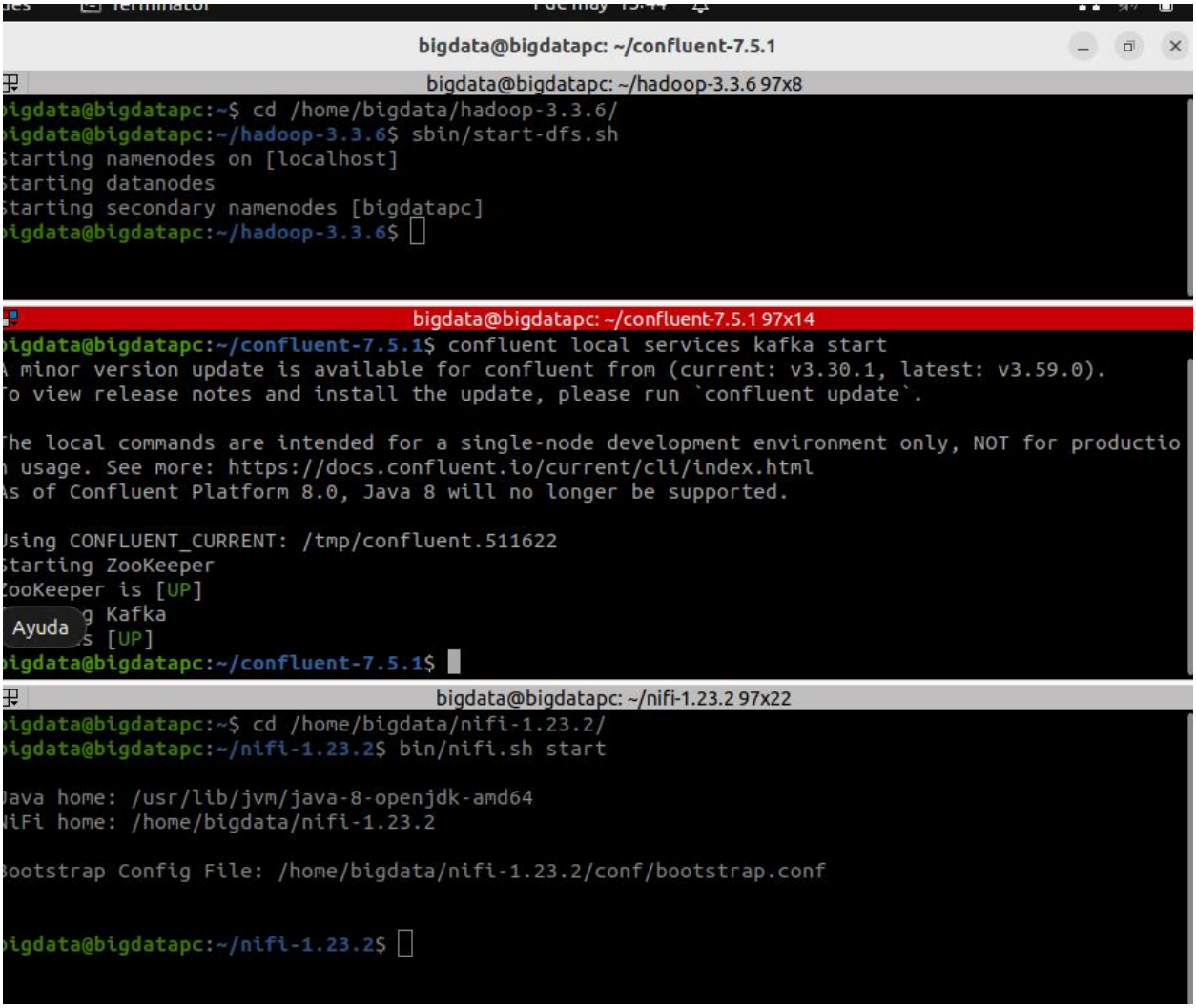
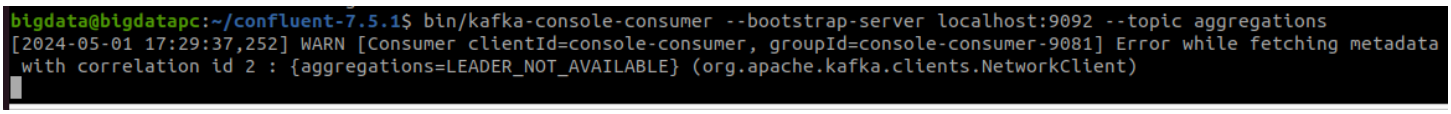
Proyecto Final

SPRINT 4

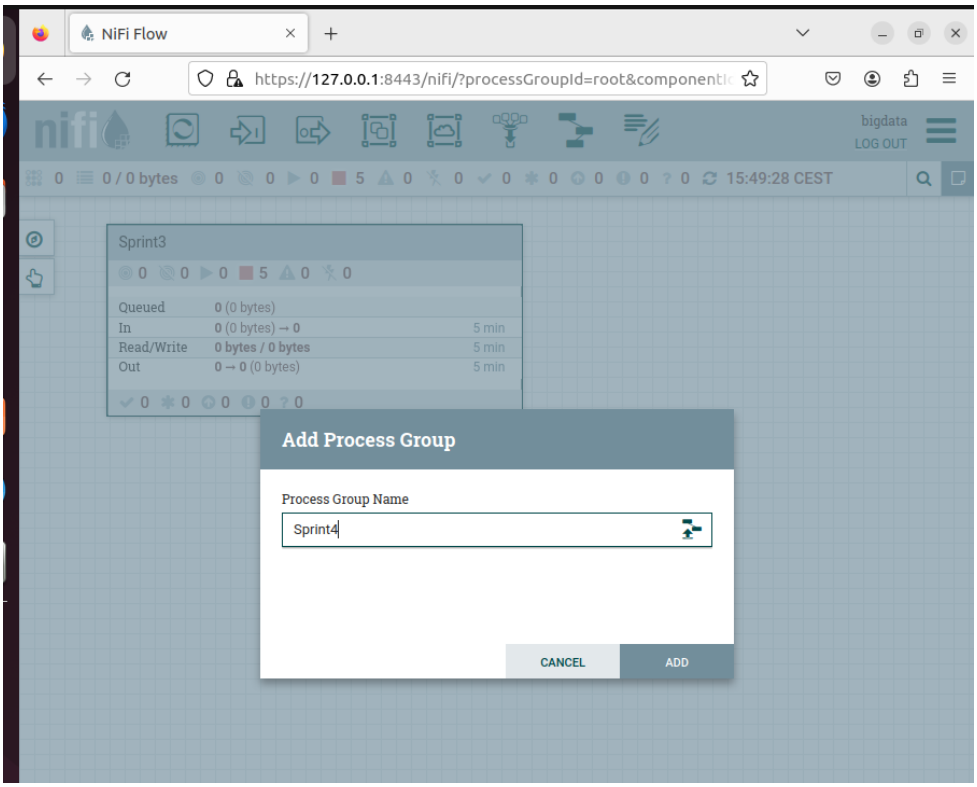
Arrancar HDFS, NIFI y la plataforma de Confluent.



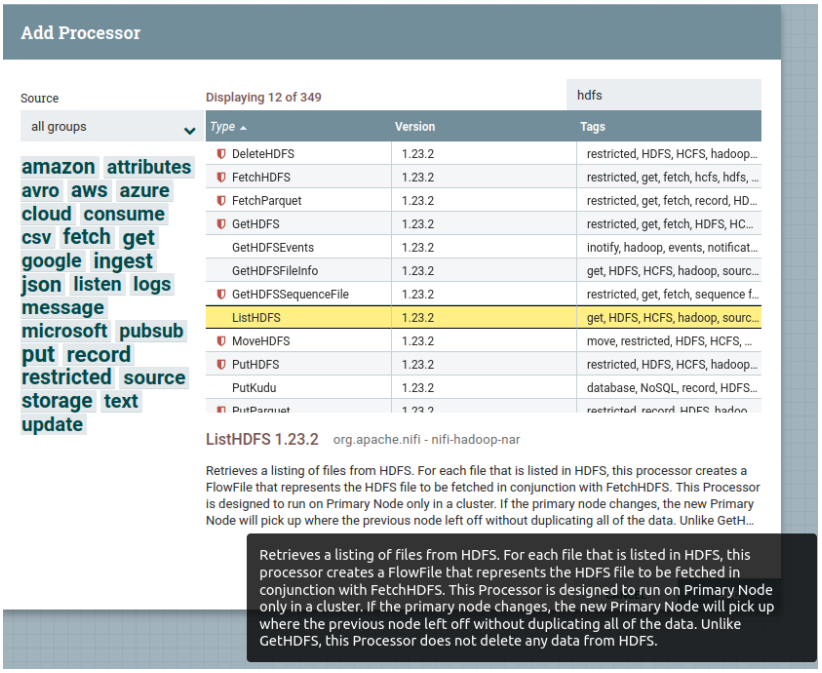
Se pone un consumidor donde se volcaran las agregaciones que provienen de la ruta HDFS. Este lo volveremos a usar más adelante cuando lo hayamos definido en NiFi.



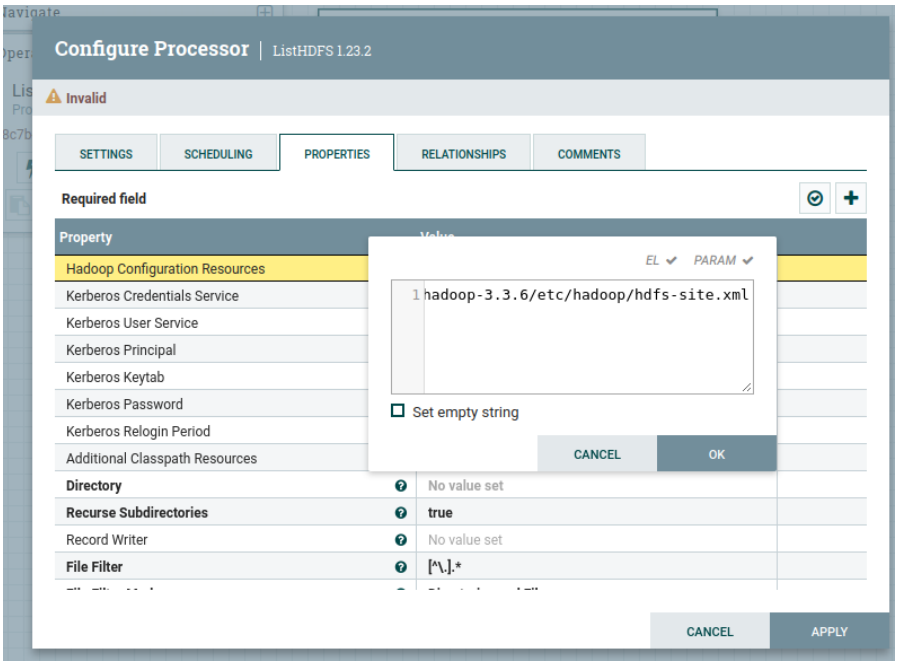
Accedemos a NiFi y creamos un nuevo Proccess Group.



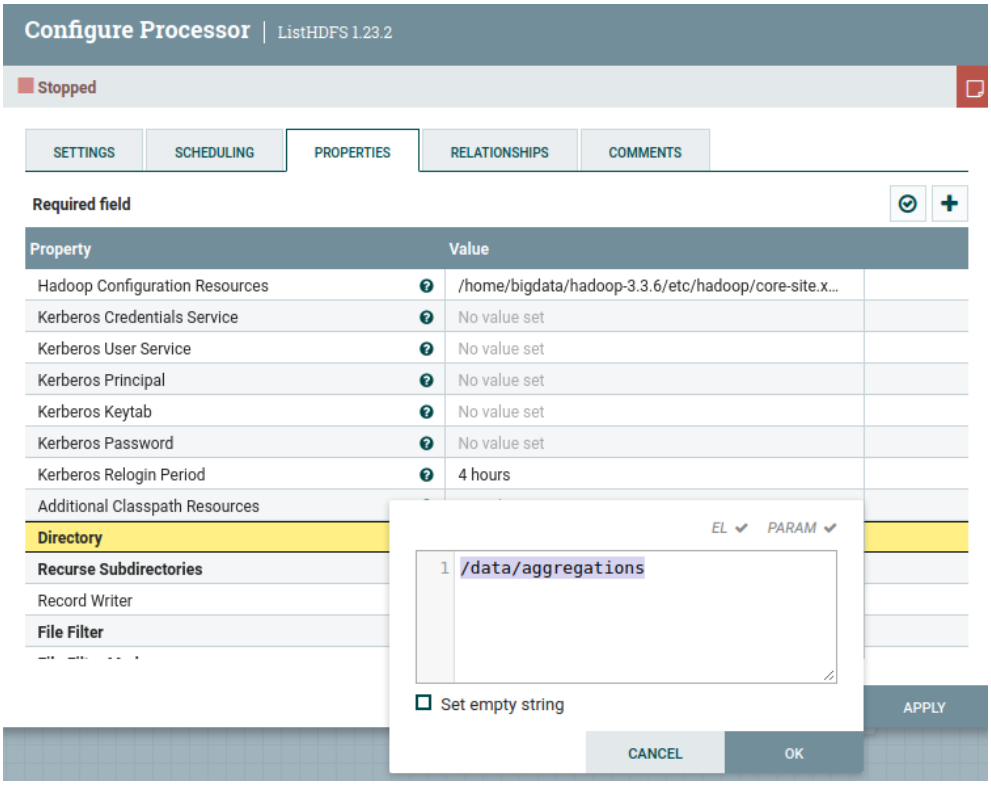
El primer procesador que se añadirá es para monitorear datos de una carpeta HDFS, en esta ocasión se utilizara ListHDFS.



