Prática SESIón 3 -  
operaciones básicas con pentaho

Este documento, respondiendo al enfoque del curso principalmente práctico, contempla la realización de ejercicios para la correcta asimilación de los conceptos del proceso ETL a través de la herramienta Pentaho Data Integration. Haciendo uso de distintos ejercicios guiados, se expone cómo desarrollar transformaciones básicas y flujos de trabajo con los que poder empezar a trabajar con este tipo de tecnologías.

Contenido

[1. Introducción 1](#_Toc70970854)

[2. Instalación de PDI 1](#_Toc70970855)

[3. Ejercicio 1 – Orígenes/Destinos 2](#_Toc70970856)

[4. Ejercicio 2 – Unión de tablas 4](#_Toc70970857)

[5. Ejercicio 3 – Fórmulas 7](#_Toc70970858)

[6. Ejercicio 4 – Generación de Jobs 8](#_Toc70970859)

## Introducción

Kettle es un componente de Pentaho Data Integration (PDI) que a su vez contiene a Spoon.

Con spoon se pueden realizar procesos ETL de manera muy fácil y rápida, de hecho, se puede hacer casi de todo:

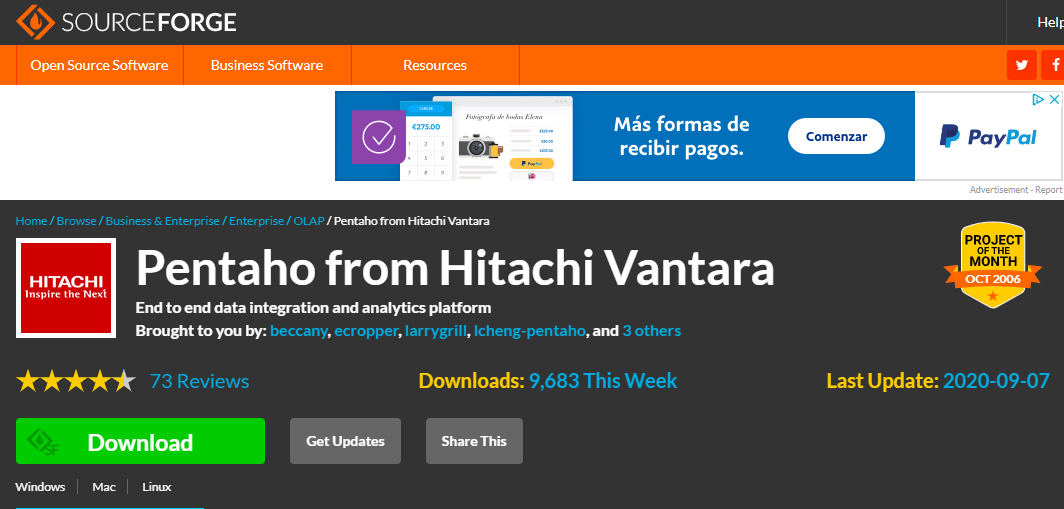
* Conexiones a los datos
* Transformaciones
* Insertar fórmulas
* Desarrollar un DW con estructura en estrella (Hechos/Dimensiones)

Y todo esto sin necesidad de programar directamente con código y sin necesidad de instalar o configurar nada para poder empezar a usarla. Es por ello, que esta herramienta resulta especialmente interesante en el ámbito educativo para formar a futuros profesionales del dato.

## Instalación de PDI

Para descargar la aplicación acceder al siguiente enlace: <https://sourceforge.net/projects/pentaho/>

