PRACTICA 2
ALMACÉNS E MINARÍA DE DATOS

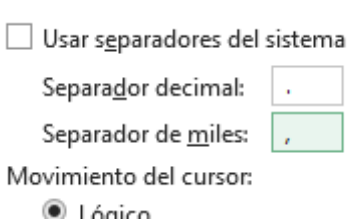
Alex Baquero Domínguez

alex.baquero@rai.usc.es

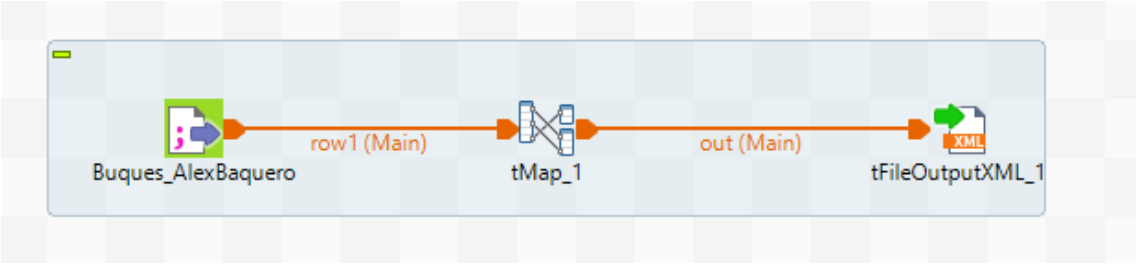
Táboa de contido

1. **Partindo dos ficheiros do Rexistro de buques 2019-2021 en formato csv y XML, convertilos no ficheiro csv en XML e viceversa, comparándolos.**

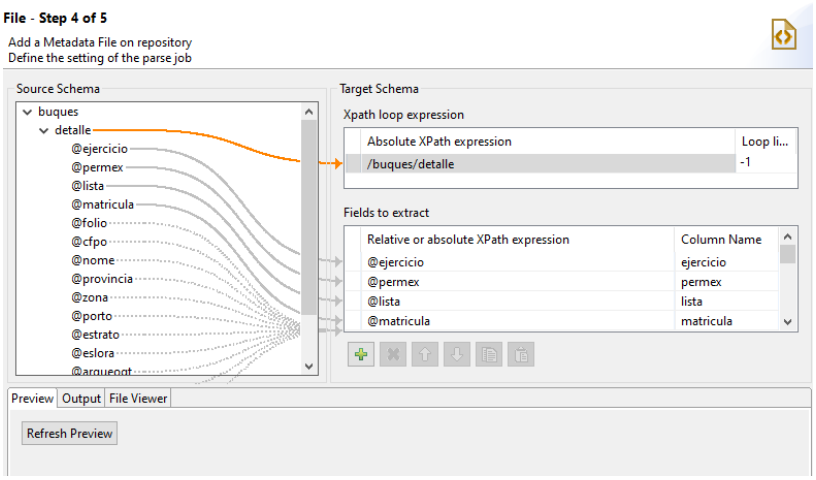
- **Conversión de CSV a XML:** No apartado de metadata, creamos un archivo *File Delimited* e cargase o arquivo Buques 2019-2021.csv. Mapeamos os atributos dandolle nome a cada columna e especificando que tipo de atributo é cada un. Finalmente, se pasa o esquema a arquivo XML estandarizado.
  Mencionar que tivemos que cambiar as , da columna estora porque daba error. Para iso, entramos no csv a través do Excel e clicamos opciones> avanzadas:



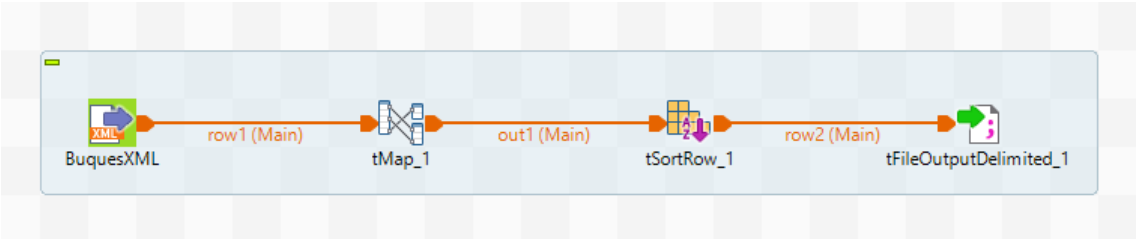
  Desclicamos a opción “Usar separadores del sistema”, poñemos un punto e aceptamos. Logo seleccionamos a columna eslora e damoslle formato como numero.