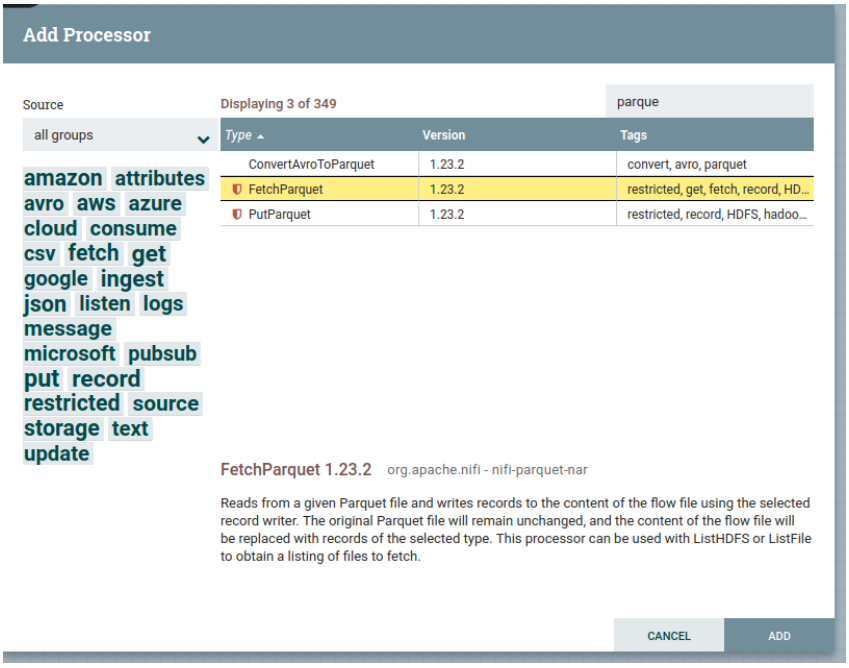
Se establecen los ficheros de configuración de HDFS.



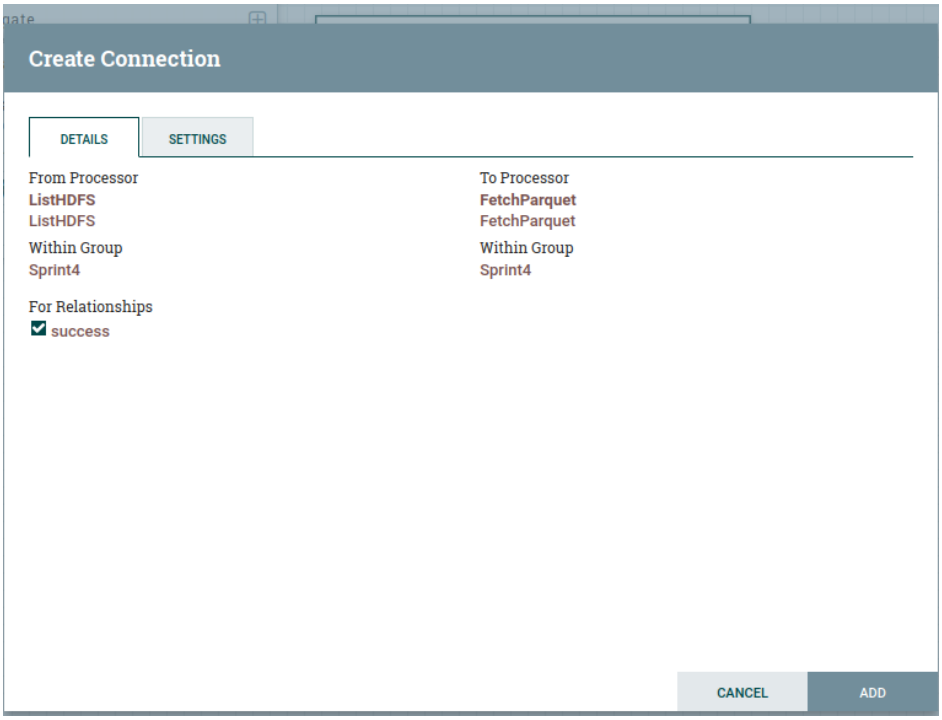
Se establece la dirección del directorio



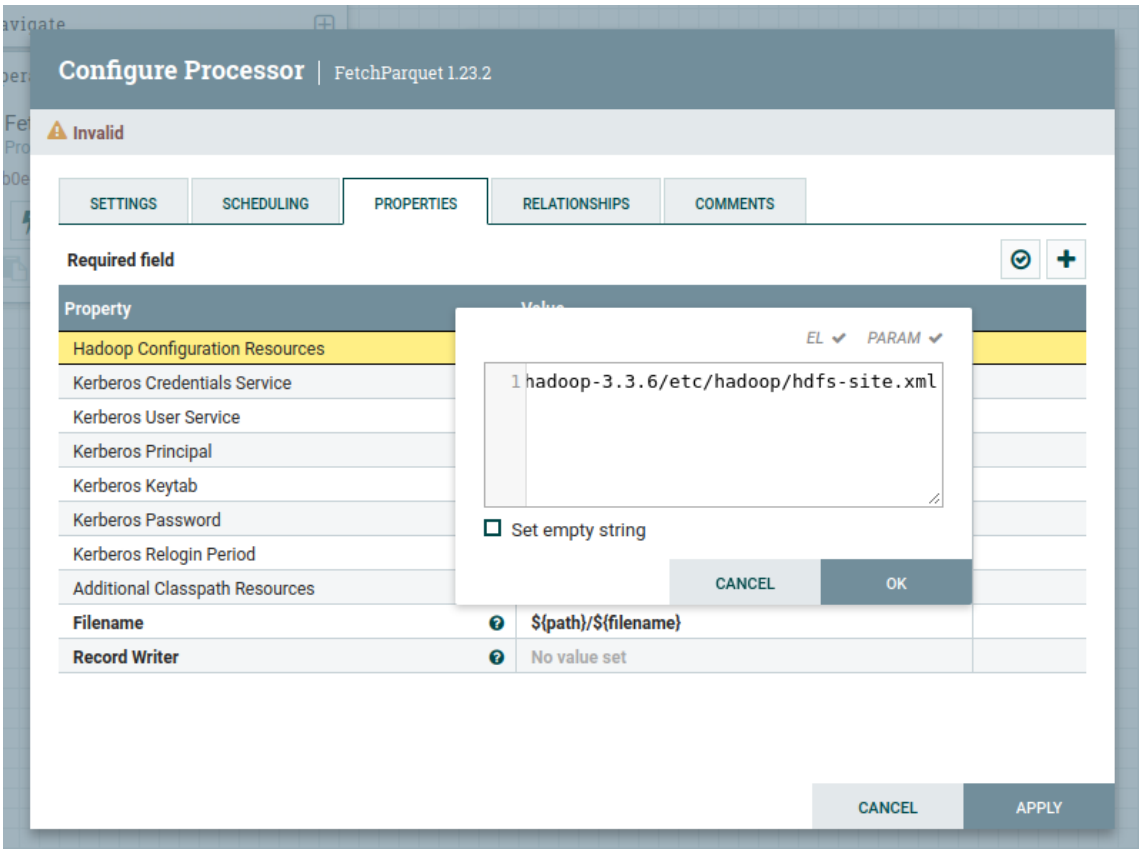
El siguiente procesador será fetch parquet para poder leer el contenido de un fichero parquet y escribir el contenido en un flow file.



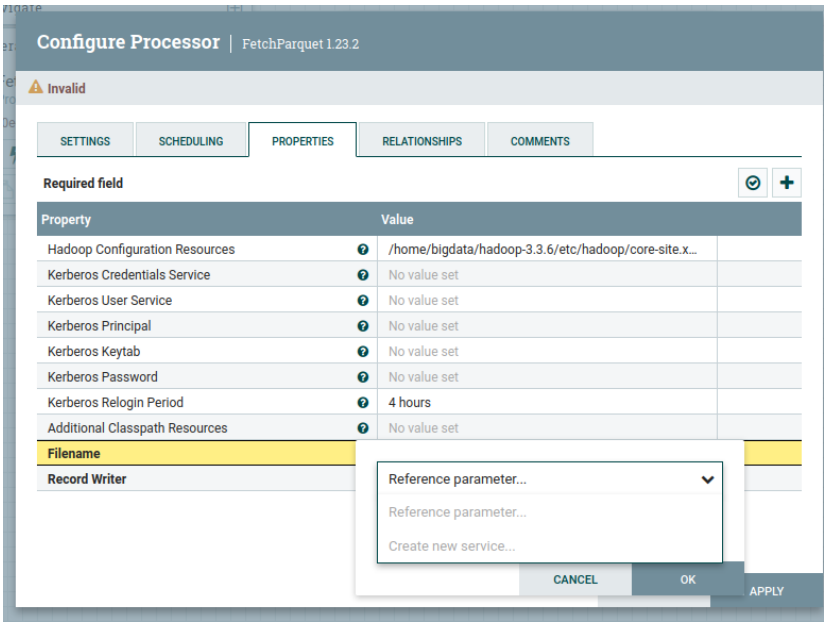
Se conecta la relación de success del anterior procesador, con este.



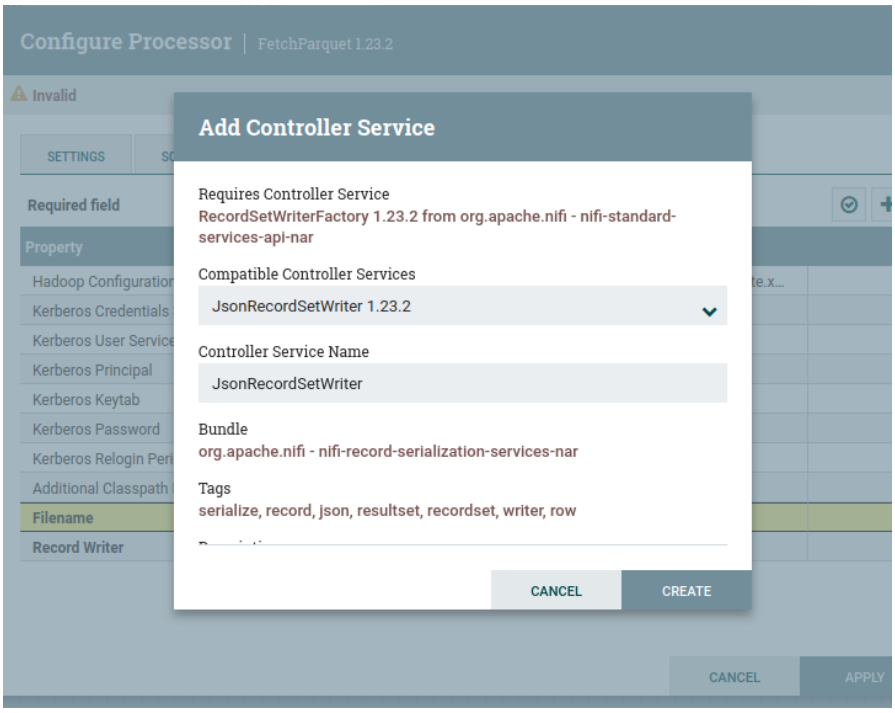
Este procesador también necesita la configuración de hadoop.