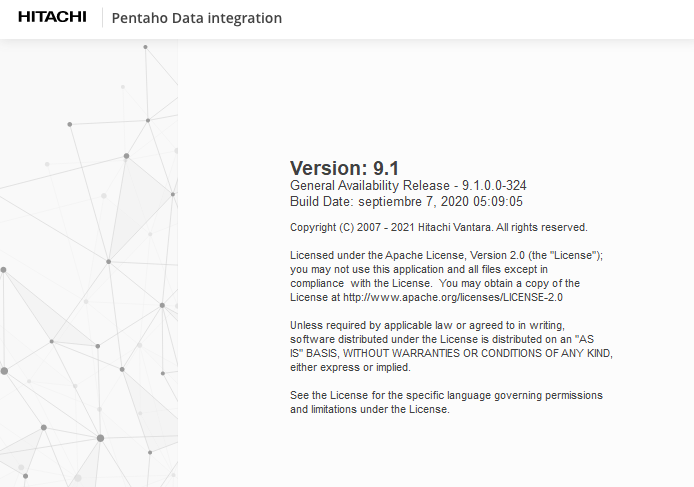
Seleccionar directamente la opción “Download” y al poco tiempo comenzará la descarga de la aplicación en un archivo .zip que únicamente tendremos que descomprimir para poder comenzar a trabajar con kettel.



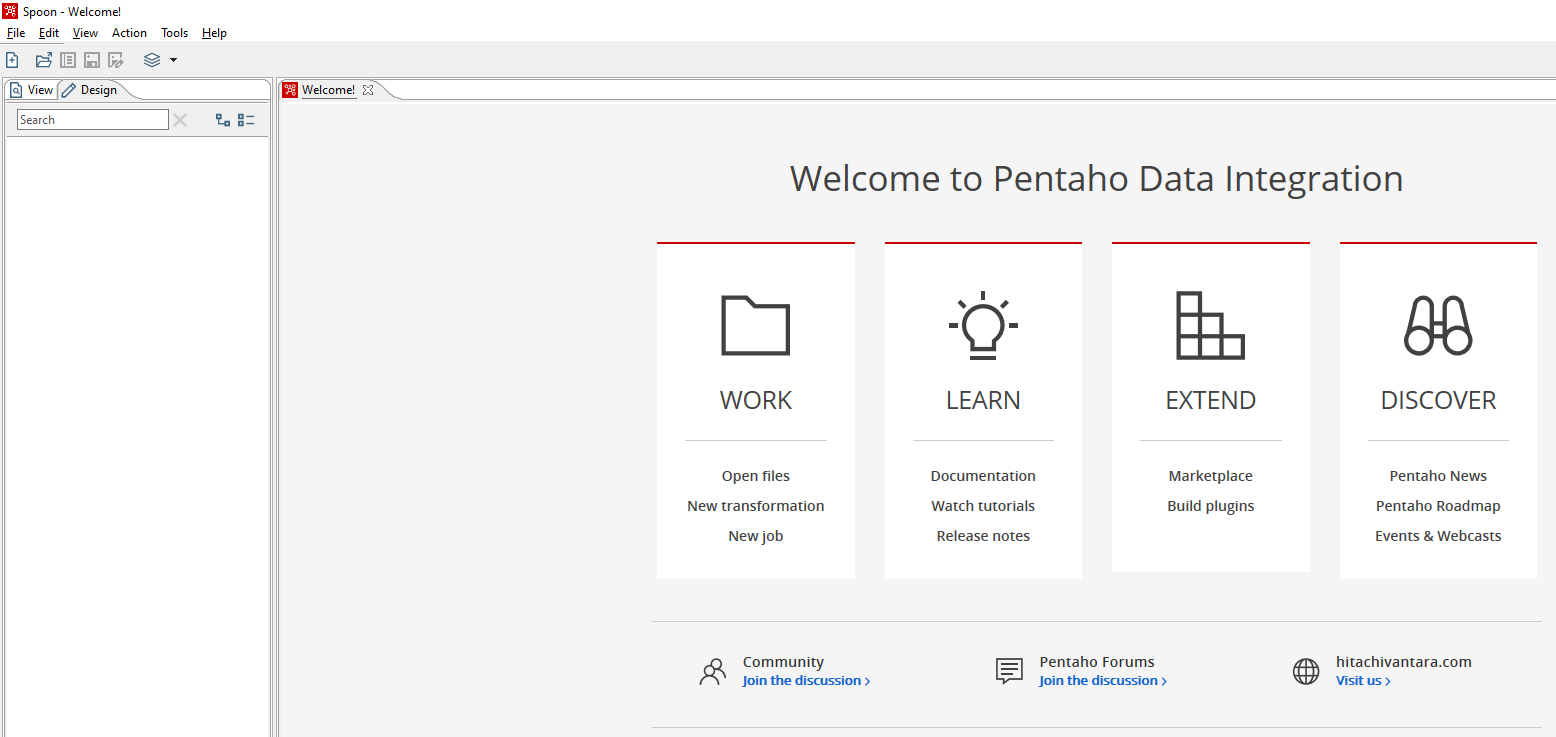
Una vez descomprimido el directorio data-integration en el lugar deseado procedemos a ejecutar el archivo Spoon.bat