- **Conversion de XML a CSV:** Facemos o mesmo que o mencionado anteriormente pero neste caso creamos un archivo *tFileInputXML* para cargar o arquivo Buques 2019-2021.xml e a saída vai ser un arquivo csv. Este apartado foi menos traballoso pois os nomes dos atributos xa veñen dados automáticamente.
  Mencionar que tivemos que poñer un -1 no Loop Limit para que lese todos as liñas do arquivo e non só as 50 primeiras liñas. Ademais de indicar o xPath de onde se encontren todos os buques:

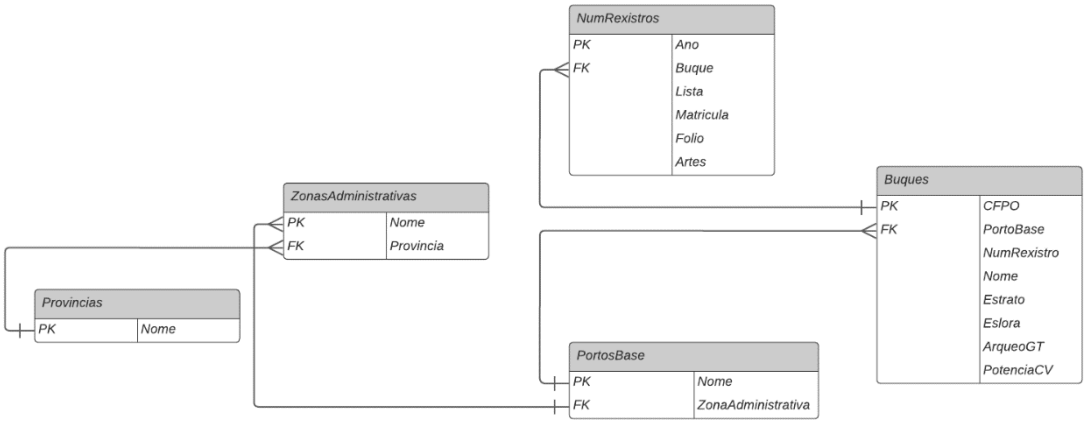


Finalmente, visualmente quédanos este resultado. Podemos ordenar os datos por un dos campos usando o tSortRow, como por exemplo por CFPO.



2. Importación do CSV a un esquema normalizado en PostgreSQL.

O esquema normalizado é o que aparece na seguinte foto:

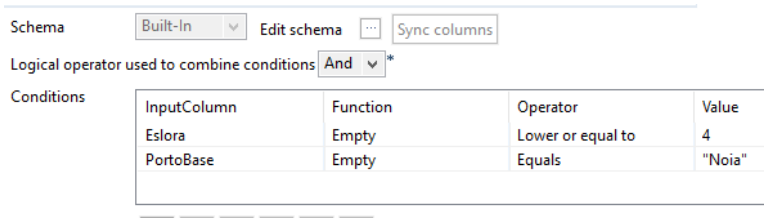


Despois de crear este esquema, creamos unha base de datos no noso PGAdmin e creamos as seguintes táboas:

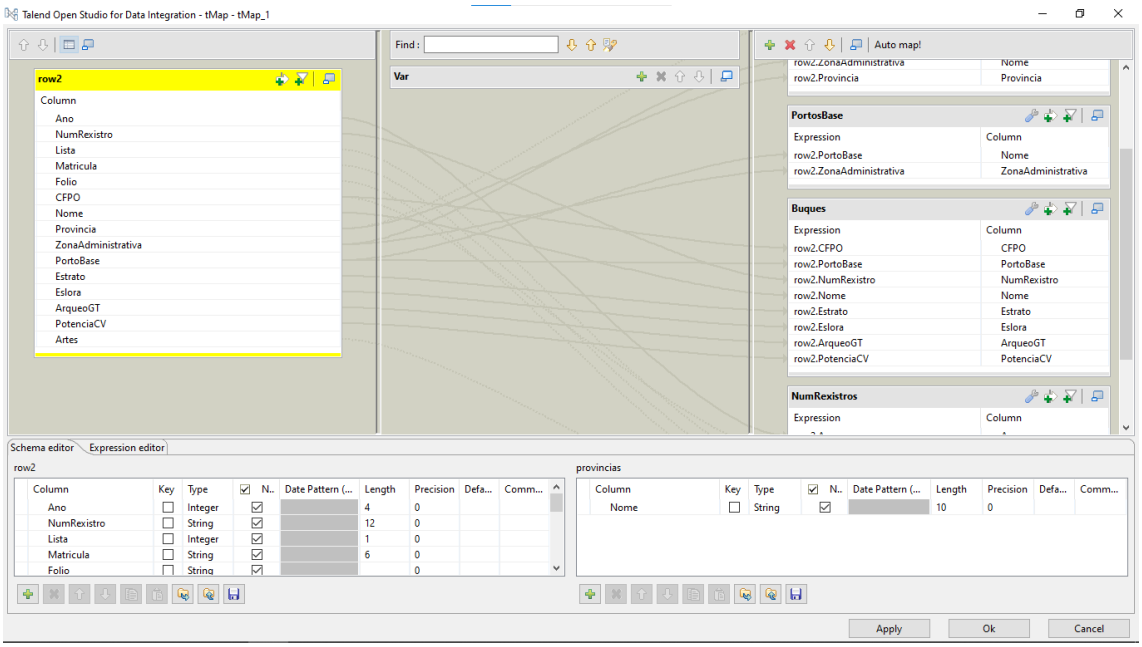
```
CREATE TABLE Provincias(
        Nome Varchar(300)  PRIMARY KEY
);
CREATE TABLE ZonasAdministrativas(
        Nome Varchar(300) PRIMARY KEY,
        Provincia Varchar(300),
        FOREIGN KEY(Provincia) REFERENCES Provincias(Nome)
);
CREATE table PortoBase(
        Nome Varchar(300) PRIMARY KEY,
        ZonaAdministrativa Varchar(300),
        FOREIGN KEY(ZonaAdministrativa) REFERENCES ZonasAdministrativas(Nome)
);
CREATE TABLE Buques(
```

```
        CFPO varchar(300) PRIMARY KEY,
        PortoBase Varchar(300),
        NumRexistro Varchar(300),
        Nome Varchar(300),
        Estrato decimal,
        Eslora decimal,
        ArqueoGT decimal,
        PotenciaCV decimal,
        FOREIGN KEY(PortoBase) REFERENCES PortoBase(nome)
);
CREATE TABLE NumRexistros(
        Ano int,
        Buque varchar(300),
        Lista int,
        Matricula Varchar(300),
        Folio Varchar(300),
        Artes Varchar(300),
        FOREIGN KEY(Buque) REFERENCES Buques(CFPO)
);
```

Comenzamos abrindo unha conexion á base de datos de PostgresSQL (onde temos creadas as táboas normalizadas) e leendo os datos do fichero .csv. Antes de mappear os atributos a cada unha das 5 táboas creadas, filtramos os datos que teñan unha lonxitude inferior a 4 metros e conserven os datos do porto Noia:

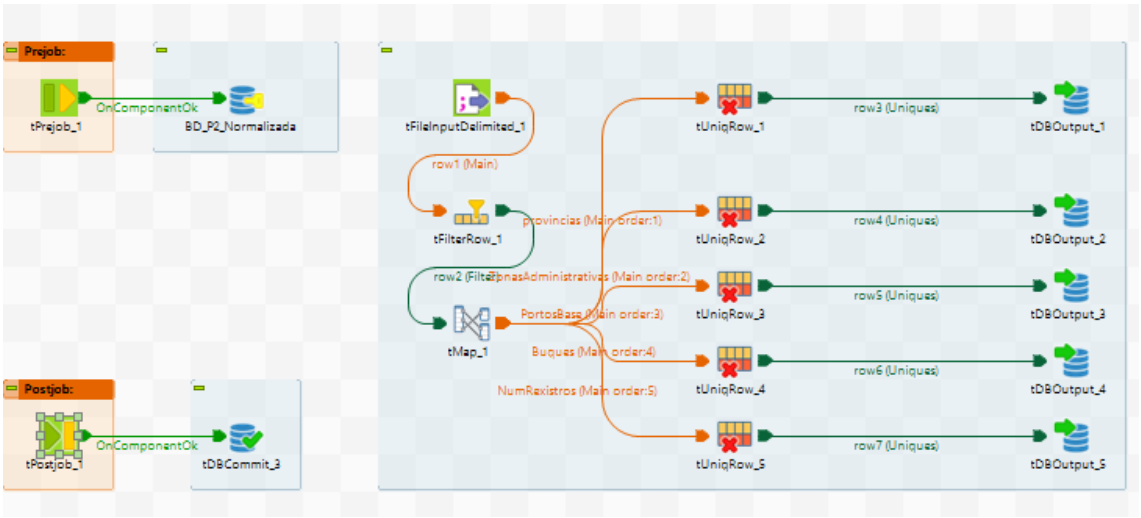


Os datos mappeanse e pasanse por un filtro *unique* para intentar elementos dupilicados:



Nos elementos do tDBOutput, ponse que as táboas deben ser creadas e borradas cada vez que se execute e eliximos o metodo insertar.

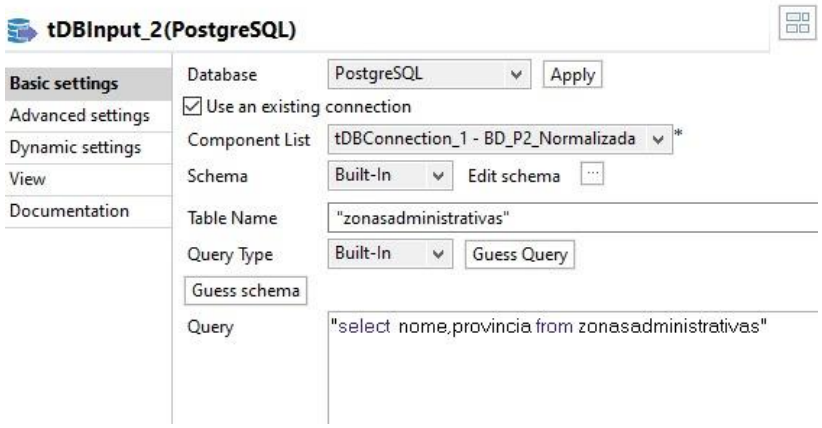
Finalmente, facemos commit e pechamos a conexion da base de datos.

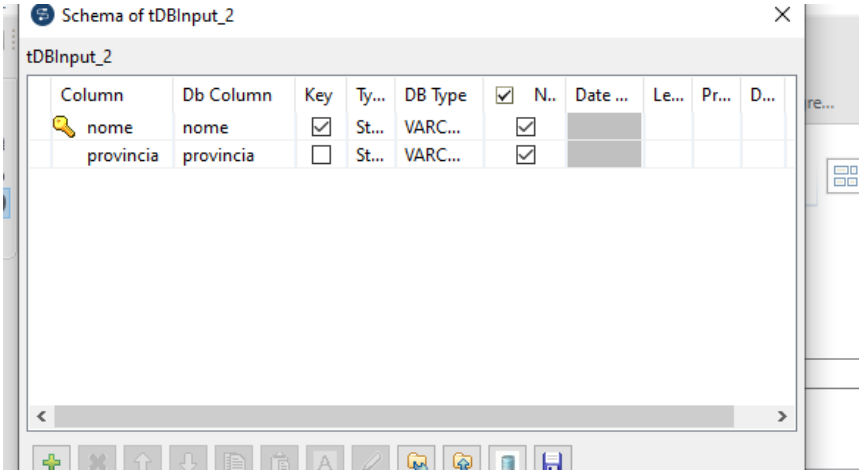


3. Calculo do total de arqueo por provincia e comparación dos resultado usando talend e PostgreSQL.

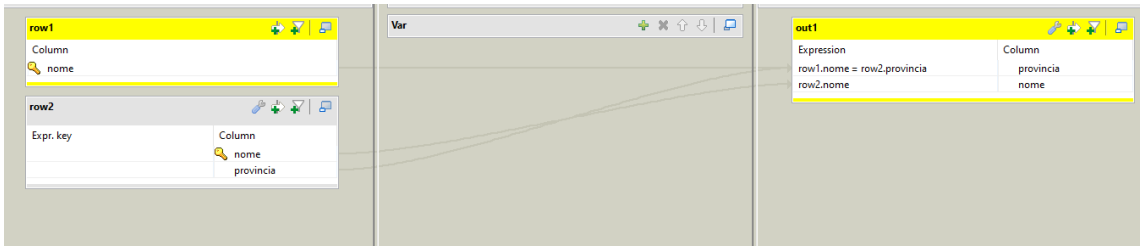
Comenzamos conectandonos a base de datos e vamos unindo as táboas a través dos mapeos e os distintos tDBInput.

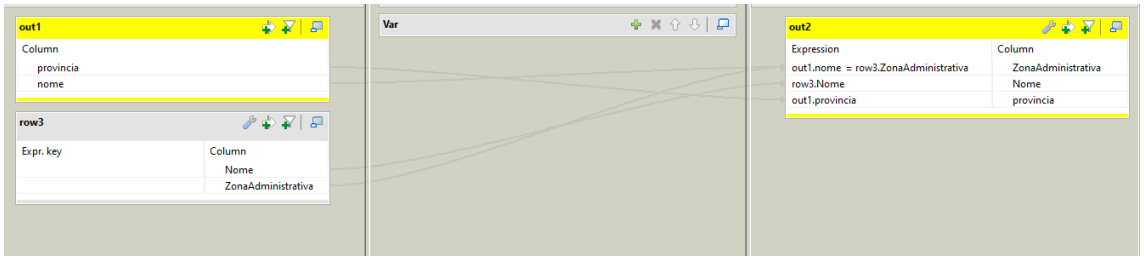
Exemplo:

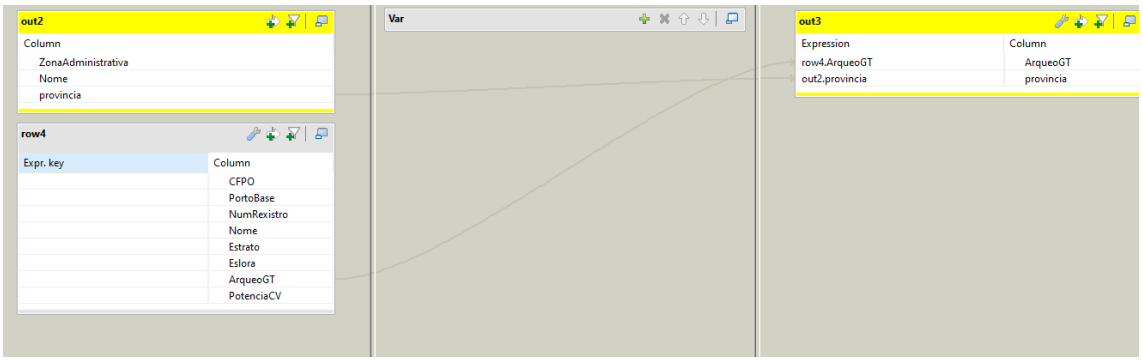




Vamos unindo as chaves primarias coas chaves foráneas:

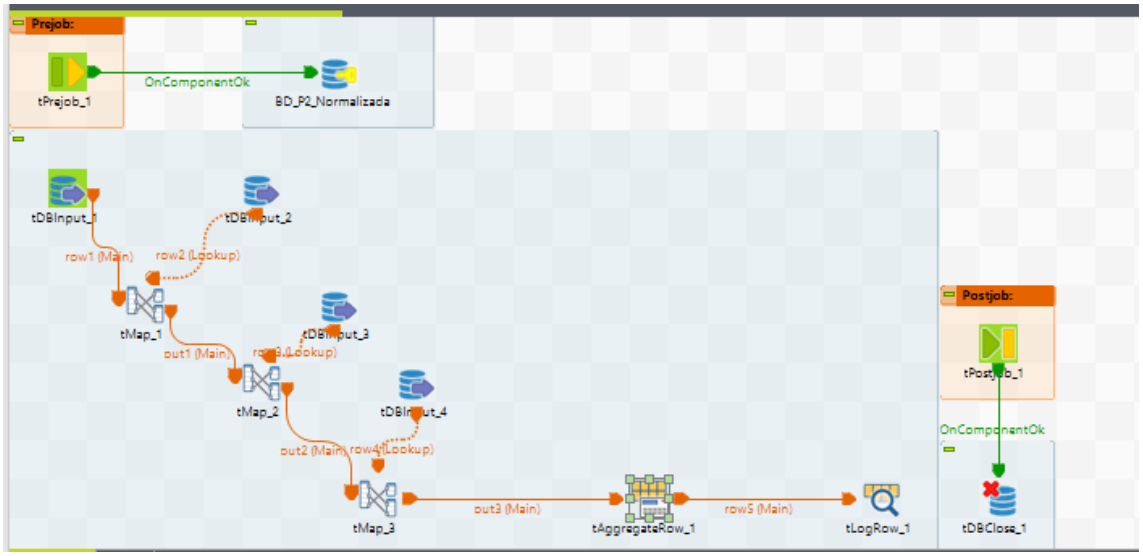






Para facer o total de arqueos, é dicir un *sum* nunha consulta SQL, usamos un tAggregateRow agregando as filas por provincia.

Finalmente, imprimimos o resultado por pantalla e cerramos a conexión na base de datos.

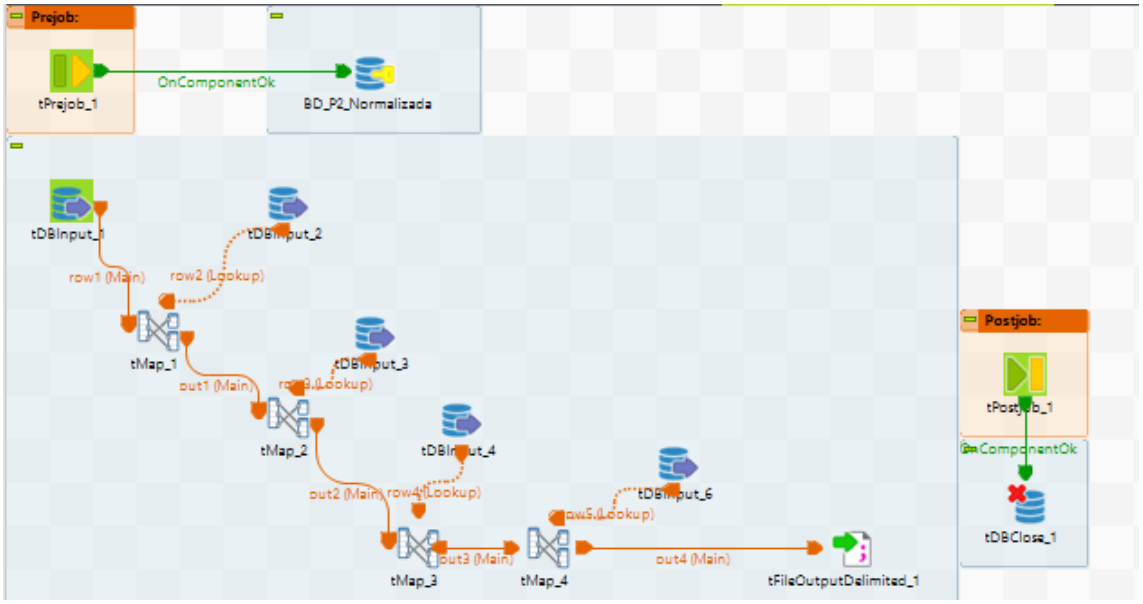


O arqueo por provincia usando o PostgresSQL sería da seguinte maneira:

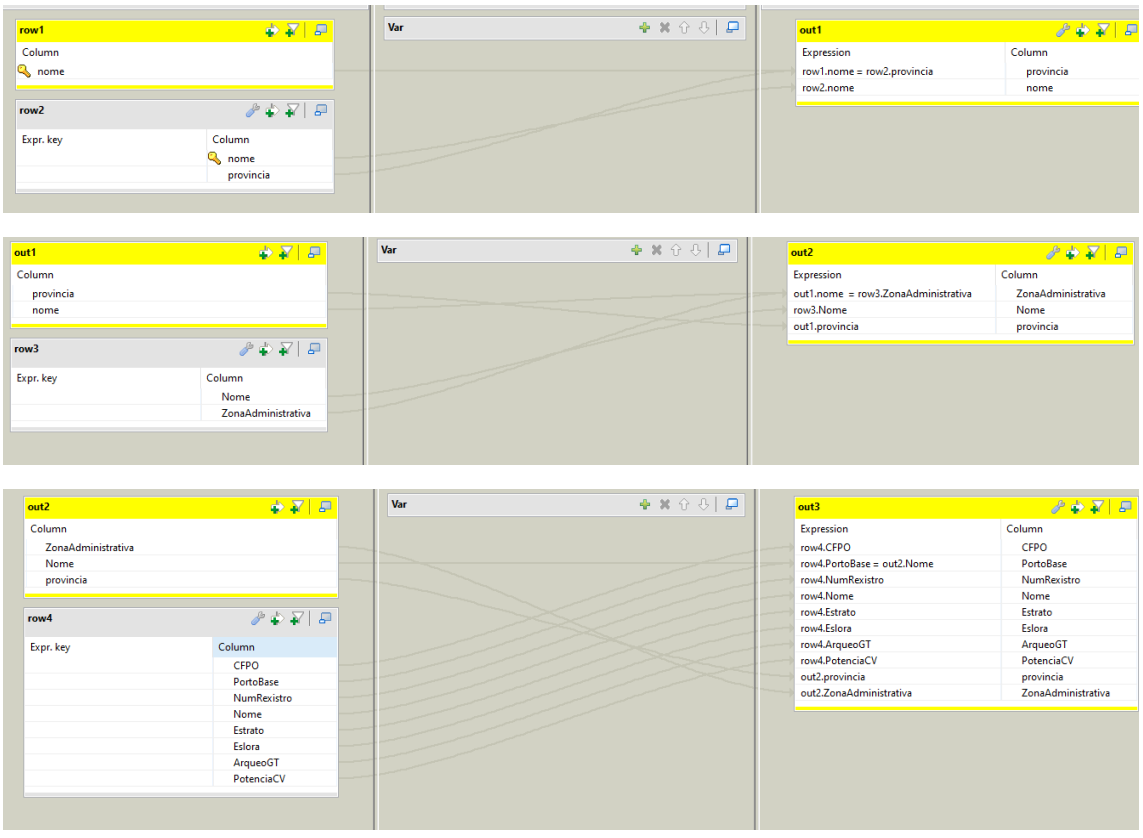
```
SELECT pr.nome, SUM(bq.arqueogt)
from provincias pr,
 zonasadministrativas za,
 portobase pb,
 buques bq
where pr.nome = za.provincia AND za.nome = pb.zonaadministrativa AND pb.nome = bq.portobase
GROUP BY pr.nome;
```

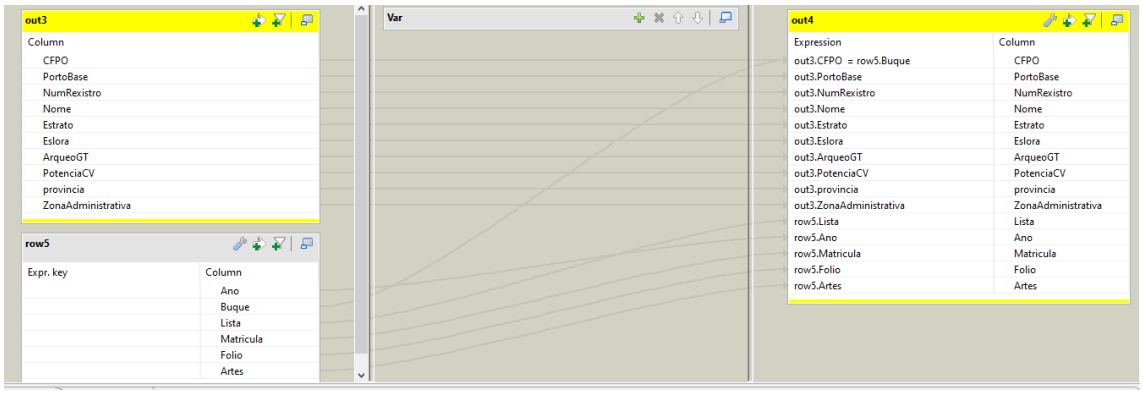
Comprobamos que coincide que en ambos A Coruña ten 51157.959 arqueos, Lugo 23848.31 arqueos e Pontevedra 80551 arqueos.

4. **A través da anterior BBDD como entrada, importamos nun esquema desnormalizado que simule o arquivo orixinal.**



Conectámonos a base de datos e importamos os datos a través dos tDBInput, os cales vamos mappeando pouco a pouco unindo as chaves primarias coas foraneas:

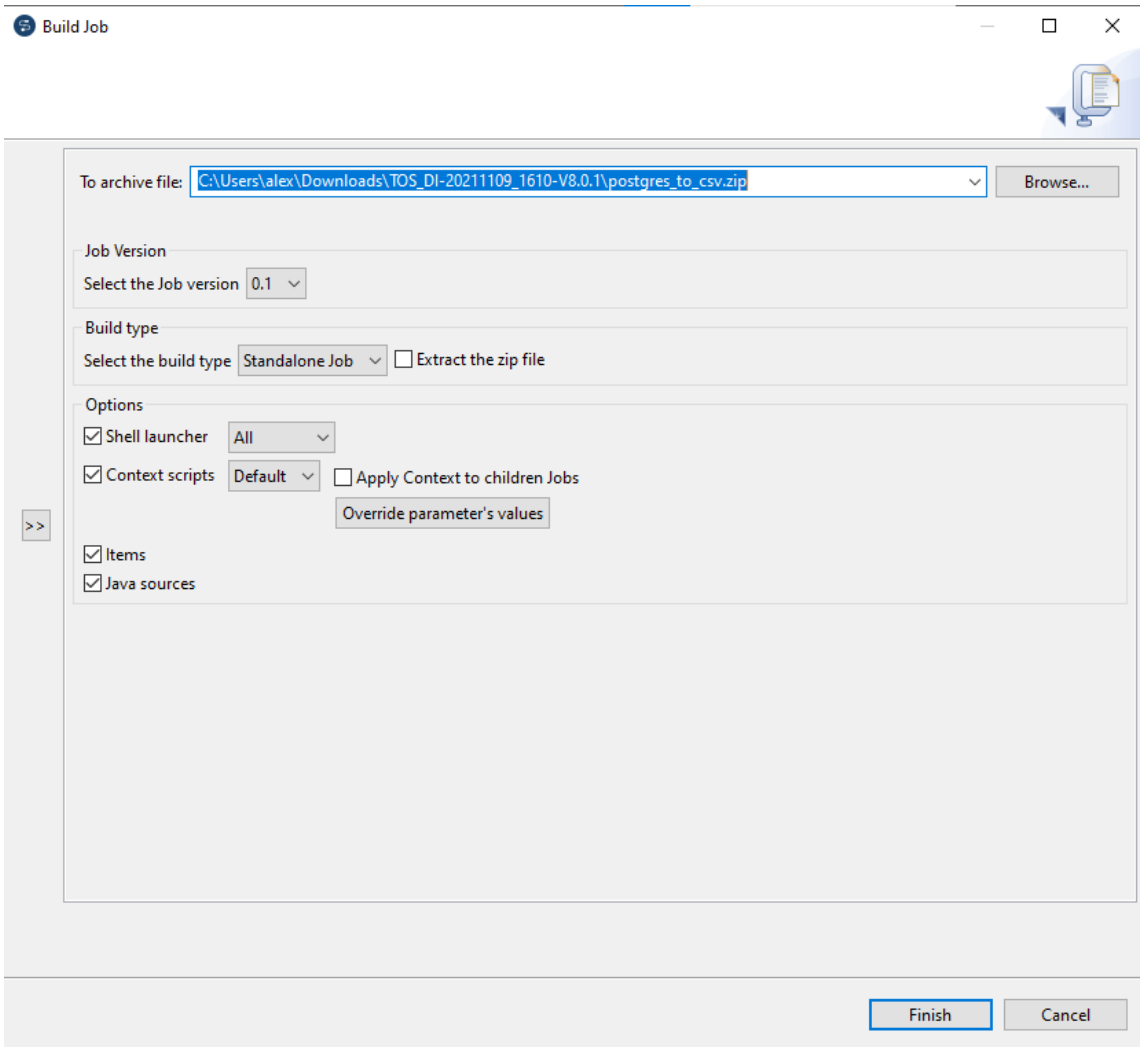




Finalmente, o ultimo mappeo conten todas columnas do esquema desnormalizado do ficheiro orixinal, o cal introducese nun archivo .csv e cerrase a conexion coa base de datos

5. Execución dede la línea de comandos

Grazas a función *Build a job,* podese agregar todos os arquivos necesarios para executar o job do talend nun arquivo, incluidos os arquivos .baty e .sh xunto con calquera arquivo de parámetros de contexto u outros arquivos relacionados. Este arquivo é un .zip que ten un jar coa execución do arquivo.



Debido as dependencias da carpeta lib/, este JAR non se pode executar sobre Java. Podes executalo a través duns scripts para os distintos sistemas operativos: Windows, MacOs, Linux..

Nome	Data de modificación	Tipo	Tamaño
amd_p2	23/10/2022 23:15	Cartafol de ficheiros	
items	23/10/2022 23:15	Cartafol de ficheiros	
src	23/10/2022 23:15	Cartafol de ficheiros	
log4j2.xml	23/10/2022 23:14	Documento XML	1 KB
postgres_to_csv_0_1.jar	23/10/2022 23:15	Executable Jar File	64 KB
postgres_to_csv_run.bat	23/10/2022 23:14	Archivo por lotes ...	1 KB
postgres_to_csv_run.ps1	23/10/2022 23:14	Script de Window...	1 KB
postgres_to_csv_run.sh	23/10/2022 23:14	Shell Script	1 KB

Abrimos o .bat: