PráCtica sesión 1 Spark -

Ejercicios RDDs

El objetivo de este documento es poner en práctica lo que se ha ido viendo durante la sesión número 1 de Spark, donde hemos practicado con las operaciones sobre RDDs.

Acciones y transformaciones sobre RDDs simples.

**Índice**

[Ejercicio propuesto 1 2](#__RefHeading___Toc614_2103982880)

[Ejercicio propuesto 2 3](#__RefHeading___Toc616_2103982880)

[Ejercicio propuesto 3. 4](#__RefHeading___Toc618_2103982880)

[Ejercicio propuesto 4 5](#__RefHeading___Toc620_2103982880)

[Ejercicio propuesto 5 6](#__RefHeading___Toc622_2103982880)

[Ejercicio propuesto 6 6](#__RefHeading___Toc624_2103982880)

## Ejercicio propuesto 1

A partir de la lista siguiente:

['Alicante','Elche','Valencia','Madrid','Barcelona','Bilbao','Sevilla']

1. Usando lo que hemos aprendido hasta ahora, quédate sólo con las ciudades que tengan la letra 'e' en su nombre y muestralas
2. Muestra las ciudades que tienen la letra ‘e’ y muestra el número de veces que aparece en cada nombre. Por ejemplo (Elche,2)
3. Podrías quedarte solo con las ciudades que tengan una sola e?
4. Han pasado una nueva lista pero no han separado bien las ciudades.. podrias volver a contar cuantas e hay en cada ciudad?

ciudades\_mal = [['Alicante.Elche','Valencia','Madrid.Barcelona','Bilbao.Sevilla'],['Murcia','San Sebastián','Melilla.Merida']]

|  |  |
| --- | --- |
| SOLUCIÓN a) | llistaCiutats = ['Alicante','Elche','Valencia','Madrid','Barcelona','Bilbao','Sevilla']  rdd\_llista = sc.parallelize(llistaCiutats)  # Exercici 1.1  #res1=list(filter(lambda x: x.find('e')>=0,llistaCiutats))  #res1  rdd\_res1 =rdd\_llista.filter(lambda x: x.find('e')>=0)  rdd\_res1.collect() |
| SOLUCIÓN b) | # Exercici 1.2  #res2=list(map(lambda x: print("(" + x + "," + str(x.lower().count('e')) + ")"),llistaCiutats))  rdd\_res2 = rdd\_res1.map(lambda x: (x,x.lower().count('e')))  rdd\_res2.collect() |
| SOLUCIÓN c) | # Exercici 1.3  #res3=list(filter(lambda x: x.lower().count('e')==1,llistaCiutats))  #res3  rdd\_res31 = rdd\_res2.filter(lambda x: x[1]==1)  rdd\_res31.collect()  #rdd\_res3 = rdd\_llista.filter(lambda x: x.lower().count('e')==1)  #rdd\_res3.collect() |
| SOLUCIÓN d) | ciutatsMal = [['Alicante.Elche','Valencia','Madrid.Barcelona','Bilbao.Sevilla'],['Murcia','San Sebastián','Melilla.Merida']]  #res = list(map(lambda y: print("(" + y + "," + str(y.lower().count('e')) + ")") , sum(map(lambda x: x.split('.'),sum(ciutatsMal,[])),[])))  rdd\_llistaMal = sc.parallelize(ciutatsMal)  rdd\_res4 = rdd\_llistaMal.flatMap(lambda x: x).flatMap(lambda y: y.split('.')).map(lambda z: (z,z.lower().count('e')))  rdd\_res4.collect() |

## Ejercicio propuesto 2

Dada una lista de nombres

Lista de nombres:

Juan,Jimena, Luis, Cristian, Laura, Lorena,Cristina, Jacobo, Jorge

Agruparlos según su inicial

*Por ejemplo de la lista: Laura, Luis, Juan el resultado sería:*

*L Laura,Luis*

*J Juan*

|  |  |
| --- | --- |
| SOLUCIÓN | llistaNoms = ['Juan','Jimena', 'Luis', 'Cristian', 'Laura', 'Lorena','Cristina', 'Jacobo', 'Jorge']  rdd\_noms = sc.parallelize(llistaNoms)  rdd\_separat = rdd\_noms.groupBy(lambda x: x[0])  rdd\_separat.collect() |

b) De la lista original obtén una muestra sin repetir valores

|  |  |
| --- | --- |
| SOLUCIÓN | rdd\_noms.sample(False,0.3).collect() |

c) Devuelve una muestra de datos de aproximadamente la mitad de registros que la lista original con datos que no se repitan

|  |  |
| --- | --- |
| SOLUCIÓN | rdd\_noms.sample(False,0.5).collect() |

## Ejercicio propuesto 3.

Dada las listas:

Inglés: hello, table, angel, cat, dog, animal,chocolate,dark,doctor,hospital,computer

Español: hola,mesa,angel,gato,perro,animal,chocolate,oscuro, doctor,hospital,ordenador

Usando las transformaciones que hemos aprendido realiza las siguientes tareas:

1. Obtén las palabras que se dicen igual en inglés y en español
2. Obtén las palabras que en español son distintas que en inglés
3. Obtén una única lista con las palabras en ambos idiomas, que son distintas entre ellas
4. Haz dos grupos con todas las palabras, uno con las que empiezan por vocal y otro que empiecen por consonante

|  |  |
| --- | --- |
| SOLUCIÓN a) |  |
| SOLUCIÓN b) |  |
| SOLUCIÓN c) |  |
| SOLUCIÓN d) |  |

## Ejercicio propuesto 4

Dada una lista de elementos desordenados y algunos repetidos, devolver una muestra de 5 elementos, que estén en la lista, sin repetir y ordenados descendentemente

lista = 4,6,34,7,9,2,3,4,4,21,4,6,8,9,7,8,5,4,3,22,34,56,98

b) Selecciona el elemento mayor de la lista resultante

|  |  |
| --- | --- |
| SOLUCIÓN a) |  |
| SOLUCIÓN b) |  |

## Ejercicio propuesto 5

Dado una lista de vocales: vocales = a,e,i,o,u

a) Une todas las letras en un único valor dando como resultado:

A-E-I-O-U

b) Sabiendo que la ‘a’ suma 1 la ‘e’ suma 2 la ‘i’ suma 3 la ‘o’ suma 4 y la ‘u’ suma 5, y que cualquier consonante resta 1...

Dada esta lista:

'a','x','r','e','i','i','a','m','p''o','u','y','r'

Se pide sumar el total de cada particion y ademas que por cada partición se sumen 10 extra.

|  |  |
| --- | --- |
| SOLUCIÓN a) |  |
| SOLUCIÓN b) |  |

## Ejercicio propuesto 6

Dada la frase ¿Hola que tal estas?

Cuenta cuantas veces se repite cada carácter

*Por ejemplo:  H 1, a 3...*

|  |  |
| --- | --- |
| SOLUCIÓN |  |

b) Crea un rdd a partir del fichero 'texto\_grande.txt' como podríamos hacer una aproximación de las lineas que hay?

|  |  |
| --- | --- |
| SOLUCIÓN b) |  |

c) Y una aproximación de las frases distintas?

|  |  |
| --- | --- |
| SOLUCIÓN c) |  |