Universidad Internacional de Valencia

Master en Big Data y Data Science

Problema 1: Gasto sin tarjeta de crédito

Procesamiento de datos masivos: Spark

Autor:
Adrián Hernández Padrón
Julio 2022

Universidad Internacional de Valencia

1. Código

Para empezar tenemos que leer los datos de entrada, esto lo hacemos iniciando la sesión del programa e indicandole la ruta del archivo con los datos la cual hemos guardado en la variable entrada. Entonces ya podemos trabajar con estos datos, el primer paso dentro

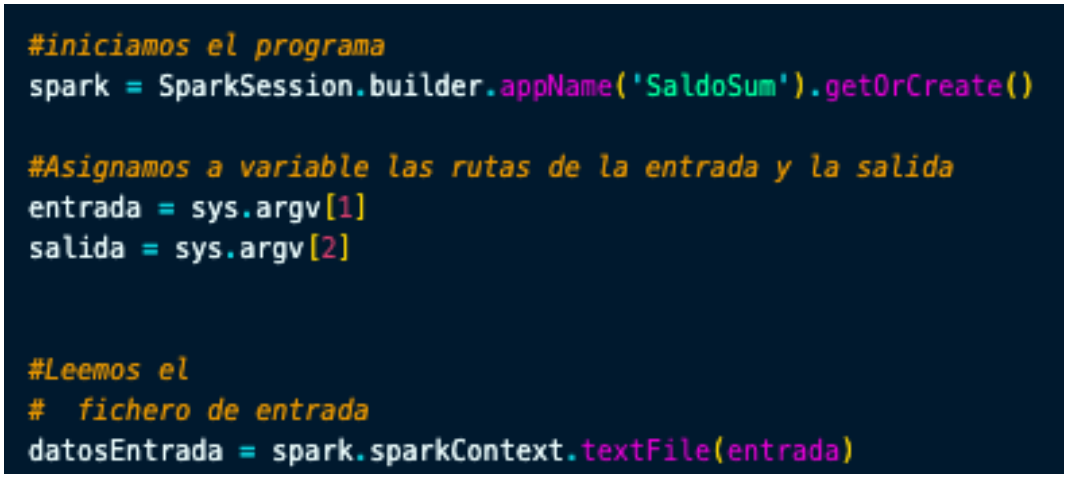


Figura 1: Con este código ya hemos cargado los datos de entrada guardados en el fichero entrada1.txt

de la función será crear una variable vacía, que en este caso llamamos clavevalor, donde guardaremos la respuesta del map y separar las lineas haciendo uso del .split() separando por el salto de línea. De esta manera en la nueva variable, new_linea, la información estará ya separada por filas. Cuando ejecutamos el bucle for, vamos recorriendo fila por fila y separando nuevamente entre el nombre, metodo y salto y guardamos nuestros pares clave-valor.

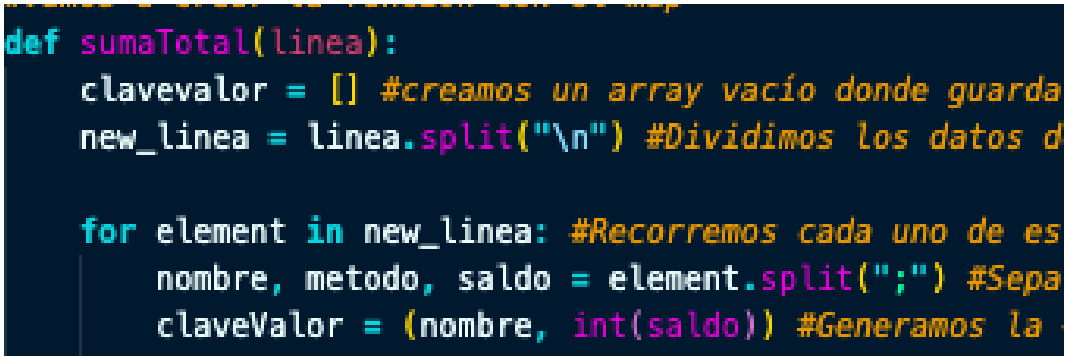


Figura 2: Vemos como guardamos nuestros pares clave-valor(saldo(str)-valor(int) los cuales serán la salida del ejercicio.)

Ya lo último que nos queda hacer dentro del map es separar la información según el método de pago, esto se hace con un if-else. Dentro del else, tratamos el caso de una persona que haya comprado unicamente con tarjeta de crédito. Con esto ya preparamos la salida de los datos del map, correctamente clasificados. Lo que queda ahora es sumar estos datos, esto lo hacemos haciendo uso del reduceByKey sobre la siguiente función. Por tanto solo queda aplicar el flatMap y el reduceByKey con sus correspondientes funciones para obtener la salida deseada, a esta salida le aplicaremos un map nuevamente para poder devolver la respuesta como se indicó en el enunciado. Por último, guardamos la salida haciendo uso de saveAsTextFile en la ruta introducida por consola.

