

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

1. Código

Para empezar tenemos que leer los datos de entrada, esto lo hacemos iniciando la sesión del programa e indicándole 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

```
#iniciamos el programa
spark = SparkSession.builder.appName('SaldoSum').getOrCreate()

#Asignamos a variable las rutas de la entrada y la salida
entrada = sys.argv[1]
salida = sys.argv[2]

#Leemos el
# fichero de entrada
datosEntrada = spark.sparkContext.textFile(entrada)
```

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 líneas 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, método y salto y guardamos nuestros pares clave-valor.

```
def sumaTotal(linea):
    clavevalor = [] #creamos un array vacío donde guarda
    new_linea = linea.split("\n") #Dividimos los datos d

    for element in new_linea: #Recorremos cada uno de es
        nombre, metodo, saldo = element.split(";") #Sepa
        claveValor = (nombre, int(saldo)) #Generamos la
```

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 únicamente 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.

```
if metodo != 'Tarjeta de crédito': #Esta clave-valor
    clavevalor.append(claveValor)
else:
    if nombre not in clavevalor: #Para cubrir el
        claveValor = (nombre, 0)
        clavevalor.append(claveValor)
```

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.

```
def sumaSaldo(suma1, suma2):
    return suma1 + suma2
```

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

```
suma = datosEntrada.flatMap(sumaTotal).reduceByKey(sumaSaldo)
#Para obtener la respuesta deseada vamos a iterar sobre "suma" usando
resultado = suma.map(lambda linea: linea[0]+";"+str(linea[1]))
resultado.repartition(1).saveAsTextFile(salida)
```

Figura 5: Al trabajar con pocos datos, guardamos la salida usando repartitio(1) para guardar la información solamente en una partición y así evitar la creación 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/Problema1/salida1
```

Con la entrada:

```
Alice;Tarjeta de crédito;100
Alice;Efectivo;150
Alice;Bizum;200
Bob;Tarjeta de crédito;201
```

La salida que nos devuelve el programa esta guardada en la carpeta salida1/part-000000:

```
(base) adrihp@MacBook-Air-de-Adrian-2 Problema1 % cat salida1/part-000000
Alice;350
Bob;0
```