Fuentes de datos y aprovisionamiento

Práctica 2 - Apache Nifi

José Manuel Bustos Muñoz

Índice

1. Flujo 1

Desde el directorio de “nifi”, vamos a “/bin” y con el comando “./nifi.sh start” arrancamos Apache Nifi.

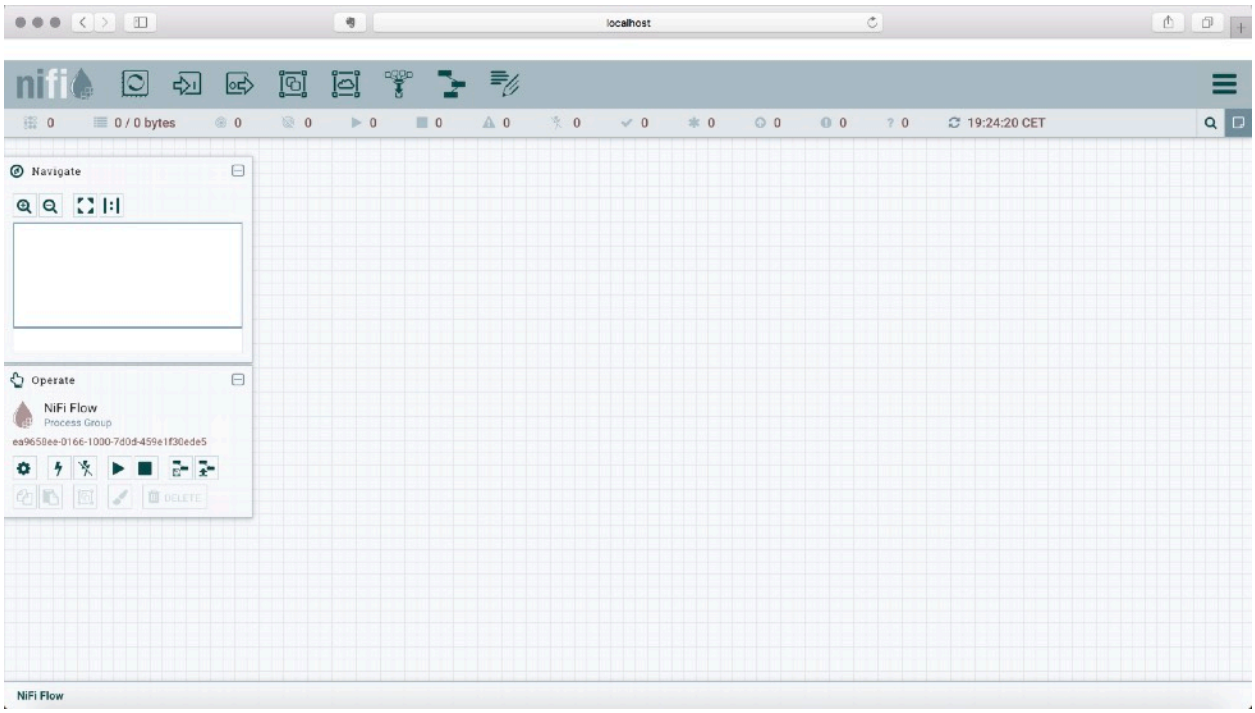
```
MBP-de-Jose:nifi-1.8.0 josemanuel$ ls
LICENSE                 content_repository      logs
NOTICE                  database_repository     provenance_repository
README                  docs                    run
bin                     flowfile_repository     state
conf                    lib                     work
MBP-de-Jose:nifi-1.8.0 josemanuel$ cd bin
MBP-de-Jose:bin josemanuel$ ls
dump-nifi.bat   nifi-env.sh     run-nifi.bat
nifi-env.bat    nifi.sh         status-nifi.bat
MBP-de-Jose:bin josemanuel$ ./nifi.sh start

Java home: /Library/Java/JavaVirtualMachines/jdk1.8.0_162.jdk/Contents/Home/
NiFi home: /Users/josemanuel/server/nifi-1.8.0

Bootstrap Config File: /Users/josemanuel/server/nifi-1.8.0/conf/bootstrap.conf


MBP-de-Jose:bin josemanuel$
```

Una vez arrancamos nifi, accedemos a la aplicación desde “http://localhost:8080/nifi" desde el navegador web.



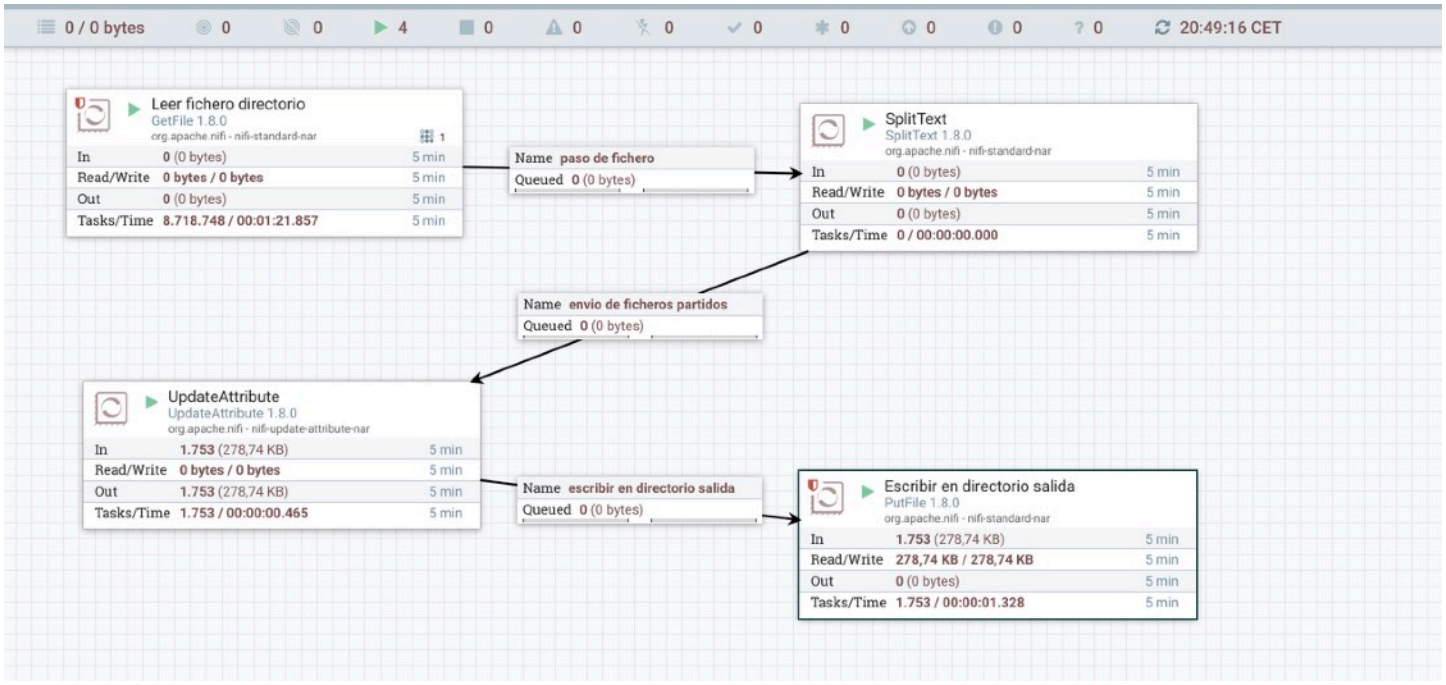
El fichero para trabajar en la práctica tiene el siguiente aspecto:

UAM_Clientes

ERP	J	JACKSON	8388 SOUTH CALIFORNIA ST.	TUCSON	AZ	85708	267-3352		ALLENTON	MI	48002	810	710-0470
ERP	FRANK	DIETSCH	5064 E METAIRIE AVE.	BRANDSVILLA	MO	65687	252-5592	1176 E THAYER ST.	COLUMBIA	MO	65215	557	291-9571
ERP	MARLENE	GALARZA	9314 E PERIANDER ST.	NEWTON	IA	50208	790-0416	803 EAST HENDERSON ST.	SUNBURG	MN	56289	320	471-7635
ERP	CHRIS	LINDERSMIT	2406 NORTH 8 TH AVENUE	MONT ALTO	PA	17237	335-8495	2733 SOUTHWEST EDMOND ST.	QUICKSBURG	VA	22847	540	217-2594

La primera columna “ERP” es el atributo que servirá de discriminado, que representa la fuente origen de ese registro.

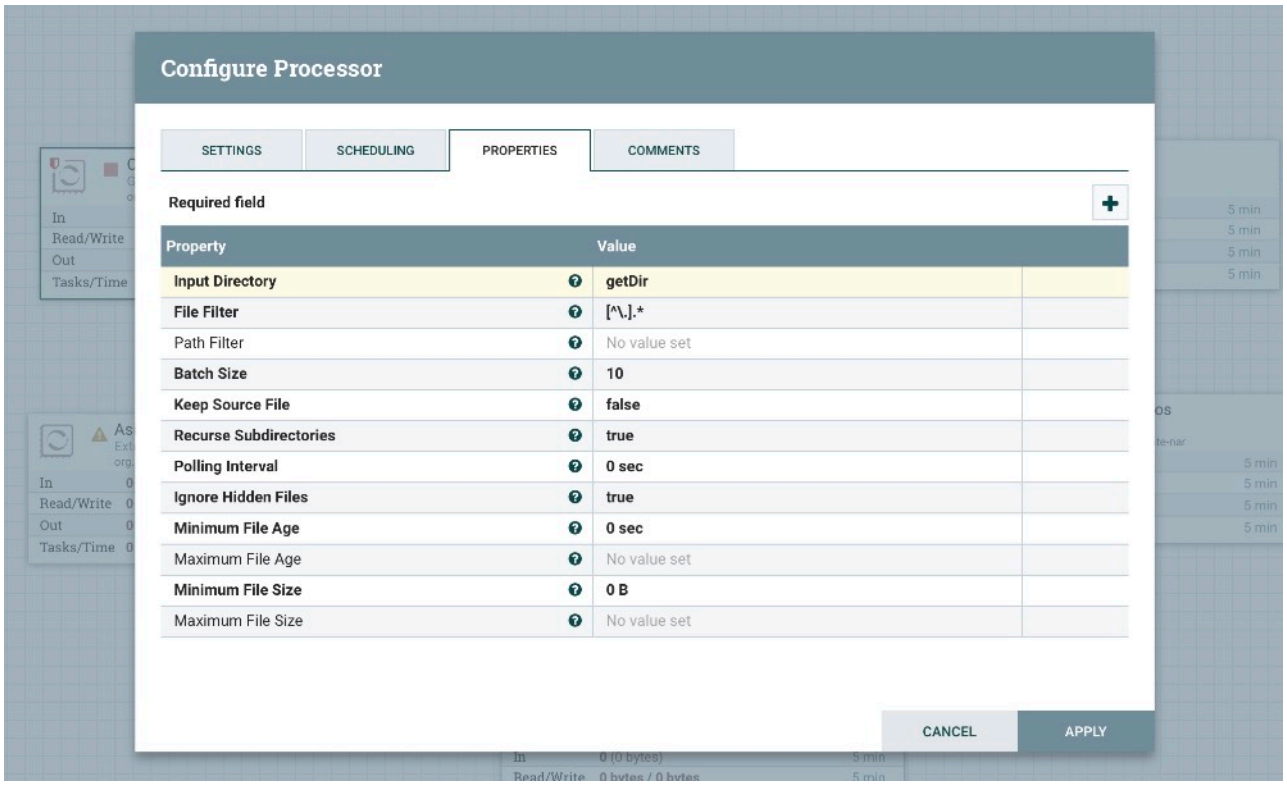
Una vez en Nifi presentamos el primer flujo, en el cual intervienen los siguientes procesadores: GetFile, SplitText, UpdateAttribute, PutFile.
Este flujo cogerá del directorio que se configure como entrada los ficheros que en él se dejen, lo recorrerá y partirá por cada registro del fichero, y creará un fichero por cada uno de esos registros dejando los ficheros resultantes en el directorio que se configure como directorio de salida.
En la imagen se puede ver como con el flujo ejecutado al final en el "PutFile" se han realizado 1753 tareas que serían los ficheros generados.



