Objetivo de la práctica

Queremos realizar un programa en python con Spark que muestre cuál es el personaje más popular de los comic de Marvel, entendiendo popularidad como el personajes que coaparece más veces en los comic.

Dataset utilizado

Para ello nos disponemos de dos dataset, Marvel-names.txt y Marvel-graph.txt:
Marvel-names.txt :

- En dicho dataset encontramos un listado de nombre de héroes e id que corresponden con dichos héroes.

Marvel-graph.txt:

- En dónde el primer número es el identificador de un personaje/superhéroe, y a continuación todos los IDs de otros personajes con los que coaparece en la historia a la que hace referencia esa línea.

Descripción del código

Lo primero que vamos a realizar es importar las librerías que necesitamos para trabajar con spark en python y para poder recuperar los parámetros introducidos por terminal

```
from pyspark import SparkConf, SparkContext
import sys
```

Una vez importado las librerías que necesitamos vamos a realizar es una función que lea el fichero Marvel-names.txt y devuelve un lista, de heroes ordenados por id.

```
def loadHeroesNames():
    heroesNames = {}
    with open("Marvel-names.txt") as f:
        for line in f:
            fields = line.split('"')
            heroesNames[int(fields[0])] = fields[1]
    return heroesNames
```

- Abrimos el fichero, y por cada línea añadimos un héroe a nuestro listado de heroesName.

Creamos un objetos conf que es de tipo SparkConf() donde le introducimos un nombre para nuestra App en nuestro caso sera Heroes_Popular y donde queremos que se ejecute el programa, en nuestro caso en local.
Creamos un objeto SparkCotext el cual utilizamos para crear el RDD de los nombre de los heroes con la función anteriormente creada

```
conf = SparkConf().setMaster("local").setAppName("Heroes_Popular)
sc = SparkContext(conf = conf)

nameDict = sc.broadcast(loadHeroesNames())
```

- Guardamos el RDD de los nombre los heroes y su id en nameDict

Abrimos el archivo Marvel-graph.txt y lo guardamos como un RDD en lines.

Sobre dicho RDD lines vamos a proceder a realizar las transformaciones.

```
lines = sc.textFile("Marvel-graph.txt")

heroesNumConcurrencia = lines.map(lambda x: (int(x.split()[0]), len(x.split())-1))#  Modificamos el RDD para que

heroesCounts = heroesNumConcurrencia.reduceByKey(lambda x, y: x + y) # Agrupamos por ID de personaje por si hubi

flipped = heroesCounts.map( lambda x : (x[1], x[0])) # Para poder ordenar por popularidad devemos hacer un flip

sortedHeroes = flipped.sortByKey(ascending= False) # Ordenamos por popularidad de manera descendente 1 posicion

sortedHeroesWithNames = sortedHeroes.map(lambda countHeroes : (nameDict.value[countHeroes[1]], countHeroes[0]))
lstPopuMarvel=sortedHeroesWithNames.collect() # Convertimos el objeto RRD Spark a una lista de python
```

- Primera transformación HeroesNumConcurrencia
  - Realizamos un map por cada línea del RDD anterior se ejecuta la función lambda x-> dicha función modifica el RDD para que sea de tipo diccionario clave ID heroe, número de héroes que aparece con el en la saga (para obtener el número de personajes que aparecen en la línea realizamos un len(x.split()-1) es decir por cada línea separamos por espacio en blanco y devolvemos el tamaño de la lista a la cual le restamos 1 por el Héroe principal del cómic)
- Segunda trasformación HeroesCount
  - Realizamos un reduceByKey para evitar que hubiera personajes repetidos y en el caso de que los hubiera sumamos la popularidad del héroe ejemplo (3,10), (3,1)= (3,11)
- Tercera transformación flipped
  - Como su nombre indica vamos a realizar un intercambio de posiciones entre la id y la popularidad (id posicion 0 ahora sera la posicion 1 y la popularidad se encontraba en la posición 1 y ahora será la posición 0)
  - Dicha acción la realizamos para poder ordenar los datos extraídos
- Cuarta transformación sortedHeroes
  - Ordenamos la lista de más popular a menor popular
- Quinta trasformacion sortedHeroesWithNames
  - Trasformamos el id del personaje a su nombre para que sea más sencillo interpretar los datos por un humano
- Sexta trasformacion lstPopuMarvel
  - Realizamos un collect() para transformar el objeto de un RDD de spark a una lista de python

Por último mostramos los datos obtenidos

```
numeroSuperheroes=int(sys.argv[1])
#print(lstPopuMarvel[:int(numeroSuperheroes)])
if(numeroSuperheroes<len(lstPopuMarvel) and (numeroSuperheroes>0)):
    print ( lstPopuMarvel[:numeroSuperheroes])
```

- Para ello recuperamos el valor que nos introdujeron por parámetro posicional al ejecutar el código, transformamos dicho valor a entero y lo guardamos en la variable numeroSuperheroes
- Comprobamos que el valor es numérico y que se mayor que 0 y menor que la longitud de la lista de héroes
- Si es así mostramos el número de héroes según el parámetro introducido 12 = lista de los 12 heroes mas populares

Ejemplo de ejecución

Guardamos el código anterior en nuestra carpeta de spark como Marvel.py

En IOs, abrimos un terminal en la carpeta de spark y ejecutamos el siguiente comando *sudo spark-submit Marvel.py 12*

```
MacBook-Pro-de-Alfredo-2:Apache-Spark-codigo adrianGallego$ sudo spark-submit Marvel.py 12
```

y obtenemos un listado con los 12 superhéroes más populares del mundo marvel y el número de superhéroes con los que ha coaparecido

```
[('CAPTAIN AMERICA', 1933), ('SPIDER-MAN/PETER PAR', 1741), ('IRON MAN/TONY STARK ', 1528), ('THING/BENJAMIN
 J. GR', 1426), ('WOLVERINE/LOGAN ', 1394), ('MR. FANTASTIC/REED R', 1386), ('HUMAN TORCH/JOHNNY S', 1371),
('SCARLET WITCH/WANDA ', 1345), ('THOR/DR. DONALD BLAK', 1289), ('BEAST/HENRY &HANK& P', 1280), ('VISION ',
1263), ('INVISIBLE WOMAN/SUE ', 1244)]
```