PráCtica sesión 2 Spark -

proyecto 2

El objetivo del siguiente proyecto, es evaluar lo visto en la sesión y en los ejercicios.

**Sumario**

[Ejercicio final Repaso (Proyecto 2) 1](#__RefHeading___Toc631_2103982880)

## Ejercicio final Repaso (Proyecto 2)

Trabajamos en un colegio y tenemos las notas de Matemáticas, Inglés y Física de los alumnos del colegio en 3 documentos txt, a partir de estos ficheros:

*Notas\_Fisica.txt*

*Notas\_Mates.txt*

*Notas\_Ingles.txt*

a) Crea 3 RDD de pares, uno para cada asignatura, con los alumnos y sus notas

|  |  |
| --- | --- |
| SOLUCIÓN | rdd\_Fisica = sc.textFile("Notas\_Fisica.txt").map(lambda x: x.split(',')).map(lambda y: (y[0],float(y[1])))  rdd\_Ingles = sc.textFile("Notas\_Ingles.txt").map(lambda x: x.split(',')).map(lambda y: (y[0],float(y[1])))  rdd\_Mates = sc.textFile("Notas\_Mates.txt").map(lambda x: x.split(',')).map(lambda y: (y[0],float(y[1]))) |

b) Crea un solo RDD con todas las notas

|  |  |
| --- | --- |
| SOLUCIÓN | rdd\_Total = rdd\_Fisica.union(rdd\_Ingles).union(rdd\_Mates) |

c) ¿Cuál es la nota más baja que ha tenido cada alumno?

|  |  |
| --- | --- |
| SOLUCIÓN | rdd\_Total.reduceByKey(lambda x,y: min(x,y)).collect() |

d) ¿Cuál es la nota media de cada alumno?

|  |  |
| --- | --- |
| SOLUCIÓN | # La primera no és moltbona, perquè sempre divideix per 3. La segona sí que és bona  #rdd\_mitj = rdd\_Total.reduceByKey(lambda x,y: x+y).map(lambda y: (y[0],y[1]/3))  rdd\_Total.groupByKey().mapValues(lambda x: sum(x)/len(x)).collect() |

e) ¿En qué asignatura suspende más gente?

|  |  |
| --- | --- |
| SOLUCIÓN | # El més fàcil seria comprovar separadament cada rdd. He intentat muntar un altre rdd que ens puga donar directament el resultat  rdd\_NotesAssig = sc.parallelize([\  ('Fisica',rdd\_Fisica.filter(lambda x: x[1]<5.0).count()),\  ('Ingles',rdd\_Ingles.filter(lambda x: x[1]<5.0).count()),\  ('Mates',rdd\_Mates.filter(lambda x: x[1]<5.0).count()) ])  rdd\_NotesAssig.takeOrdered(1, key = lambda x: -x[1]) |

f) Total de notables o sobresalientes por alumno

|  |  |
| --- | --- |
| SOLUCIÓN | # Ens ho dóna en forma de diccionari, però crec que és vàlid  rdd\_Total.filter(lambda x: x[1]>=7).countByKey() |

g) ¿Qué alumno no se ha presentado a inglés?

|  |  |
| --- | --- |
| SOLUCIÓN | rdd\_Total.subtractByKey(rdd\_Ingles).keys().distinct().collect() |

h) ¿A cuántas asignaturas se ha presentado cada alumno?

|  |  |
| --- | --- |
| SOLUCIÓN | rdd\_Total.groupByKey().map(lambda x:(x[0],len(x[1]))).collect()  # ens apareix Rocio que s'ha presentat a 4 perquè apareix 2 vegades en Fisica |

i) Obtén un RDD con cada alumno con sus notas

|  |  |
| --- | --- |
| SOLUCIÓN | rdd\_notesAlumnes = rdd\_Total.groupByKey()  # bucle per a poder fer list de les notes  [(x,list(y)) for x,y in rdd\_notesAlumnes.collect()] |