Y si no hay ningún inconveniente deberías poder visualizar la ventana de carga de la aplicación



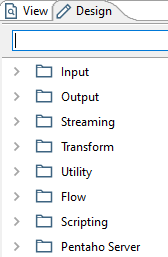
Una vez finalizada la carga ya podrás ver el entorno de trabajo de Spoon.



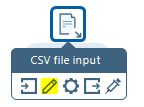
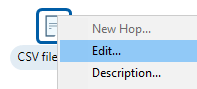
## Ejercicio 1 – Orígenes/Destinos

Para poder realizar los siguientes ejercicios vamos a necesitar descargarnos la carpeta Recursos\_ETL disponible en Moodle en Tema 3, donde estarán los ficheros .csv con los que vamos a trabajar.

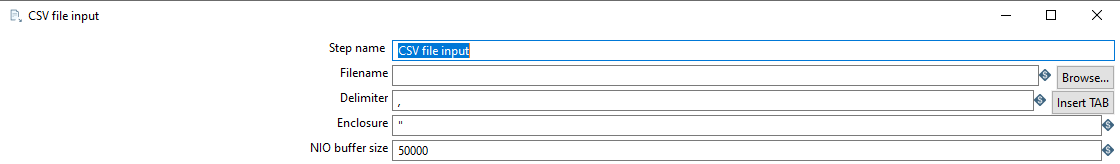
El primer componente que vamos a necesitar es de tipo Input y para leer ficheros .csv debemos seleccionar CSV file input. Los componentes se localizan en la carpeta design:



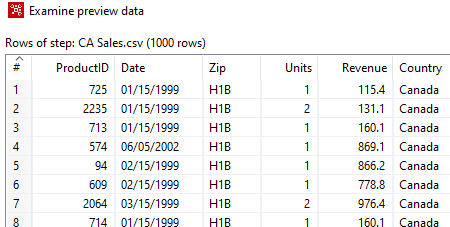
Una vez puesto en el área de trabajo seleccionamos editar las propiedades de nuestro componente para indicar el arhivo *.csv* de origen. Para acceder a las propiedades de edición de nuestro componente podemos situarnos sobre el componente y esperar que se despliegue un menú con varias opciones o bien, pulsamos botón derecho y le damos a propiedades:

1.  b) 

Ambas opciones nos llevan a la misma pantalla de edición del componente:



Mediante el botón *Browse…* seleccionamos el archivo CA Sales.csv que encontraremos dentro de Recursos\_ETL. Dejamos el resto de campos como están y pulsamos Get Fileds para traernos los campos. Le damos a Preview y si todo es correcto deberíamos visualizarlos sin problemas. Para finalizar le damos a Ok.

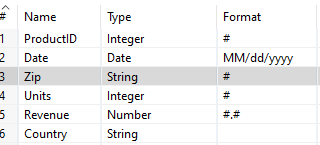


Repetimos la misma acción con el resto de orígenes:

* + DE Sales.csv
  + FR Sales.csv
  + MX Sales.csv

Por último, añadimos un componente de tipo Output para guardar los valores, en este caso vamos a elegir el Text file Output. Indicamos el directorio, nombre y extensión del archivo: “Ventas.csv”.

