

## Problema 3

Dado un dataset que contenga entradas con la forma "persona;método\_pago;dinero\_gastado", crea un programa llamado personaYMetodosDePago que:

- Por cada persona indique en cuántas compras pagó más de 1500 euros con un medio de pago diferente a tarjeta de crédito. La solución se tiene que guardar en un archivo llamado comprasSinTDCMayorDe1500.
- Por cada persona indique en cuántas compras pagó menos o igual a 1500 euros con un medio de pago diferente a tarjeta de crédito. La solución se tiene que guardar en un archivo llamado comprasSinTDCMenorIgualDe1500.

Se valorará positivamente la eficiencia del programa, por ejemplo, no usar transformaciones innecesarias.

Ejemplo:

Entrada

Alice;Tarjeta de crédito;1000  
Alice;Tarjeta de crédito;1800  
Alice;Tarjeta de crédito;2100  
Bob;Bizum;2000  
Alice;Bizum;1000  
Bob;Tarjeta de crédito;1100

Salida (a)

Alice;0  
Bob;1

Salida (b)

Alice;1  
Bob;0

Notar que Alice y Bob solo hacen una compra con pago diferente a tarjeta de crédito.

### 3.1 Solución

- Creamos nuestro context spark
- Agregamos las rutas del archivo input y output

```
from pyspark import SparkContext, SparkConf

# Crear un contexto de Spark
conf = SparkConf().setAppName("problema3")
sc = SparkContext(conf=conf)

# Constantes
PATH_INPUT = './problema3/casoDePrueba3.txt'
PATH_OUT = './problema3/output'
```

- Leemos la data con método textFile
- Separamos la información por separador ';' y generamos lista de tuplas de rdd
- # 1. lectura de datos
- input\_rdd = sc.textFile(PATH\_INPUT)
-

```
- # 2. Procesamiento de datos
-
- # 2.1 Convirtiendo datos en tupla
- tuple_rdd = (input_rdd
-     .map(lambda line: line.split(";"))
-     .map(lambda x: (x[0], x[1], float(x[2]))))
- )
```

- Para las compras sin TDC mayores a 1500 generamos una función map que se encargue de realizar la tarea
- Finalmente agrupamos la información sumalizando los datos

```
def processComprasSinTDCMayorDe1500(rdd):
    """Procesa las compras sin TDC mayores a 1500"""

    # aplico funcion de mapeo
    process_rdd = rdd.map(lambda x: (x[0], 1 if x[1]!='Tarjeta de
    crédito' and x[2]>1500 else 0))

    # agrupamiento
    group_rdd = process_rdd.reduceByKey(lambda x,y: x+y)
    return group_rdd
```

- Para procesar las compras sin TDC con montos menores e iguales a 1500 se emplea función maper
- Agrupamos y sumizamos los datos

```
- def processcomprasSinTDCMenoroIgualDe1500(rdd):
-     """Procesa las compras sin TDC menores o iguales a 1500"""
-
-     # aplico funcion de mapeo
-     process_rdd = rdd.map(lambda x: (x[0], 1 if x[1]!='Tarjeta de
-     crédito' and x[2]<=1500 else 0))
-
-     # agrupamiento
-     group_rdd = process_rdd.reduceByKey(lambda x,y: x+y)
-     return group_rdd
```

- Se recupera rdd de las funciones y por último escribimos resultados en rutas definidas

```
# 2.2 Procesando compras sin TDC mayores a 1500
compras_rdd = processComprasSinTDCMayorDe1500(tuple_rdd)

# 2.3 Procesando compras sin TDC menores o iguales a 1500
reduce_rdd = processComprasSinTDCMenorIgualDe1500(tuple_rdd)

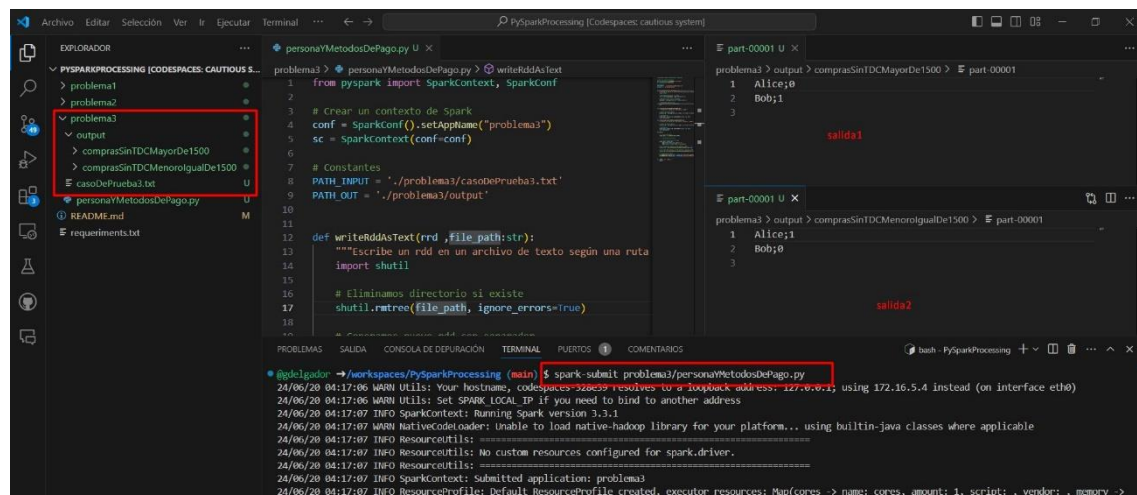
# 3. Escritura de datos
writeRddAsText(compras_rdd, f"{PATH_OUT}/comprasSinTDCMayorDe1500")
writeRddAsText(reduce_rdd,
f"{PATH_OUT}/comprasSinTDCMenorIgualDe1500")
```

## 3.2 Ejecución Programa

Se ejemplifica comando de ejecución desde el repositorio

```
spark-submit problema3/ personaYMetodosDePago.py
```

## 3.3 Evidencia Ejecución



```

@helgader: ~/workspaces/PySparkProcessing (main) $ spark-submit problema3/personaYMetodosDePago.py
24/06/20 04:17:06 WARN Utils: Your hostname, codespaces-92859 resolves to a loopback address: 127.0.0.1; using 172.16.5.4 instead (on interface eth0)
24/06/20 04:17:07 INFO SparkContext: Running Spark version 3.3.1
24/06/20 04:17:07 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
24/06/20 04:17:07 INFO ResourceUtils: =====
24/06/20 04:17:07 INFO ResourceUtils: No custom resources configured for spark.driver.
24/06/20 04:17:07 INFO ResourceUtils: =====
24/06/20 04:17:07 INFO SparkContext: Submitted application: problema3
24/06/20 04:17:07 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 1, script: , vendor: , memory ->

```