PráCtica sesión 4 Spark -

proyecto 3

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

**Sumario**

[Práctica guiada Spark Streaming (Proyecto 4) 2](#__RefHeading___Toc631_2103982880)

[Ejercicio 1 Trabajando con Sockets 2](#__RefHeading___Toc371_311506310)

[Ejercicio 2 Bizums. Trabajando con Spark Streaming y Ficheros 4](#__RefHeading___Toc373_311506310)

## Práctica guiada Spark Streaming (Proyecto 4)

**Práctica guiada Spark Streaming**

**(SE DEBE ADJUNTAR IMAGEN DEL CONTADOR DE NOMBRE Y APELLIDO UNA VEZ FINALIZADO EL PROCESO)**

### Ejercicio 1 Trabajando con Sockets

**1.- Inicia spark Streaming con 2 hilos (pyspark --master local[2])**

Para ello abriremos una consola y escribiremos

**pyspark --master local[2]**

**2. Abre una terminal en un puerto concreto el que quieras donde escribiremos lo que queramos**

Ahora abriremos otra consola en un puerto concreto por ejemplo 4444 para ello escribiremos:

**nc -n -lkv 4444**

**3. Paralelamente iniciaremos Spark Streaming e iremos recogiendo todo lo que escribimos en la terminal e irá contando las veces que aparece cada palabra.**

**Por ejemplo escribiremos nuestro nombre o apellidos indistintamente y en tiempo real spark Streaming ira contando las veces que hemos escrito cada cosa.**

3..a. Definiremos un tiempo por ejemplo de 10 segundos en el StreamingContext, para que cada 10 segundos nos vaya haciendo el conteo.

StreamingContext(sc,10)

3.b. Vamos a indicarle que queremos leer de un socket que esta en el puerto 4444:

socketTextStream(‘localhost’,4444)

3.c. Con flatMap igual que ya hemos hecho otras veces, vamos a aplanar las palabras con split

linea.split(“ “)

3.d. Haremos un contador, para hacer esto usaremos un RDD de pares, haremos una tupla de (palabra,1), para despues agregar por clave y sumar el valor así cada vez que aparezca una palabra, sumaremos 1.

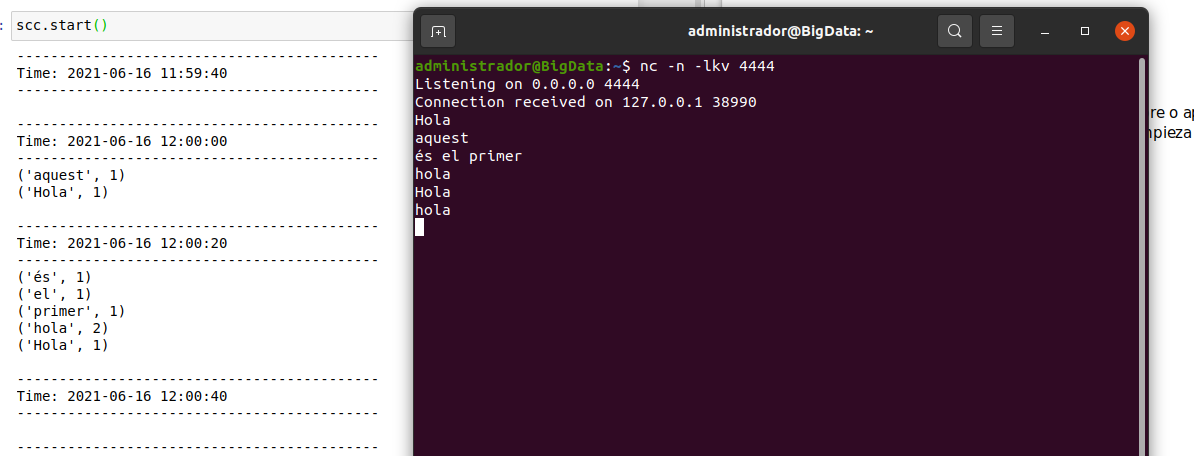
3.e. Haremos un print de las palabras y su contador:

wordCount.pprint()

3.f. Inicializamos sparkStreaming

scc.start()

3.g. Comenzamos a escribir nuestro nombre o apellidos en la consola cada X tiempo, veremos que nuestro contador empieza a funcionar.



### Ejercicio 2 Bizums. Trabajando con Spark Streaming y Ficheros

**SE DEBEN ADJUNTAR IMAGENES DEL PROCESO**

**Análisis de bizum en tiempo real.**

**Vamos a simular los bizums que nos llegan a nuestra cuenta y estos, se van a almacenar en un directorio en formato texto.**

**En tiempo real queremos saber cual es el bizum más alto por cada persona y concepto.**

1. Vamos a crearnos una estructura de datos como hemos visto en dataframes, con las siguientes columnas:

Nombre(string),Cantidad(Integer),Concepto(string)

2. Vamos a crearnos el punto de entrada de spark Streaming, esta ves desde ficheros csv, para ello vamos a hacerlo del siguiente modo:

spark.readStream.option(“sep”,”;”) #Para indicarle el separador en los csv

.schema(mi\_esquema)#Le pasaremos el esquema que hemos creado

.csv(“bizums/recibidos”) #El directorio donde va a ir leyendo los bizums que recibimos

3. Nos interesa en tiempo real quedarnos con los bizum agregados por nombre y concepto cuyo importe sea el mayor, para hacer esto podemos hacer uso de groupBy y las agregaciones que hemos visto en la seccion de spark SQL.

4.- Iremos mostrando en la consola la información en tiempo real para ello haremos:

query = mayorBizum.writeStream.outputMode(“complete”).format(“console”)

5.- Tenemos adjuntos en el Moodle, una carpeta con bizums sin enviar, y un código python que se encarga de enviar bizums llamado **Bizums.py**

Crearemos una carpeta llamada: **Sin\_llegar** (donde estarán los csv de los bizum que hemos descargado)

Y otra carpeta llamada: **recibidos**

6. Ejecutaremos nuestro codigo de Spark Streaming para que empiece a detectar si llegan bizums

query.start()

7. Ejecutaremos el código Bizums.py para empezar a recibir bizums...(el codigo va enviando aleatoriamente, a veces dará error, otras veces enviará bien el bizum… debemos ir viendo en consola cuando vamos recibiendo bizums, como se va actualizando la información)

|  |  |
| --- | --- |
|  |  |