V Big Data and Machine Learning Bootcamp

Data Processing

PRACTICA (2/3)

Punto 1.2 (CON KAFKA)

- Nos desplazamos al directorio donde se encuentran los ejecutables de Kafka

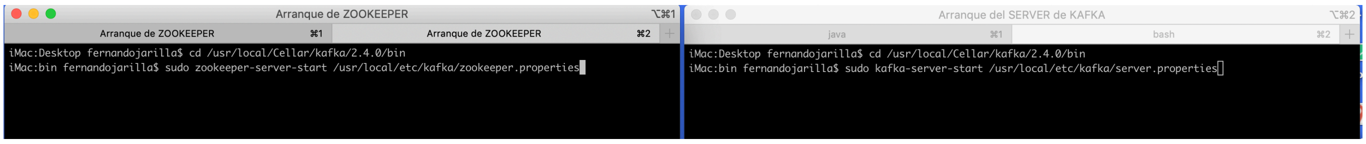
  `$ cd /usr/local/Cellar/kafka/2.4.0/bin`

- Lanzamos el servidor de zookeeper

  `$ sudo zookeeper-server-start /usr/local/etc/kafka/zookeeper.properties`

- A continuación, lanzamos el servidor de kafka

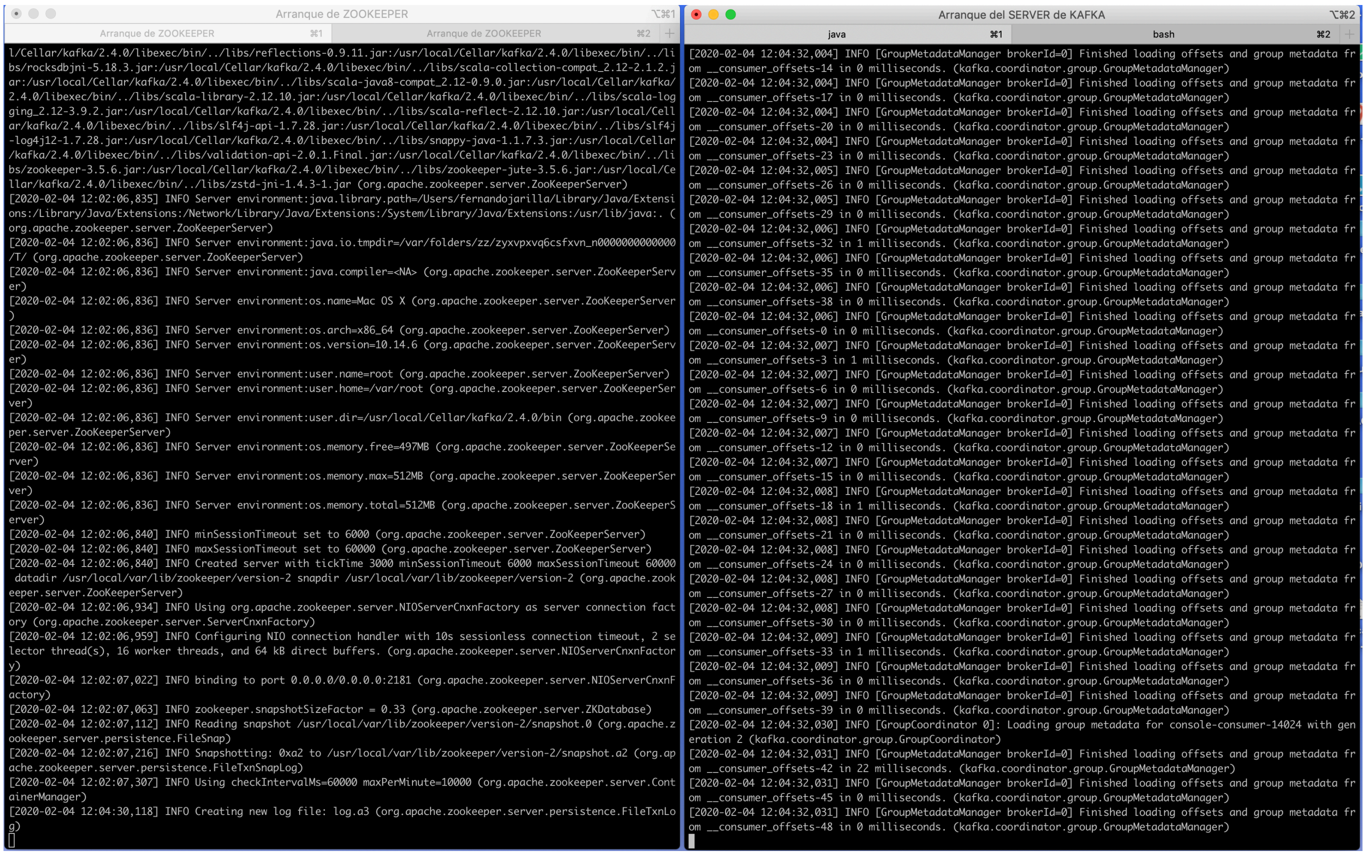
  `$ sudo kafka-server-start /usr/local/etc/kafka/server.properties`



```
iMac:Desktop fernandojarilla$ cd /usr/local/Cellar/kafka/2.4.0/bin
iMac:bin fernandojarilla$ sudo zookeeper-server-start /usr/local/etc/kafka/zookeeper.properties
```

```
iMac:Desktop fernandojarilla$ cd /usr/local/Cellar/kafka/2.4.0/bin
iMac:bin fernandojarilla$ sudo kafka-server-start /usr/local/etc/kafka/server.properties
```