Tal y como tenemos ahora parametrizados los orígenes si intentamos unir, por ejemplo, los registros de CA Sales y DE Sales nos saltará un Warning ya que el campo Zip son de diferente tipo en cada origen. En este caso simplemente cambiamos en DE Sales el tipo de campo en a String.



Ya podemos unir todos los orígenes con el componente de destino posicionando el puntero sobre uno de estos componentes y seleccionando la opción mostrada en la siguiente captura:



Una vez hecha las modificaciones, lanzamos nuestra primera transformación con F9 o la opción Run:



**Ejercicio**: Captura de pantalla del esquema de la transformación realizada con el check marcado como válido tras la ejecución.

|  |
| --- |
|  |

## Ejercicio 2 – Unión de tablas

En el siguiente ejercicio vamos a ver cómo realizar una join entre dos orígenes de datos distintos. Para ello, vamos a utilizar la salida del ejercicio anterior y vamos a cruzarlo con la información del archivo bi\_product.csv disponible en la carpeta Recursos\_ETL.

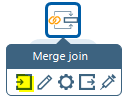
Creamos una nueva transformación y añadimos dos componentes CSV file input.

* Ventas: origen resultado del ejercicio anterior.
* Productos: origen bi\_product.csv

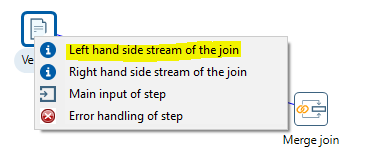
Indicamos como separador “;” en ambos orígenes.

Si con el preview todo es correcto, nos traemos un objeto de tipo Join  Merge Join.

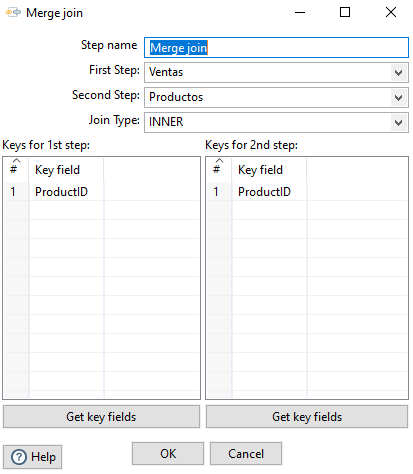
Con el componente ya dispuesto en nuestra área de trabajo nos posicionamos sobre el join y seleccionamos enlazar con origen:



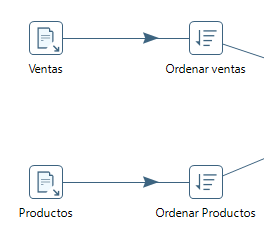
Unimos con el origen ventas e indicamos que será Left hand:



Hacemos lo propio con el otro origen pero en este caso indicando que será Right hand. Completamos este paso y editamos el componente. Con GetFields nos traemos los campos de cada colección, quedándonos con los que vamos a hacer la join:



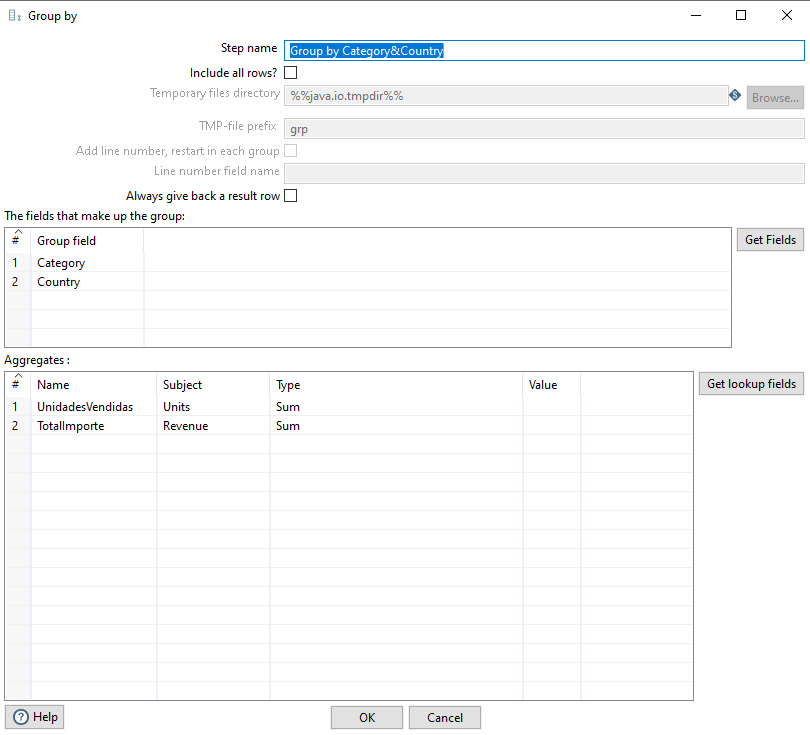
Si le damos a ok nos dirá que es necesario que las entradas estén ordenadas. Para ello, añadimos dos componentes: Transform  Sort rows, conectamos con cada origen respectivamente y en sus propiedades de edición le damos a Get Fields.



