**Comparación de procesamiento en Python y Mongo con el paradigma Map Reduce.**

Para el presente trabajo se toma el archivo Delito\_Violencia\_Intrafamiliar.csv que contiene 74402 registros. Este archivo contiene 20 campos o columnas pero se considera que las más importantes o relevantes para trabajar son:

DEPARTAMENTO,MUNICIPIO,DIA,BARRIO,ZONA,EDAD,SEXO,ESTADO CIVIL y ESCOLARIDAD

Los primeros campos a ser descartados son Código DANE, 2015 y el país. Estos campos resultan ser innecesarios.

Los campos FECHA, HORA, CLASE SITIO, ARMA EMPLEADA, MÓVIL AGRESOR, MÓVIL VICTIMA, CLASE EMPLEADO Y PROFESIÓN son algo relevantes, sin embargo, los nombres parecen ser muy prometedores pero no lo es tanto, ya que al indagar más, se ve que por ejemplo en el campo FECHA, hay muchos campos sin llenar, esto entorpece el procesamiento, además, lo único que aporta en realidad FECHA es el mes. Del mismo estilo son la mayoría de los otros campos, por ese motivo no son tenidos en cuenta.

En este trabajo se pretende generar resultados como:

Generar información del número de casos de violencia intrafamiliar que se presenten por municipio de acuerdo a un Departamento seleccionado y con el municipio, sacar los casos que se presenten en cada barrio, mostrar los porcentajes de los casos por días, zonas, estado civil, escolaridad y mostrar el número de casos por género en el municipio y el departamento.

**Alejandro, si quiere defina el tipo dato para cada campo y le pone valores de la escala que quiera, también si es necesario hace las preguntas y alguna descripción de los datos. Tengo dudas con algunos campos, por ejemplo, el SEXO es de la persona que cometió el acto de violencia y es la victima (hay que tener eso claro). Además, puede hacer todas las adecuaciones que considere necesarias de acuerdo con el documento que el profesor nos pasó.**

**METODOLOGÍA:**

La idea es generar la información antes mencionada, pero, además, para ese procesamiento se requiere tomar los tiempos para saber cuánto se toman Python y mongo en hacer el mapeo y la reducción.

Se toma el archivo de violencia intrafamiliar y se particiona en varios archivos, para esto se usa una función logarítmica que me permitirá establecer la cantidad máxima de archivos a crear del archivo completo. Como resultado final, se obtiene 13 archivos particionados, cada uno tendrá cierta cantidad de registros. Con los archivos particionados es que se hará el procesamiento. Por cada archivo se hace un procesamiento, por lo tanto, si quiero hacer mapeo y reducción sobre un campo en específico, por ejemplo, sacar el número de casos de violencia que cada departamento tuvo, en realidad salen 13 resultados y esos 13 resultados al sumarse dan el resultado esperado.

Todos los resultados del archivo se tienen que sumar para dar bien los resultados, para comprobar que los resultados estén bien, se hacen consultas con Mongo pero no con cada archivo particionado, sino con el archivo completo, esto hará que se evidencia que al partir el archivo, no se estén omitiendo ni sumando registros.

Por último, se toman los resultados en general y se muestra información según se especificó anteriormente, además, se mostrarán los gráficos pertenecientes a los tiempos de cada procesamiento tanto para Python como para Mongo y se hará el speed up de python con respecto a mongo para ver si hay alguna mejora.

Es de aclarar que a mongo le toma un tiempo conectarse a la base de datos, ese tiempo no hace parte del procesamiento, sin embargo, se ha dejado en los gráficos. Para tener algo más acertado, se ha medido el tiempo que mongo tarde en conectarse a la base de datos y subir un archivo de 20 registros con 3 campos y ha dado un resultado alrededor de 0,45 segundos. Entonces para los resultados que se vean de Mongo con respecto a los tiempos, habría que restar 0,45. Claro, este tiempo debe variar si se hacen pruebas en computadoras diferentes.

**Resultados de tiempos para 10 iteraciones:**