Foto 1. Comandos antes de ejecutarse*



```
l/Cellar/kafka/2.4.0/libexec/bin/../libs/reflections-0.9.11.jar:/usr/local/Cellar/kafka/2.4.0/libexec/bin/../li
bs/rocksdbjni-5.18.3.jar:/usr/local/Cellar/kafka/2.4.0/libexec/bin/../libs/scala-collection-compat_2.12-2.1.2.j
ar:/usr/local/Cellar/kafka/2.4.0/libexec/bin/../libs/scala-java8-compat_2.12-0.9.0.jar:/usr/local/Cellar/kafka/
2.4.0/libexec/bin/../libs/scala-library-2.12.10.jar:/usr/local/Cellar/kafka/2.4.0/libexec/bin/../libs/scala-log
ging_2.12-3.9.2.jar:/usr/local/Cellar/kafka/2.4.0/libexec/bin/../libs/scala-reflect-2.12.10.jar:/usr/local/Cell
ar/kafka/2.4.0/libexec/bin/../libs/slf4j-api-1.7.28.jar:/usr/local/Cellar/kafka/2.4.0/libexec/bin/../libs/slf4j
-log4j12-1.7.28.jar:/usr/local/Cellar/kafka/2.4.0/libexec/bin/../libs/snappy-java-1.1.7.3.jar:/usr/local/Cellar
/kafka/2.4.0/libexec/bin/../libs/validation-api-2.0.1.Final.jar:/usr/local/Cellar/kafka/2.4.0/libexec/bin/../li
bs/zookeeper-3.5.6.jar:/usr/local/Cellar/kafka/2.4.0/libexec/bin/../libs/zookeeper-jute-3.5.6.jar:/usr/local/Ce
llar/kafka/2.4.0/libexec/bin/../libs/zstd-jni-1.4.3-1.jar (org.apache.zookeeper.server.ZooKeeperServer)
[2020-02-04 12:02:06,835] INFO Server environment:java.library.path=/Users/fernandojarilla/Library/Java/Extensi
ons:/Library/Java/Extensions:/Network/Library/Java/Extensions:/System/Library/Java/Extensions:/usr/lib/java:. (
org.apache.zookeeper.server.ZooKeeperServer)
[2020-02-04 12:02:06,836] INFO Server environment:java.io.tmpdir=/var/folders/zz/zyxvpxvq6csfxvn_n0000000000000
/T/ (org.apache.zookeeper.server.ZooKeeperServer)
[2020-02-04 12:02:06,836] INFO Server environment:java.compiler=<NA> (org.apache.zookeeper.server.ZooKeeperServ
er)
[2020-02-04 12:02:06,836] INFO Server environment:os.name=Mac OS X (org.apache.zookeeper.server.ZooKeeperServer
)
[2020-02-04 12:02:06,836] INFO Server environment:os.arch=x86_64 (org.apache.zookeeper.server.ZooKeeperServer)
[2020-02-04 12:02:06,836] INFO Server environment:os.version=10.14.6 (org.apache.zookeeper.server.ZooKeeperServ
er)
[2020-02-04 12:02:06,836] INFO Server environment:user.name=root (org.apache.zookeeper.server.ZooKeeperServer)
[2020-02-04 12:02:06,836] INFO Server environment:user.home=/var/root (org.apache.zookeeper.server.ZooKeeperSer
ver)
[2020-02-04 12:02:06,836] INFO Server environment:user.dir=/usr/local/Cellar/kafka/2.4.0/bin (org.apache.zookee
per.server.ZooKeeperServer)
[2020-02-04 12:02:06,836] INFO Server environment:os.memory.free=497MB (org.apache.zookeeper.server.ZooKeeperSe
rver)
[2020-02-04 12:02:06,836] INFO Server environment:os.memory.max=512MB (org.apache.zookeeper.server.ZooKeeperSer
ver)
[2020-02-04 12:02:06,836] INFO Server environment:os.memory.total=512MB (org.apache.zookeeper.server.ZooKeeperS
erver)
[2020-02-04 12:02:06,840] INFO minSessionTimeout set to 6000 (org.apache.zookeeper.server.ZooKeeperServer)
[2020-02-04 12:02:06,840] INFO maxSessionTimeout set to 60000 (org.apache.zookeeper.server.ZooKeeperServer)
[2020-02-04 12:02:06,840] INFO Created server with tickTime 3000 minSessionTimeout 6000 maxSessionTimeout 60000
 datadir /usr/local/var/lib/zookeeper/version-2 snapdir /usr/local/var/lib/zookeeper/version-2 (org.apache.zook
eeper.server.ZooKeeperServer)
[2020-02-04 12:02:06,934] INFO Using org.apache.zookeeper.server.NIOServerCnxnFactory as server connection fact
ory (org.apache.zookeeper.server.ServerCnxnFactory)
[2020-02-04 12:02:06,959] INFO Configuring NIO connection handler with 10s sessionless connection timeout, 2 se
lector thread(s), 16 worker threads, and 64 kB direct buffers. (org.apache.zookeeper.server.NIOServerCnxnFactor
y)
[2020-02-04 12:02:07,022] INFO binding to port 0.0.0.0/0.0.0.0:2181 (org.apache.zookeeper.server.NIOServerCnxnF
actory)
[2020-02-04 12:02:07,063] INFO zookeeper.snapshotSizeFactor = 0.33 (org.apache.zookeeper.server.ZKDatabase)
[2020-02-04 12:02:07,112] INFO Reading snapshot /usr/local/var/lib/zookeeper/version-2/snapshot.0 (org.apache.z
ookeeper.server.persistence.FileSnap)
[2020-02-04 12:02:07,216] INFO Snapshotting: 0xa2 to /usr/local/var/lib/zookeeper/version-2/snapshot.a2 (org.ap
ache.zookeeper.server.persistence.FileTxnSnapLog)
[2020-02-04 12:02:07,307] INFO Using checkIntervalMs=60000 maxPerMinute=10000 (org.apache.zookeeper.server.Cont
ainerManager)
[2020-02-04 12:04:30,118] INFO Creating new log file: log.a3 (org.apache.zookeeper.server.persistence.FileTxnLo
g)
```

```
[2020-02-04 12:04:32,004] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-14 in 0 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,004] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-17 in 0 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,004] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-20 in 0 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,004] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-23 in 0 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,005] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-26 in 0 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,005] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-29 in 0 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,006] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-32 in 1 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,006] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-35 in 0 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,006] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-38 in 0 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,006] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-0 in 0 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,007] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-3 in 1 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,007] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-6 in 0 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,007] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-9 in 0 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,007] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-12 in 0 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,007] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-15 in 0 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,008] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-18 in 1 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,008] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-21 in 0 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,008] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-24 in 0 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,008] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-27 in 0 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,008] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-30 in 0 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,009] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-33 in 1 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,009] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-36 in 0 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,009] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-39 in 0 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,030] INFO [GroupCoordinator 0]: Loading group metadata for console-consumer-14024 with gen
eration 2 (kafka.coordinator.group.GroupCoordinator)
[2020-02-04 12:04:32,031] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-42 in 22 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,031] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-45 in 0 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
[2020-02-04 12:04:32,031] INFO [GroupMetadataManager brokerId=0] Finished loading offsets and group metadata fr
om __consumer_offsets-48 in 0 milliseconds. (kafka.coordinator.group.GroupMetadataManager)
```

