Tarea evaluable CE_5075 7.1

Big data aplicado



APARTADO 2

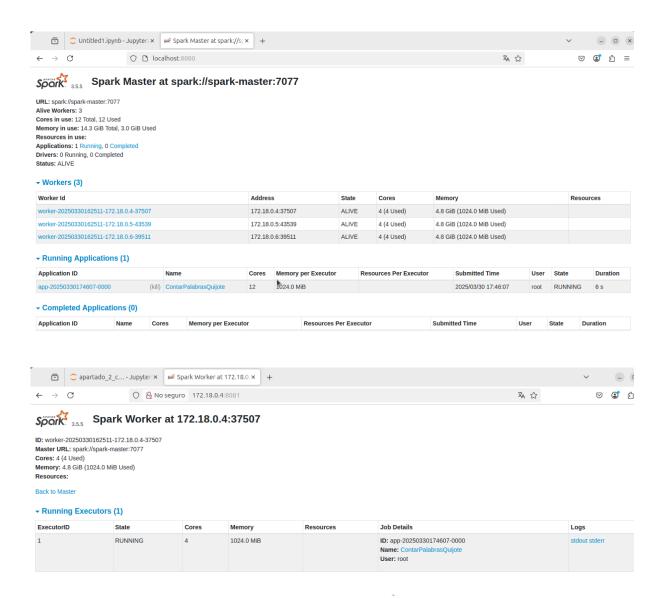
Crea un cuaderno de Jupyter en el nodo con JupyterLab del clúster que has creado y escribe el código con PySpark y RDDs (no puedes usar DataFrames de Spark SQL) para contar las ocurrencias de todas las palabras que aparecen en *El Quijote* que empiezan y terminan con vocal (mayúscula o minúscula).

Dejaré el cuaderno de Jupyter que contiene el código ejecutado entre los archivos de la entrega. Pero aún así aquí muestro algunas capturas del cuaderno.

```
◎ ↑ ↓ ☆ ♀ ▮
from pyspark import SparkContext
import re
 sc = SparkContext('spark://spark-master:7077', 'ContarPalabrasQuijote')
 /usr/local/lib/python3.9/dist-packages/pyspark/bin/load-spark-env.sh: line 68: ps: command not found
 Setting default log level to "WARN"
 To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel)
 25/03/30 17:46:06 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes wh
ere applicable
SparkContext
Spark UI
                        v3.5.5
                        spark://spark-master:7077
Master
                        ContarPalabrasQuijote
AppName
```

```
# Carga el archivo como RDD
rdd = sc.textFile('quijote.txt')
# Expresión regular para palabras que comienzan y terminan con vocal (no distingue mayúsculas de minúsculas)
vowel_pattern = r'(?i)^[aeiouáéíóú].*[aeiouáéíóú]$'
    rdd.flatMap(lambda line: re.findall(r'\b\w+\b', line)) # Pasa todas las palabras a una lista
.filter(lambda word: re.match(vowel_pattern, word)) # Filtra las palabras que coinciden con la regex
                                                                    # Pasa a minúsculas y establece el contador a 1
         .map(lambda word: (word.lower(), 1))
         .reduceByKey(lambda x, y: x + y)
                                                                    # Reduce por palabra (clave) sumando los valores
# Muestra el contador por palabra
                                                                                                                       ◎ ↑ ↓ 占 ♀ ▮
for word, count in word_counts.collect():
    print(f'{word}: {count}')
[Stage 1:>
                                                                            (0 + 2) / 2]
adulacio: 2
ora: 252
asi: 576
esto: 497
espanto: 12
enviado: 7
encuentro: 7
este: 394
alando: 4
alguna: 210
acabo: 36
osaba: 6
aquella: 219
acabado: 18
```

Y a continuación adjunto las capturas de pantalla del nodo master y de uno de los worker para comprobar que la aplicación está en funcionamiento.



2