Ahora vamos a agrupar por País y Categoría, para ello incorporamos un componente de tipo Statistics  Group by a nuestra ETL.

Para este tipo de componentes también es necesario ordenar las entradas, por lo que la forma de proceder es la misma que como se vio en la join anterior pero en esta ocasión, en el sort solo dejaremos los mismos campos que se utilizan en la agregación.

Volvemos a nuestro componente de Group By e indicamos los campos por los que vamos a agrupar, en este caso Country y category. En Aggregates, ponemos los campos que vamos a agregar junto con el tipo de agregación, en este caso Suma.



Una vez completado, ejecutamos preview para ver si es correcto.

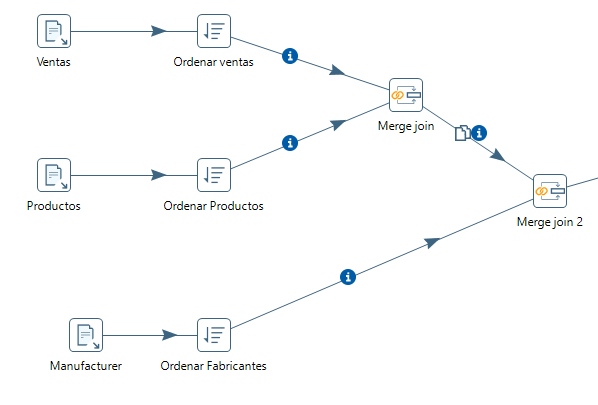
Por último, añadimos un Text file Output como final de nuestra ETL y le damos a F9 o run.

**Ejercicio**: Captura de pantalla del esquema de la transformación realizada con el check marcado como válido tras la ejecución.