Foto 2. Comandos de arranque se servers ejecutados

- Una vez lanzados los servers, haré uso del tópico creado en el punto anterior que llamé **practica**

- A continuación, crearé el **PRODUCTOR**

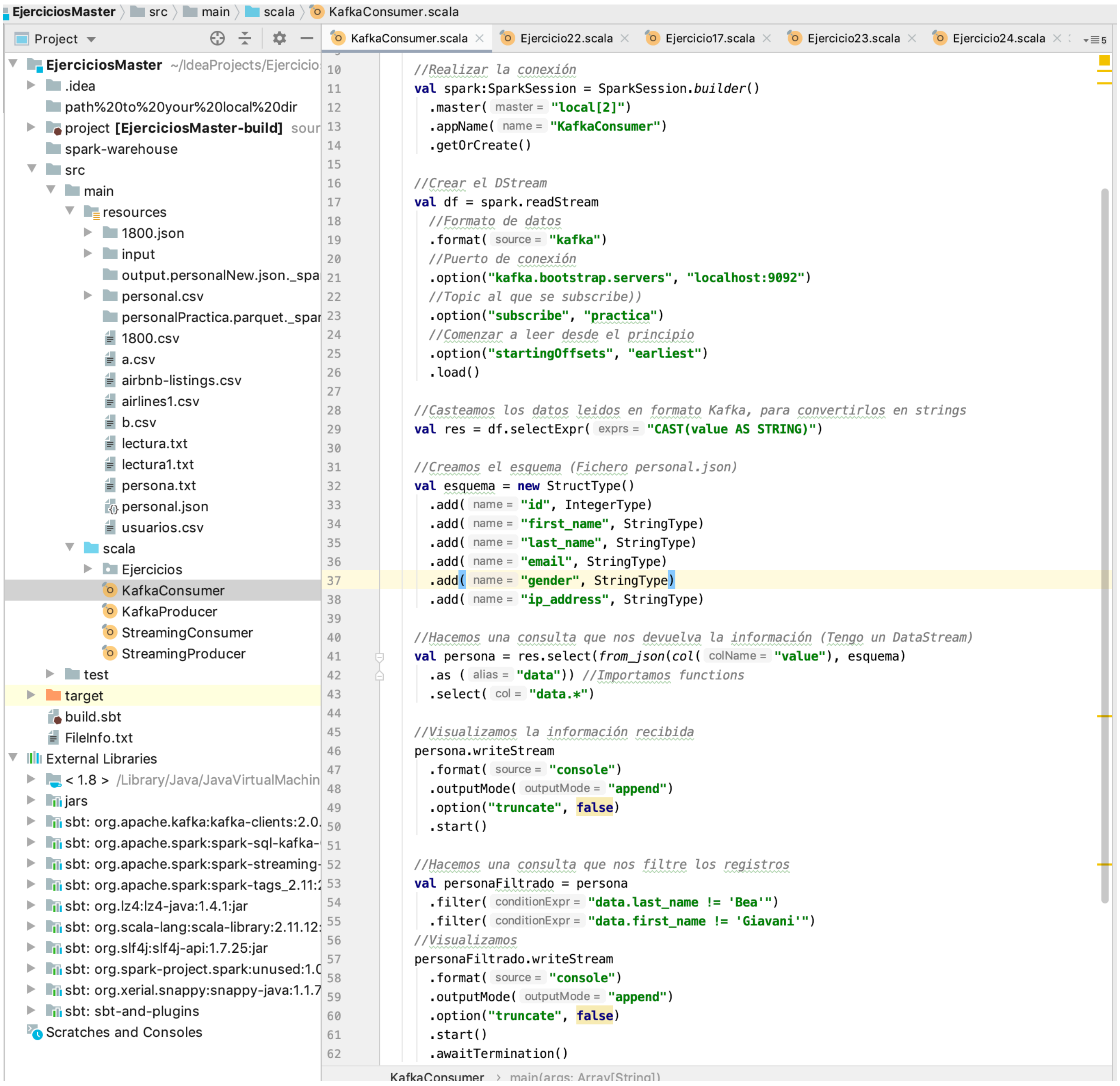
```
$ sudo kafka-console-producer --broker-list localhost:9092 --topic practica
```

- Una vez todo preparado, el comando a ejecutar para que el productor envíe el fichero que queremos leer será:

```
$ cat /users/fernandojarilla/Desktop/personal.json | sudo kafka-console-producer --broker-list localhost:9092 --topic practica
```

- Con todo el entorno listo, nos vamos al IDE de IntelliJ y dentro del proyecto donde hemos hecho el curso (en mi caso **EjerciciosMaster**), creamos una nueva **ScalaClass** como **Object** que llamaremos **KafkaConsumer**

- El programa generado es el siguiente:



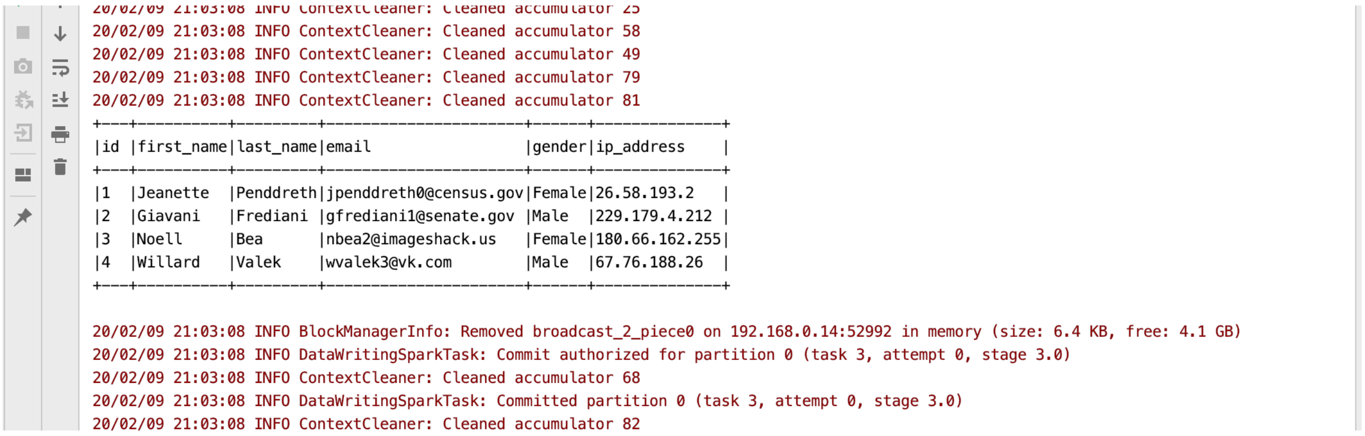
EjerciciosMaster 〉 src 〉 main 〉 scala 〉 KafkaConsumer.scala
Project
KafkaConsumer.scala
Ejercicio22.scala
Ejercicio17.scala
Ejercicio23.scala
Ejercicio24.scala
KafkaConsumer 〉 main(args: Array[String])

```
EjerciciosMaster ~/IdeaProjects/Ejercicio
  .idea
  path%20to%20your%20local%20dir
  project [EjerciciosMaster-build] sour
  spark-warehouse
  src
    main
      resources
        1800.json
        input
        output.personalNew.json._spa
        personal.csv
        personalPractica.parquet._spar
        1800.csv
        a.csv
        airbnb-listings.csv
        airlines1.csv
        b.csv
        lectura.txt
        lectura1.txt
        persona.txt
        personal.json
        usuarios.csv
      scala
        Ejercicios
        KafkaConsumer
        KafkaProducer
        StreamingConsumer
        StreamingProducer
    test
  target
  build.sbt
  FileInfo.txt
External Libraries
  < 1.8 > /Library/Java/JavaVirtualMachin
  jars
  sbt: org.apache.kafka:kafka-clients:2.0.
  sbt: org.apache.spark:spark-sql-kafka-
  sbt: org.apache.spark:spark-streaming-
  sbt: org.apache.spark:spark-tags_2.11:2
  sbt: org.lz4:lz4-java:1.4.1:jar
  sbt: org.scala-lang:scala-library:2.11.12:
  sbt: org.slf4j:slf4j-api:1.7.25:jar
  sbt: org.spark-project.spark:unused:1.0
  sbt: org.xerial.snappy:snappy-java:1.1.7
  sbt: sbt-and-plugins
Scratches and Consoles
```

```scala
    //Realizar la conexión
    val spark:SparkSession = SparkSession.builder()
      .master( master = "local[2]")
      .appName( name = "KafkaConsumer")
      .getOrCreate()

    //Crear el DStream
    val df = spark.readStream
      //Formato de datos
      .format( source = "kafka")
      //Puerto de conexión
      .option("kafka.bootstrap.servers", "localhost:9092")
      //Topic al que se subscribe))
      .option("subscribe", "practica")
      //Comenzar a leer desde el principio
      .option("startingOffsets", "earliest")
      .load()

    //Casteamos los datos leidos en formato Kafka, para convertirlos en strings
    val res = df.selectExpr( exprs = "CAST(value AS STRING)")

    //Creamos el esquema (Fichero personal.json)
    val esquema = new StructType()
      .add( name = "id", IntegerType)
      .add( name = "first_name", StringType)
      .add( name = "last_name", StringType)
      .add( name = "email", StringType)
      .add( name = "gender", StringType)
      .add( name = "ip_address", StringType)

    //Hacemos una consulta que nos devuelva la información (Tengo un DataStream)
    val persona = res.select(from_json(col( colName = "value"), esquema)
      .as ( alias = "data")) //Importamos functions
      .select( col = "data.*")

    //Visualizamos la información recibida
    persona.writeStream
      .format( source = "console")
      .outputMode( outputMode = "append")
      .option("truncate", false)
      .start()

    //Hacemos una consulta que nos filtre los registros
    val personaFiltrado = persona
      .filter( conditionExpr = "data.last_name != 'Bea'")
      .filter( conditionExpr = "data.first_name != 'Giavani'")
    //Visualizamos
    personaFiltrado.writeStream
      .format( source = "console")
      .outputMode( outputMode = "append")
      .option("truncate", false)
      .start()
      .awaitTermination()
```

