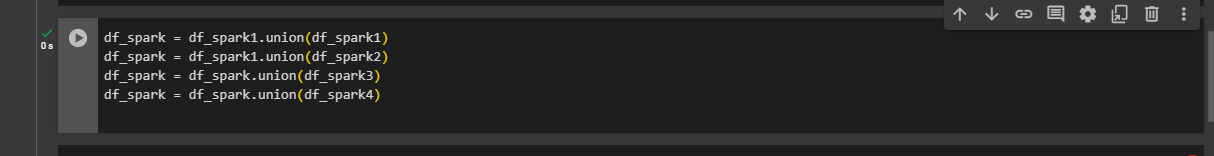


Consumo de recursos de máquina:



Se concatenan los 4 archivos

tiempo de ejecución: 0s

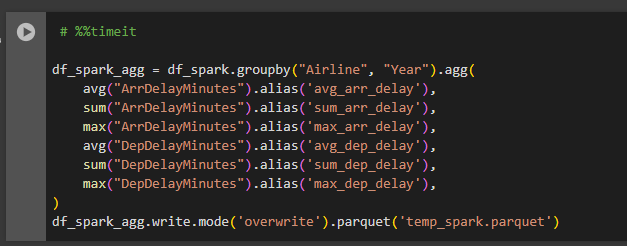


Consumo de recursos de máquina:



Se agrupa la información de los archivos:

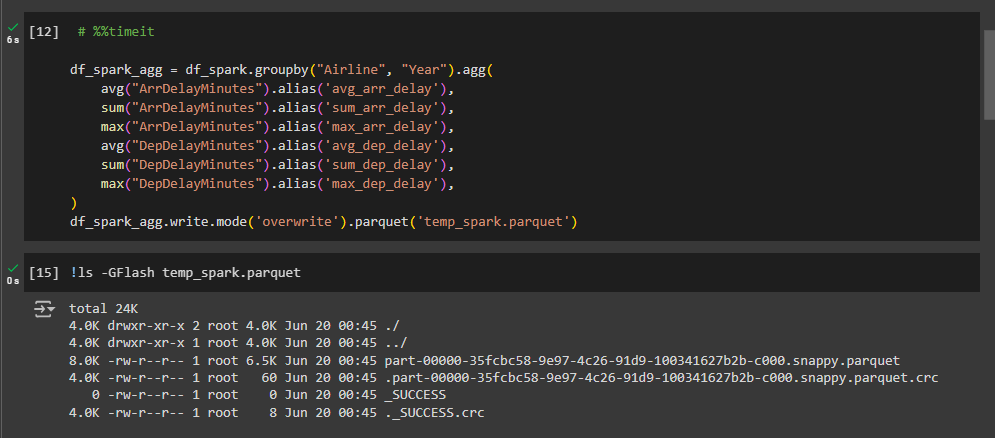
tiempo de ejecución: 6s



Consumo de recursos de máquina:



Se crea el archivo con Spark y se lee los archivos creados por la librería



Consumo de recursos de máquina:



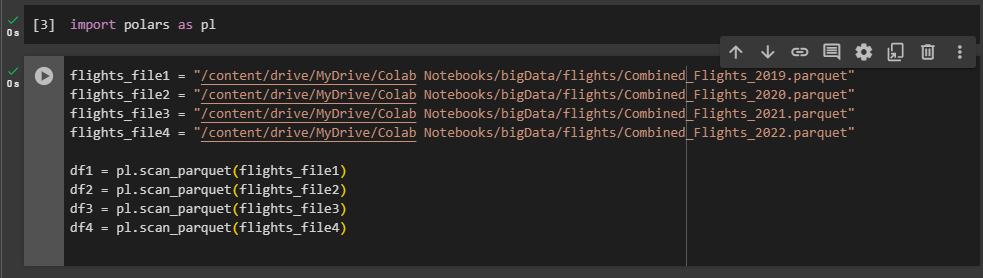
Resultado:

Se evidencia un menor tiempo de ejecución y un uso mucho menor de máquina el cual permite ejecutar los 4 archivos