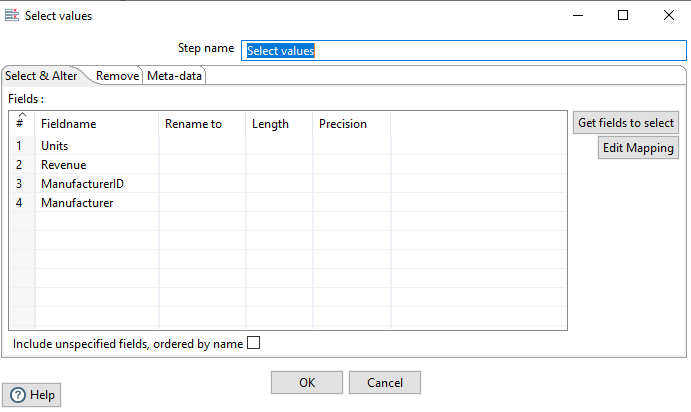
|  |
| --- |
|  |

## Ejercicio 3 – Fórmulas

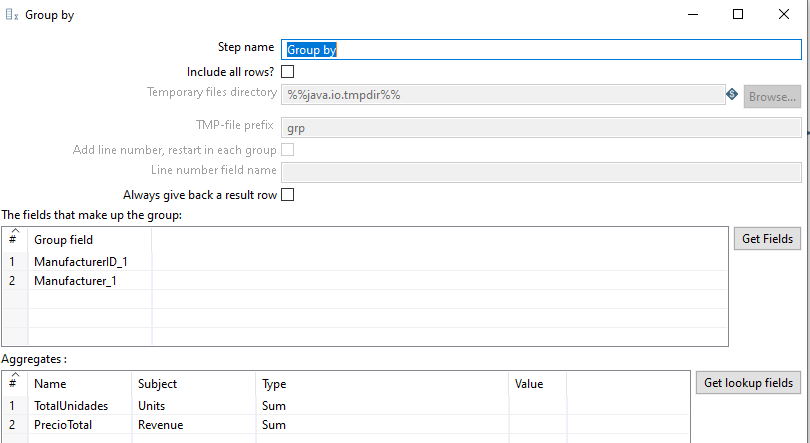
Partiendo de la base del ejercicio anterior añadimos un nuevo origen bi\_manufacturer.csv que unimos con la salida del join que une ventas y productos, haciéndolo por el campo ManufaturerID.



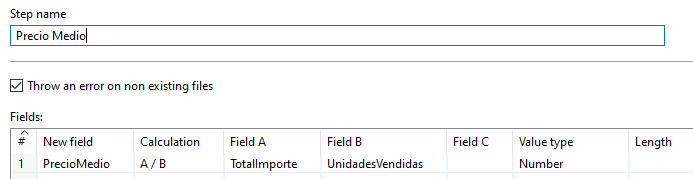
Para conocer un nuevo componente, añadimos TransformSelect Values que nos va a permitir quedarnos con aquella información que realmente consideramos importante y que colocaremos después de la nueva join. Nos quedamos con los siguientes campos:



A continuación del componente Select Values, añadimos un nuevo componente del tipo group by (recordad que en este tipo de componentes es necesario una ordenación previa) y lo configuramos como se muestra en la siguiente captura:



Después del Group By añadimos un componente Transform  Calculator:



Este componente nos va a permitir realizar operaciones sobre nuestros datos, a través de la columna Calculation la cual nos ofrece un amplio conjunto de operaciones que escoger. En este caso seleccionamo: A/B donde A será el importe Total y B el número de unidades vendidas, simplemente queremos calcular el precio medio de cada unidad vendida para cada fabricante. Con preview vemos que todo es correcto.

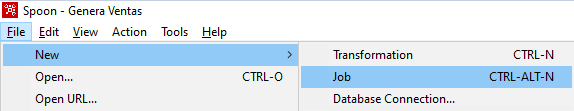
Finalmente añadimos un componente de salida Text file Output. Ejecutamos con F9 o run.

**Ejercicio**: Captura de pantalla del esquema de la transformación realizada con el check marcado como válido tras la ejecución.

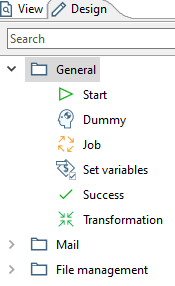
|  |
| --- |
|  |

## Ejercicio 4 – Generación de Jobs

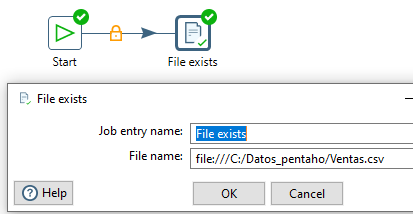
Por último, vamos a crear un job que nos va a permitir coordinar la ejecución de las tres transformaciones realizadas anteriormente a través de un solo proceso. Para ello, nos vamos a File y seleccionamos New  Job:



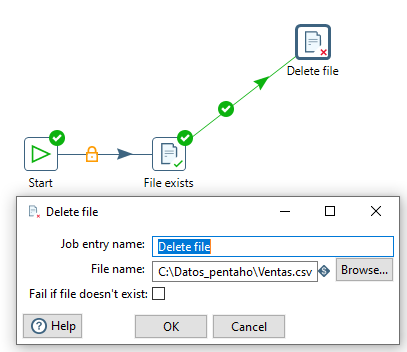
Si nos fijamos ahora en el área de Design, veremos que aparecen nuevos elementos con los que diseñar nuestro Job.



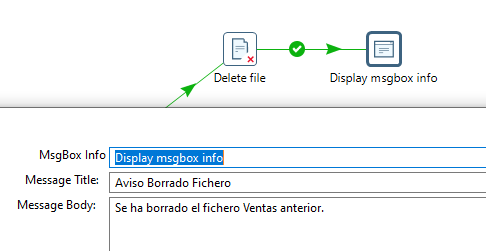
El primer componente a incorporar será “Start” que dará origen a nuestro flujo. El siguiente componente será File exists que buscará en el directorio donde guarda el resultado nuestra primera transformación para el informe de ventas. (\*) Es importante que esta transformación no tenga en el elemento Text file output la opción *Include date in filename?* Chequeada para que nos mantenga el nombre.



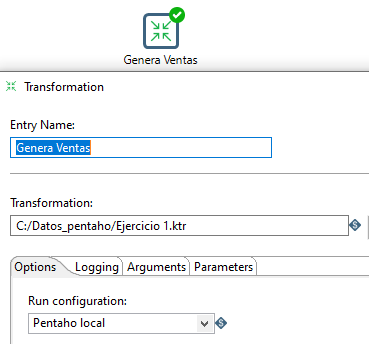
Añadimos un componente Delete Files, para que en el caso de que el fichero exista lo borremos.



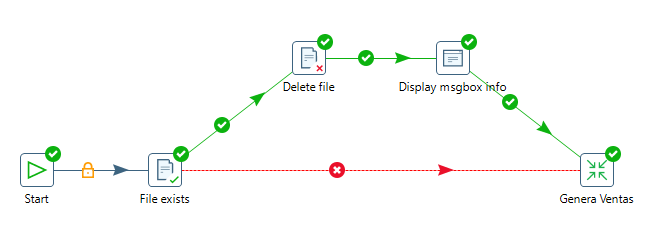
Añadimos un elemento Display msgbox info para avisarnos de que ha borrado el fichero de ventas que existía anteriormente.



A continuación, añadimos nuestra primera transformación, la que hicimos en el ejercicio 1, a través del elemento Transformation:

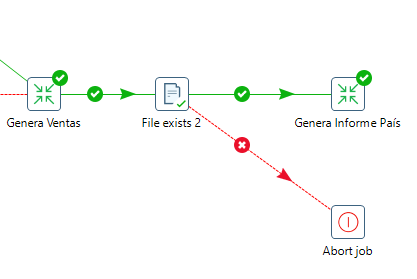


Y unimos tanto la salida del Display msgbox info como la de File exists con nuestra transformación.

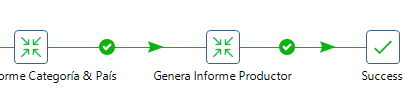


Ahora vamos a añadir nuestras otras dos transformaciones, pero como estas dependen del Informe de ventas que genera la primera transformación añadimos un componente File exists, para que en el caso de que no exista Aborte el job. Para esto último utilizamos el componente Abort Job. Por tanto, en este punto realizamos los siguientes pasos:

1. Añadir File exists y comprobar que existe Ventas.csv
   1. Unir la salida exitosa con un nuevo componente de transformación para el trabajo que realizamos en el ejercicio 2: Generar Informe Categoría & País.
   2. Unir la salida errónea con Abortar Job



Por último añadimos un nuevo componente para la transformación que contendrá el ejercicio 3 y la salida de éste lo unimos con el componente Success para indicar que todo ha ido bien.



**Ejercicio**: Captura de pantalla del esquema del job realizado con el check marcado como válido tras la ejecución.

|  |
| --- |
|  |

Para hacer el seguimiento de esta práctica enviar la memoria a [clopez@teralco.com](mailto:clopez@teralco.com)