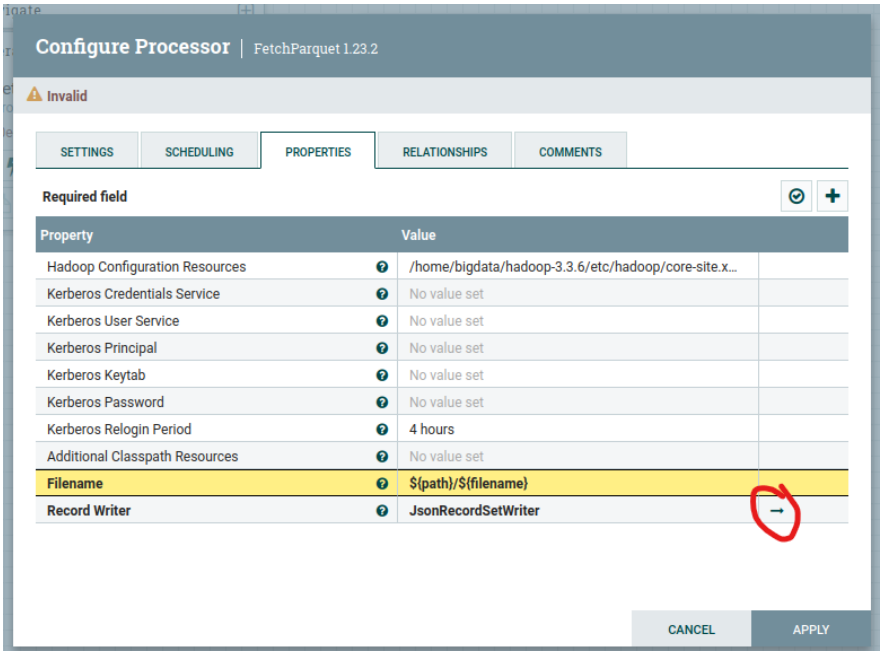
Se necesita configurar un record writer, se le da a la opción de créate new service.



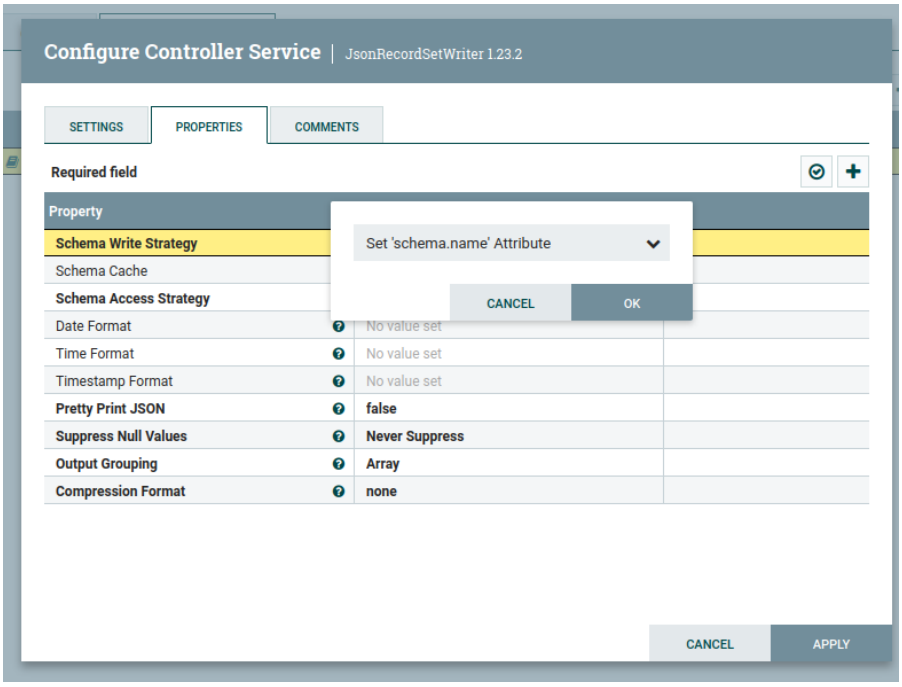
Se selecciona uno de JSON.

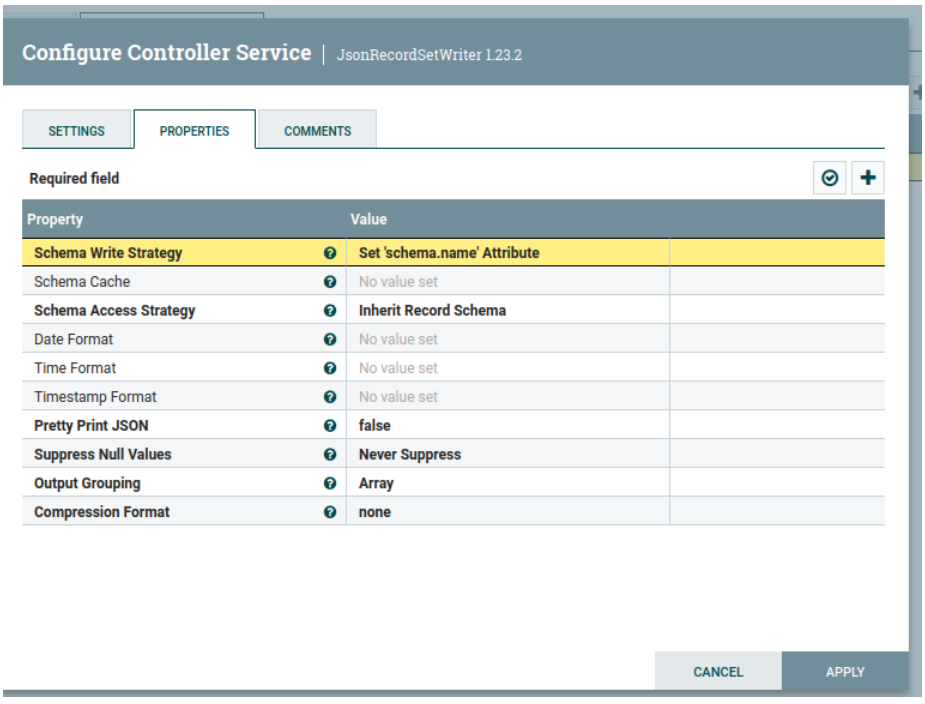


Le damos a la flecha para configurarlo.

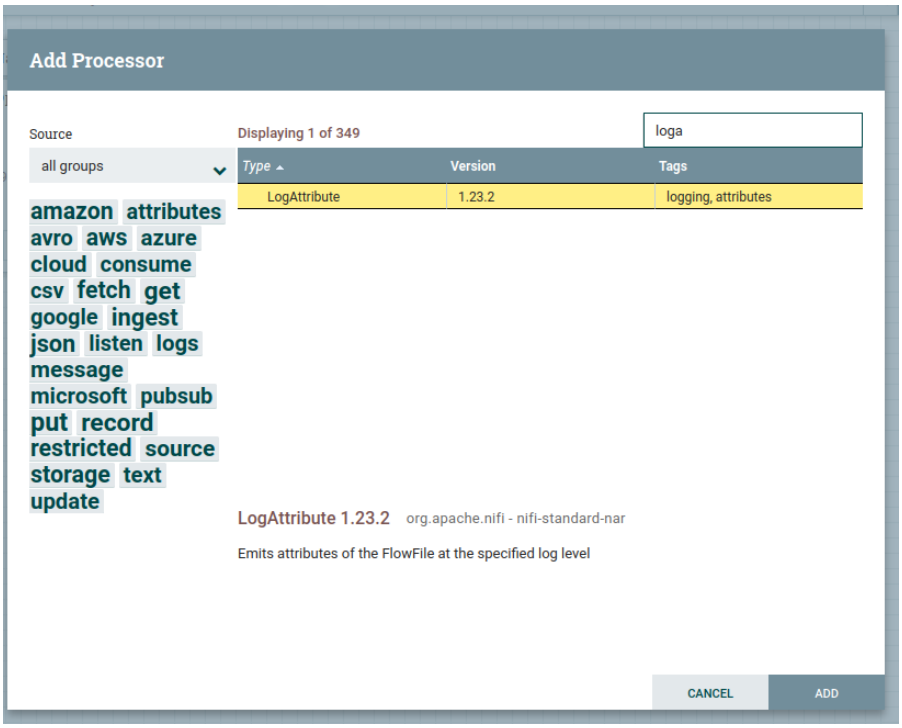


Se establece el schema.name para propagar el esquema de los flows files.

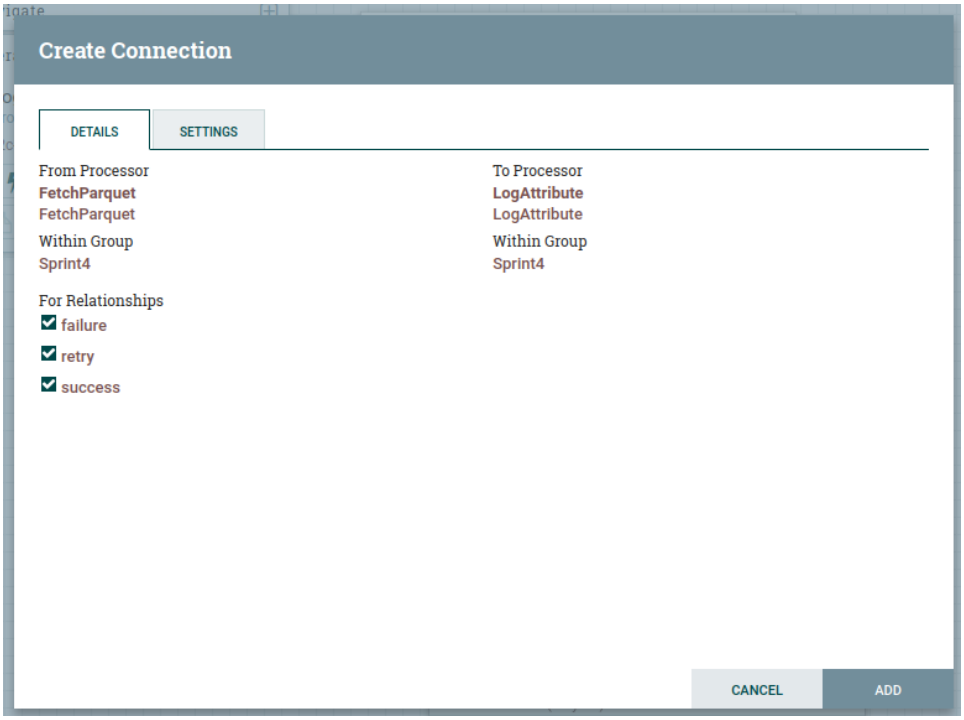




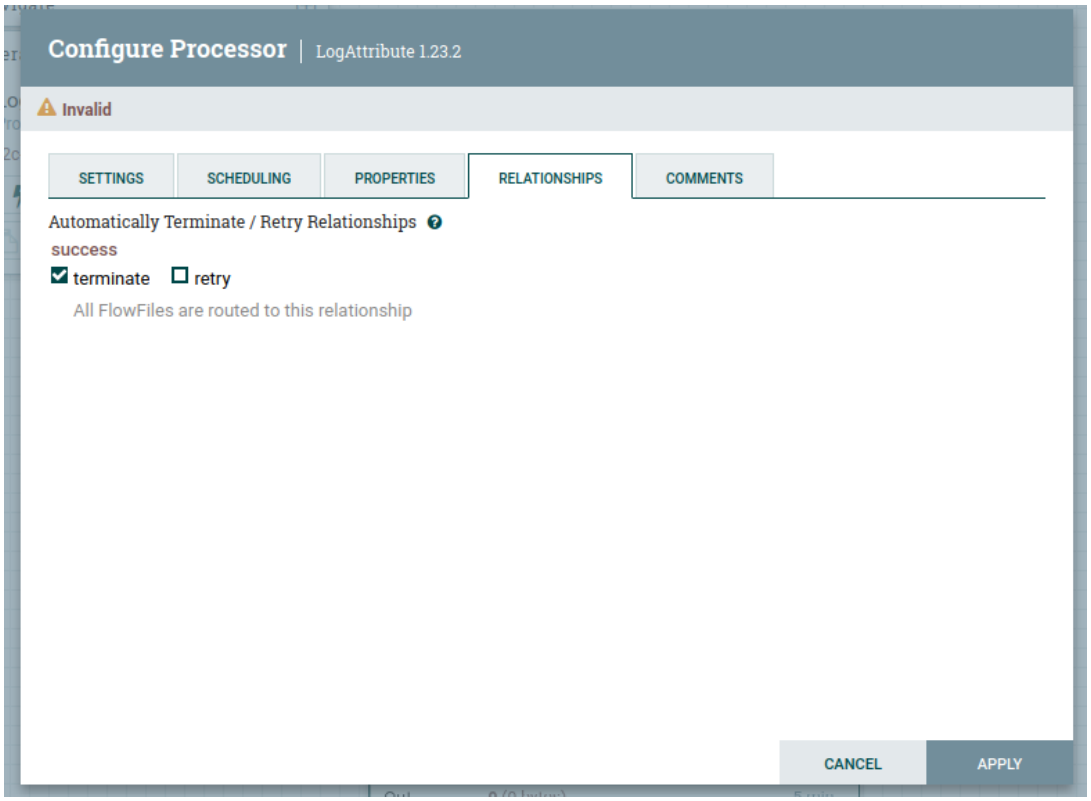
Agregamos otro procesador de log attribute, en caso de que algo falle tener el registro.