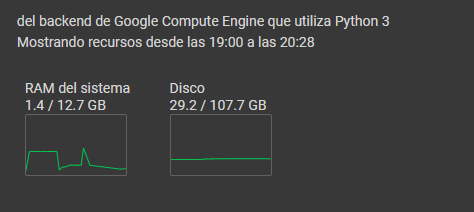
Uso de Polar:

se importa la librería y se cargan los archivos

tiempo de ejecución: 0s

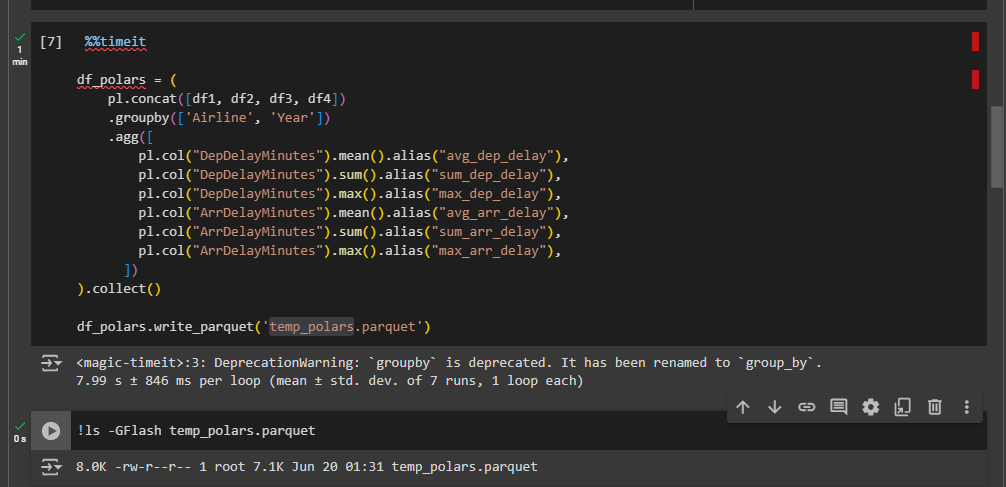
****

Consumo de máquina:

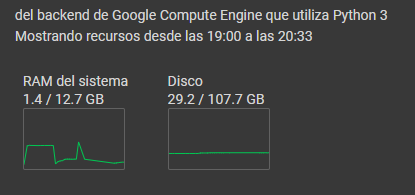
****

Se agrupan los archivos y los campos, posteriormente se ejecuta un resumen simple de los archivos generados

tiempo de ejecución: 1m

****

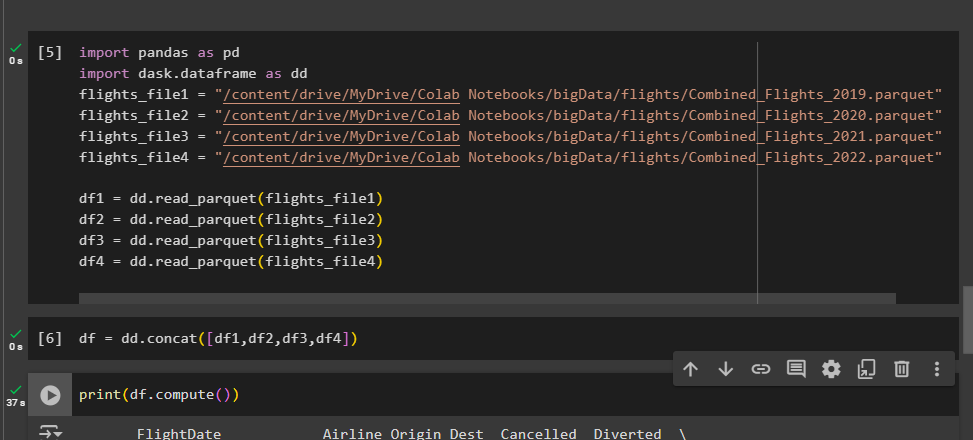
Consumo de recursos de máquina:



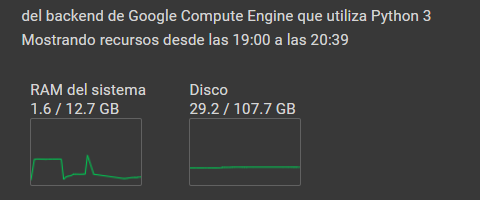
Uso de Dask:

Se importa la librería y se cargan los archivos

tiempo de ejecución: 0s

****

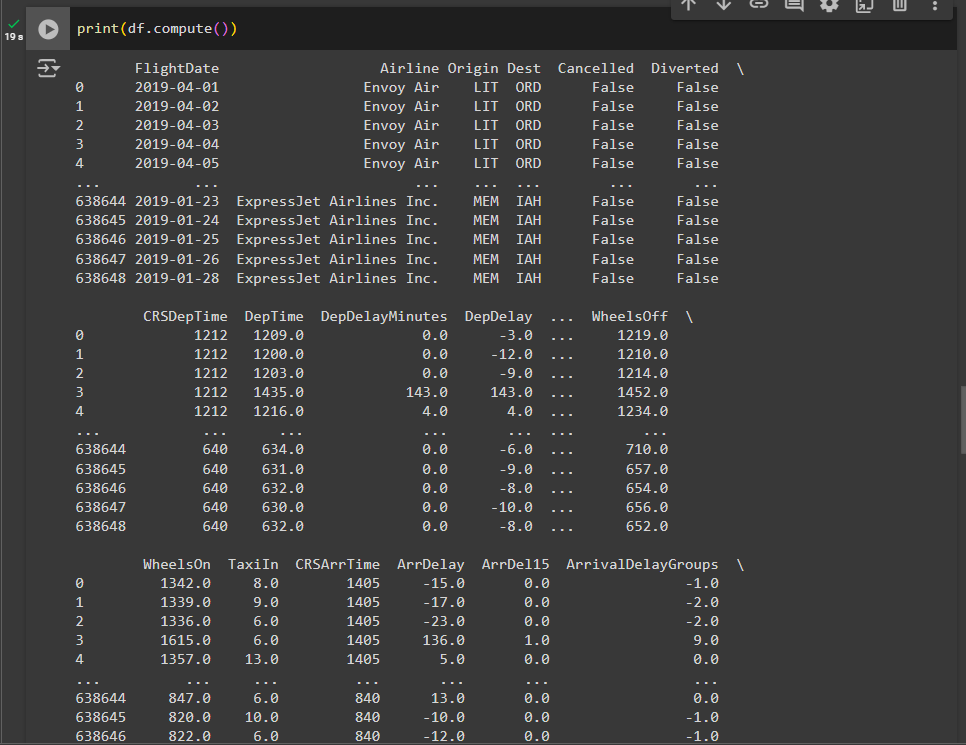
Consumo de recursos de máquina:

****

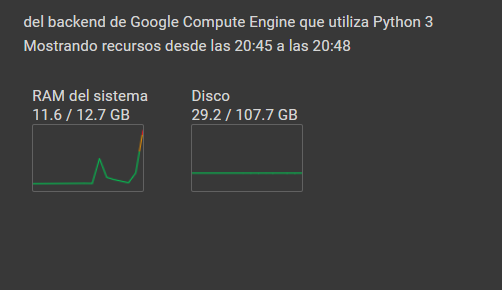
Resultado:

Alimprimir los resultados de los 4 archivos supera la capacidad máxima de hardware, igualmente con 3 archivos e incluso con 2 por lo que se implementa solo con 1 archivo

tiempo de ejecución: 19s



Consumo de recursos de máquina:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **librería** | **tiempo usado** | **Max maquina usada** | **Max archivos cargados** | **puntaje (1-10)** |
| Pandas | 29s | 5 gb | 1 | 5 |
| Polar | 60s | 1.4 gb | 4 | 7 |
| Spark | 16s | 2.1 gb | 4 | 9 |
| Dask | 19s | 11.6 gb | 1 | 6 |

Se evidencia que la mejor librería en cuestión de tiempos y eficiencia en cuanto a consumo de máquina es Spark ya que, aunque consume un poco más de máquina que Polar se demora mucho menos procesando los 4 archivos y los peores son Dask y Pandas con la diferencia que Dask si logra cargar los 4 archivos, pero supera el máximo consumo de hardware al momento de procesar la información e imprimirla.