

PROGRAMAÇÃO DE ENTRADAS STREAMING EM CLUSTERS

Segundo projeto de pesquisa



Equipe



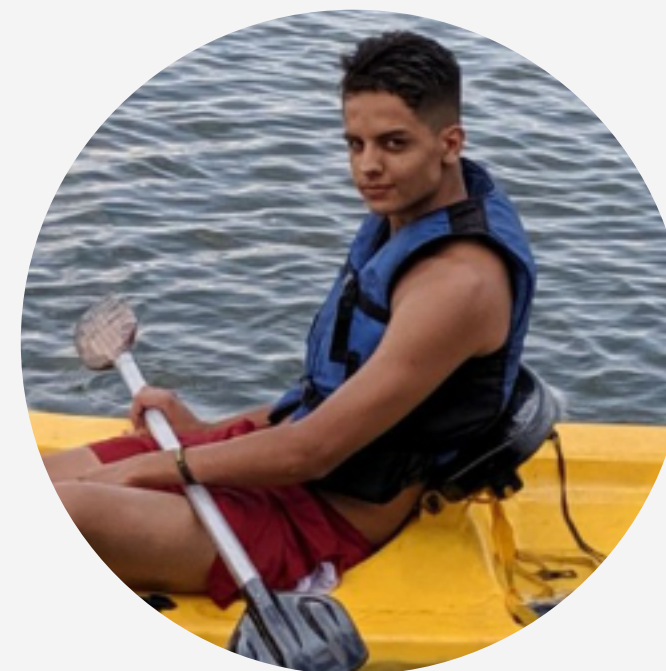
Eduardo Afonso

19/0012307



Thiago Paiva

19/0020377



Rafael Cleydson

19/0019085



Problema

Contar palavras

Contar palavras que começam com S

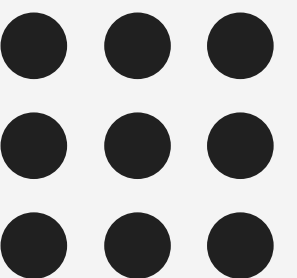
Contar palavras que começam com P

Contar palavras que começam com R

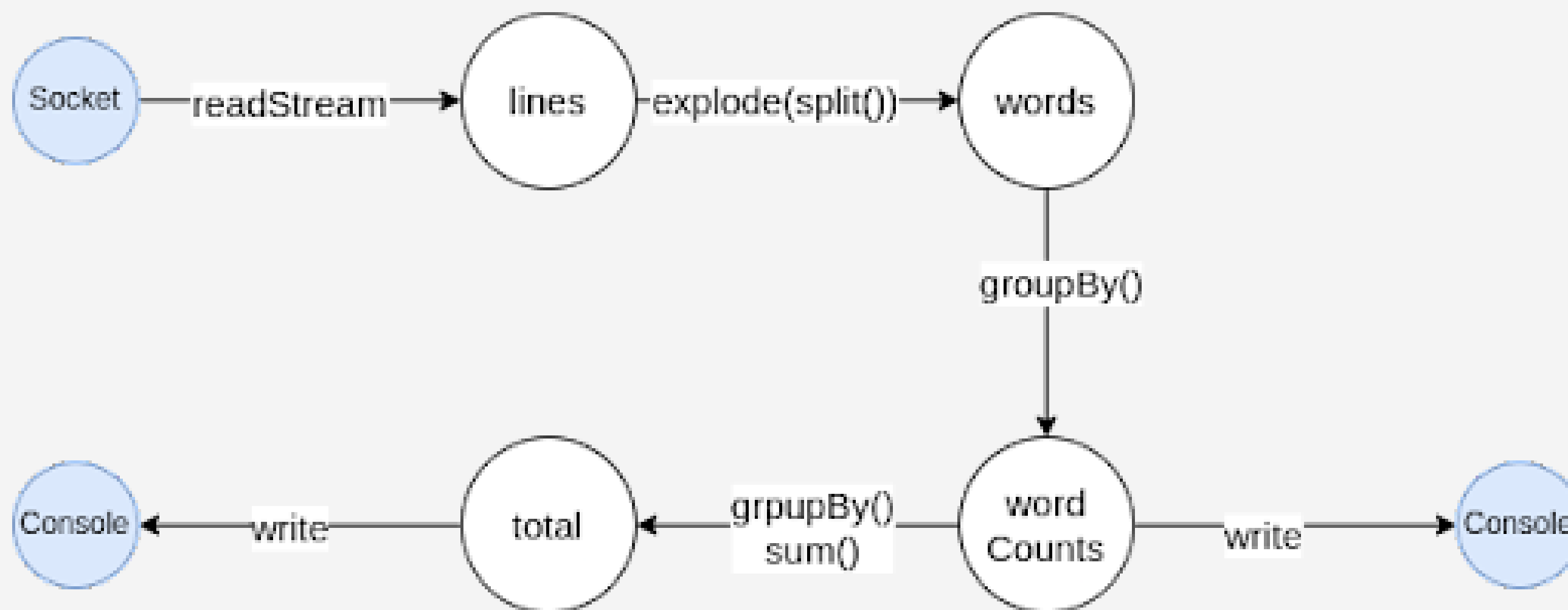
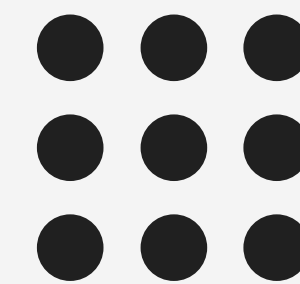
Contar palavras com 6 caracteres