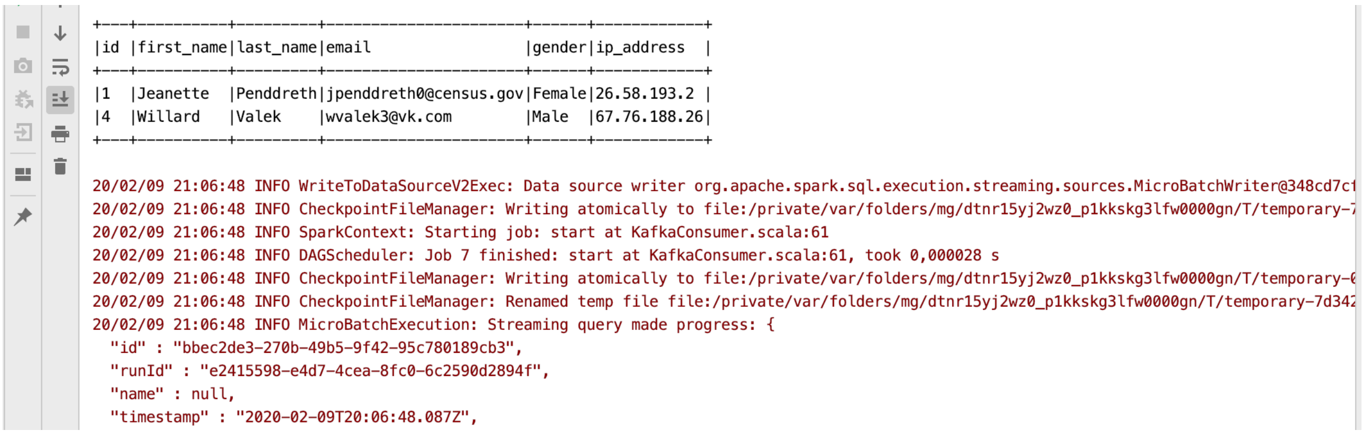
Foto 3: Aplicación con Streaming - Kafka

- Una vez ejecutada la aplicación y lanzado el productor, recogemos los resultados.



```
20/02/09 21:03:08 INFO ContextCleaner: Cleaned accumulator 25
20/02/09 21:03:08 INFO ContextCleaner: Cleaned accumulator 58
20/02/09 21:03:08 INFO ContextCleaner: Cleaned accumulator 49
20/02/09 21:03:08 INFO ContextCleaner: Cleaned accumulator 79
20/02/09 21:03:08 INFO ContextCleaner: Cleaned accumulator 81
+---+----------+---------+-----------------------+------+--------------+
|id |first_name|last_name|email                  |gender|ip_address    |
+---+----------+---------+-----------------------+------+--------------+
|1  |Jeanette  |Penddreth|jpenddreth0@census.gov|Female|26.58.193.2   |
|2  |Giavani   |Frediani |gfrediani1@senate.gov |Male  |229.179.4.212 |
|3  |Noell     |Bea      |nbea2@imageshack.us   |Female|180.66.162.255|
|4  |Willard   |Valek    |wvalek3@vk.com        |Male  |67.76.188.26  |
+---+----------+---------+-----------------------+------+--------------+

20/02/09 21:03:08 INFO BlockManagerInfo: Removed broadcast_2_piece0 on 192.168.0.14:52992 in memory (size: 6.4 KB, free: 4.1 GB)
20/02/09 21:03:08 INFO DataWritingSparkTask: Commit authorized for partition 0 (task 3, attempt 0, stage 3.0)
20/02/09 21:03:08 INFO ContextCleaner: Cleaned accumulator 68
20/02/09 21:03:08 INFO DataWritingSparkTask: Committed partition 0 (task 3, attempt 0, stage 3.0)
20/02/09 21:03:08 INFO ContextCleaner: Cleaned accumulator 82
```

Foto 4: Fichero .json recibido



```
+---+----------+---------+-----------------------+------+------------+
|id |first_name|last_name|email                  |gender|ip_address  |
+---+----------+---------+-----------------------+------+------------+
|1  |Jeanette  |Penddreth|jpenddreth0@census.gov|Female|26.58.193.2 |
|4  |Willard   |Valek    |wvalek3@vk.com        |Male  |67.76.188.26|
+---+----------+---------+-----------------------+------+------------+

20/02/09 21:06:48 INFO WriteToDataSourceV2Exec: Data source writer org.apache.spark.sql.execution.streaming.sources.MicroBatchWriter@348cd7c1
20/02/09 21:06:48 INFO CheckpointFileManager: Writing atomically to file:/private/var/folders/mg/dtnr15yj2wz0_p1kksg3lfw0000gn/T/temporary-7
20/02/09 21:06:48 INFO SparkContext: Starting job: start at KafkaConsumer.scala:61
20/02/09 21:06:48 INFO DAGScheduler: Job 7 finished: start at KafkaConsumer.scala:61, took 0,000028 s
20/02/09 21:06:48 INFO CheckpointFileManager: Writing atomically to file:/private/var/folders/mg/dtnr15yj2wz0_p1kksg3lfw0000gn/T/temporary-0
20/02/09 21:06:48 INFO CheckpointFileManager: Renamed temp file file:/private/var/folders/mg/dtnr15yj2wz0_p1kksg3lfw0000gn/T/temporary-7d342
20/02/09 21:06:48 INFO MicroBatchExecution: Streaming query made progress: {
  "id" : "bbec2de3-270b-49b5-9f42-95c780189cb3",
  "runId" : "e2415598-e4d7-4cea-8fc0-6c2590d2894f",
  "name" : null,
  "timestamp" : "2020-02-09T20:06:48.087Z",
```