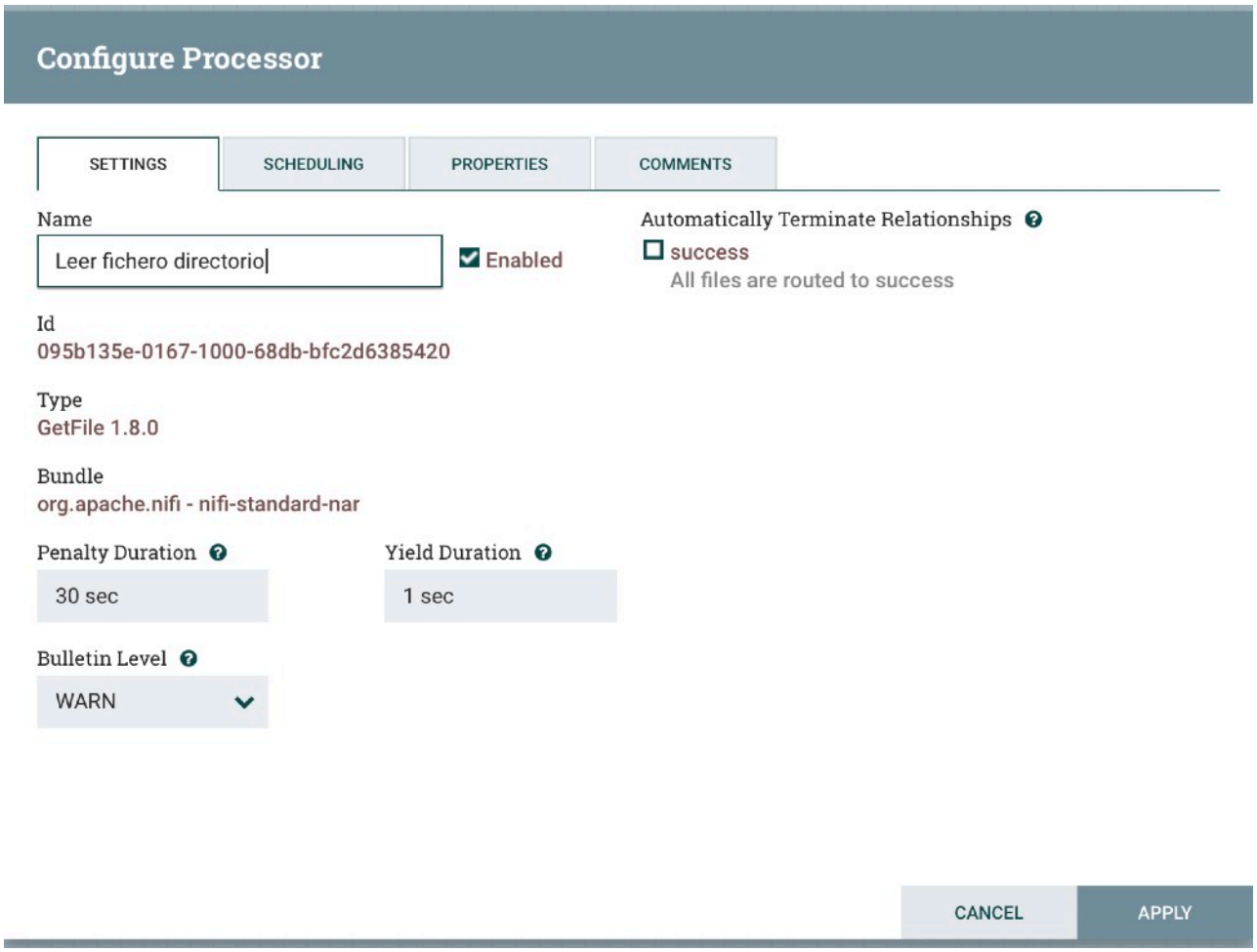
Colocamos en el directorio de entrada "../nifi/getDir" el fichero "UAM_Clientes.csv" para hacer la prueba.

```
MBP-de-Jose:server josemanuel$ cd nifi-1.8.0/
MBP-de-Jose:nifi-1.8.0 josemanuel$ ls
LICENSE                 database_repository     provenance_repository
NOTICE                  docs                    putDir
README                  flowfile_repository     run
bin                     getDir                  state
conf                    lib                     work
content_repository      logs
MBP-de-Jose:nifi-1.8.0 josemanuel$ cd getDir
MBP-de-Jose:getDir josemanuel$ ls
UAM_Clientes.csv
MBP-de-Jose:getDir josemanuel$
```

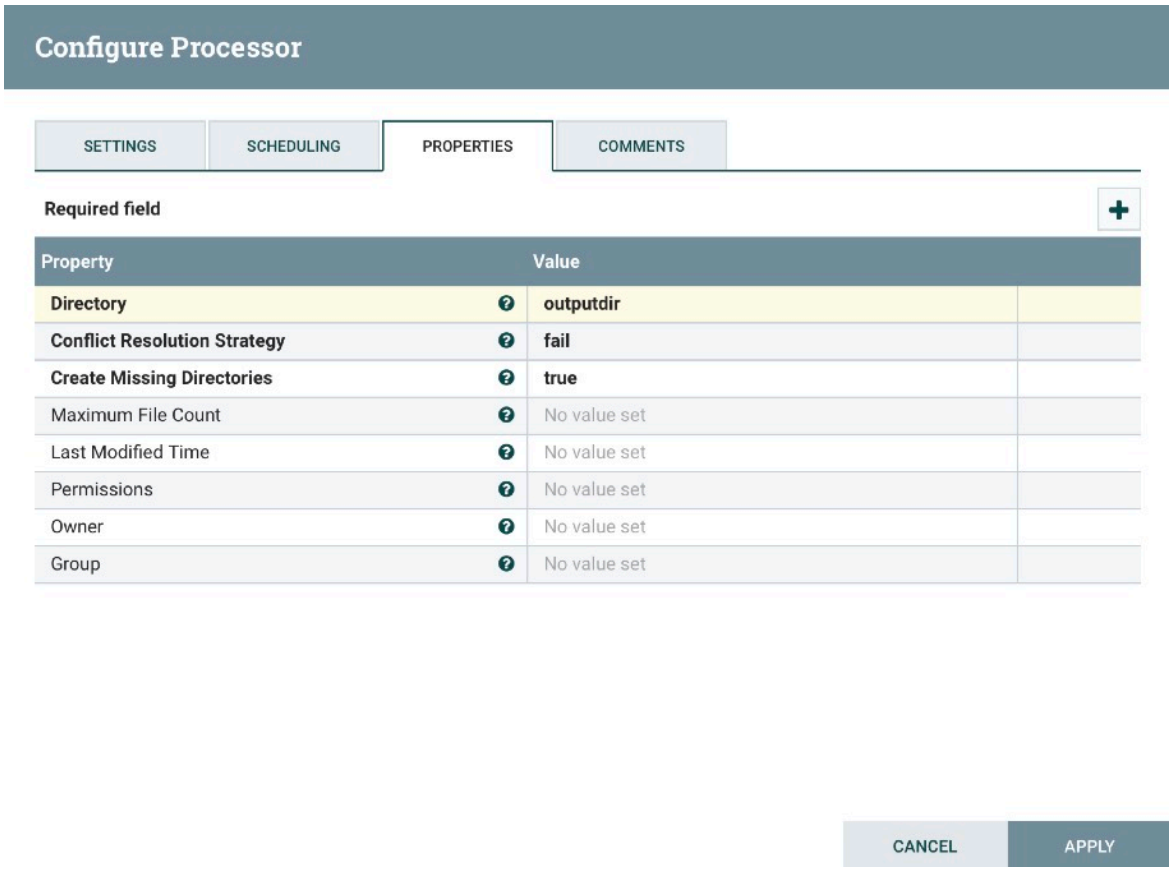
En las propiedades del procesador GetFile se configura el directorio de entrada:



También se le da nombre, y algunas otras propiedades:



En el procesador final “PutFile” se configura la propiedad del directorio de salida donde se dejarán los ficheros resultantes del flujo:



Una vez ejecutamos el flujo y llega a su final, si listamos el directorio de salida se ve como están todos los ficheros resultantes. En este caso serían 1753 ficheros, que corresponde con el número de registros del fichero original.

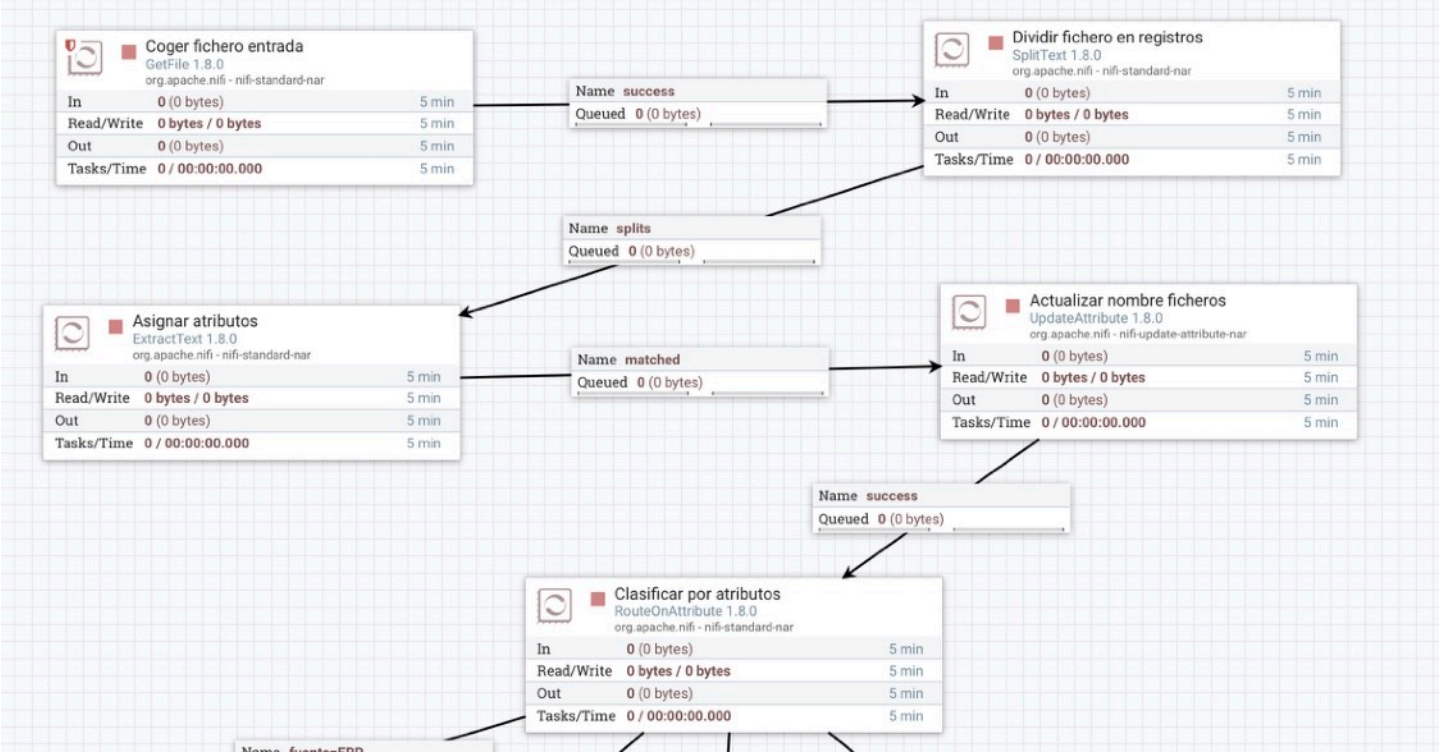
```
MBP-de-Jose:nifi-1.8.0 josemanuel$ cd putDir
MBP-de-Jose:putDir josemanuel$ ls | wc -l
    1753
MBP-de-Jose:putDir josemanuel$
```

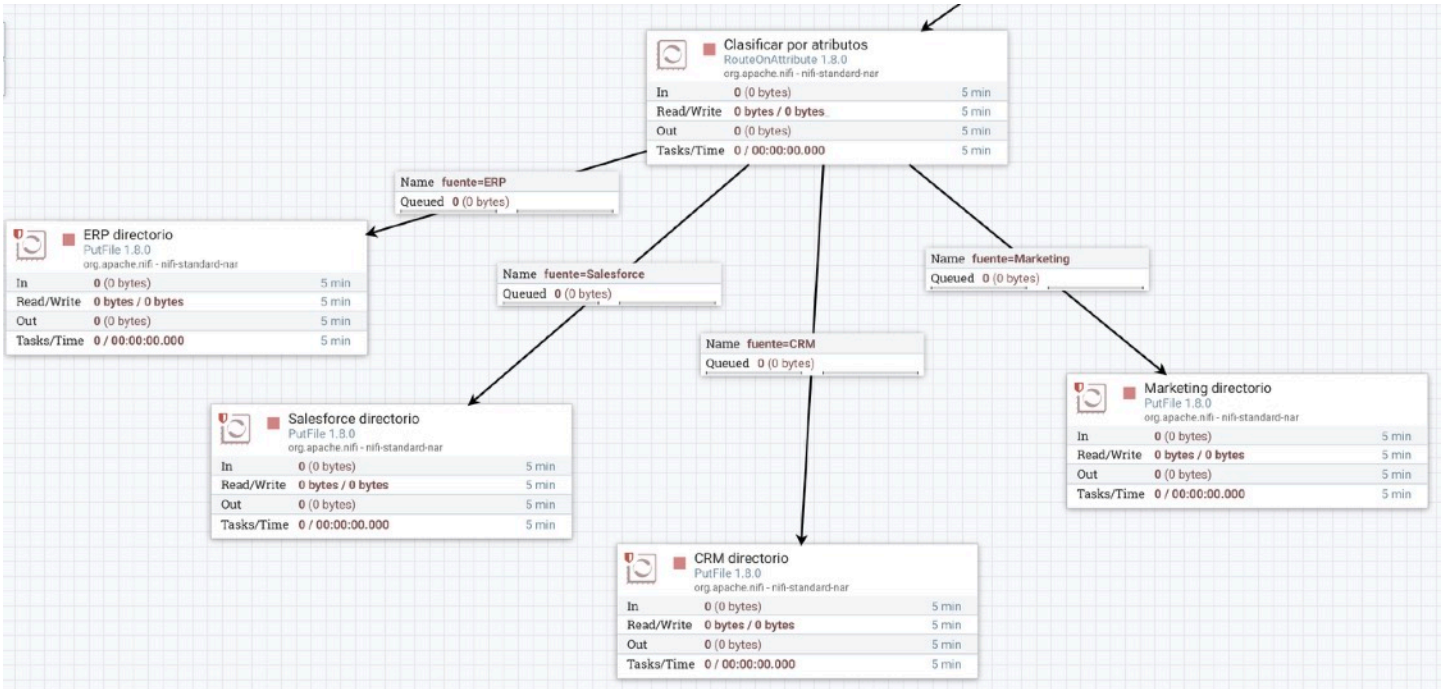
```
MBP-de-Jose:nifi-1.8.0 josemanuel$ cd getDir
MBP-de-Jose:getDir josemanuel$ ls
UAM_Clientes.csv
MBP-de-Jose:getDir josemanuel$ ls
MBP-de-Jose:getDir josemanuel$ cd ..
MBP-de-Jose:nifi-1.8.0 josemanuel$ ls putDir
00079467-5fd7-481d-ac3e-6b8678cd4c78-UAM_Clientes.csv
005424af-3fd1-4c29-b335-a846fb2da810-UAM_Clientes.csv
0061caed-882b-46ca-a51f-bee439694dc1-UAM_Clientes.csv
00a06146-70bb-4ecb-b570-e5797461a4c3-UAM_Clientes.csv
```

Vemos como ejemplo uno de los ficheros resultantes, viendo que efectivamente sólo contiene uno de los registros del fichero original:

2. Flujo 2

En el flujo 2 vamos a partir del mismo fichero, se va a dividir por registro de nuevo, pero la diferencia es que según el valor de la fuente de origen (primer atributo del fichero original) se guardará en un directorio de salida distinto. Para lograr esto se añaden al flujo los procesadores: ExtractText y RouteOnAttribute. Con ellos se creará una variable que relacione el registro con su fuente, y posteriormente se discriminará por este valor para ir a un directorio de salida u otro. Con el procesador "RouteOnAttribute" discriminamos y sacamos 4 flujos distintos hacia 4 procesadores PutFile donde cada uno guarda lo que le llegue en un directorio de salida distinto.