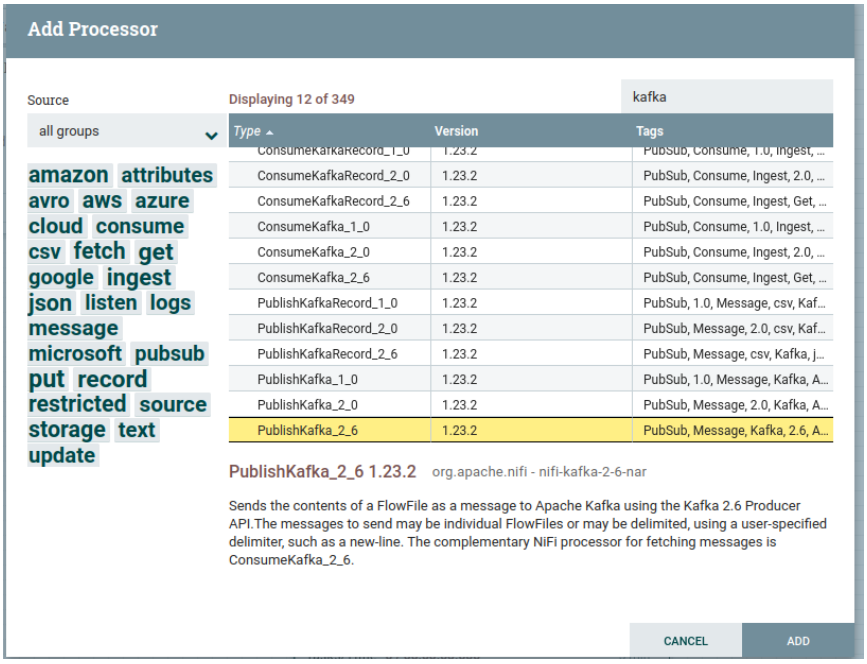
Se relaciona el fetchparquet con el log attribute de manera que registre todos los eventos de fallo, success y retry.



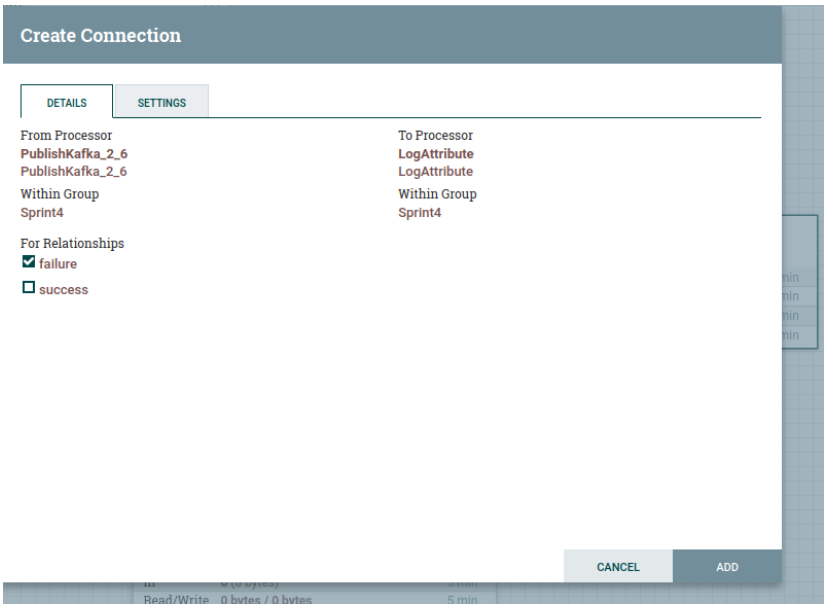
En el log attribute para remover la señal de warning solo debemos terminar la relación de success.



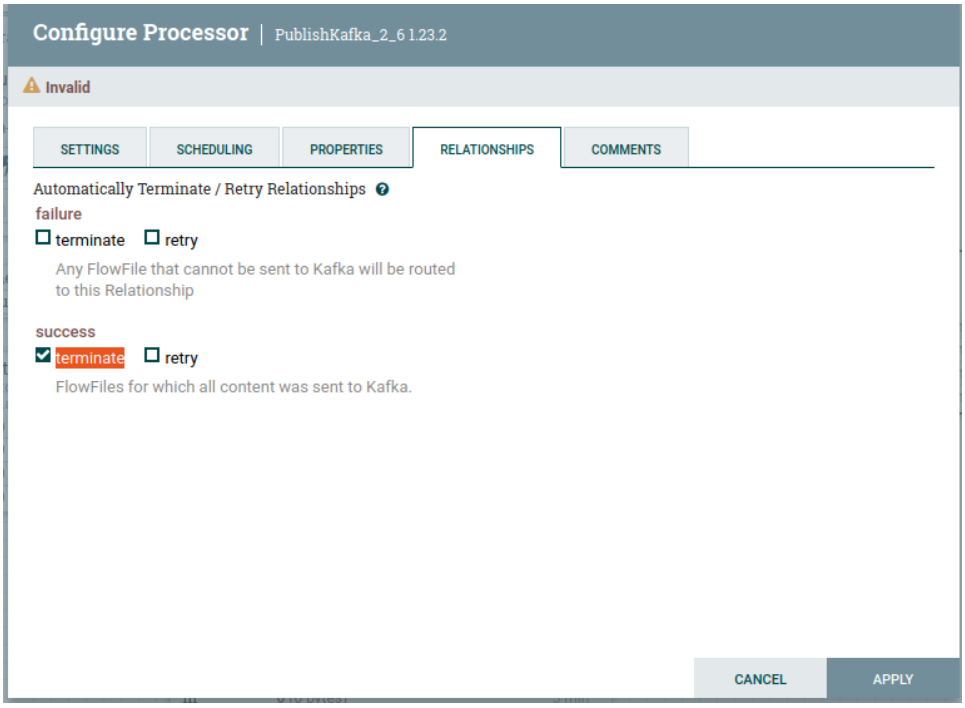
Para volcar el fichero tipo json en un topic de kakfa, usaremos el publish Kafka 2.6 que envía el contenido de un flow file como un mensaje.



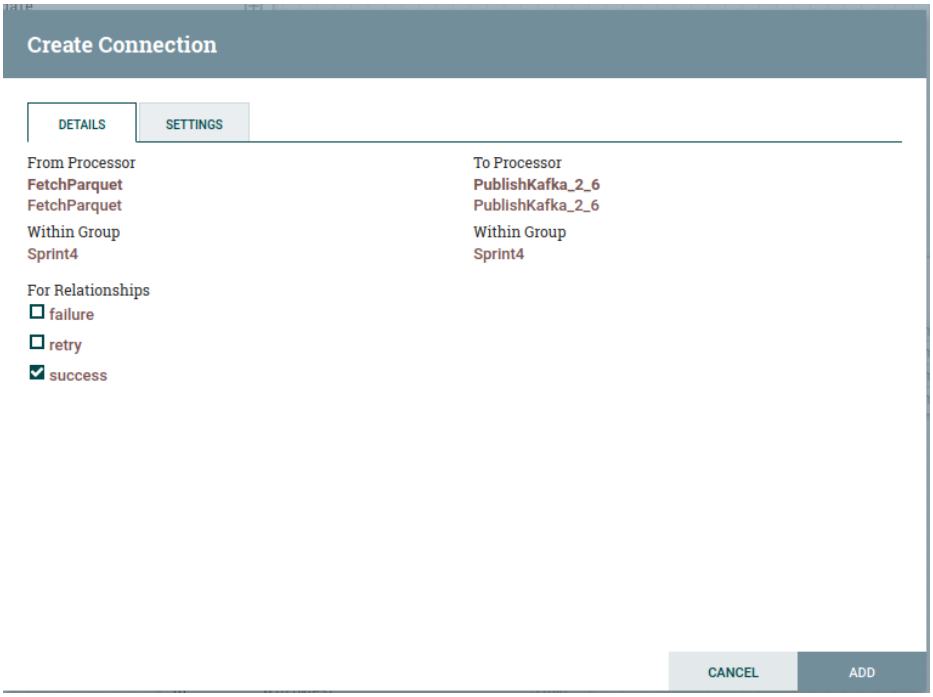
Configuramos la relación de publish Kafka con log attribute cuando ocurra un error.



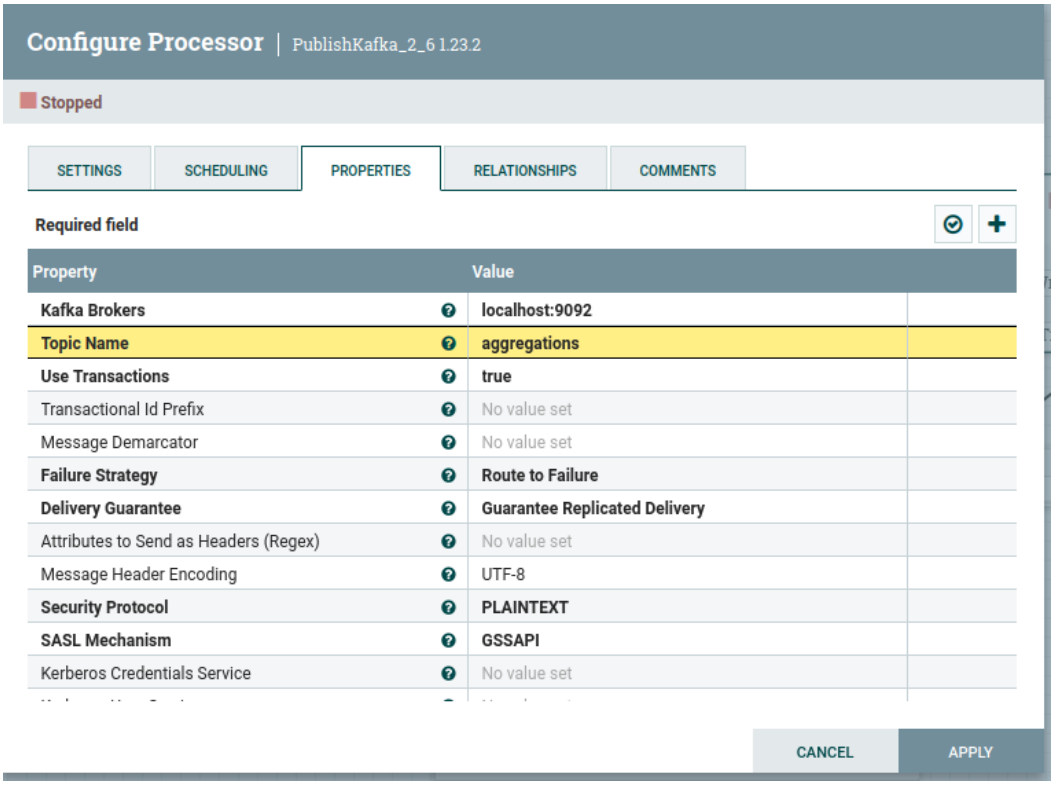
Y la relación de success se termina.