Contar palavras com 8 caracteres

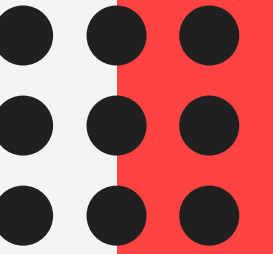
Contar palavras com 11 caracteres



Solução Socket - Parte 1

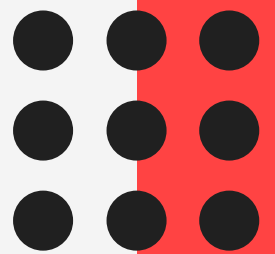


Solução Socket - Conexão



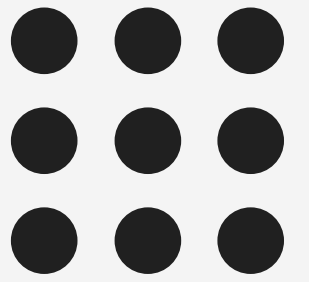
```
1  from pyspark.sql import SparkSession
2  from pyspark.sql.functions import split, explode, lit, col, upper
3
4  SOCKET_HOST = "localhost"
5  SOCKET_PORT = "9999"
6
7  spark = SparkSession \
8      .builder \
9      .appName("P2 - PSPD - Socket") \
10     .getOrCreate()
11
12  lines = spark \
13     .readStream \
14     .format("socket") \
15     .option("host", SOCKET_HOST) \
16     .option("port", SOCKET_PORT) \
17     .load()
```

Solução Socket - Contagem



```
19 # Split the lines into words
20 words = lines.select(
21     explode(
22         split(lines.value, "\s+")
23     ).alias('word')
24 )
25 words = words.select(upper(words.word).alias('word'))
26
27 # Generate running word count
28 wordCounts = words.groupBy("word").count()
29
30 def foreach_batch_func(df, _):
31     """Find the total number of words and write it and wordsCount in console"""
32     total = df \
33         .groupBy() \
34         .sum() \
35         .select(lit('TOTAL').alias('key'), col('sum(count)').alias('value'))
36
37     df.write.format('console').save()
38     total.write.format('console').save()
39
40 # Sink
41 query = wordCounts \
42     .writeStream \
43     .outputMode("complete") \
44     .foreachBatch(foreach_batch_func) \
45     .start()
46
47 query.awaitTermination()
```