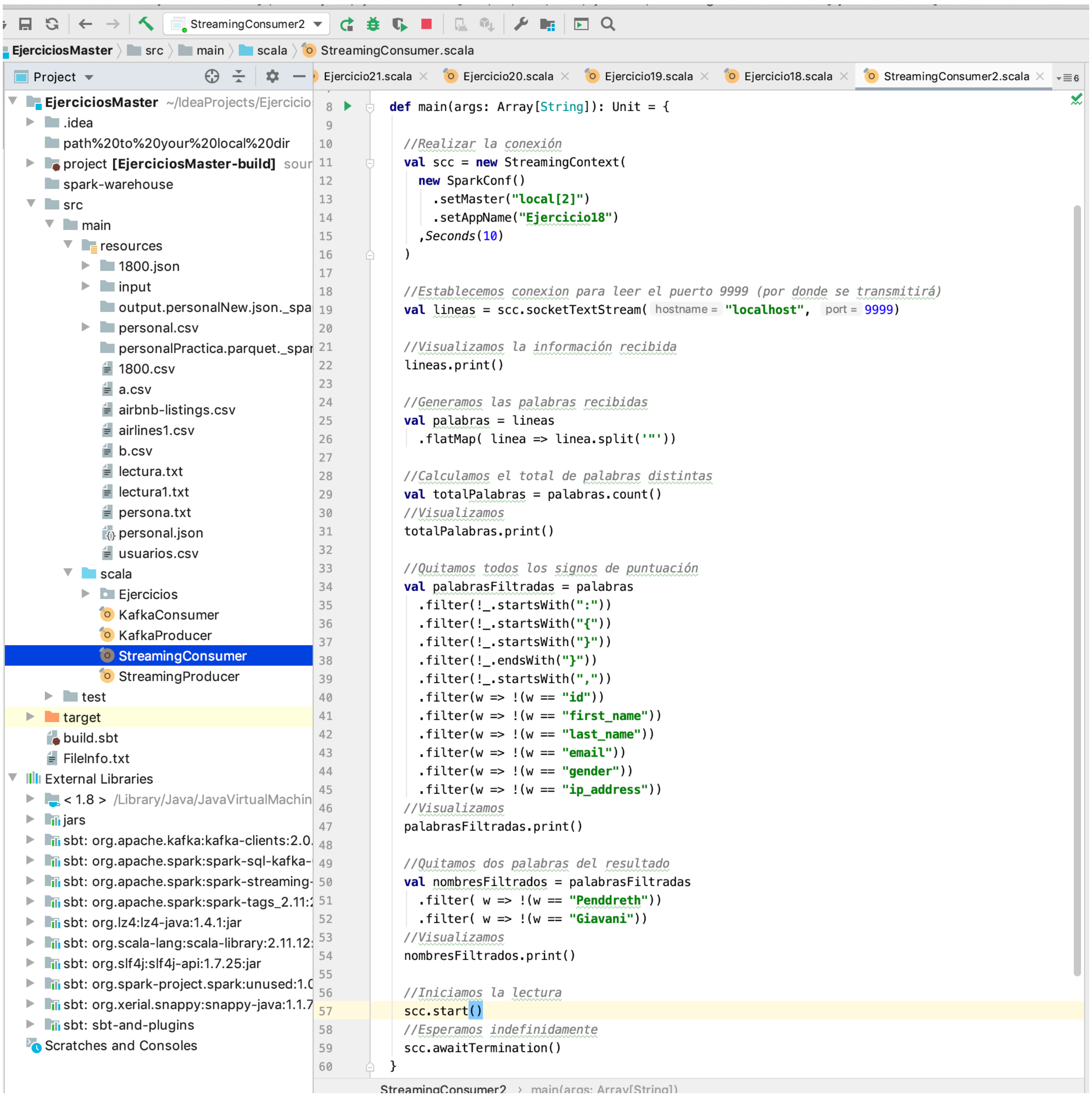
Foto 5: Datos una vez filtrados

Punto 1.2 (CON STREAMING)

- Generamos el comando a enviar como productor.

```
iMac:Desktop fernandojarilla$ cat personal.json | nc -l 9999
```

- Lanzamos la aplicación que es la siguiente y hemos generado con el IDE de IntelliJ de forma similar al punto anterior.