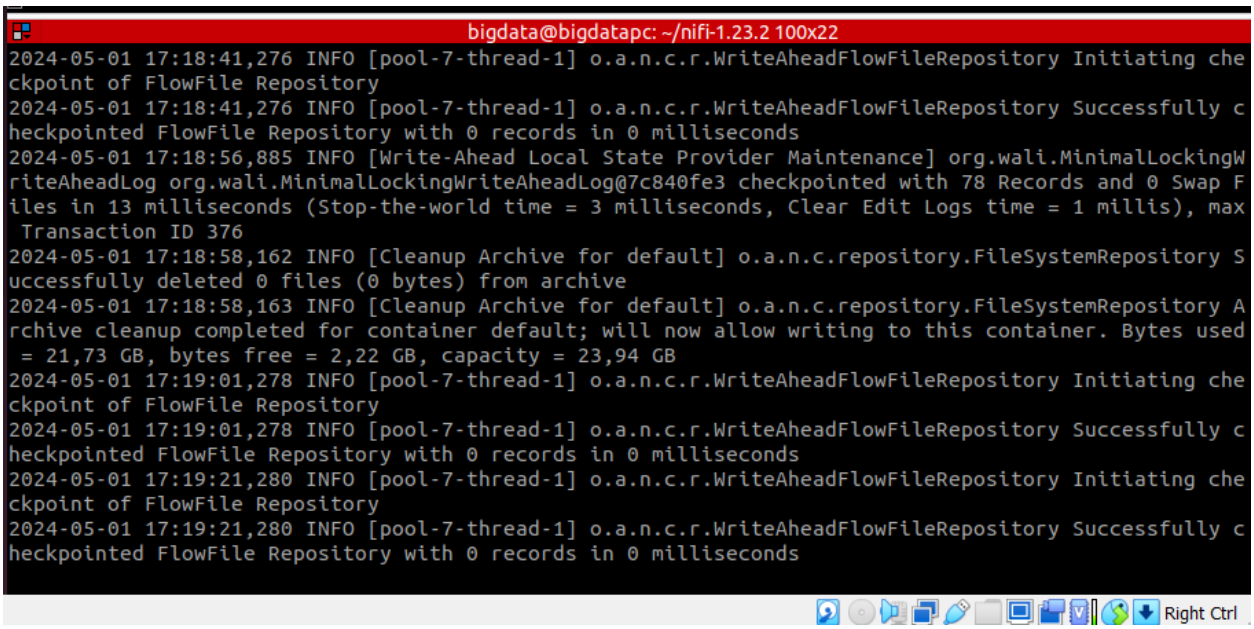
Como este procesador necesita una entrada le indicamos la relación con el fetch parquet y de esta manera cuando sea success este pase los resultados.



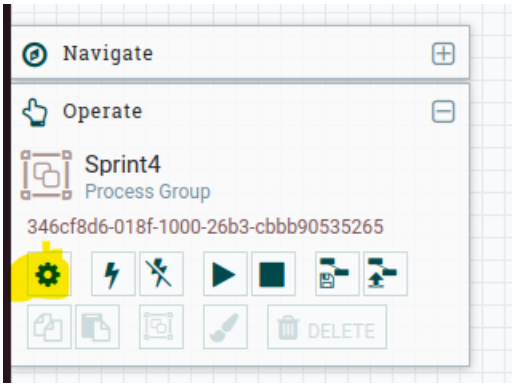
Se configura el topic name con el nombre que se había puesto en el consumidor anteriormente.

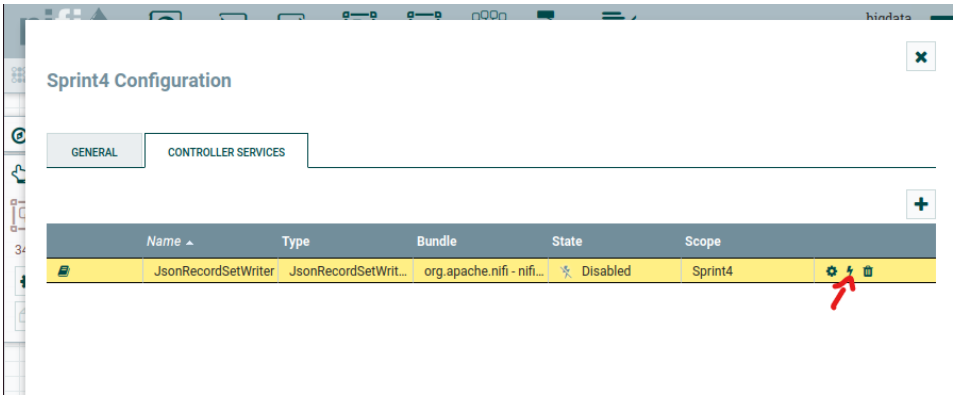


Emitimos los logs de nifi para ver cómo van fluyendo los datos, los atributos y contenidos del flow file.

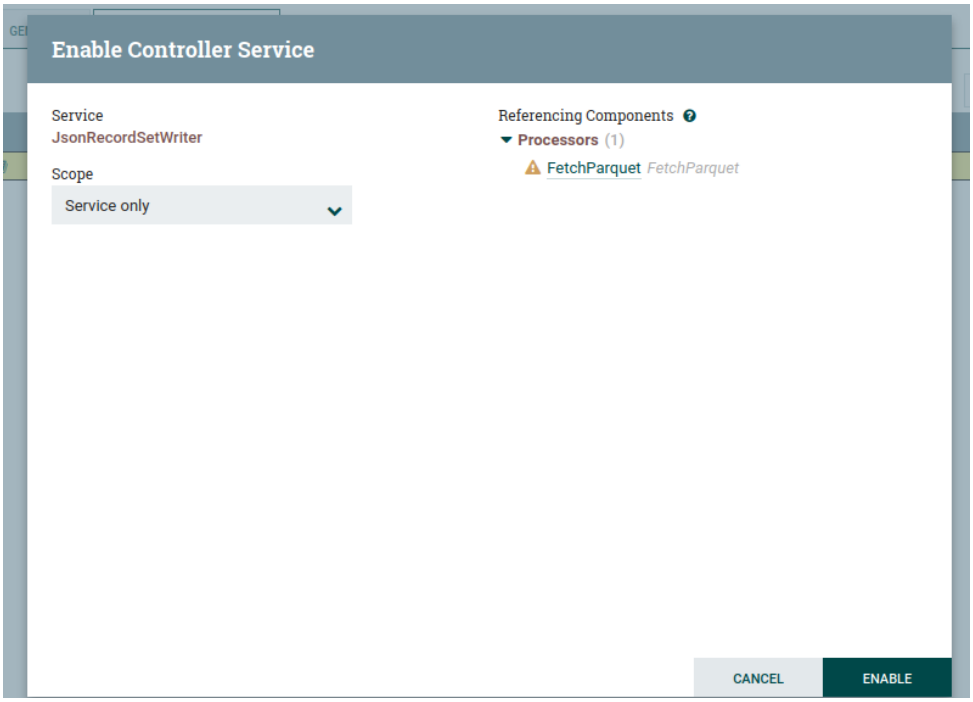


Volvemos a la interfaz y vemos que servicios están sin activar.

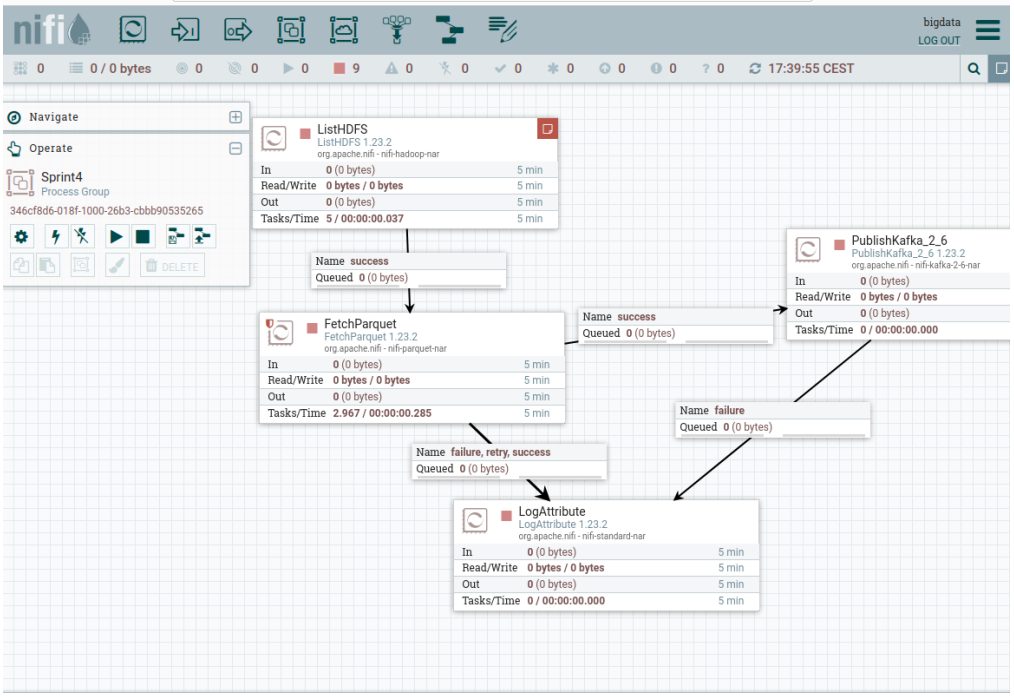




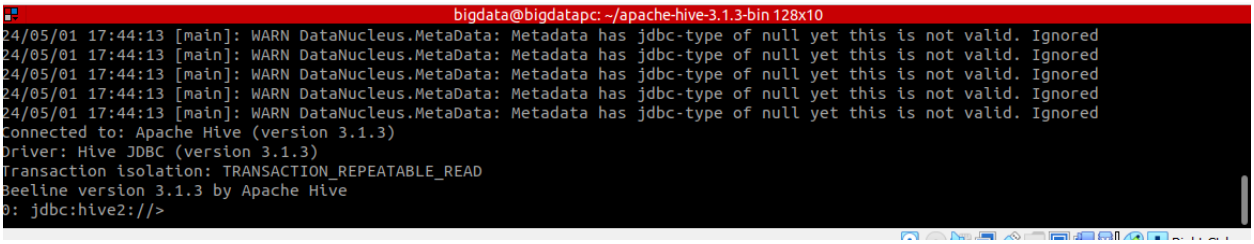
Y lo activamos.



Luego de correrlo, vemos que tenemos un error en el ListHDFS debido a que no encuentra el archivo al que nos estamos refiriendo.



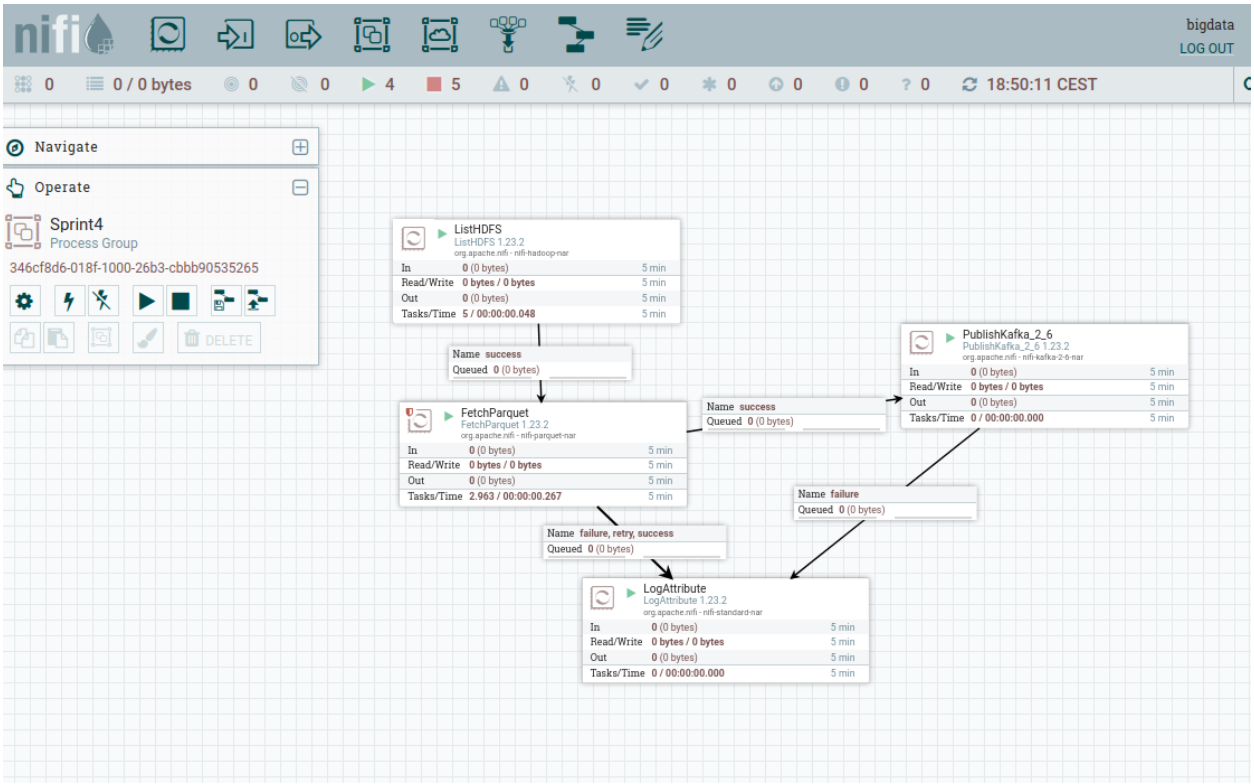
Por lo que vamos a Hive a levnatar una consola.