Comandos Exec Socket

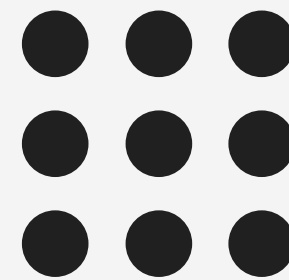


```
nc -lk 9999 < {filePath}
```

```
bin/pyspark
```



Solução Socket Saída

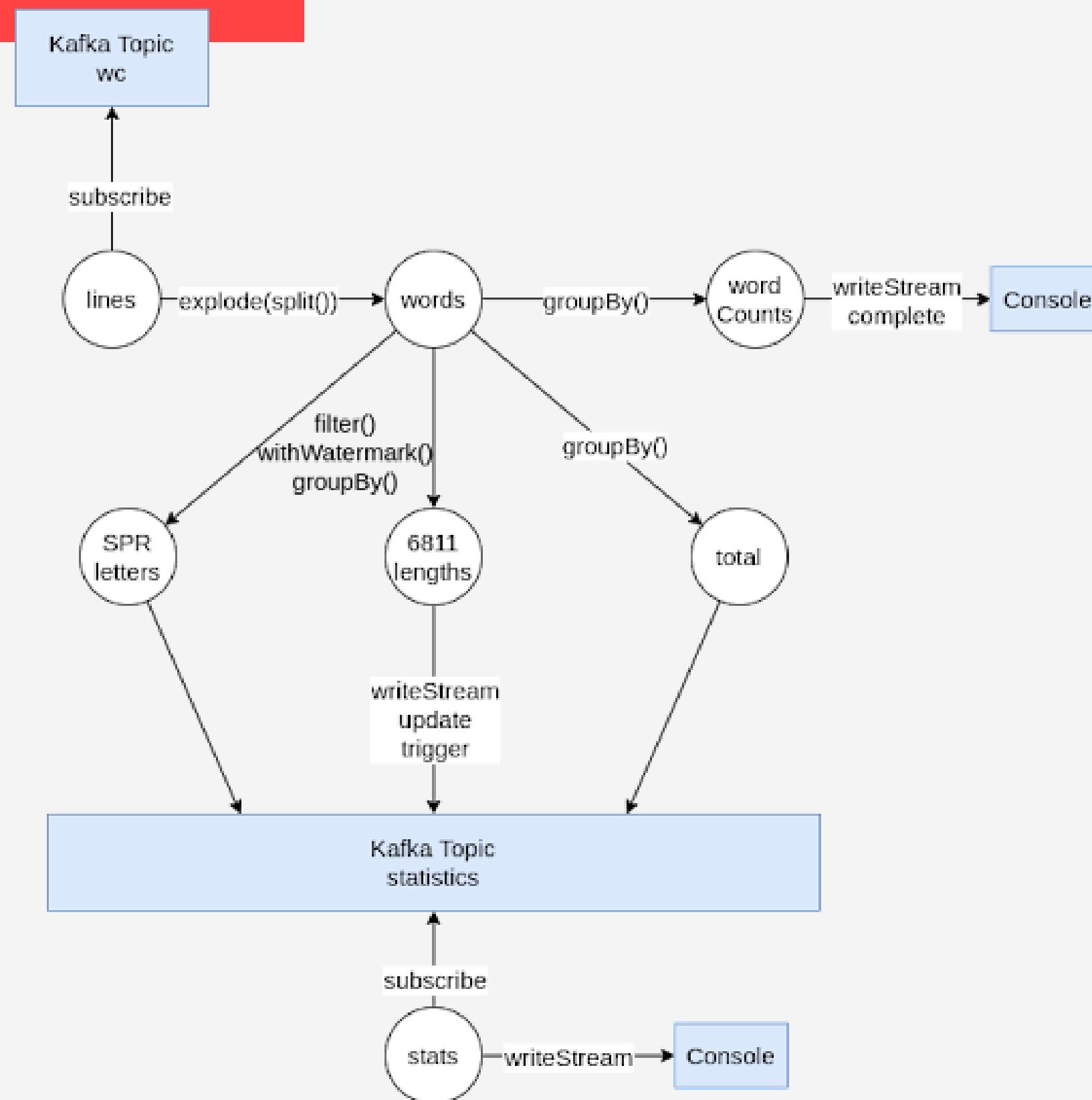
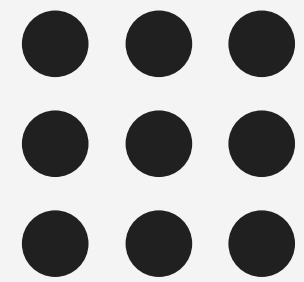


word count	
YTSGWVEV	1
DZ	1
VDXZW	1
WNZPUDOPF	1
FEKZUVF	1
LT	1
EVVHRWKETXG	1
DAAPTIXRX	1
GFSZXJVEE	1
VWEXCUXOQW	1
LEB	1
NBKP	1
K	17
ATPPYCWGX	1
PZKVVNI	1
FMXBZKE	1
MM	1
KOTWSZIBIDT	1
RDJY	1
BRKCICSACE	1
only showing top 20 rows	

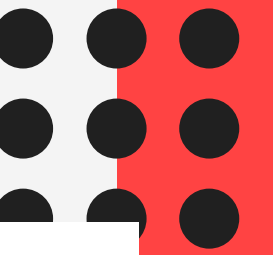
key	value
TOTAL	19462001
S	170323
R	169696
P	170119
Batch: 8	
key	value
8	368409
11	367776
6	368207
Batch: 9	
key	value
TOTAL	100000001



Solução Kafka - Parte 2

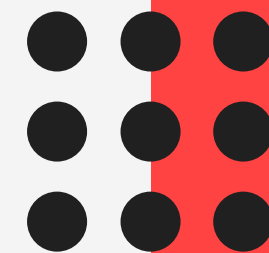


Solução Kafka - Conexão



```
1  from pyspark.sql import SparkSession
2  from pyspark.sql.functions import length, explode, split, substring, upper, window
3
4  INTERVAL = '3 seconds'
5  KAFKA_SERVER = 'localhost:9092'
6  WORDS_TOPIC = 'wc'
7  STATS_TOPIC = 'statistics'
8
9  spark = SparkSession \
10     .builder \
11     .appName("P2 - PSPD - Transformer") \
12     .getOrCreate()
13
14  # Create DataFrame representing the stream of input lines and subscribe it in kafka topic
15  lines = spark \
16     .readStream \
17     .format("kafka") \
18     .option("kafka.bootstrap.servers", KAFKA_SERVER) \
19     .option("subscribe", WORDS_TOPIC) \
20     .option('includeTimestamp', 'true') \
21     .load()
22
```

Solução Kafka - Contagem

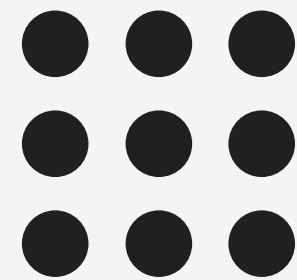


```
23 # Split the lines into words
24 words = lines.select(
25     explode(
26         split(lines.value, "\s+").alias("word"),
27         lines.timestamp
28     )
29 words = words.select(upper(words.word).alias('word'), words.timestamp)
30
31 # Group words
32 wordCounts = words.groupBy("word").count()
33
34 # Count the total of words readed
35 total = words \
36     .groupBy() \
37     .count() \
38     .selectExpr("'TOTAL' as key", "CAST(count AS STRING) as value")
39
```

```
40 # Count the words that startswith S, P and R
41 letters = words \
42     .filter(upper(substring(words.word, 0, 1)).isin(["S", "P", "R"])) \
43     .withWatermark("timestamp", INTERVAL) \
44     .groupBy(
45         window(words.timestamp, INTERVAL, INTERVAL),
46         upper(substring(words.word, 0, 1)).alias("key"),
47     ) \
48     .count() \
49     .selectExpr("key", "CAST(count AS STRING) as value")
50
51 # Count the words that has length 6, 8 and 11
52 lengths = words \
53     .filter(length(words.word).isin([6, 8, 11])) \
54     .withWatermark("timestamp", INTERVAL) \
55     .groupBy(
56         window(words.timestamp, INTERVAL, INTERVAL),
57         length(words.word).alias("key")
58     ) \
59     .count() \
60     .selectExpr("CAST(key AS STRING)", "CAST(count AS STRING) as value")
61
62 # Sink
```



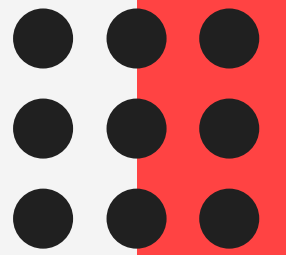
Solução Kafka - Sink



```
62 # Sinks
63 qW = wordCounts \
64     .writeStream \
65     .outputMode("complete") \
66     .format("console") \
67     .start()
68
69 qT = total \
70     .writeStream \
71     .outputMode("complete") \
72     .format("kafka") \
73     .option("kafka.bootstrap.servers", KAFKA_SERVER) \
74     .option('topic', STATS_TOPIC) \
75     .option('checkpointLocation', '/tmp/spark/total-stats') \
76     .start()
77
78 qLen = lengths \
79     .writeStream \
80     .outputMode("update") \
81     .format("kafka") \
82     .option("kafka.bootstrap.servers", KAFKA_SERVER) \
83     .option('topic', STATS_TOPIC) \
84     .option('checkpointLocation', '/tmp/spark/len-stats') \
85     .trigger(processingTime=INTERVAL) \
86     .start()
87
88 qLet = letters \
89     .writeStream \
90     .outputMode("update") \
91     .format("kafka") \
92     .option("kafka.bootstrap.servers", KAFKA_SERVER) \
93     .option('topic', STATS_TOPIC) \
94     .option('checkpointLocation', '/tmp/spark/let-stats') \
95     .trigger(processingTime=INTERVAL) \
96     .start()
97
```

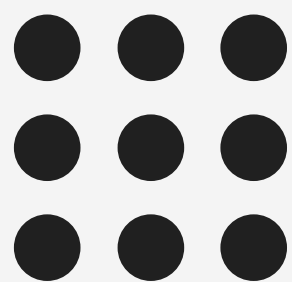


Solução Kafka - Visualização das Estatísticas



```
1  from pyspark.sql import SparkSession
2
3  KAFKA_SERVER = 'localhost:9092'
4  STATS_TOPIC = 'statistics'
5
6  spark = SparkSession \
7      .builder \
8      .appName("P2 - PSPD - Stats Consumer") \
9      .getOrCreate()
10
11 stats = spark \
12     .readStream \
13     .format("kafka") \
14     .option("kafka.bootstrap.servers", KAFKA_SERVER) \
15     .option("subscribe", "statistics") \
16     .load()
17
18 q = stats \
19     .selectExpr("CAST(key AS STRING)", "CAST(value AS STRING)") \
20     .writeStream \
21     .format('console') \
22     .outputMode('append') \
23     .trigger(processingTime='3 seconds') \
24     .start()
25
26 q.awaitTermination()
```

Solução Kafka Saída



word count		key value	
TOTAL 19462001		S 170323	
R 169696		P 170119	
Batch: 8			
key value			
8 368409			
11 367776			
6 368207			
Batch: 9			
key value			
TOTAL 100000001			

only showing top 20 rows



**Obrigado pela
atenção**