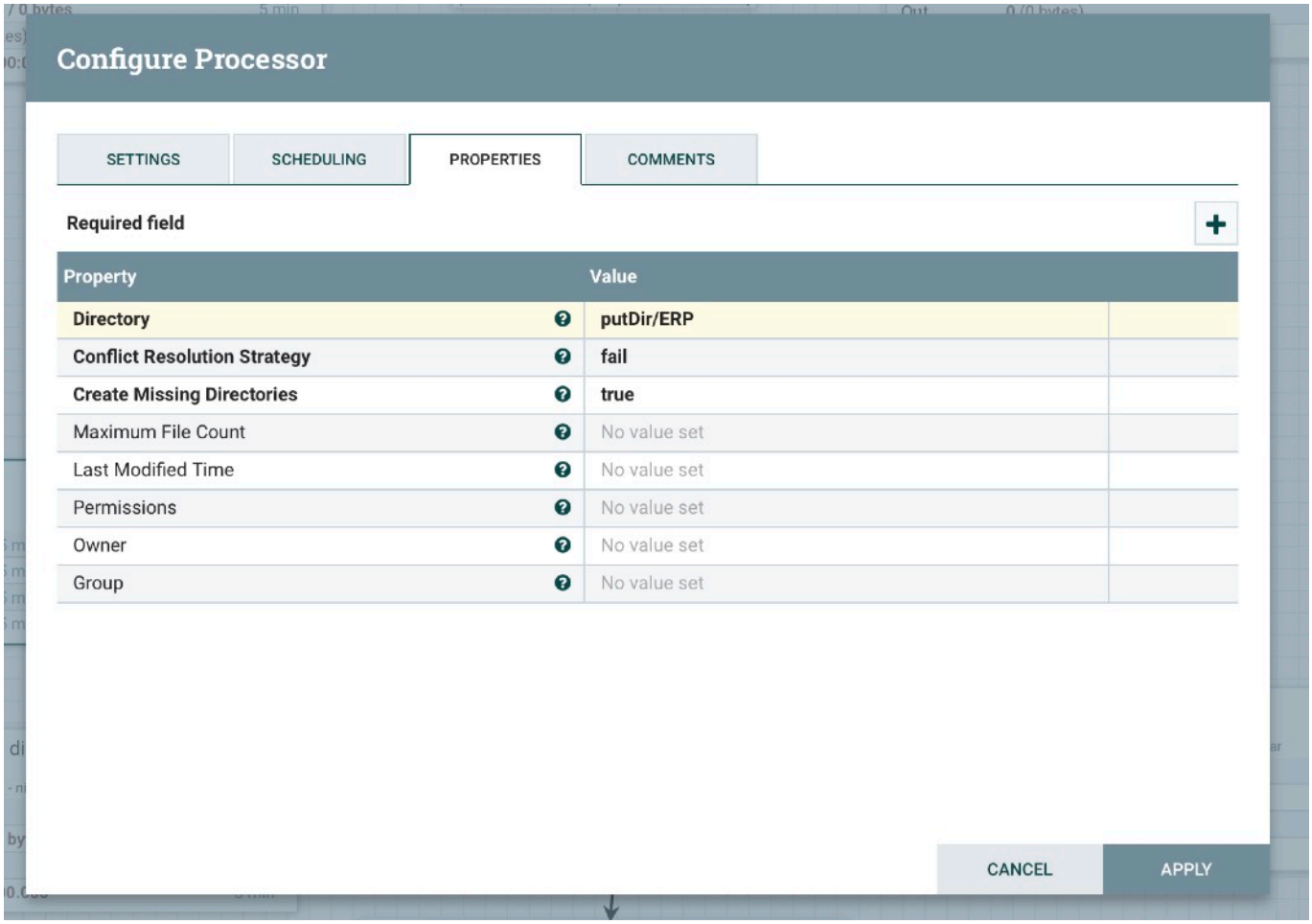




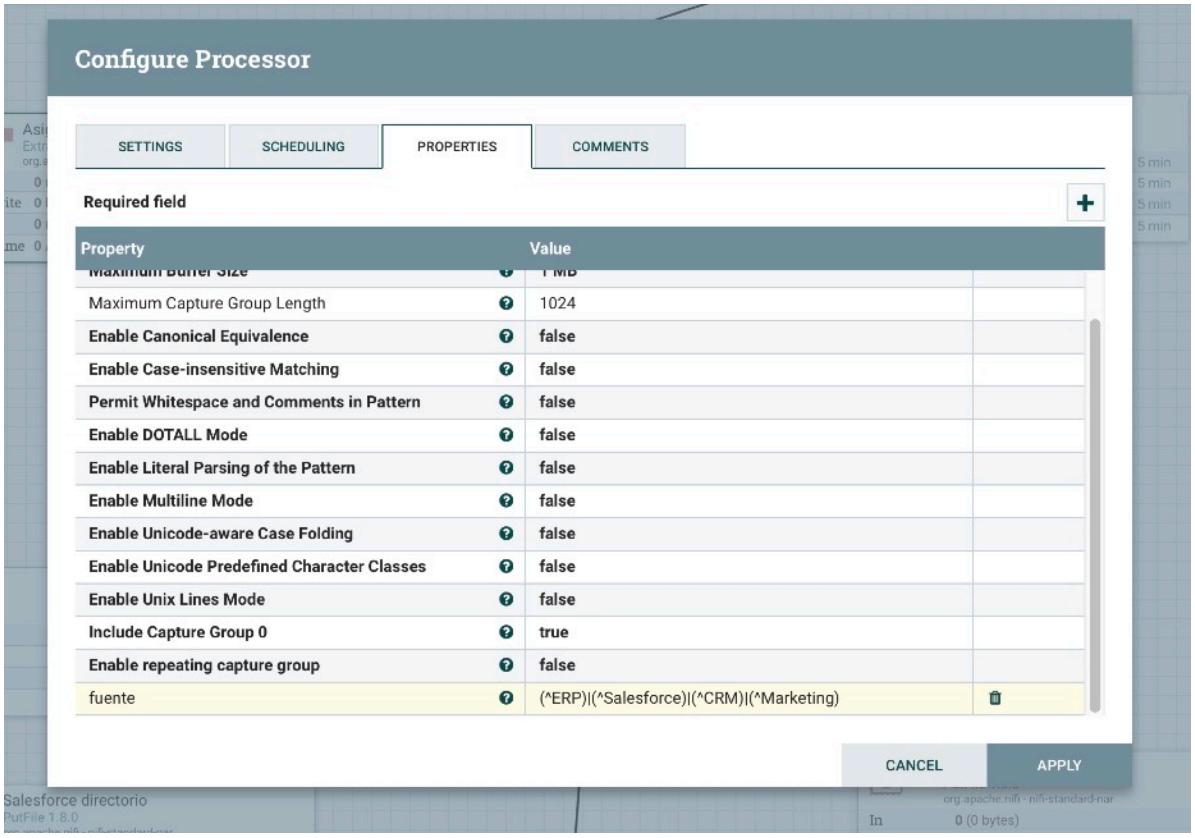
Creamos los 4 directorios en la carpeta “putDir” de salida de Nifi. Serían 4 directorios porque los valores de la fuente del fichero son: ERP, Salesforce, CRM y Marketing.

```
content_repository      logs
MBP-de-Jose:nifi-1.8.0 josemanuel$ cd putDir
MBP-de-Jose:putDir josemanuel$ ls
CRM             ERP             Marketing       Salesforce
MBP-de-Jose:putDir josemanuel$
```

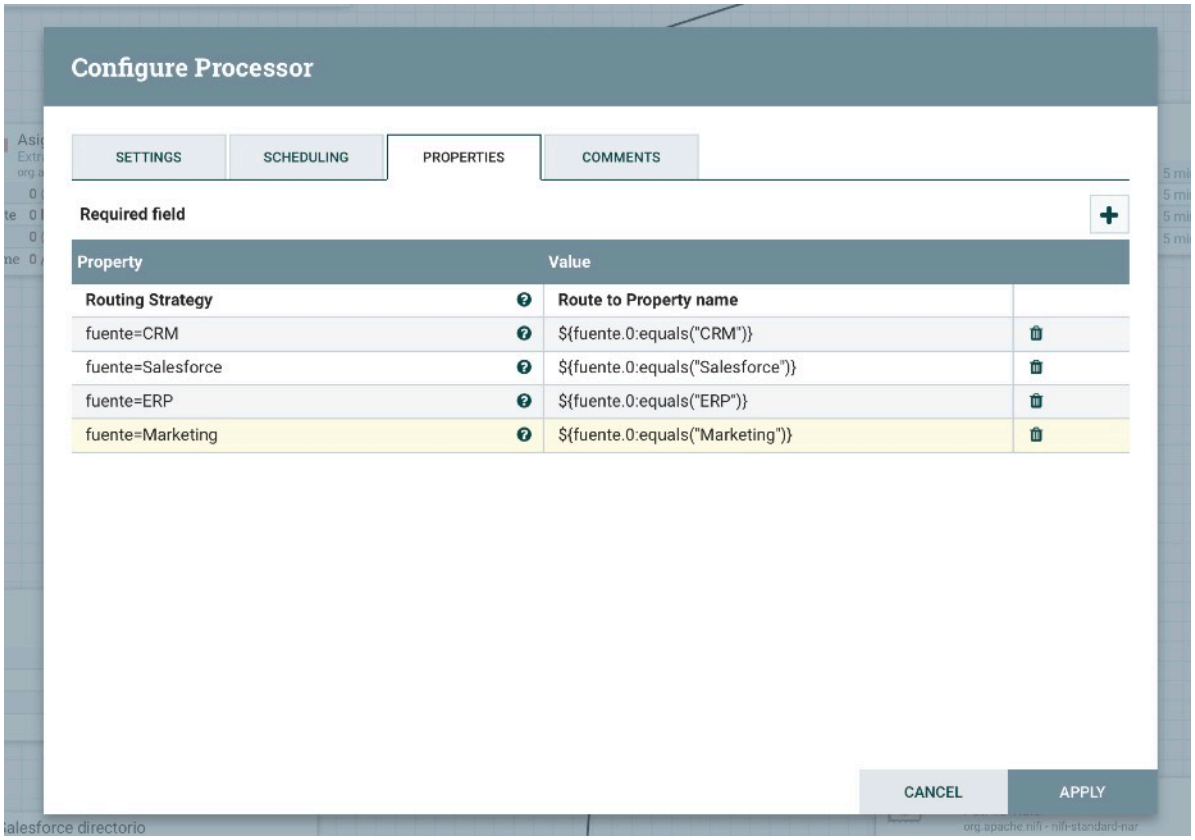
Cada uno de los putFile se configura con uno de estos directorios de salida:



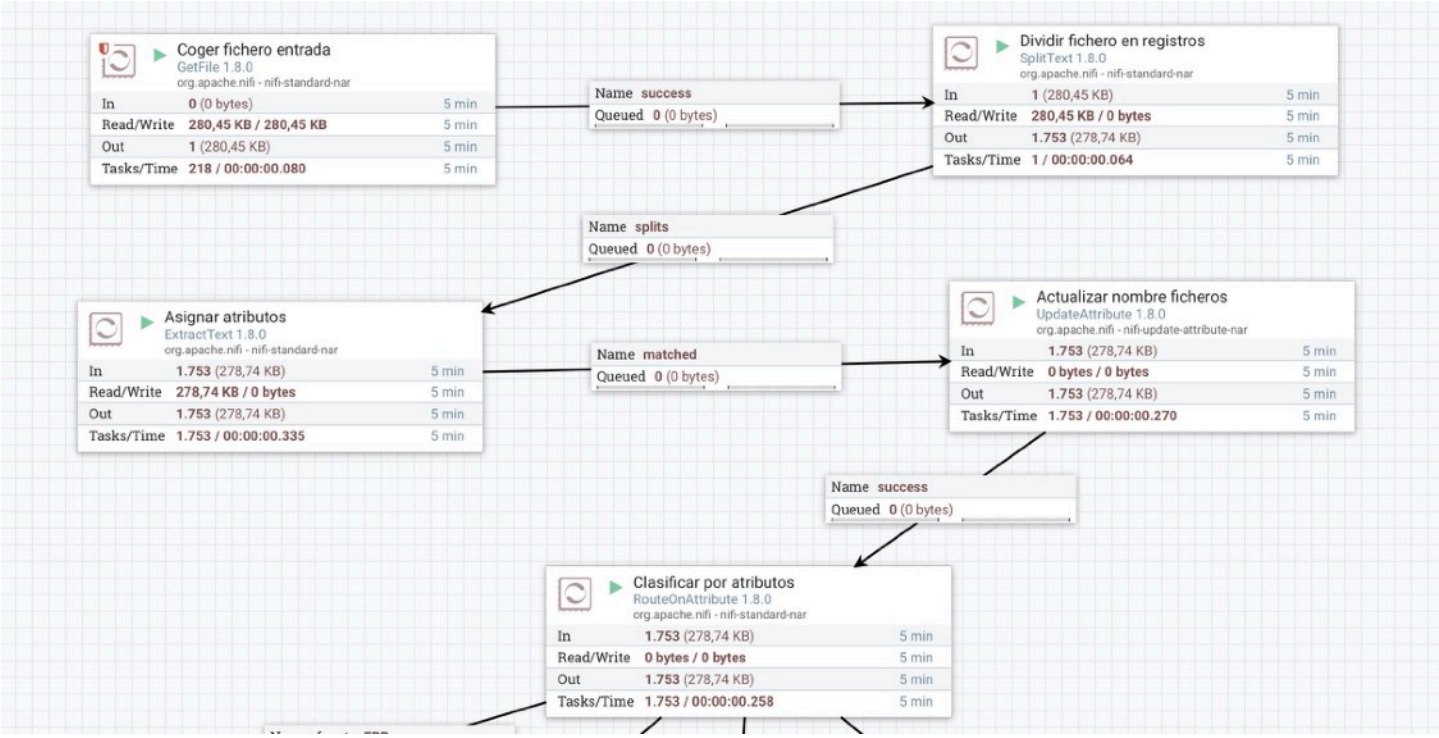
En el procesador “ExtractText” es donde se crea la variable “fuente” que puede tomar los valores comentados: ERP, Salesforce, CRM y Marketing.