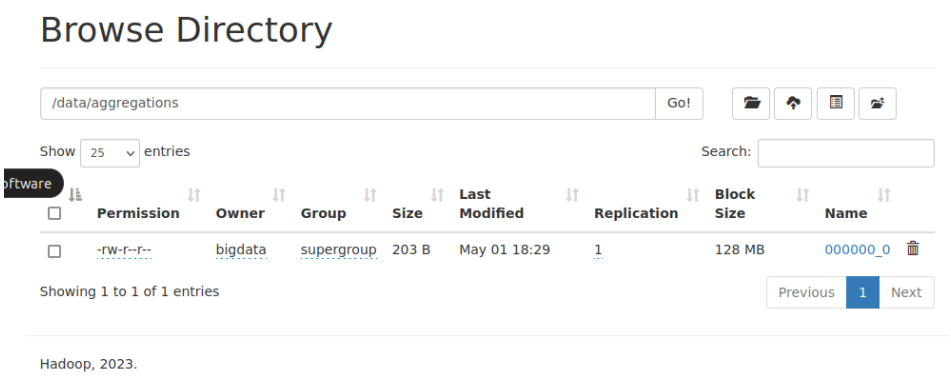


Se generara una tabla que se llame aggregations y que contiene los datos que está esperando la ruta hdfs.

```
CREATE TABLE aggregations

STORED AS parquet

LOCATION '/data/aggregations/'

AS

SELECT

   a.country,

   COUNT(*) AS total_rutas_sin_paradas

FROM

   routes r

JOIN

   aerolineas a ON r.airlineID = a.airlineID

JOIN

   aeropuertos aero ON r.sourceAirportID = aero.airportID

WHERE

   r.stops = 0

   AND a.active = FALSE

   AND aero.altitude > 5000

GROUP BY

   a.country;
```

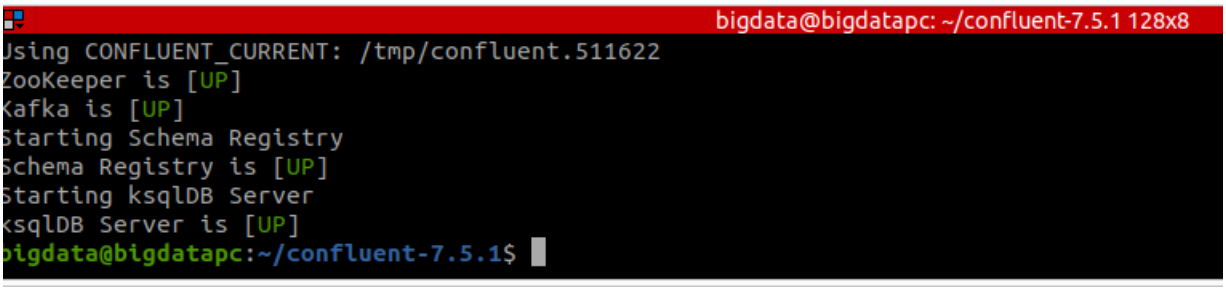
```
bigdata@bigdatapc: ~/apache-hive-3.1.3-bin 128x21
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [?:1.8.0_382]
        at java.lang.Thread.run(Thread.java:750) [?:1.8.0_382]
24/05/01 18:29:27 [pool-21-thread-1]: WARN hadoop.MemoryManager: Total allocation exceeds 50,00% (130.809.856 bytes) of heap mem
ory
Scaling row group sizes to 97,46% for 1 writers
2024-05-01 18:29:28,006 Stage-3 map = 100%,  reduce = 0%
2024-05-01 18:29:29,022 Stage-3 map = 100%,  reduce = 100%
Ended Job = job_local1234549444_0001
24/05/01 18:29:29 [HiveServer2-Background-Pool: Thread-89]: WARN metastore.ObjectStore: datanucleus.autoStartMechanismMode is se
t to unsupported value null . Setting it to value: ignored
Moving data to directory /data/aggregations
24/05/01 18:29:29 [HiveServer2-Background-Pool: Thread-89]: WARN metastore.HiveMetaStore: Location: hdfs://localhost:9000/data/a
ggregations specified for non-external table:aggregations
24/05/01 18:29:29 [HiveServer2-Background-Pool: Thread-89]: WARN metastore.ObjectStore: datanucleus.autoStartMechanismMode is se
t to unsupported value null . Setting it to value: ignored
MapReduce Jobs Launched:
Stage-Stage-3:  HDFS Read: 6711110 HDFS Write: 249 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
No rows affected (12,28 seconds)
0: jdbc:hive2://>
```



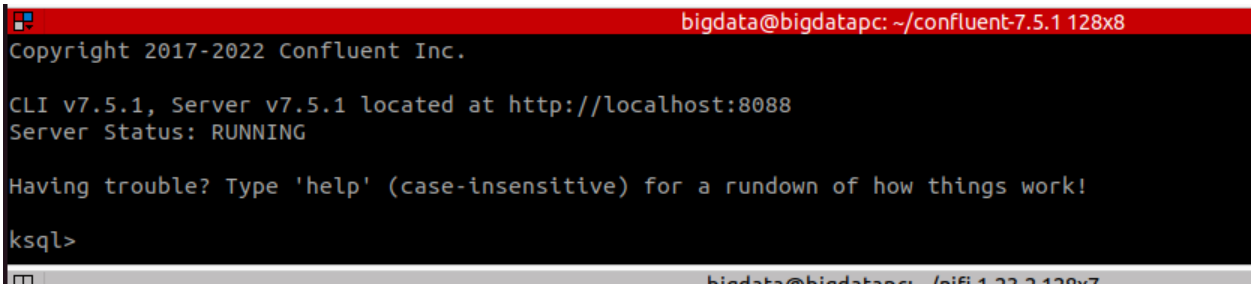


SPRINT 5

Arrancar KSql y todos los servicios requeridos para ello.

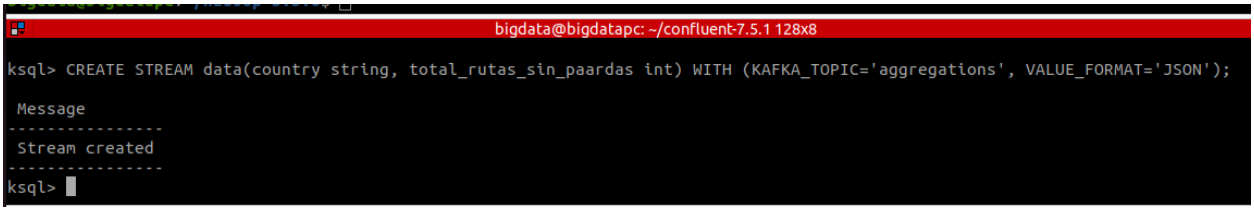


Subimos una consola de ksql

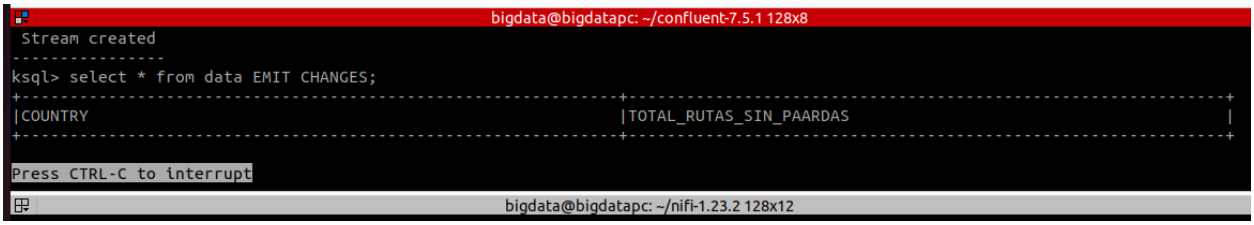


Usamos este commando para crear el stream

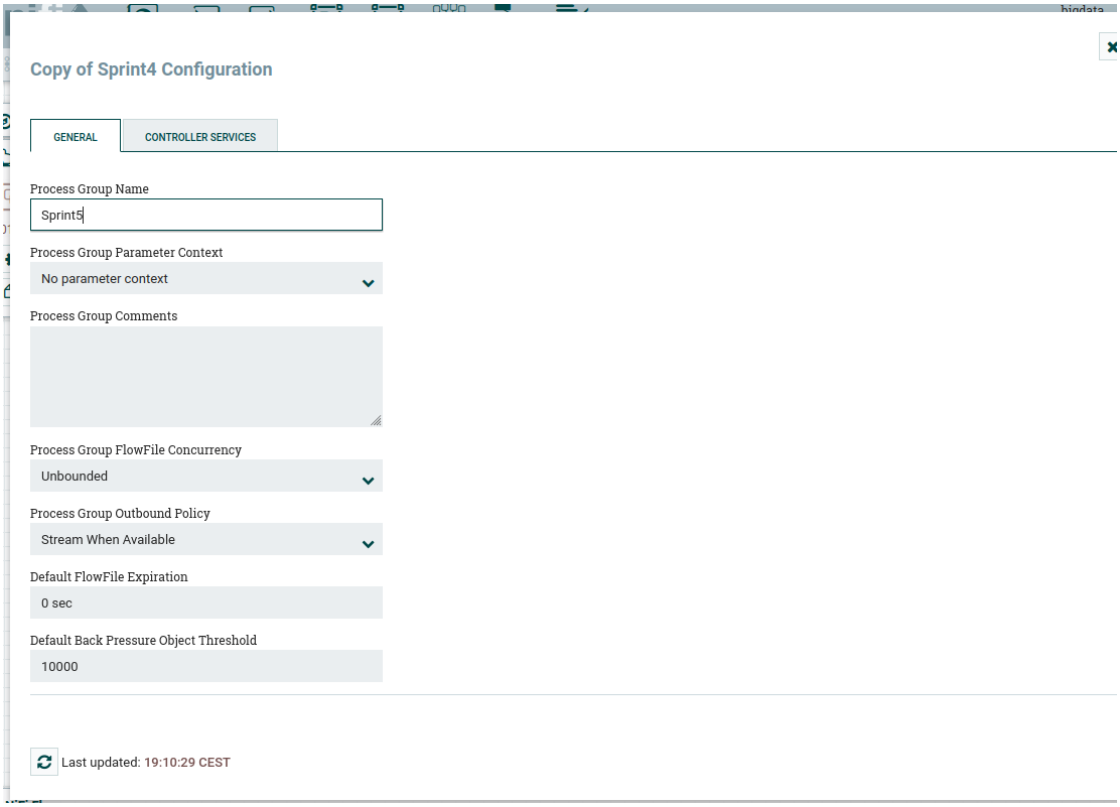
CREATE STREAM data(country string, total_rutas_sin_paardas int) WITH (KAFKA_TOPIC='aggregations', VALUE_FORMAT='JSON');