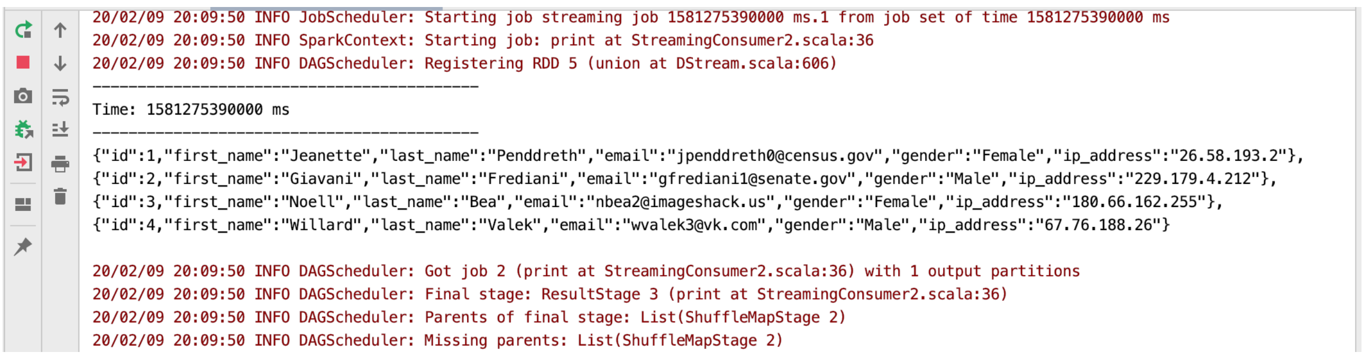


StreamingConsumer2
EjerciciosMaster › src › main › scala › StreamingConsumer.scala
Project
Ejercicio21.scala
Ejercicio20.scala
Ejercicio19.scala
Ejercicio18.scala
StreamingConsumer2.scala
EjerciciosMaster ~/IdeaProjects/Ejercicio
.idea
path%20to%20your%20local%20dir
project [EjerciciosMaster-build] sour
spark-warehouse
src
main
resources
1800.json
input
output.personalNew.json._spa
personal.csv
personalPractica.parquet._spar
1800.csv
a.csv
airbnb-listings.csv
airlines1.csv
b.csv
lectura.txt
lectura1.txt
persona.txt
personal.json
usuarios.csv
scala
Ejercicios
KafkaConsumer
KafkaProducer
StreamingConsumer
StreamingProducer
test
target
build.sbt
FileInfo.txt
External Libraries
< 1.8 > /Library/Java/JavaVirtualMachin
jars
sbt: org.apache.kafka:kafka-clients:2.0.
sbt: org.apache.spark:spark-sql-kafka-
sbt: org.apache.spark:spark-streaming-
sbt: org.apache.spark:spark-tags_2.11:2
sbt: org.lz4:lz4-java:1.4.1:jar
sbt: org.scala-lang:scala-library:2.11.12:
sbt: org.slf4j:slf4j-api:1.7.25:jar
sbt: org.spark-project.spark:unused:1.0
sbt: org.xerial.snappy:snappy-java:1.1.7
sbt: sbt-and-plugins
Scratches and Consoles
StreamingConsumer2 › main(args: Array[String])

```scala
def main(args: Array[String]): Unit = {

  //Realizar la conexión
  val scc = new StreamingContext(
    new SparkConf()
      .setMaster("local[2]")
      .setAppName("Ejercicio18")
    ,Seconds(10)
  )

  //Establecemos conexion para leer el puerto 9999 (por donde se transmitirá)
  val lineas = scc.socketTextStream( hostname = "localhost",  port = 9999)

  //Visualizamos la información recibida
  lineas.print()

  //Generamos las palabras recibidas
  val palabras = lineas
    .flatMap( linea => linea.split('"'))

  //Calculamos el total de palabras distintas
  val totalPalabras = palabras.count()
  //Visualizamos
  totalPalabras.print()

  //Quitamos todos los signos de puntuación
  val palabrasFiltradas = palabras
    .filter(!_.startsWith(":"))
    .filter(!_.startsWith("{"))
    .filter(!_.startsWith("}"))
    .filter(!_.endsWith("}"))
    .filter(!_.startsWith(","))
    .filter(w => !(w == "id"))
    .filter(w => !(w == "first_name"))
    .filter(w => !(w == "last_name"))
    .filter(w => !(w == "email"))
    .filter(w => !(w == "gender"))
    .filter(w => !(w == "ip_address"))
  //Visualizamos
  palabrasFiltradas.print()

  //Quitamos dos palabras del resultado
  val nombresFiltrados = palabrasFiltradas
    .filter( w => !(w == "Penddreth"))
    .filter( w => !(w == "Giavani"))
  //Visualizamos
  nombresFiltrados.print()

  //Iniciamos la lectura
  scc.start()
  //Esperamos indefinidamente
  scc.awaitTermination()
}
```

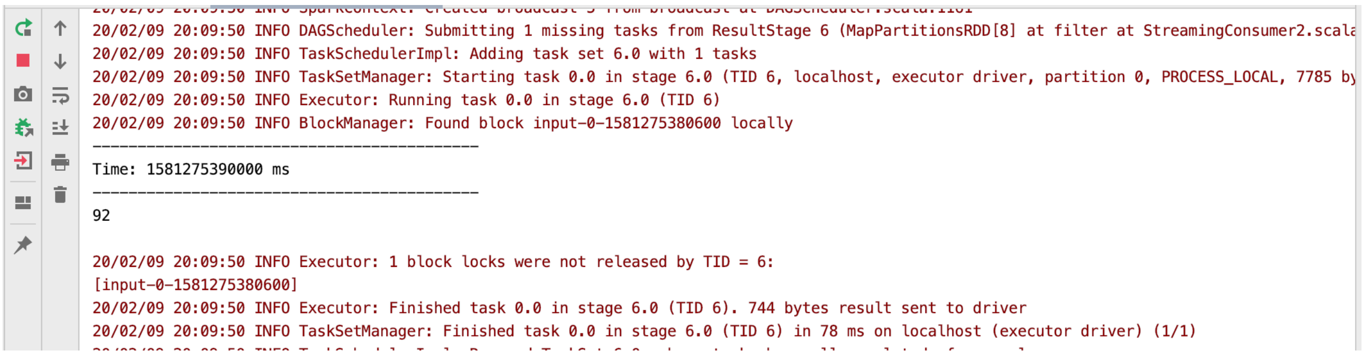
Foto 1: Programa fuente

- Tras la ejecución los resultados son:



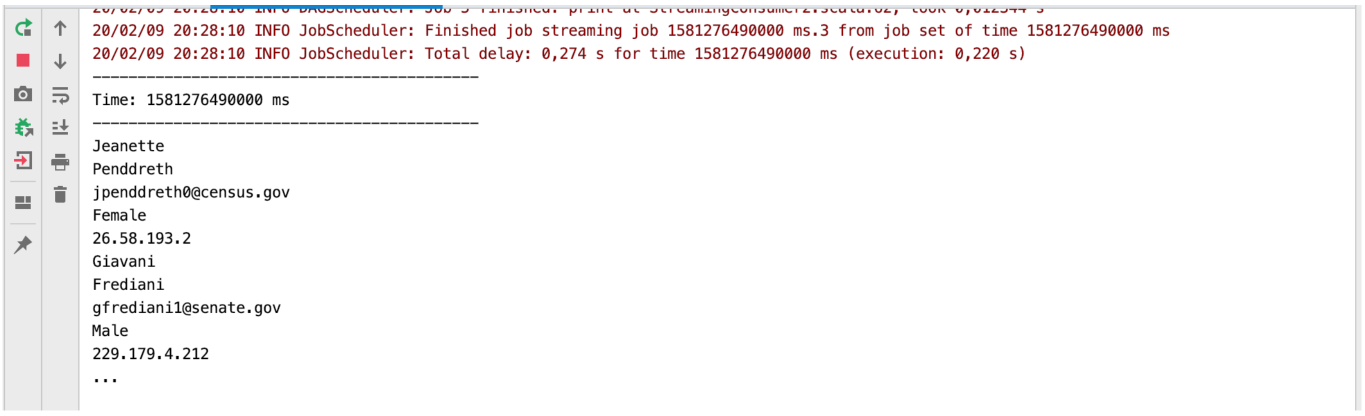
```
20/02/09 20:09:50 INFO JobScheduler: Starting job streaming job 1581275390000 ms.1 from job set of time 1581275390000 ms
20/02/09 20:09:50 INFO SparkContext: Starting job: print at StreamingConsumer2.scala:36
20/02/09 20:09:50 INFO DAGScheduler: Registering RDD 5 (union at DStream.scala:606)
-------------------------------------------
Time: 1581275390000 ms
-------------------------------------------
{"id":1,"first_name":"Jeanette","last_name":"Penddreth","email":"jpenddreth0@census.gov","gender":"Female","ip_address":"26.58.193.2"},
{"id":2,"first_name":"Giavani","last_name":"Frediani","email":"gfrediani1@senate.gov","gender":"Male","ip_address":"229.179.4.212"},
{"id":3,"first_name":"Noell","last_name":"Bea","email":"nbea2@imageshack.us","gender":"Female","ip_address":"180.66.162.255"},
{"id":4,"first_name":"Willard","last_name":"Valek","email":"wvalek3@vk.com","gender":"Male","ip_address":"67.76.188.26"}

20/02/09 20:09:50 INFO DAGScheduler: Got job 2 (print at StreamingConsumer2.scala:36) with 1 output partitions
20/02/09 20:09:50 INFO DAGScheduler: Final stage: ResultStage 3 (print at StreamingConsumer2.scala:36)
20/02/09 20:09:50 INFO DAGScheduler: Parents of final stage: List(ShuffleMapStage 2)
20/02/09 20:09:50 INFO DAGScheduler: Missing parents: List(ShuffleMapStage 2)
```

