

```
In [ ]: import pandas as pd
        from pyspark.sql import *
```

```
In [ ]: #Cargo mis archivos de prueba, no pertenece al ejercicio

spark = SparkSession.builder.getOrCreate()
sc = spark.sparkContext

recetas = spark.read.csv("recetas.csv", header=True).rdd
recetas = recetas.map(lambda x: (pd.to_numeric(x[0], "coerce"), x[1], x[2]))

ingredientes = spark.read.csv("ingredientes.csv", header=True).rdd
ingredientes = ingredientes.map(lambda x: (pd.to_numeric(x[0], "coerce"), x[1], pd.to_numeric(x[2], "coerce")))
```

```
In [ ]: #recetas: (ID_Receta, Nombre, Categoría)
        #ingredientes : (ID_Receta, Ingrediente, Cantidad_Kg)
```

```
In [ ]: #A: Listar todos los ingredientes que aparecen en alguna receta que
        #usa "pollo" indicando en cuantas
        #recetas el ingrediente y pollo aparecen juntos. El formato de salida
        #es (ingrediente, cantidad de recetas
        #en que aparece junto con pollo). Por ejemplo, la papa aparece en
        #10 recetas con pollo, por lo que
        #tendríamos (papa, 10).

recetas_con_pollo = ingredientes.filter(lambda x: x[1] == "pollo").map(
    lambda x: (x[0], 1)) # (id_receta, 1)
recetas_sin_pollo = ingredientes.filter(lambda x: x[1] != "pollo") # saca los pollos de las recetas, no tiene sentido

# contar el pollo con si mismo

id_e_ingrediente = recetas_sin_pollo.map(lambda x: (x[0], x[1])) # (id_receta, ingrediente)
recetas_que_usan_pollo = id_e_ingrediente.join(recetas_con_pollo) # (id_receta, (ingrediente, 1))

# los ingredientes no se repiten dentro de una misma receta, así que se que todos los ingredientes que aparecen
# n veces es porque estan en n recetas. Ademas como hice un inner join con las recetas que tienen pollo, se que
# por cada aparicion del ingrediente ese ingrediente esta con pollo

recetas_que_usan_pollo.map(lambda x: (x[1][0], 1)).reduceByKey(lambda x, y: x+y).collect()
```

```
In [ ]: #B: Queremos obtener todos los nombres de recetas Mediterráneas q
ue no tengan ni papa ni pollo entre
#sus ingredientes.

recetas_mediterraneas = recetas.filter(lambda x:x[2]=="Mediterrán
ea").map(lambda x:(x[0],x[1]))#(id receta,nombre)

recetas_con_papa_o_pollo = ingredientes.filter(lambda x: (x[1]=="
pollo")or(x[1]=="papa")).\
    map(lambda x: (x[0],1))

#Saco todas las recetas que usaron pollo o papa, para eso primero
busco las que lo usaron y se las saco a
#ingredientes original, no puedo simplemente filtrar por si tiene
papa o pollo porque sino solo estoy sacando
#esos ingredientes, y queda el resto de la receta como si no los
tuviera, cuando originalmente si los tenia
#Cons subtract me aseguro que no tenga recetas que usaron pollo
o papa, despues distinct me deja solo
#los registros que no se repiten, para que no tenga la receta mas
de 1 vez
recetas_sin_papa_ni_pollo = ingredientes.map(lambda x:(x[0],1)).s
ubtract(recetas_con_papa_o_pollo).distinct()

#Hago un join para tener el nombre para las recetas que
recetas_pedidas = recetas_sin_papa_ni_pollo.join(recetas_mediterr
aneas)#(id_receta,(1,nombre))

recetas_pedidas.map(lambda x:x[1][1]).collect()
```

```
In [ ]:
```