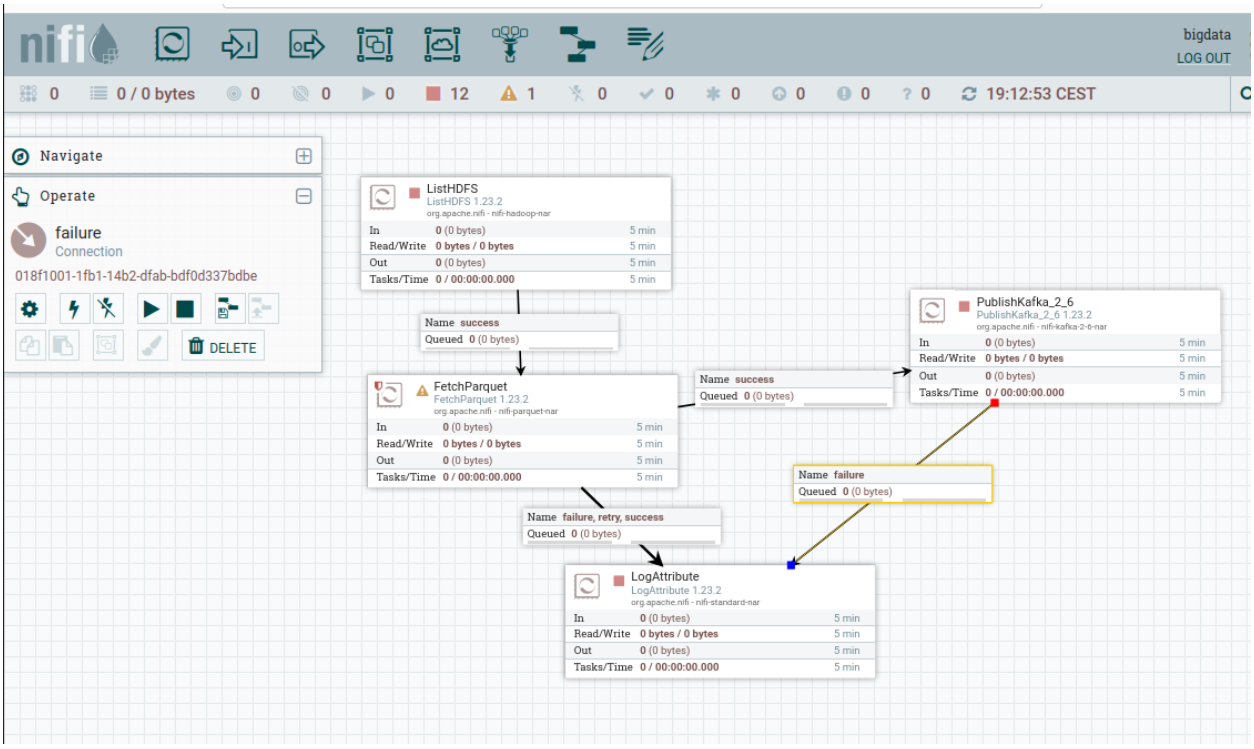
Ejecutamos este query para que sea continua de forma permanente los cambios que lleguen.



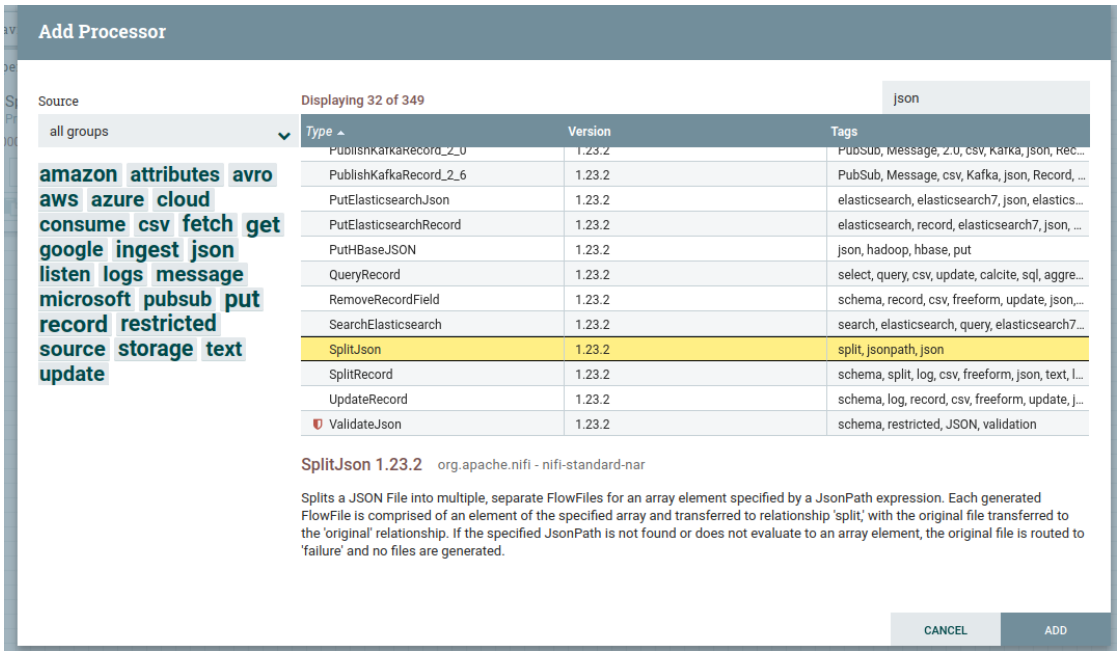
Vamos a NiFi y copiamos y pegamos el process group que teníamos para el 4 y le cambiamos el nombre



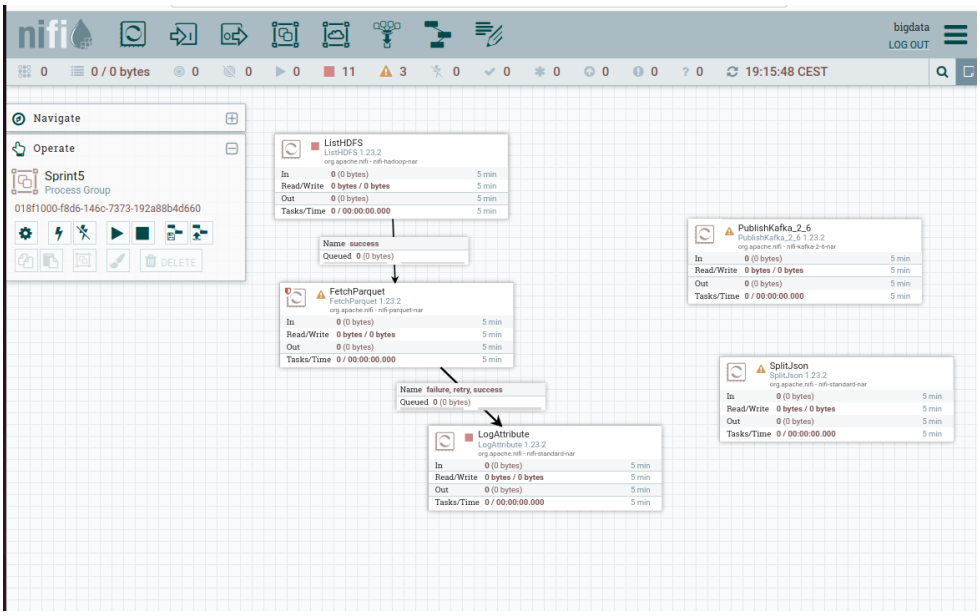
Se elimina la relación entre log attribute y publish Kafka para poder añadir un nuevo procesador.



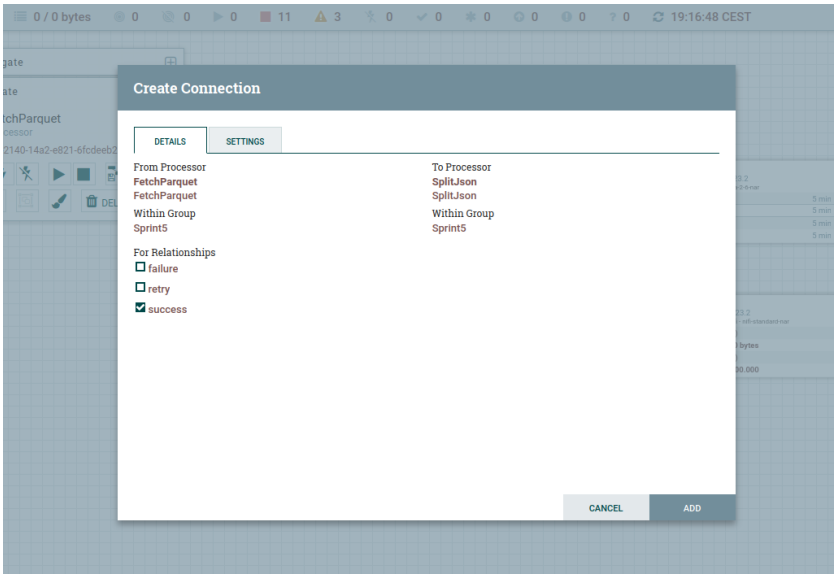
Usamos este llamado Split json para separar un flow file en una raíz de elementos.



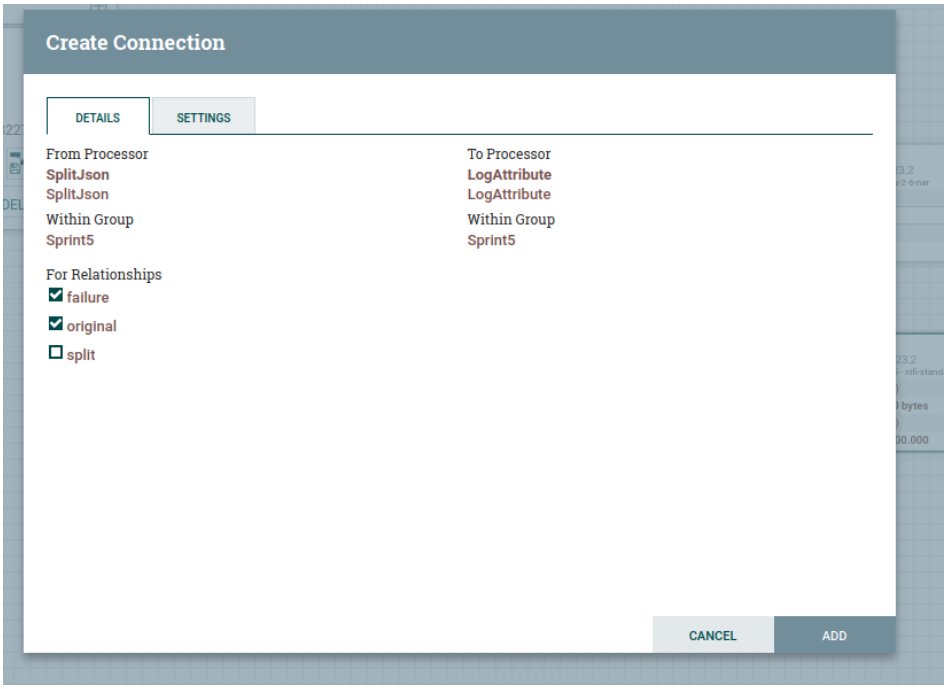
Cambiamos la relación de fecth parquet para que no llegue directamente a publish Kafka si no que pase por Split json y tengamos diferentes elementos de salida.



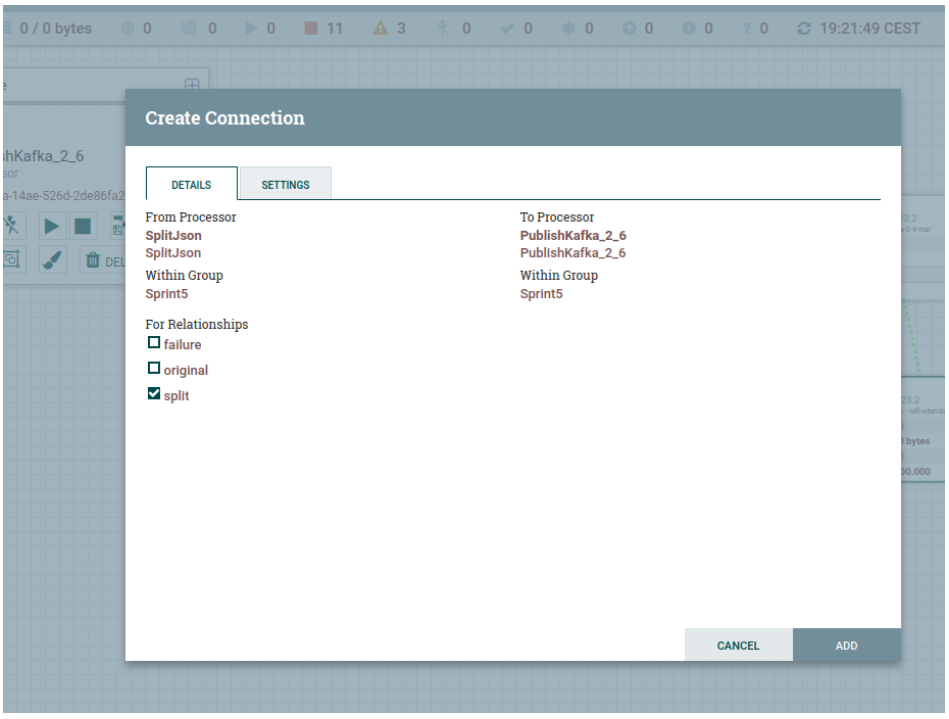
Cuando sea de éxito, lo relacionamos.



