

```
StreamingEntregaplementacion <
StreamingEntregaplementacion <

Batch: 3

-----
| key | value | topic | partition | offset | timestamp | timestampType |
-----
| null | [78 22 62 79 74 6...] | devices | 0 | 936 | 2021-02-18 18:32: | 0 |
| null | [78 22 62 79 74 6...] | devices | 0 | 935 | 2021-02-18 18:32: | 0 |
| null | [78 22 62 79 74 6...] | devices | 0 | 934 | 2021-02-18 18:32: | 0 |
| null | [78 22 62 79 74 6...] | devices | 0 | 937 | 2021-02-18 18:32: | 0 |
| null | [78 22 62 79 74 6...] | devices | 0 | 938 | 2021-02-18 18:32: | 0 |
| null | [78 22 62 79 74 6...] | devices | 0 | 939 | 2021-02-18 18:32: | 0 |
| null | [78 22 62 79 74 6...] | devices | 0 | 940 | 2021-02-18 18:32: | 0 |
| null | [78 22 62 79 74 6...] | devices | 0 | 941 | 2021-02-18 18:32: | 0 |
| null | [78 22 62 79 74 6...] | devices | 0 | 942 | 2021-02-18 18:32: | 0 |
| null | [78 22 62 79 74 6...] | devices | 0 | 943 | 2021-02-18 18:32: | 0 |
| null | [78 22 62 79 74 6...] | devices | 0 | 944 | 2021-02-18 18:32: | 0 |
| null | [78 22 62 79 74 6...] | devices | 0 | 945 | 2021-02-18 18:32: | 0 |
| null | [78 22 62 79 74 6...] | devices | 0 | 946 | 2021-02-18 18:32: | 0 |
| null | [78 22 62 79 74 6...] | devices | 0 | 947 | 2021-02-18 18:32: | 0 |
| null | [78 22 62 79 74 6...] | devices | 0 | 948 | 2021-02-18 18:32: | 0 |
```

Nos quedamos con el valor y lo convertimos de array de bytes a Stringtype y el esquema lo obtengo de la case class que tengo definida. (imagen 4)

timestamp	app	id	antenna_id	bytes
1615673626	SKYPE	00000000-0000-0000-0000-00000000-0000-0000	00000000-0000-0000-0000-00000000-0000-0000	9848
1615673626	FACEBOOK	00000000-0000-0000-0000-00000000-0000-0000	00000000-0000-0000-0000-00000000-0000-0000	2529
1615673626	TELEGRAM	00000000-0000-0000-0000-00000000-0000-0000	00000000-0000-0000-0000-00000000-0000-0000	1080
1615673626	FACEBOOK	00000000-0000-0000-0000-00000000-0000-0000	00000000-0000-0000-0000-00000000-0000-0000	1416
1615673626	SKYPE	00000000-0000-0000-0000-00000000-0000-0000	00000000-0000-0000-0000-00000000-0000-0000	3787
1615673626	SKYPE	00000000-0000-0000-0000-00000000-0000-0000	00000000-0000-0000-0000-00000000-0000-0000	7345
1615673626	FACETIME	00000000-0000-0000-0000-00000000-0000-0000	00000000-0000-0000-0000-00000000-0000-0000	8581
1615673626	FACETIME	00000000-0000-0000-0000-00000000-0000-0000	00000000-0000-0000-0000-00000000-0000-0000	9784
1615673626	FACEBOOK	00000000-0000-0000-0000-00000000-0000-0000	00000000-0000-0000-0000-00000000-0000-0000	3850
1615673626	FACETIME	00000000-0000-0000-0000-00000000-0000-0000	00000000-0000-0000-0000-00000000-0000-0000	2068
1615673626	SKYPE	00000000-0000-0000-0000-00000000-0000-0000	11111111-1111-1111-1111-11111111-1111-1111	7399
1615673626	FACETIME	00000000-0000-0000-0000-00000000-0000-0000	11111111-1111-1111-1111-11111111-1111-1111	2218
1615673626	FACEBOOK	00000000-0000-0000-0000-00000000-0000-0000	11111111-1111-1111-1111-11111111-1111-1111	5372
1615673626	TELEGRAM	00000000-0000-0000-0000-00000000-0000-0000	11111111-1111-1111-1111-11111111-1111-1111	3696

```

C:\Users\evaes\j\jdk\corretto-1.8.0_262\bin\java.exe ...

id| name| email| quota|
-----|-----|-----|-----|
00000000-0000-0000-0000-000000000000| andres| andres@gmail.com| 2000000|
00000000-0000-0000-0000-000000000000| paco| paco@gmail.com| 3000000|
00000000-0000-0000-0000-000000000000| juan| juan@gmail.com| 1000000|
00000000-0000-0000-0000-000000000000| fede| fede@gmail.com| 5000|
00000000-0000-0000-0000-000000000000| gorka| gorka@gmail.com| 2000000|
00000000-0000-0000-0000-000000000000| luis| luis@gmail.com| 2000000|
00000000-0000-0000-0000-000000000000| eric| eric@gmail.com| 3000000|
00000000-0000-0000-0000-000000000000| carlos| carlos@gmail.com| 1000000|
00000000-0000-0000-0000-000000000000| david| david@gmail.com| 3000000|
00000000-0000-0000-0000-000000000000| juanchug| juanchug@gmail.com| 3000000|
00000000-0000-0000-0000-000000000000| charo| charo@gmail.com| 3000000|
00000000-0000-0000-0000-000000000000| delicias| delicias@gmail.com| 10000000|
00000000-0000-0000-0000-000000000000| nilagros| nilagros@gmail.com| 2000000|
00000000-0000-0000-0000-000000000000| antonio| antonio@gmail.com| 10000000|
00000000-0000-0000-0000-000000000000| sergio| sergio@gmail.com| 10000000|
00000000-0000-0000-0000-000000000000| maria| maria@gmail.com| 10000000|
00000000-0000-0000-0000-000000000000| cristine| cristine@gmail.com| 3000000|
00000000-0000-0000-0000-000000000000| lucia| lucia@gmail.com| 3000000|
00000000-0000-0000-0000-000000000000| carlota| carlota@gmail.com| 2000000|
00000000-0000-0000-0000-000000000000| emilio| emilio@gmail.com| 2000000|
  
```

```

StreamingEngineImplementation X
-----
Batch: 1
-----
| timestamp | app | id | antenna_id | bytes | name | email | quota |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 16163677630 | TELEGRAM | 000000000-0000-0000-... | 111111111-1111-1111-... | 6947 | delicias | delicias@gmail.com | 10000000 |
| 16163677630 | SKYPE | 000000000-0000-0000-... | 000000000-0000-0000-... | 8543 | luis | luis@gmail.com | 200000 |
| 16163677630 | SKYPE | 000000000-0000-0000-... | 000000000-0000-0000-... | 4964 | eric | eric@gmail.com | 300000 |
| 16163677630 | FACETIME | 000000000-0000-0000-... | 000000000-0000-0000-... | 9848 | andres | andres@gmail.com | 200000 |
| 16163677630 | TELEGRAM | 000000000-0000-0000-... | 000000000-0000-0000-... | 6435 | juanchun | juanchun@gmail.com | 300000 |
| 16163677630 | FACETIME | 000000000-0000-0000-... | 22222222-2222-2222-... | 7712 | carlota | carlota@gmail.com | 200000 |
| 16163677630 | FACEBOOK | 000000000-0000-0000-... | 22222222-2222-2222-... | 4643 | emilio | emilio@gmail.com | 300000 |
| 16163677630 | FACETIME | 000000000-0000-0000-... | 111111111-1111-1111-... | 8872 | miguelos | miguelos@gmail.com | 200000 |
| 16163677630 | FACETIME | 000000000-0000-0000-... | 000000000-0000-0000-... | 1448 | fede | fede@gmail.com | 5000 |
| 16163677630 | FACETIME | 000000000-0000-0000-... | 22222222-2222-2222-... | 8367 | lucia | lucia@gmail.com | 300000 |
| 16163677630 | FACETIME | 000000000-0000-0000-... | 11111111-1111-1111-... | 5911 | charo | charo@gmail.com | 300000 |
| 16163677630 | SKYPE | 000000000-0000-0000-... | 111111111-1111-1111-... | 2570 | antonio | antonio@gmail.com | 1000000 |
| 16163677630 | TELEGRAM | 000000000-0000-0000-... | 000000000-0000-0000-... | 9505 | gorka | gorka@gmail.com | 200000 |
| 16163677630 | TELEGRAM | 000000000-0000-0000-... | 000000000-0000-0000-... | 4044 | carlos | carlos@gmail.com | 100000 |
| 16163677630 | FACEBOOK | 000000000-0000-0000-... | 000000000-0000-0000-... | 6916 | david | david@gmail.com | 300000 |
| 16163677630 | FACEBOOK | 000000000-0000-0000-... | 22222222-2222-2222-... | 6862 | cristina | cristina@gmail.com | 300000 |
| 16163677630 | TELEGRAM | 000000000-0000-0000-... | 000000000-0000-0000-... | 7588 | paco | paco@gmail.com | 300000 |
| 16163677630 | TELEGRAM | 000000000-0000-0000-... | 11111111-1111-1111-... | 4186 | sergio | sergio@gmail.com | 1000000 |
| 16163677630 | TELEGRAM | 000000000-0000-0000-... | 22222222-2222-2222-... | 8490 | maria | maria@gmail.com | 1000000 |
| 16163677630 | FACEBOOK | 000000000-0000-0000-... | 000000000-0000-0000-... | 6422 | juan | juan@gmail.com | 100000 |

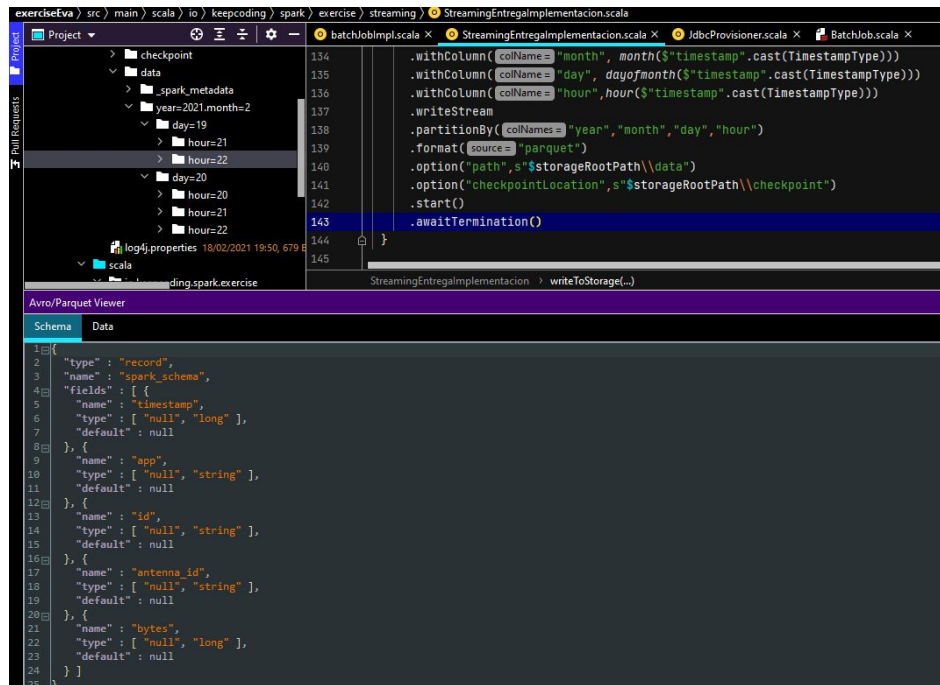
```