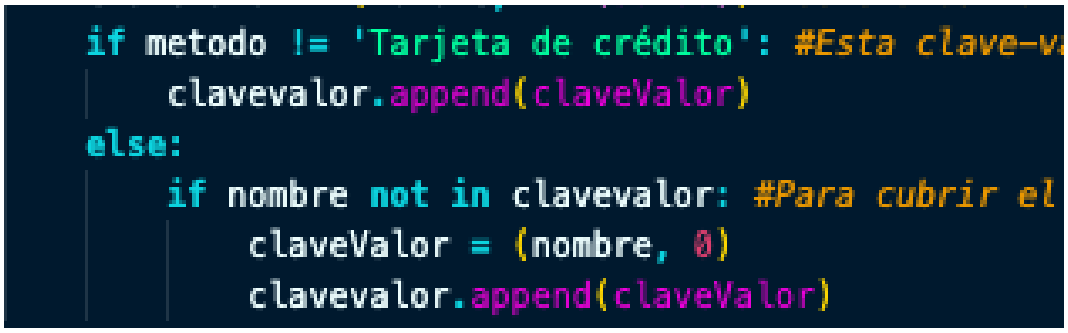


Figura 3: Dentro del else realizamos un if que agregará (Nombre,0) en caso de que el nombre no se encuentre en la variable de salida.

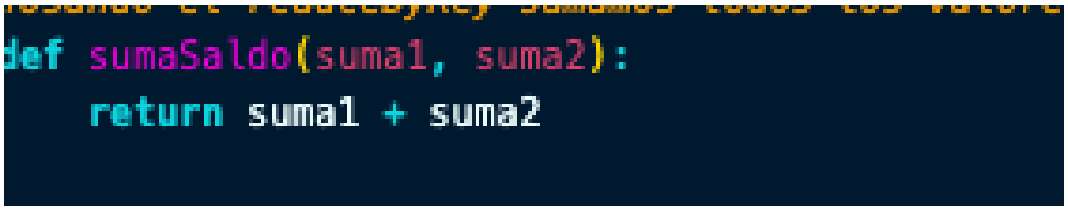


Figura 4: Función que ejecuta el reduceByKey para poder sumar todos los gastos según el nombre.

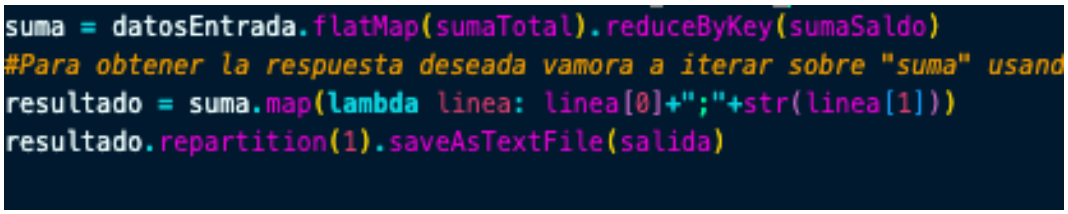
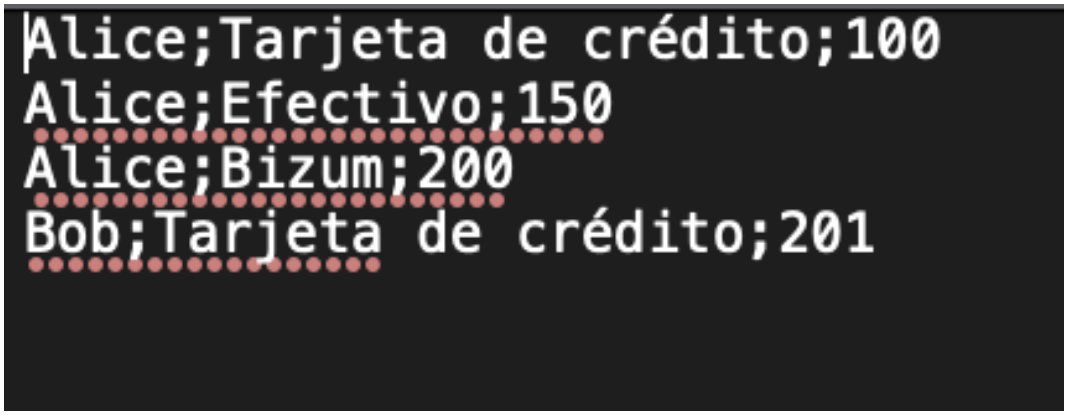


Figura 5: Al trabajar con pococs datos, guardamos la salida usando reparittio(1) para guardar la información solamente en una partición y asi evitar la creacion de particiones vacías.

2. Ejecución y resultados

Para la ejecución del programa, escribimos el siguiente código por la consola de comandos:
$spark-submit personaGastosSinTarjetaCredito.py file : /Users/adrihp/Master/MBID03/scriptsSpark/Probl$
$/Users/adrihp/Master/MBID03/scriptsSpark/Problema1/salida1$

Con la entrada:



La salida que nos devuelve el programa esta guardada en la carpeta salida1/part-00000: