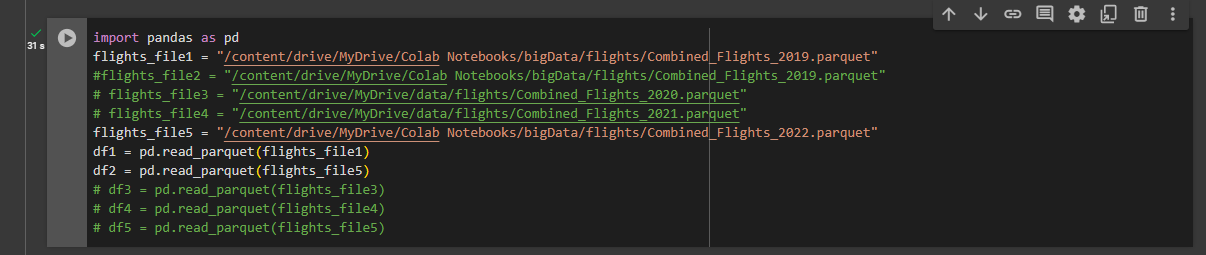
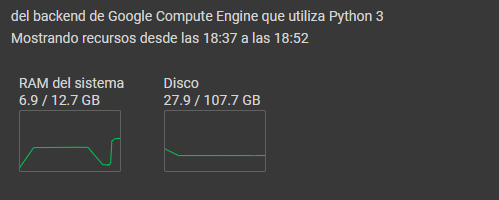
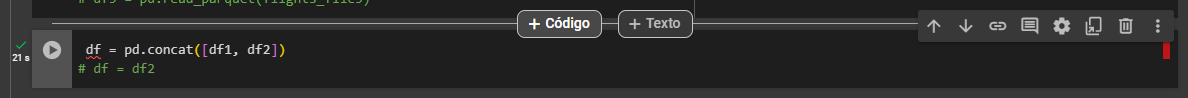
**uso de librería panda:**  
  
**cargue de archivos**

tiempo de ejecución de carga: 31s



Consumo de recursos de máquina con 2 archivos:



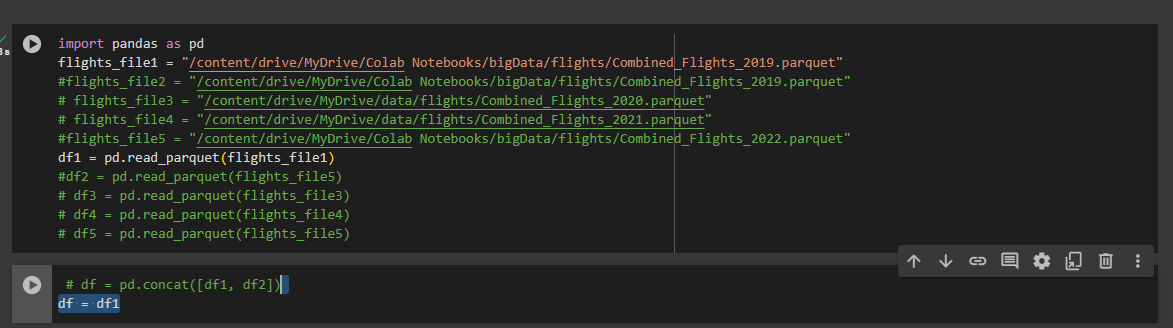
Tiempo de ejecución: 21s  


**Resultado en prueba:**  
se evidencia que al concatenar los dos archivos con panda consume toda la ram posible por ende se procede a usar solo un archivo sin concatenar

**Uso librería panda 1 archivo:**

**cargue de archivo**

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

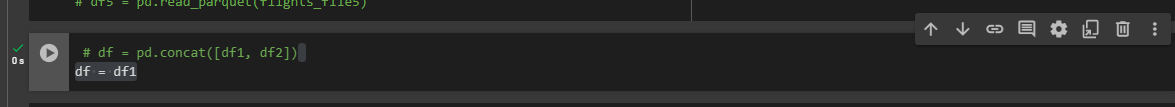


Consumo de recursos de máquina:

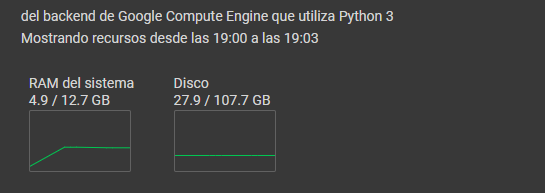


**concatenación**

tiempo de ejecución: 0s

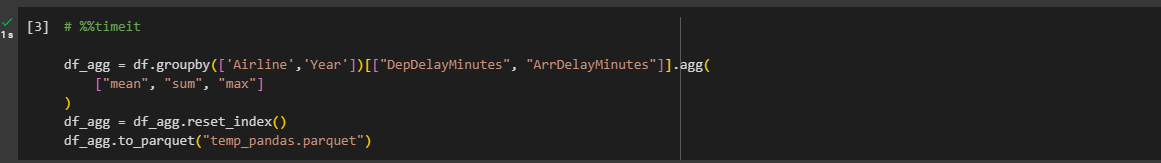


Consumo de recursos de máquina:

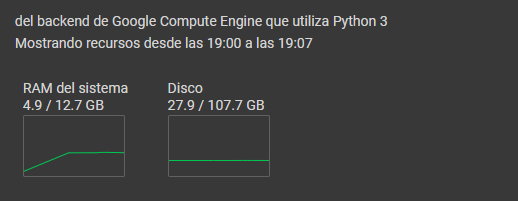


**agrupación de la información carga**

tiempo de ejecución: 1s

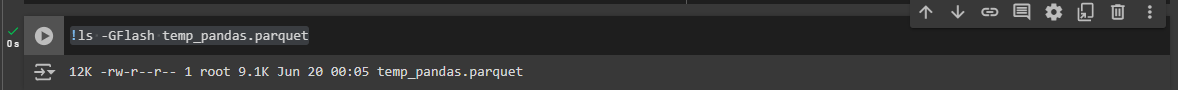


Consumo de recursos de máquina:

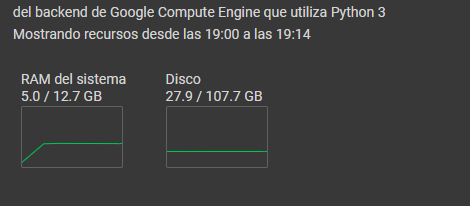


**Enumeración de lo archivos cargados y guarde del dataFrame**

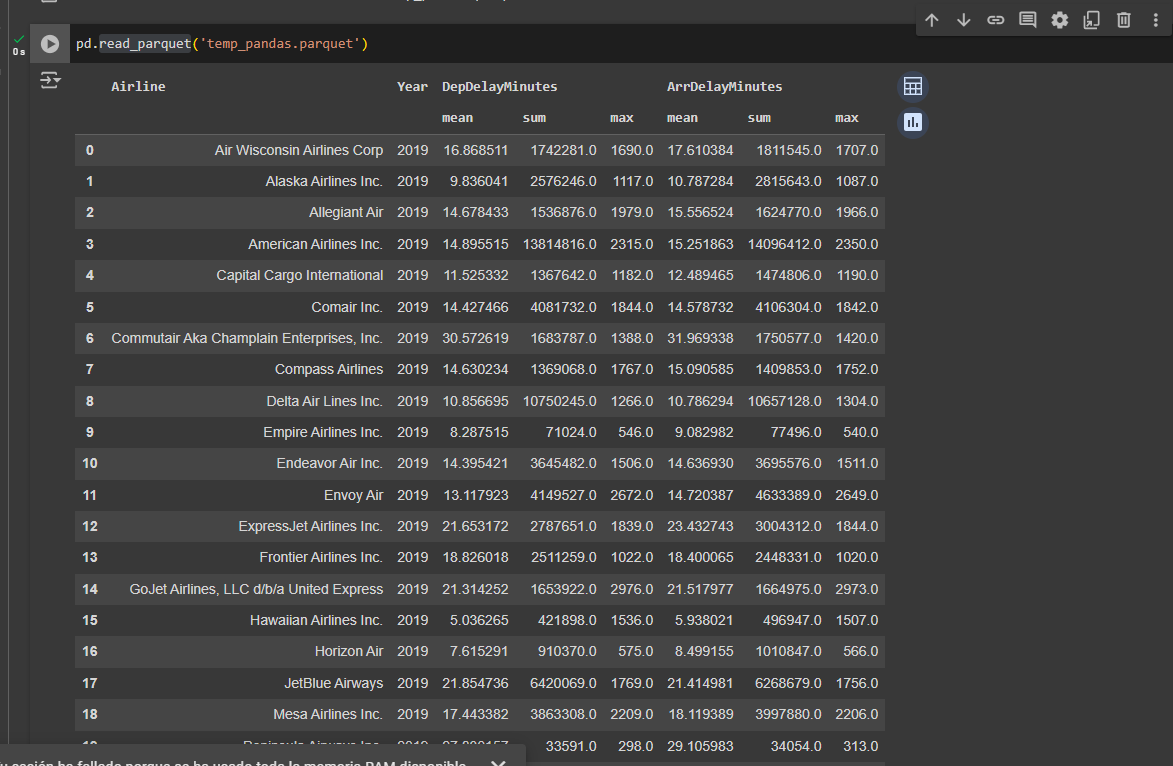
tiempo de ejecución: 0s

****

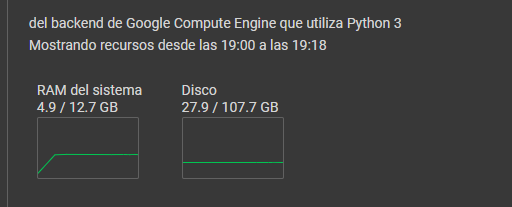
Consumo de recursos de máquina:



**lectura del archivo creado con panda**

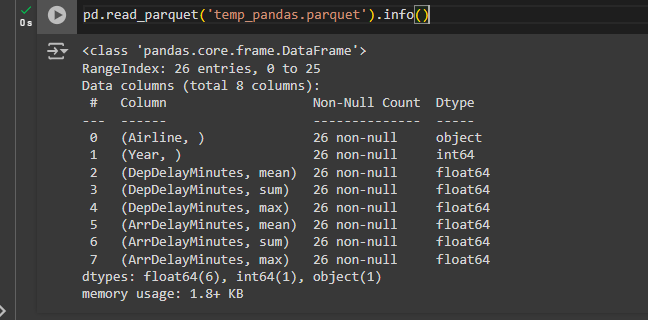
tiempo de ejecución: 0s ****

Consumo de recursos de máquina:

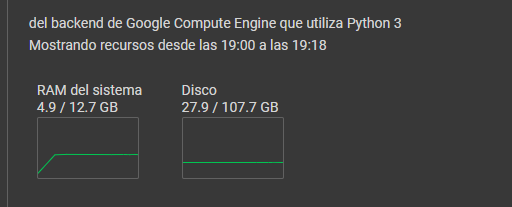


**se lee el archivo y se saca un resumen simple del archivo**

tiempo de ejecución: 0s

****

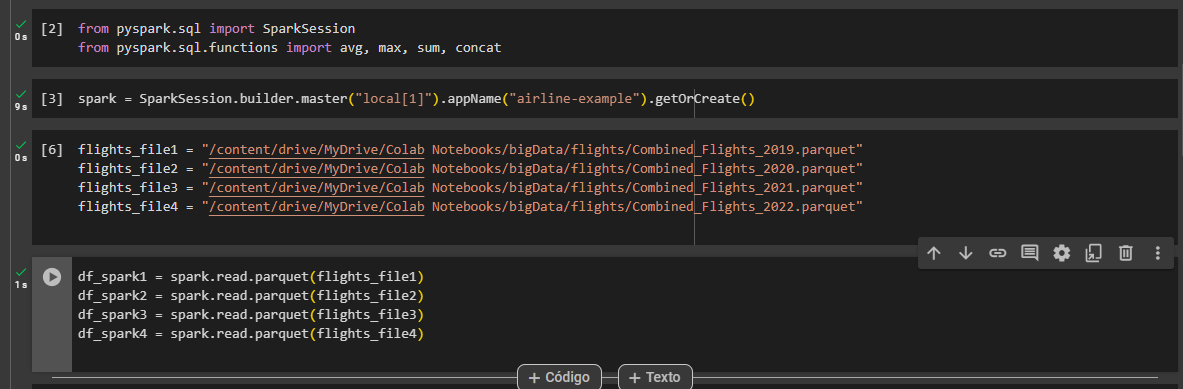
Consumo de recursos de máquina:



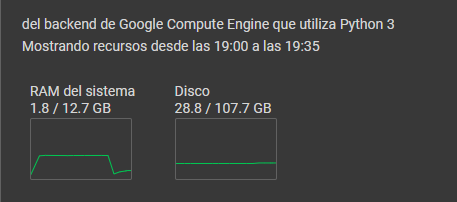
**uso de librería Spark:**

cargue de los 4 archivos

tiempo de ejecución: 10s

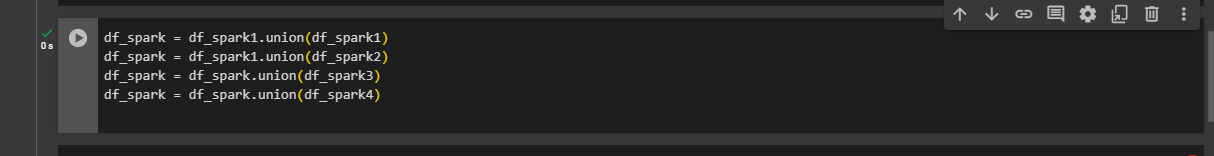


Consumo de recursos de máquina:



unión de los 4 archivos

tiempo de ejecución: 0s

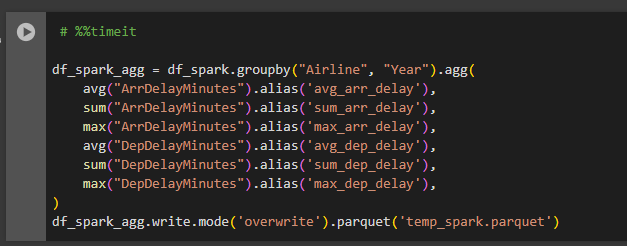


Consumo de recursos de máquina:



agrupación de la información de los archivos

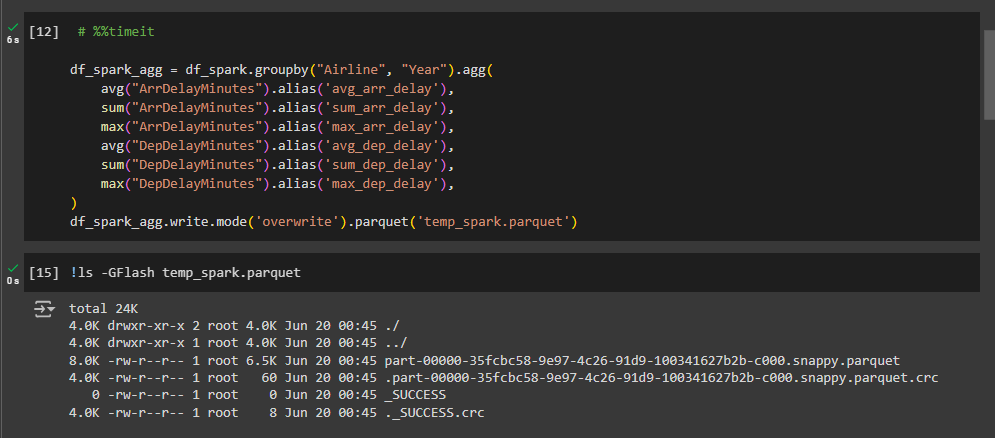
tiempo de ejecución: 6s



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:



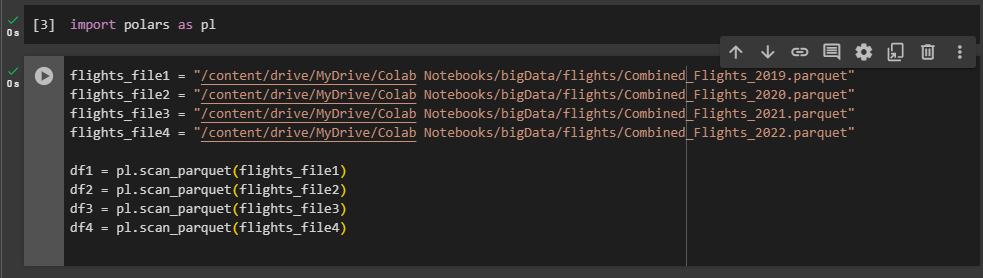
**Resultado de la prueba:**

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

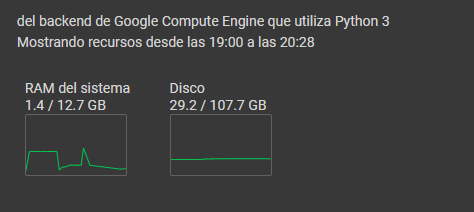
**uso de librería Polar:**

se importa la librería u se cargan los archivos

tiempo de ejecución: 0s

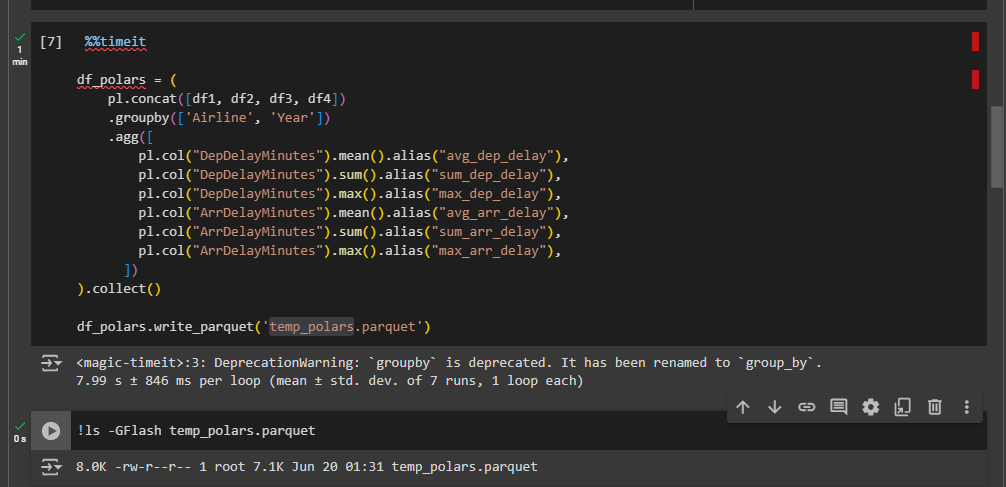
****

Consumo de recursos de máquina:

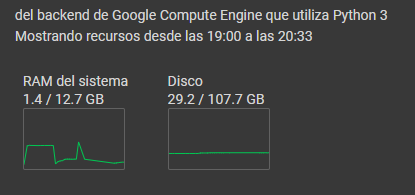
****

se agrupan los archivos y los campos después se ejecuta un resumen simple de los archivos generados

tiempo de ejecución: 1m

****

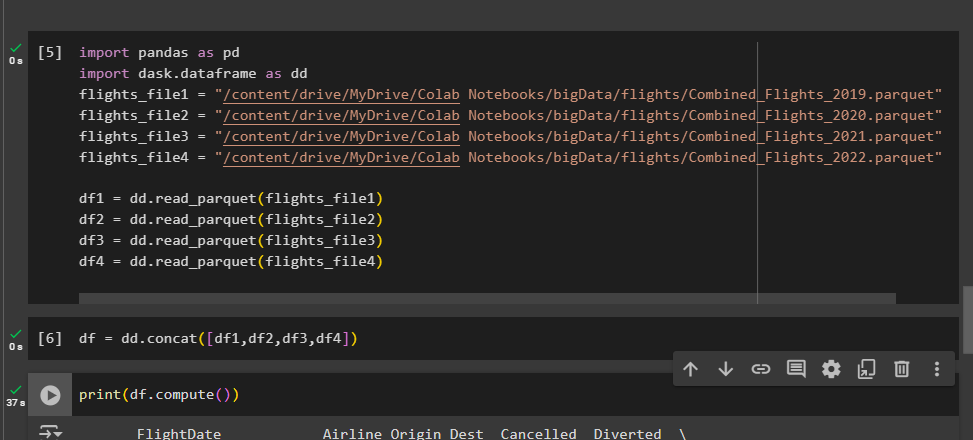
Consumo de recursos de máquina:



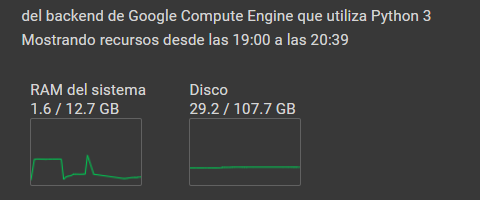
**uso de librería dask:**

se importa la librería y se cargan los archivos, también se agrupan los archivos

tiempo de ejecución: 0s

****

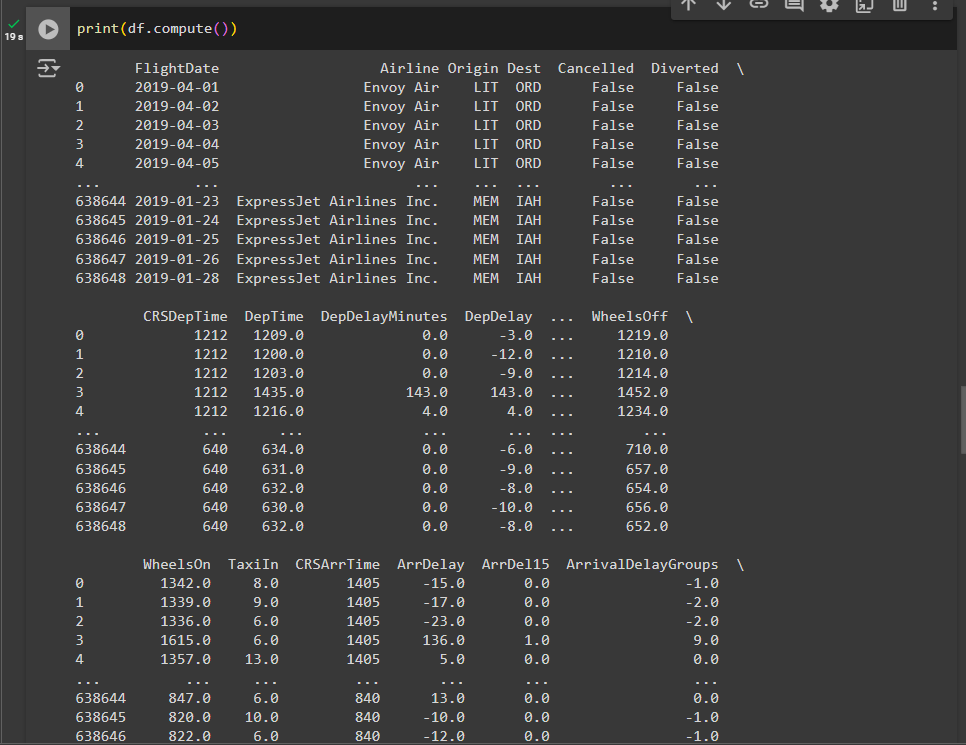
Consumo de recursos de máquina:

****

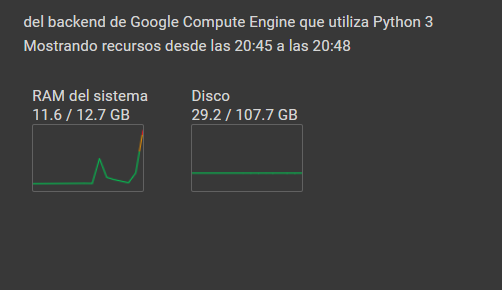
**Resultado de la prueba:**

alimprimir los resultados de los 4 archivos supera el máximo de la máquina y muere con 3 pasa tambien lo mismo 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 maquina que polar se demora mucho menos procesando los 4 archivos y los peores son dask y pandas con diferencia que dask si permite cargar los 4 archivos pero muere al momento de procesar la información e imprimirla.