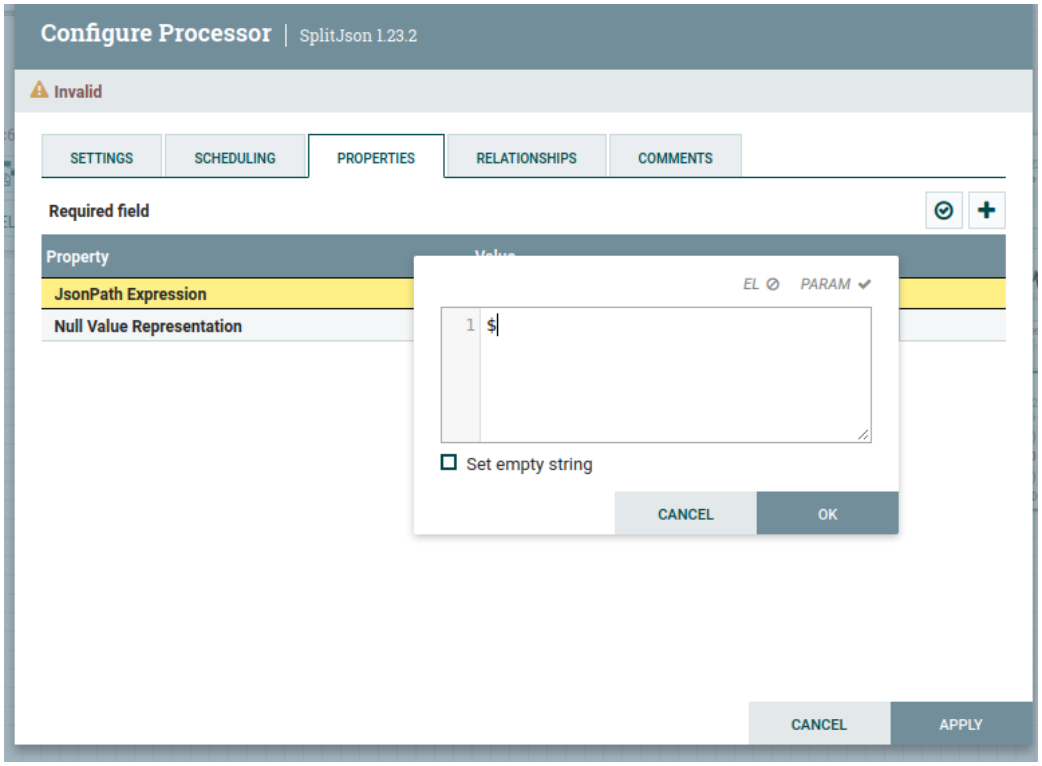
Conectamos el Split json con log attribute y de esta manera cuando falle o tenga el original lo registrara.



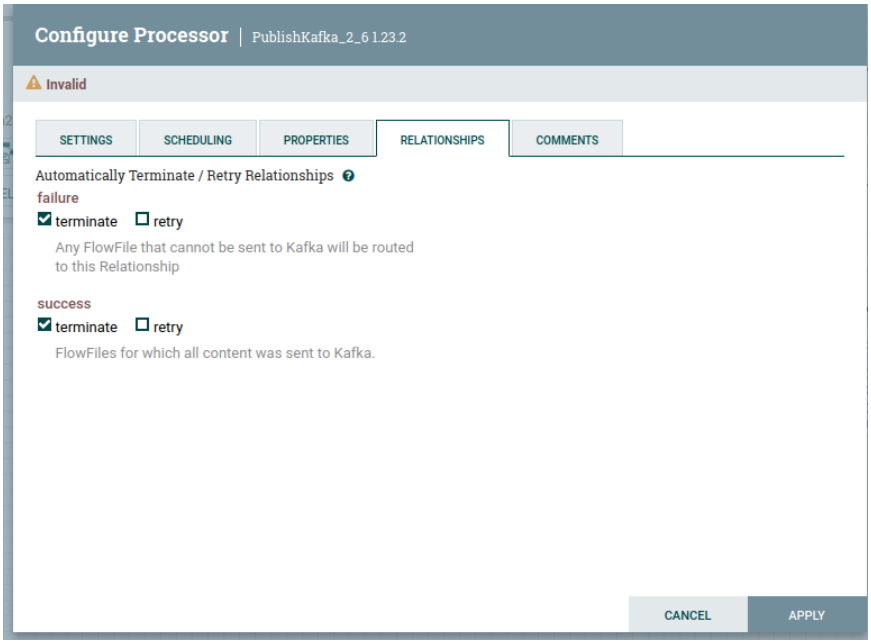
Y cuando haya hecho bien el Split, entonces lo pasa al procesador de Kafka



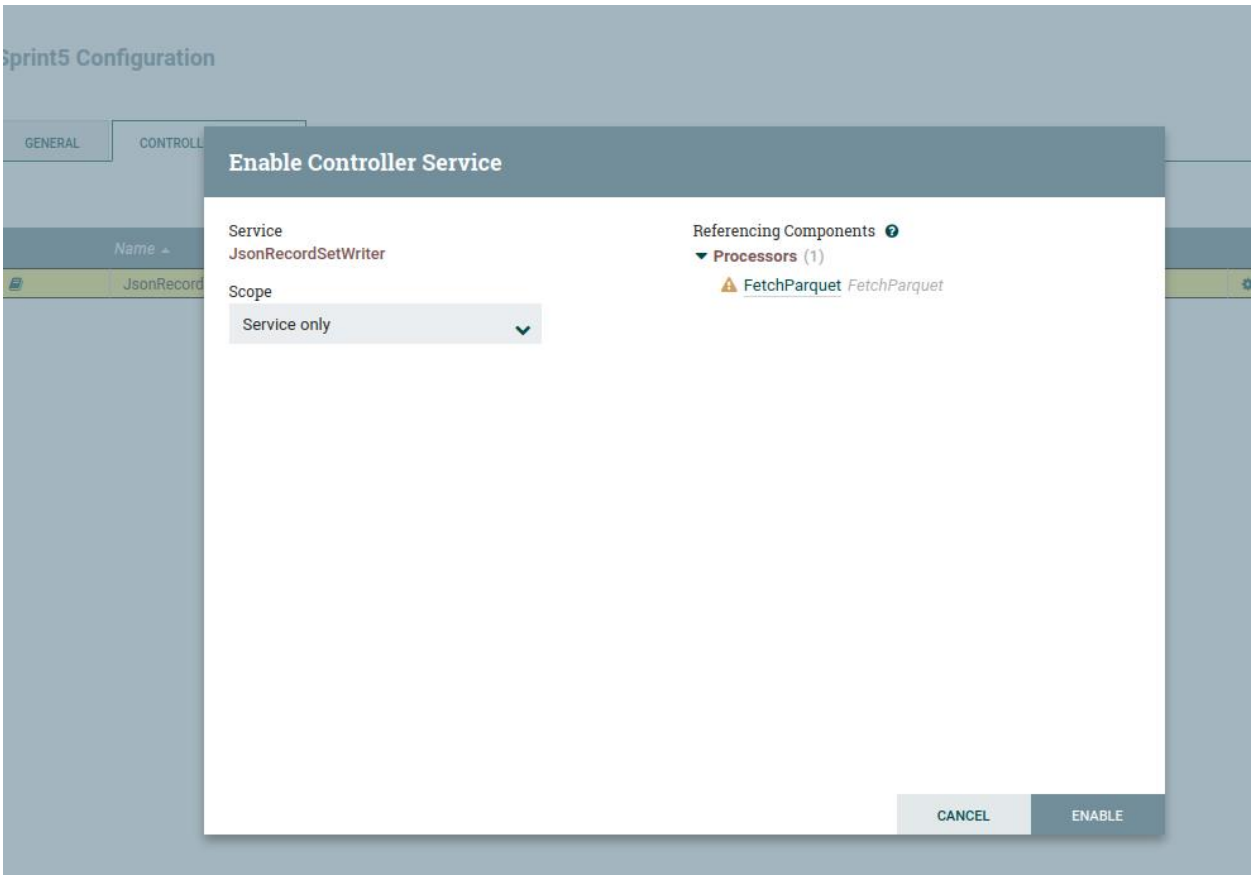
Configuramos el json file de manera que cada evento se convertirá en un flow file diferente al usar el signo de $.

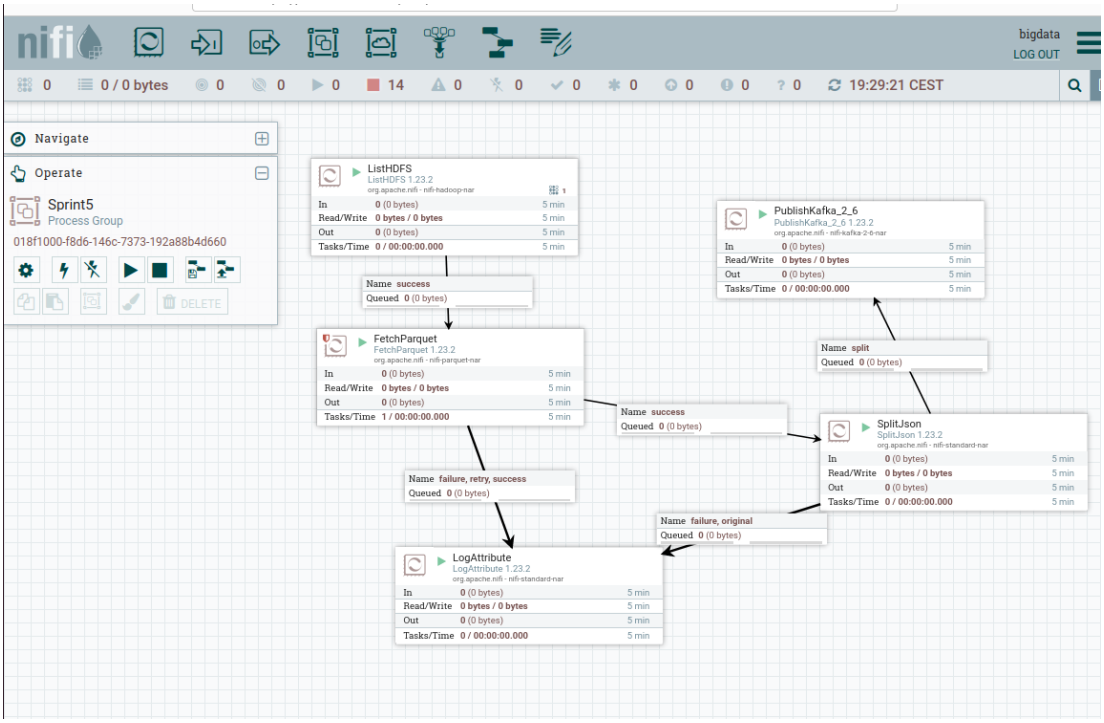


Configuramos el procesador de Kafka de manera que cuando haya un fallo este sea terminado.



Y activamos el controller para el fetch parquet


Sprint5 Configuration
GENERAL
CONTROLL
Name
JsonRecord
Enable Controller Service
Service
JsonRecordSetWriter
Scope
Service only
Referencing Components
Processors (1)
FetchParquet FetchParquet
CANCEL
ENABLE


nifi
bigdata
LOG OUT
19:29:21 CEST
Navigate
Operate
Sprint5
Process Group
018f1000-f8d6-146c-7373-192a88b4d660
DELETE
ListHDFS
ListHDFS 1.23.2
org.apache.nifi - nifi-hadoop-nar
PublishKafka_2_6
PublishKafka_2_6 1.23.2
org.apache.nifi - nifi-kafka-2-6-nar
FetchParquet
FetchParquet 1.23.2
org.apache.nifi - nifi-parquet-nar
SplitJson
SplitJson 1.23.2
org.apache.nifi - nifi-standard-nar
LogAttribute
LogAttribute 1.23.2
org.apache.nifi - nifi-standard-nar
Name success
Queued 0 (0 bytes)
Name split
Queued 0 (0 bytes)
Name failure, retry, success
Queued 0 (0 bytes)
Name failure, original
Queued 0 (0 bytes)
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 0 / 00:00:00.000 5 min