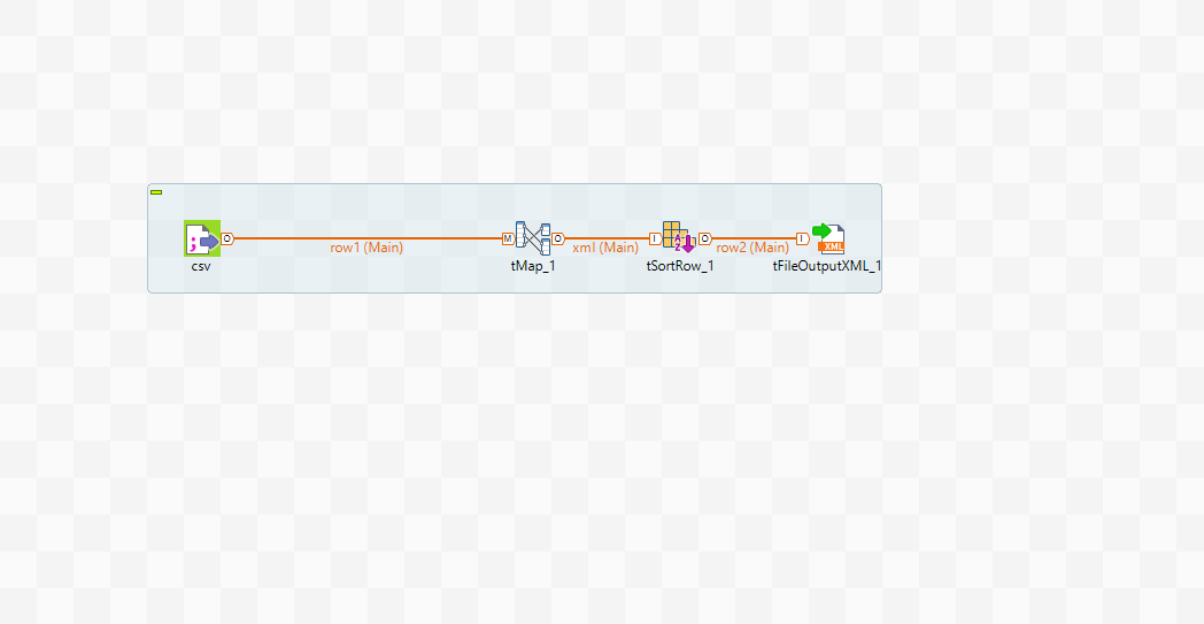
Práctica 2

Talend

CSV a XML



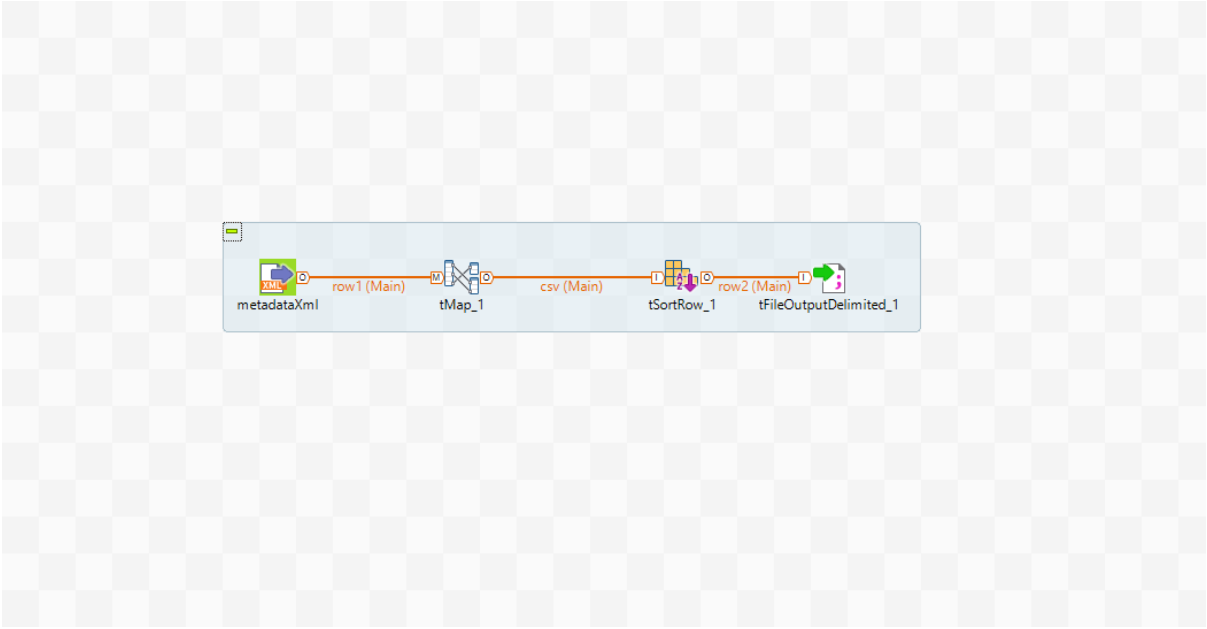
Para este ejercicio, se usan los componentes que se pueden ver en la imagen. Primero, se carga el archivo CSV especificando las propiedades de este (delimitador, comillas, etc.).

A continuación, se mapean los atributos, dándole nombres a cada columna (en este paso, además, se corrigen errores de decimales, ya que en el archivo original el elemento decimal es la coma en vez del punto, usando `row1.eslora.length() == 0 ? null : Float.valueOf(row1.eslora.replace(",", "."))`).

Se pueden ordenar los datos usando el componente tSortRow, como por ejemplo por año y por puerto.

Finalmente, se pasa el esquema al archivo XML estandarizado.

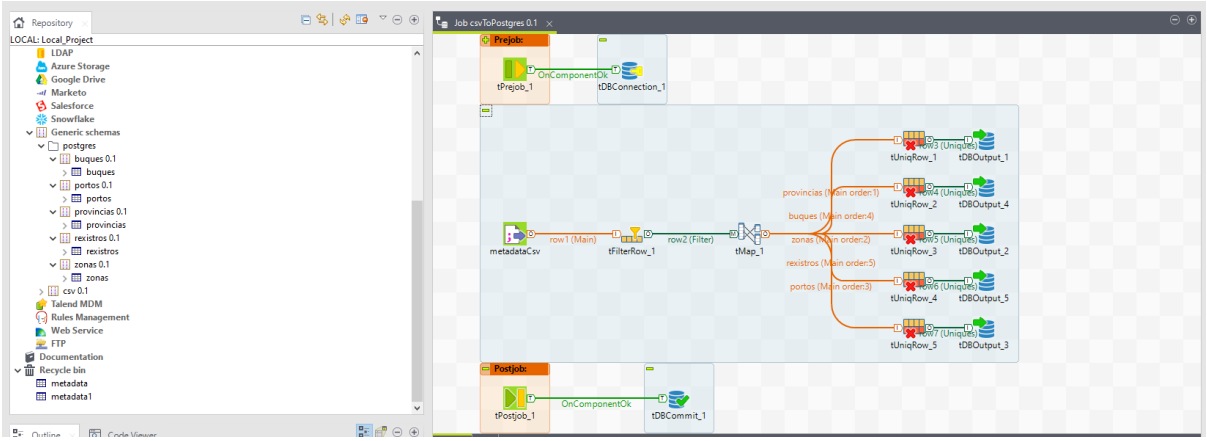
XML a CSV



El proceso es el mismo que el ejercicio anterior, pero siendo la entrada un archivo XML y la salida un archivo CSV.

Quizás la diferencia más salientable es que en el componente de lectura XML es necesario indicar un Loop Limit de -1 para que lea todos los elementos, y el XPath de donde se encuentran todos los buques. Además, el usar XML facilita el trabajo en el sentido en el que ya vienen dados los nombres de los atributos.

CSV a PostgreSQL



Este ejercicio es algo más complejo. Primero, se definen unos esquemas genéricos con las columnas de las tablas normalizadas. Estas son provincia, zona (relacionada con provincia), puertos (relacionada con zona), buques (relacionada con puertos) y registros (para cada buque y año una entrada). Los datos en la tabla de buques son número de registro, cfpo, nombre, estrato, eslora, arqueo y potencia; y en la tabla de registros se almacena lista, matrícula, folio y artes.

En cuanto al job de Talend, se comienzan definiendo unos pre y post ejecuciones. Primero, se define una conexión a la base de datos PostgreSQL para facilitar el uso del resto de componentes (ya que estos reutilizarán la conexión, sin abrir ninguna otra nueva). La post condición será de commit y cierre de conexión, ya que se desactiva el auto-commit.

El flujo principal de datos consiste en, primero, leer los datos del archivo CSV, y filtrar los elementos que tengan menor eslora de 4 metros y quedarse con los datos de un solo puerto. Esto se puede realizar con la consulta personalizada `(input_row.eslora.length() == 0 || Float.valueOf(input_row.eslora.replace(",", ".")) > 4) && input_row.porto.equals("Ribeira")`.

A continuacuón, se mapean los datos con los esquemas de salida definidos anteriormente. Cabe destacar que, estos esquemas, se han mapeado para PostgreSQL y se han definido las correctas claves primarias, junto con los tipos de datos de la base de datos.

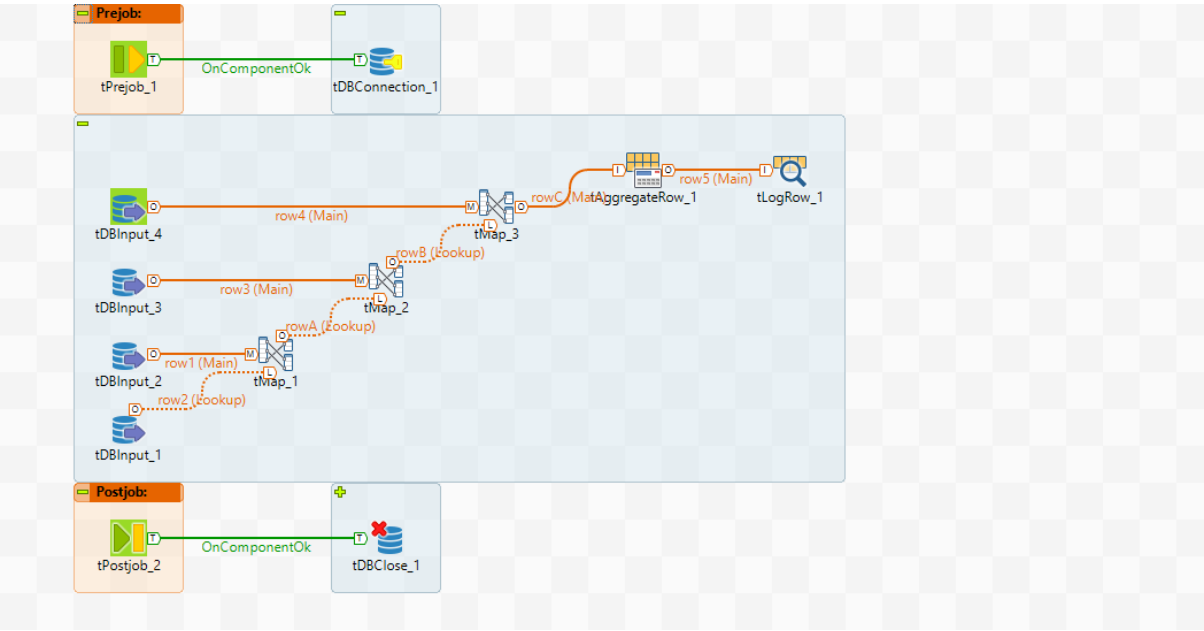
Finalmente, esta salida del mapeo se redirige a cada una de las 5 tablas creadas. Antes de insertarlas, se pasan por un filtro de unicidad, para evitar insertar provincias duplicadas o similares debido al esquema no normalizado del CSV. En los componentes de inserción en base de datos, se ha especificado que las tablas se destruyan y se creen en cada ejecución, y el método de “Insert” para los datos.

Tonelaje por Provincia

La consulta SQL equivalente de esta consulta es la siguiente:

```
SELECT p.provincia, SUM(b.arqueo)
from provincias p,
     zonas z,
     portos po,
     buques b
where p.provincia = z.provincia
  AND z.zona = po.zona
  AND po.porto = b.porto
GROUP BY p.provincia;
```

La equivalente en Talend sería:

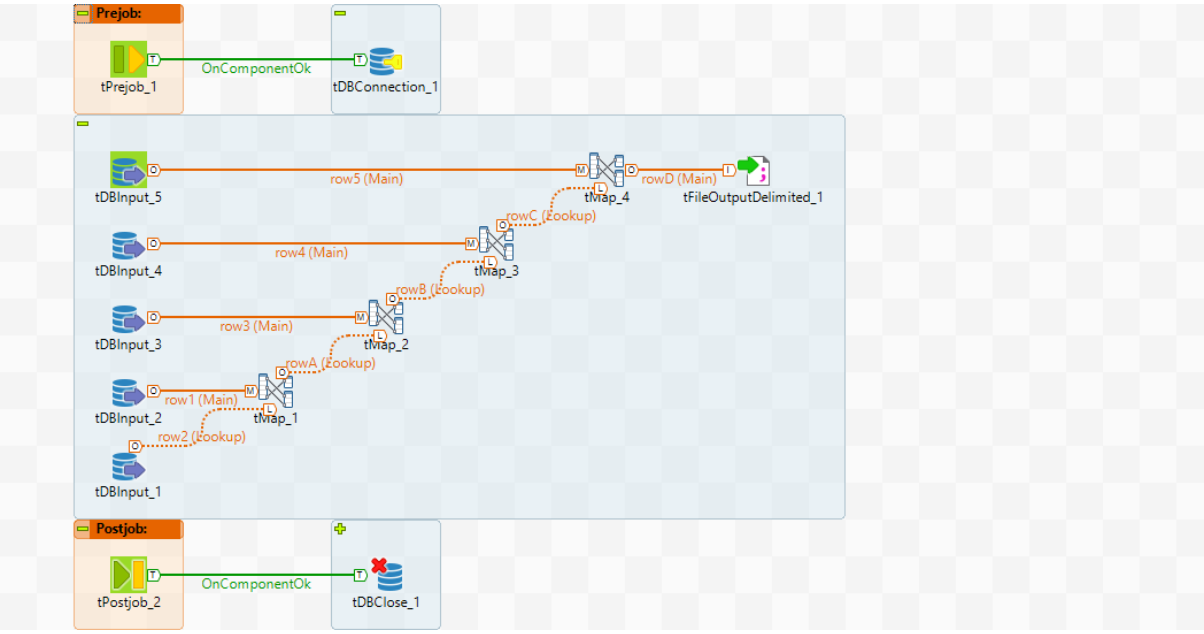


Como se puede apreciar, el ejercicio se parece al anterior en cuanto a pre y post condiciones (pero con la diferencia de que en vez de ejecutar un commit se ejecuta un cierre de conexión).

La primera parte del ejercicio es ir uniendo las tablas en una única para poder agregarlas (sería el FROM WHERE de la consulta SQL). Sobre los componentes de Postgres, cabe comentar que se han especificado los esquemas definidos en ejercicios anteriores, para facilitar el trabajar con esos datos. Sobre los componentes de mapeo, se referencian los datos de las otras tablas como claves foráneas.

A continuación, se agregan las filas por provincia ignorando valores nulos para evitar errores. Por último, se imprime el resultado por pantalla usando el componente de logging.

PostgreSQL a CSV



Este ejercicio es muy similar al anterior. La única diferencia es que también se junta la tabla de registros, y la salida de este mapeo se pasa directamente a un componente de salida como el usado en el primer ejercicio.

Línea de Comandos

Talend permite exportar los trabajos a archivos ejecutables sin necesidad de tener Talend instalado.

Esto se puede conseguir desde el panel lateral de la izquierda, haciendo click derecho en trabajo y dándole a “Build Job”. Se generará un archivo ZIP, el cual contiene un JAR con la ejecución del trabajo. Sin embargo, no se puede ejecutar directamente sobre Java debido a que faltarían dependencias de la carpeta lib/. Por eso, en el ZIP se adjuntan unos scripts para Windows, MacOS y Linux que permiten ejecutarlo a través de estos scripts.