# wordcount-spark

November 14, 2025

[ ]:
```
# si esta en EMR o Databricks, estos objetos ya están preconstruidos:
spark
```

[ ]:
```
# si esta en EMR o Databricks, estos objetos ya están preconstruidos:
sc
```

[ ]:
```
# WORDCOUNT COMPACTO
# en AWS S3
#files_rdd = sc.textFile("s3a://username_datalake/datasets/gutenberg-small/*.
 ↪txt")

# en gdrive:
files_rdd = sc.textFile("gdrive/MyDrive/st0263-252/bigdata/datasets/
 ↪gutenberg-small/*.txt")

# local:
#files_rdd = sc.textFile("../datasets/gutenberg-small/*.txt")
wc_unsort = files_rdd.flatMap(lambda line: line.split()).map(lambda word:␣
 ↪(word, 1)).reduceByKey(lambda a, b: a + b)
wc = wc_unsort.sortBy(lambda a: -a[1])
for tupla in wc.take(10):
        print(tupla)
```

[ ]:
```
# WORDCOUNT PASO A PASO
```

[ ]:
```
files = sc.textFile("gdrive/MyDrive/st0263-252/bigdata/datasets/gutenberg-small/
 ↪*.txt")
for f in files.take(10):
    print(f)
```

[ ]:
```
tokens = files.flatMap(lambda line: line.split())
for t in tokens.take(10):
    print(t)
```

[ ]:
```
wc1 = tokens.map(lambda word: (word, 1))
for c in wc1.take(10):
    print(c)
```

[ ]:
```
wc = wc1.reduceByKey(lambda a, b: a + b)
for c in wc.take(10):
    print(c)
```

[ ]:
```
wcsort = wc.sortBy(lambda a: -a[1])
for c in wcsort.take(10):
    print(c)
```