Foto 2: Fichero .json recibido por streaming



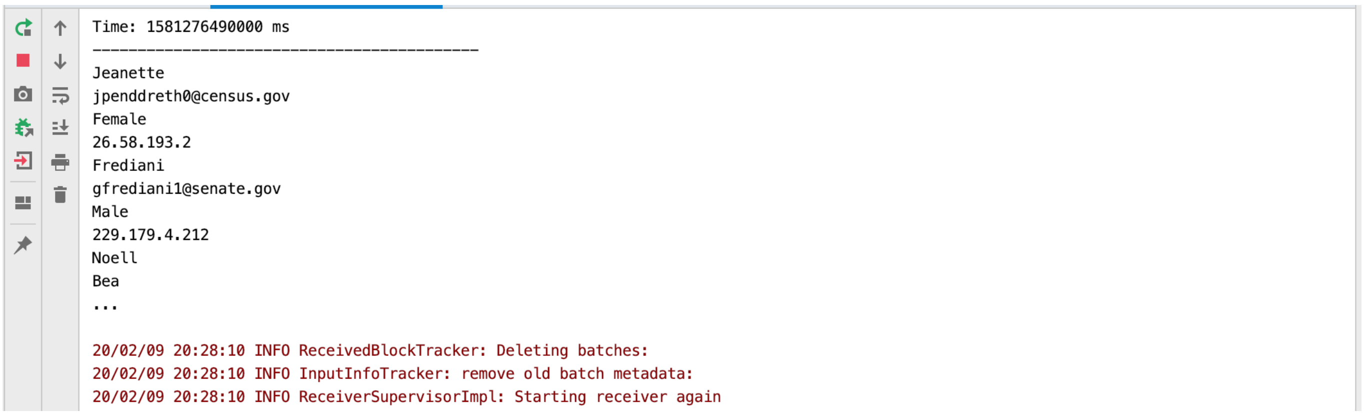
```
20/02/09 20:09:50 INFO DAGScheduler: Submitting 1 missing tasks from ResultStage 6 (MapPartitionsRDD[8] at filter at StreamingConsumer2.scala
20/02/09 20:09:50 INFO TaskSchedulerImpl: Adding task set 6.0 with 1 tasks
20/02/09 20:09:50 INFO TaskSetManager: Starting task 0.0 in stage 6.0 (TID 6, localhost, executor driver, partition 0, PROCESS_LOCAL, 7785 by
20/02/09 20:09:50 INFO Executor: Running task 0.0 in stage 6.0 (TID 6)
20/02/09 20:09:50 INFO BlockManager: Found block input-0-1581275380600 locally
-------------------------------------------
Time: 1581275390000 ms
-------------------------------------------
92

20/02/09 20:09:50 INFO Executor: 1 block locks were not released by TID = 6:
[input-0-1581275380600]
20/02/09 20:09:50 INFO Executor: Finished task 0.0 in stage 6.0 (TID 6). 744 bytes result sent to driver
20/02/09 20:09:50 INFO TaskSetManager: Finished task 0.0 in stage 6.0 (TID 6) in 78 ms on localhost (executor driver) (1/1)
```

Foto 3: Total de palabras contadas



```
20/02/09 20:28:10 INFO JobScheduler: Finished job streaming job 1581276490000 ms.3 from job set of time 1581276490000 ms
20/02/09 20:28:10 INFO JobScheduler: Total delay: 0,274 s for time 1581276490000 ms (execution: 0,220 s)
-------------------------------------------
Time: 1581276490000 ms
-------------------------------------------
Jeanette
Penddreth
jpenddreth0@census.gov
Female
26.58.193.2
Giavani
Frediani
gfrediani1@senate.gov
Male
229.179.4.212
...
```

Foto 4: Palabras del fichero



```
Time: 1581276490000 ms
-------------------------------------------
Jeanette
jpenddreth0@census.gov
Female
26.58.193.2
Frediani
gfrediani1@senate.gov
Male
229.179.4.212
Noell
Bea
...

20/02/09 20:28:10 INFO ReceivedBlockTracker: Deleting batches:
20/02/09 20:28:10 INFO InputInfoTracker: remove old batch metadata:
20/02/09 20:28:10 INFO ReceiverSupervisorImpl: Starting receiver again
```

Foto 5: Palabras tras aplicar los filtros "Penddreth" y "Giavani".