[illegible]

nuevos datos una vez realizadas las agregaciones por los elementos solicitados.

2.5. Almacenamiento en local en formato parquet de copia de la información recibida de la antena particionada por hora, día, mes y año. (imagen.8)

Por último con respecto a streaming, configuro el job para que los datos que recibe de Kafka, además de la agregación y enviarlos a postgres, sean guardados en formato parquet (en local) particionados por hora, día, mes y año.



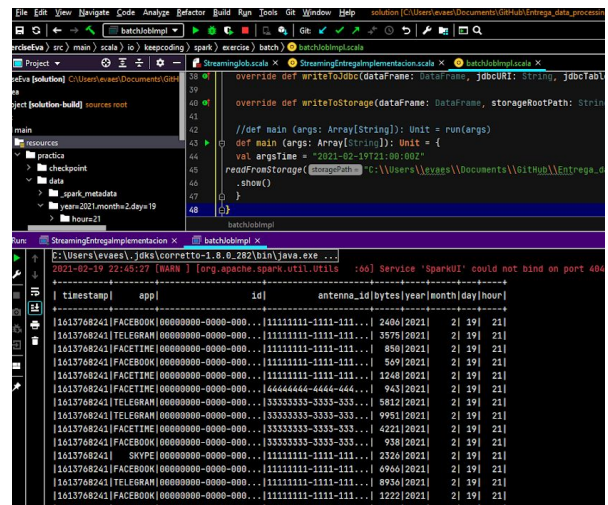
Resultados Speed Layer: En la base de datos almaceno información del consumo de datos por usuario, por aplicación y de los datos recibidos por cada antena. En storage local (o en versión futura en google storage para garantizar la permanencia de los datos) almaceno una copia de los datos de las antenas, organizados por hora, día, mes y año.

3. Batch Layer: Análisis de datos del storage

A través de la implementación de un job de SparkSQL, filtrando por hora:

3.1. Obtengo los datos del storage local.

Confirmando que puedo leer los datos de streaming guardados en local en formato parquet y cargarlos en el job de batch (imagen.9)



3.2. Accedo igualmente a los metadatos de los usuarios para poder completar la información que tengo de ellos de cara al análisis de correo y cuota de usuario por hora y combino los datos con los del storage. (imagen.10)

The screenshot shows a Java IDE with a code editor and a console window. The code in the editor is as follows:

```

antennaDF= readFromStorage( storagePath= "C:\\Users\\evaes\\Documents\\61thHub\\Entrega_data_processing\\
userMetadataDF = readAntennaMetadata( jdbcURI= "jdbc:postgresql://34.76.48.93:5432/postgres", jdbcU
antennaMetadataDF = enrichAntennaWithMetadata(antennaDF,userMetadataDF).cache()

```

The console window shows the output of a Java application, displaying a table with the following columns: timestamp, app, id, antenna_id, bytes, year, month, day, hour, name, email, and quota. The table contains 20 rows of data, including entries for FACEBOOK, TELEGRAM, and SKYPE, with details on user IDs, antenna IDs, and byte counts.

3.3. Agrego los datos por antena, aplicación y usuario (confirmando que el mail por usuario e id del móvil en este caso es único y se pueden usar indistintamente) para cada hora y los guardo en una tabla específica en SQL (bytes_hourly). (imagen.11)

The screenshot shows a PostgreSQL terminal window with a SQL query and its results. The query is:

```

postgres=> select*from bytes_hourly;

```

The results are displayed in a table with the following columns: timestamp, id, value, and type. The table contains 29 rows of data, including entries for antenna_total_bytes, app_total_bytes, and user_total_bytes, with details on timestamps, IDs, and values.

3.4. Filtro por email los usuarios que han sobrepasado la cuota por hora y los almaceno en tabla específica (use_quota_limit).

Agrego los bytes consumidos por email y filtro aquellos que sobrepasan la cuota de cada usuario, guardando en postgres los datos de los que superan esta cuota por hora. Lanzo el job teniendo en cuenta 4h distintas para obtener distintos resultados. La tabla confirmo que guarda los datos de los que sobrepasan cada hora. (imagen.12)

```
postgres=> select* from user_quota_limit;
      email      | usage | quota |      timestamp
-----+-----+-----+-----
 fede@gmail.com | 57022 |  5000 | 2021-02-19 21:00:00
(1 row)
```

```
postgres=> select* from user_quota_limit;
      email      | usage | quota |      timestamp
-----+-----+-----+-----
 fede@gmail.com | 57022 |  5000 | 2021-02-19 21:00:00
(1 row)
```

```
postgres=> select* from user_quota_limit;
      email      | usage | quota |      timestamp
-----+-----+-----+-----
 fede@gmail.com | 57022 |  5000 | 2021-02-19 21:00:00
 fede@gmail.com | 71236 |  5000 | 2021-02-19 22:00:00
(2 rows)
```

```
postgres=> select* from user_quota_limit;
      email      | usage | quota |      timestamp
-----+-----+-----+-----
 fede@gmail.com | 57022 |  5000 | 2021-02-19 21:00:00
 fede@gmail.com | 71236 |  5000 | 2021-02-19 22:00:00
 fede@gmail.com | 82488 |  5000 | 2021-02-20 22:00:00
(3 rows)
```

```
postgres=> select* from user_quota_limit;
      email      | usage | quota |      timestamp
-----+-----+-----+-----
 fede@gmail.com | 57022 |  5000 | 2021-02-19 21:00:00
 fede@gmail.com | 71236 |  5000 | 2021-02-19 22:00:00
 fede@gmail.com | 82488 |  5000 | 2021-02-20 22:00:00
 juan@gmail.com | 122829 | 100000 | 2021-02-20 21:00:00
 fede@gmail.com | 132569 |  5000 | 2021-02-20 21:00:00
 carlos@gmail.com | 111038 | 100000 | 2021-02-20 21:00:00
(6 rows)
```