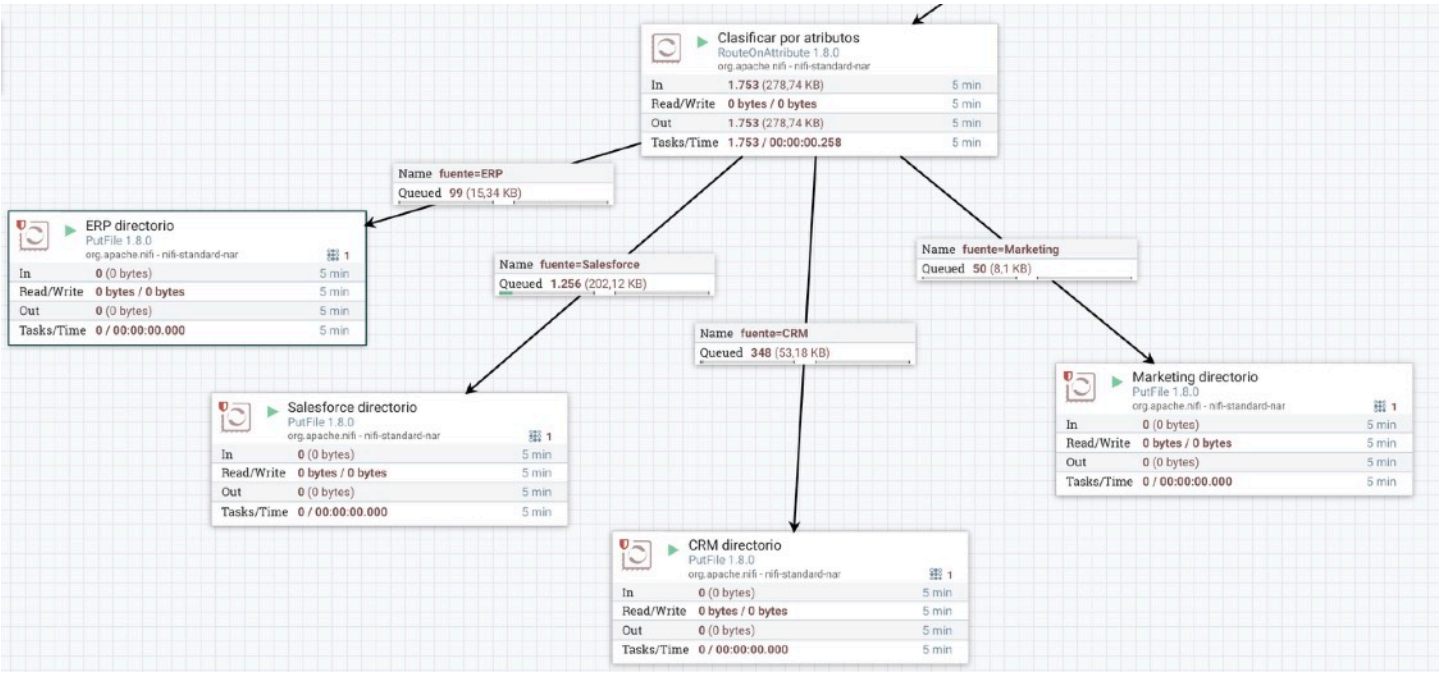
En el procesador RouteOnAttribute se definen los posibles valores de fuente, para redirigir los ficheros según el valor que les corresponda.



Ejecutamos el flujo, y se aprecia como la división vuelve a ser de 1753 registros, y este número va pasando por cada procesador.



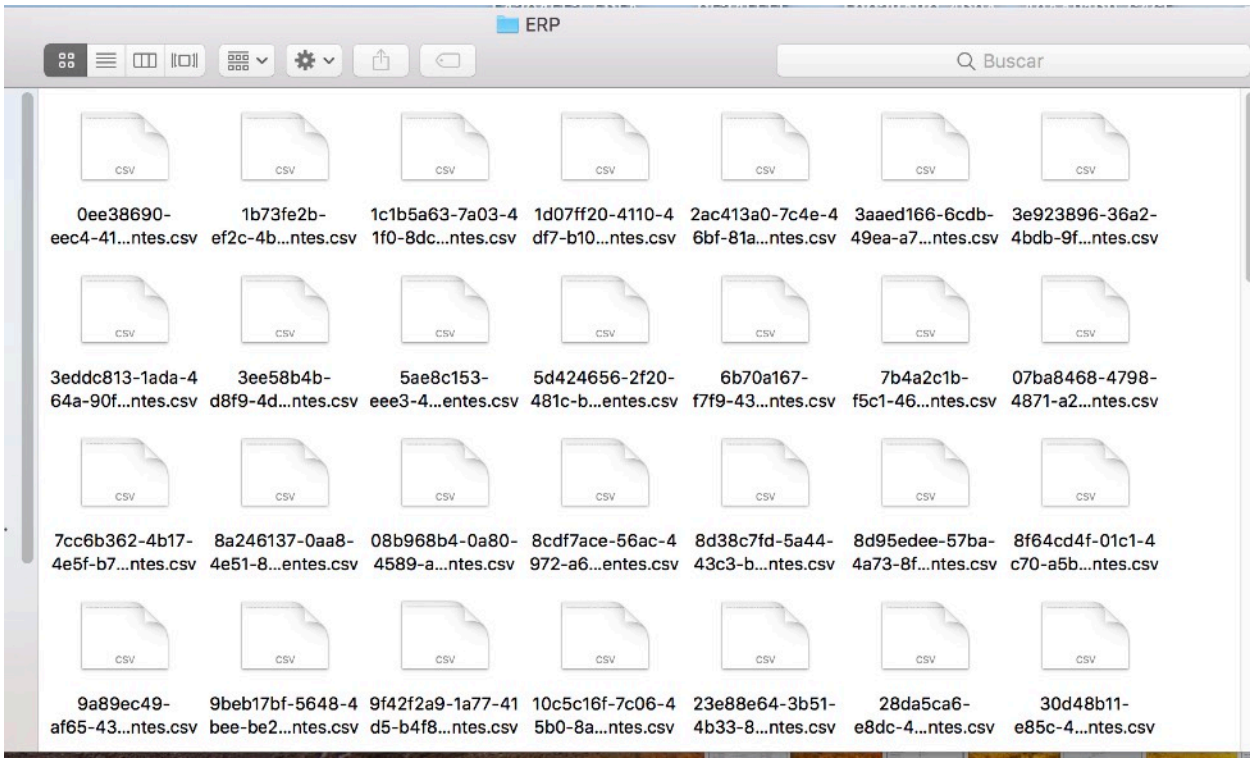
En la siguiente imagen se puede ver como están encolados los ficheros resultantes tras pasar por el procesador de RouteOnAttribute, y como para cada posible valor del discriminado hay un número distinto de ficheros:



Una vez terminado el flujo hacemos un recuento de los ficheros de cada directorio de salida:

```
MBP-de-Jose:CRM josemanuel$ pwd
/Users/josemanuel/server/nifi-1.8.0/putDir/CRM
MBP-de-Jose:CRM josemanuel$ ls | wc -l
     348
MBP-de-Jose:CRM josemanuel$ cd ..
MBP-de-Jose:putDir josemanuel$ cd ERP
MBP-de-Jose:ERP josemanuel$ ls | wc -l
      99
MBP-de-Jose:ERP josemanuel$ cd ..
MBP-de-Jose:putDir josemanuel$ cd Marketing
MBP-de-Jose:Marketing josemanuel$ ls | wc -l
      50
MBP-de-Jose:Marketing josemanuel$ cd ..
MBP-de-Jose:putDir josemanuel$ cd Salesforce
MBP-de-Jose:Salesforce josemanuel$ ls | wc -l
    1256
MBP-de-Jose:Salesforce josemanuel$
```

Por ejemplo vamos a uno de los directorios y vemos efectivamente que están sus ficheros:



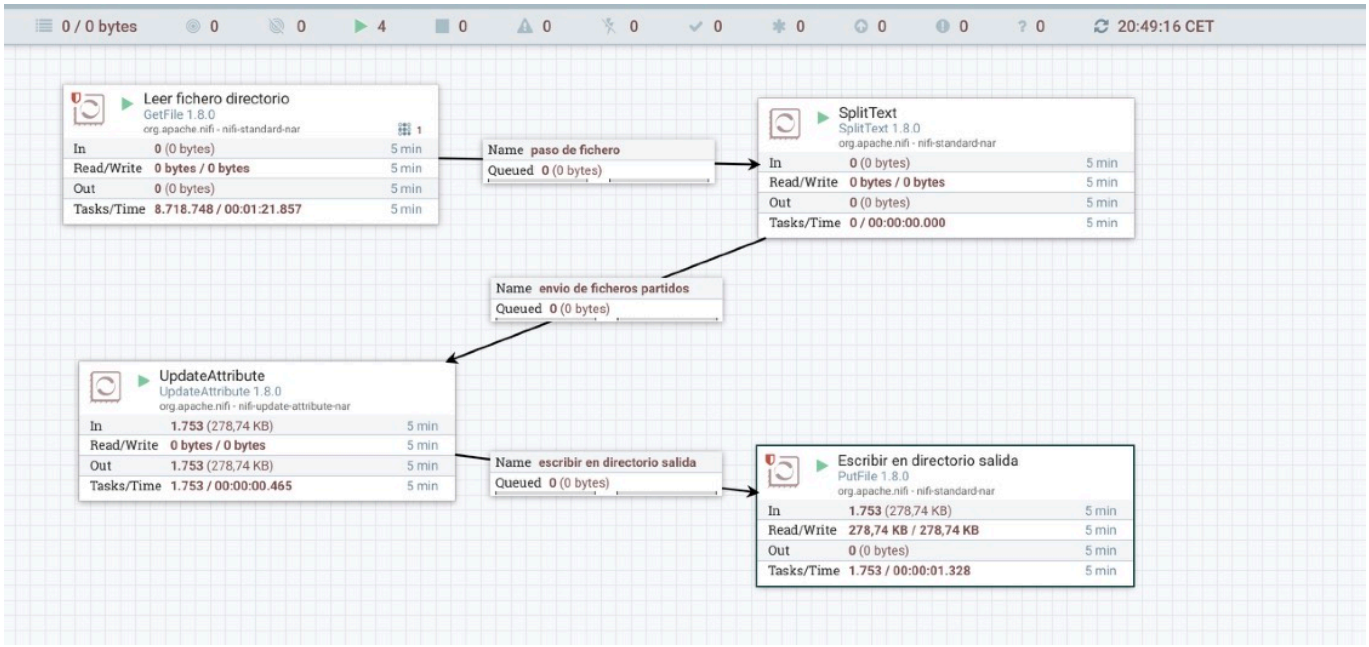
Mostramos por terminal el contenido de un par de ficheros, en este caso del directorio de Marketing y se ve como ambos registros tienen Marketing como el valor de su primer atributo:

```
MBP-de-Jose:Marketing josemanuel$ cat 03334b9f-58db-4b26-9bd4-4c68a2a70c4b-UAM_Clientes.csv
Marketing,GARY,CLAY,"7810 NORTH AVENUE ""A"".",OLNEY,MD,20832,779-4197,3954 SE OLD LEVEE ST.,LA JARA,CO,81140,719,518
-8609,117-63-1298,859-96-7834,03/03/2006,M,MBP-de-Jose:Marketing josemanuel$ cat e058d021-2cf5-499d-b4ca-eb64c9417d65
-UAM_Clientes.csv
Marketing,GARY,CLAY,"7801 NORTH AVENUE ""A"".",OLNEY,MD,20832,779-4197,3954 SE OLD LEVEE ST.,LA JARA,CO,81140,719,518
-8609,117-63-1298,859-96-7834,03/03/2006,M,MBP-de-Jose:Marketing josemanuel$
```

3. Casos teóricos

a. Incorporar cada registro final a una base de datos (qué procesadores utilizarías, con qué orden los encadenarías, aspectos que consideres más relevantes).

Hay varias BBDD de las que Apache Nifi puede tanto recoger información como introducir, incluida HBase en HDFS.
Por ejemplo pensemos que la BBDD en la que queremos introducir los ficheros generados en la práctica como registros, fuera una BBDD MongoDB.
Una vez tuviéramos en el flujo los registros/ficheros, tras el procesador “UpdateAttribute” de la imagen siguiente (flujo 1 de la práctica) podría introducirse un procesador “AttributesToJson” para convertirlos a formato JSON, propio de los documentos que trata Mongo, y posteriormente para acabar introduciendo los datos en la BBDD tenemos a disposición los procesadores “PutMongo” y “PutMongoRecord”.



b. Imaginemos que el campo discriminador inicial no es la fuente de datos sino la fecha de nacimiento (qué procesadores utilizarías, con qué orden los encadenarías, aspectos que consideres más relevantes).

En principio el flujo no lo cambiaría, en cuanto a los procesadores implicados. Dentro del procesador "ExtractText" habría que cambiar la expresión regular existente que servía para buscar la fuente, por una que sirva para encontrar el atributo fecha de nacimiento y dividir los registros según esta fecha.
Para tratar con fechas en el lenguaje de Nifi se tienen las siguientes funciones: format, toDate y now. Pienso que se podría intentar mediante expresiones regulares usando la función necesaria como format tratar el atributo fecha.

Otra opción que se han visto ejemplos es que luego en el procesador "UpdateAttribute" se transformara la fecha por ejemplo en un formato más sencillo para tratar como discriminador, como sería el valor de la fecha en timestamp, que podría ser más amigable para luego dirigir cada registro según este valor a su directorio destino pertinente.