Guía Básica de Instalación y Uso de la Herramienta de Procesos de Extracción Transformación y Carga (ETLs) Pentaho Data Integration 8

Autores: Alejandro Maté Morga

Juan Carlos Trujillo Mondéjar

Fecha: Febrero 2020

Universitat d'Alacant
Universidad de Alicante

1. Introducción

La herramienta Pentaho Data Integration (PDI) Community Edition es una herramienta de código abierto enfocada al desarrollo y ejecución de procesos de Extracción, Transformación, y Carga de datos (ETL). Esta herramienta es una de las principales fortalezas del conjunto de herramientas de Pentaho. PDI es una herramienta reconocida por múltiples especialistas como una de las mejores herramientas disponibles para la preparación, manipulación y carga de datos en el mercado, superando incluso en prestaciones a herramientas de pago ofrecidas por la competencia.

PDI funciona como un creador de flujos de datos, en los que los datos son leídos a partir de un número indeterminado de fuentes, conectados a través de uno o varios pasos de transformación, y cargados en uno o varios destinos (tablas, archivos, bases de datos documentales, etc.). Todos los datos que pasan a través de los flujos de datos son estructurados en registros, lo que permite al diseñador cuidadoso optimizar el rendimiento de los procesos.

Sin embargo, el hecho de que los datos se traten como registros no debe hacernos suponer que PDI es sólo una herramienta para trabajar con datos tradicionales. Su amplio conjunto de conectores, pasos de lectura, transformación y escritura le permiten enfrentarse tanto a conjuntos de datos empresariales típicos como a datos procedentes de fuentes de Big Data. Además, como refuerzo a esta amplia conectividad, desde las últimas versiones, PDI incluye la posibilidad de desplegar los procesos ETL sobre clusters de Apache Spark. No obstante, este proceso requiere de cierto conocimiento detallado tanto de PDI como de Spark, por lo que queda fuera de esta guía.

2. Descarga e instalación de PDI

Para comenzar es necesario tener instalado y configurado Java 1.8 o superior. A continuación, puede descargarse el software Pentaho Data Integration 8.2 (Community Edition) en la siguiente dirección:

https://sourceforge.net/projects/pentaho/files/latest/download?source=files

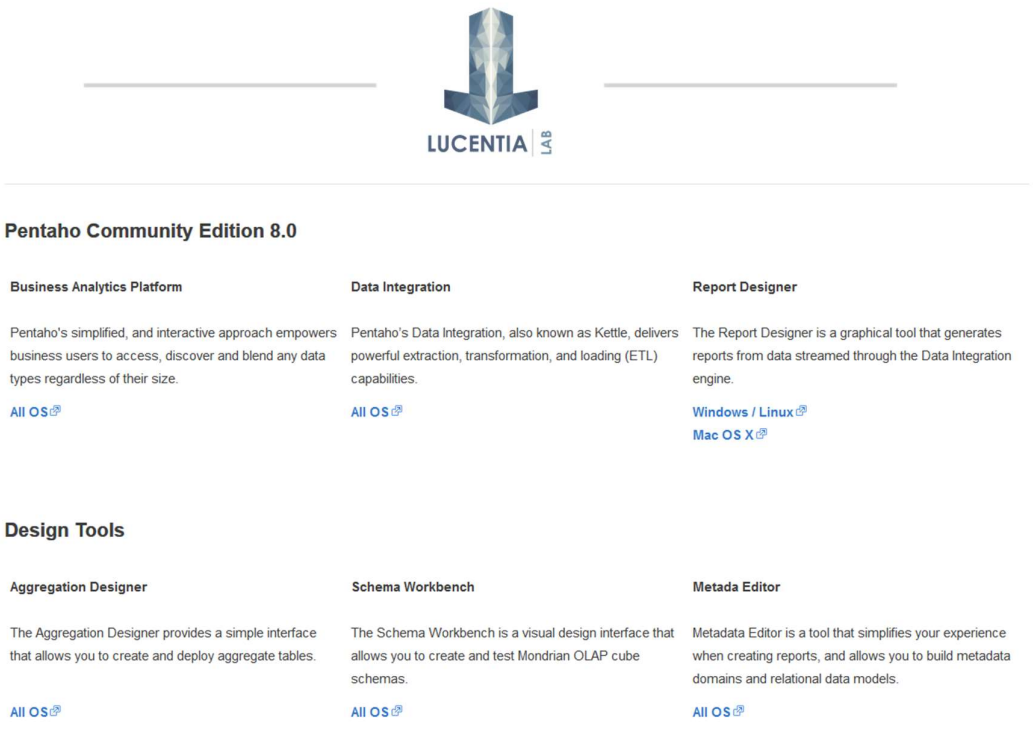


Figura 1. Repositorio online para la descarga de Pentaho Data Integration 8.0

Al extraer el archivo, obtendremos el directorio **/data-integration** cuyo contenido podemos ver a continuación:

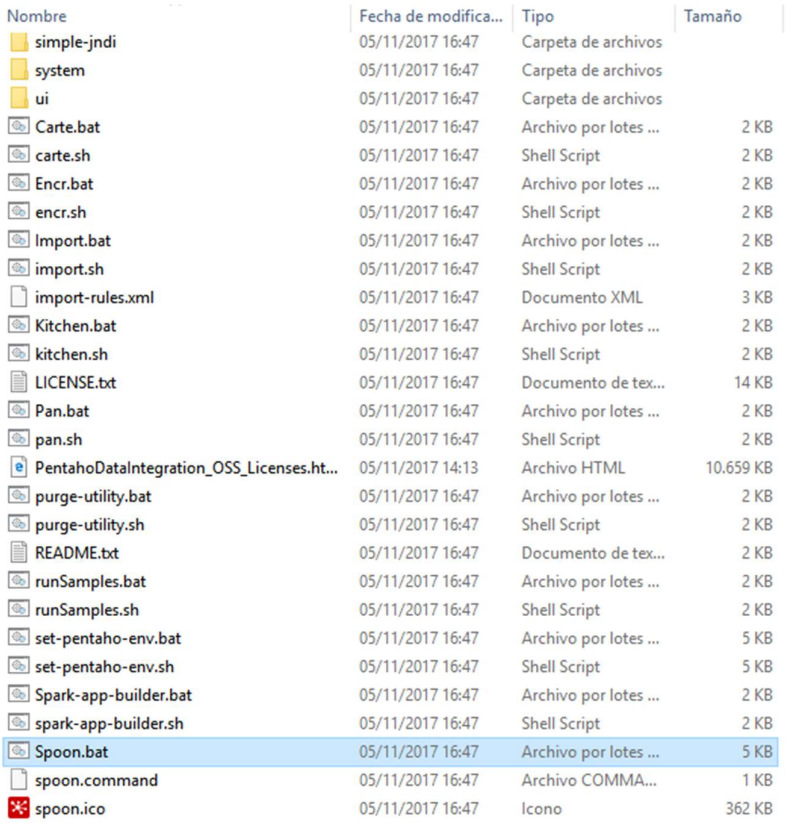


Figura 2. Directorio data-integration

En esta carpeta se encuentran todos los archivos de Pentaho Data Integration incluyendo, como veremos posteriormente, varios flujos de ejemplo de los pasos que se pueden utilizar en la herramienta para el tratamiento de datos.

Es importante tener en cuenta que, al igual que ocurre con Pentaho Server, ha de estar definida la variable de entorno **JAVA_HOME** la cual debe indicar la ruta de la instalación de Java. Pentaho ejecuta por defecto un script que busca la ruta y define dicha variable. No obstante, en algún caso podría no funcionar e impedir de esta forma el correcto arranque del servidor. En ese caso, definir dicha variable de forma manual o usando un script como "for /d %%i in ("\Program Files\Java\jdk*") do set JAVA_HOME=%%i" para asignar el
JAVA_HOME a la versión más reciente de java.

Los archivos más importantes (.bat para Windows /.sh para Linux) a tener en cuenta dentro de la carpeta de Pentaho Data Integration, son los siguientes:

- **Spoon:** Es el archivo principal para la ejecución de la interfaz gráfica de PDI. Es el archivo más comúnmente utilizado debido a la facilidad que proporciona para crear y ejecutar flujos de datos y a que PDI no está pensado para utilizarse de manera programática.
- **Pan:** Pan es un archivo diseñado para la ejecución de transformaciones por terminal. Pan toma como parámetro una transformación de PDI y la ejecuta por línea de comandos, utilizando por defecto el terminal como salida de log hasta que la transformación termina. En Linux se puede redirigir la salida de log a un archivo con “>” y ejecutar en segundo plano con el parámetro de consola “&”.
- **Kitchen:** Kitchen es similar a Pan pero en este caso es un archivo diseñado para la ejecución de trabajos o “jobs” por línea de comandos. Por lo demás su funcionalidad es idéntica a Pan.
- **Carte:** Carte es un servidor de ejecución de transformaciones o trabajos como servicio. El objetivo de Carte es posibilitar la ejecución en una máquina remota de una transformación o trabajo. Esto permite utilizar nuestra máquina como centro de diseño pero ejecutar la transformación en un servidor más potente, o simplemente en un servidor sin interfaz gráfica. Para poder hacer un seguimiento de las transformaciones, Carte levanta también una página web accesible al usuario que provee la información de log de la transformación o trabajo.

Posiblemente, el desarrollador atento esté preguntándose en este momento ¿Para qué quiero un programa de ejecución por línea de comandos en un entorno orientado al diseño y ejecución de ETLs mediante interfaz gráfica? La respuesta es que si bien la interfaz gráfica de PDI es sencilla y amigable, su estabilidad ante elevados volúmenes de datos no lo es tanto. Basta con ejecutar alguna transformación compleja que supere el orden de los millones de registros a procesar y comenzaremos a tener problemas con la aplicación gráfica. Si llegamos al orden de los miles de millones de registros que podemos encontrar en Big Data, nos encontraremos con que, indistintamente de cuánta memoria asignemos a Spoon, la interfaz gráfica y la ejecución del proceso morirán sin remedio antes de terminar con éxito. En estos casos, lo mejor es diseñar con Spoon el proceso y, una vez probado, ejecutarlo mediante Kitchen/Pan.

3. Canvas de desarrollo

A continuación se explicarán algunos de los aspectos básicos para comenzar a utilizar Pentaho Data Integration. Para poder acceder a la interfaz gráfica de diseño deberemos ejecutar **Spoon.bat,** tal y como se ha indicado anteriormente. Al abrir Spoon, tras unos momentos, veremos una interfaz como la que se muestra en la Figura 3.

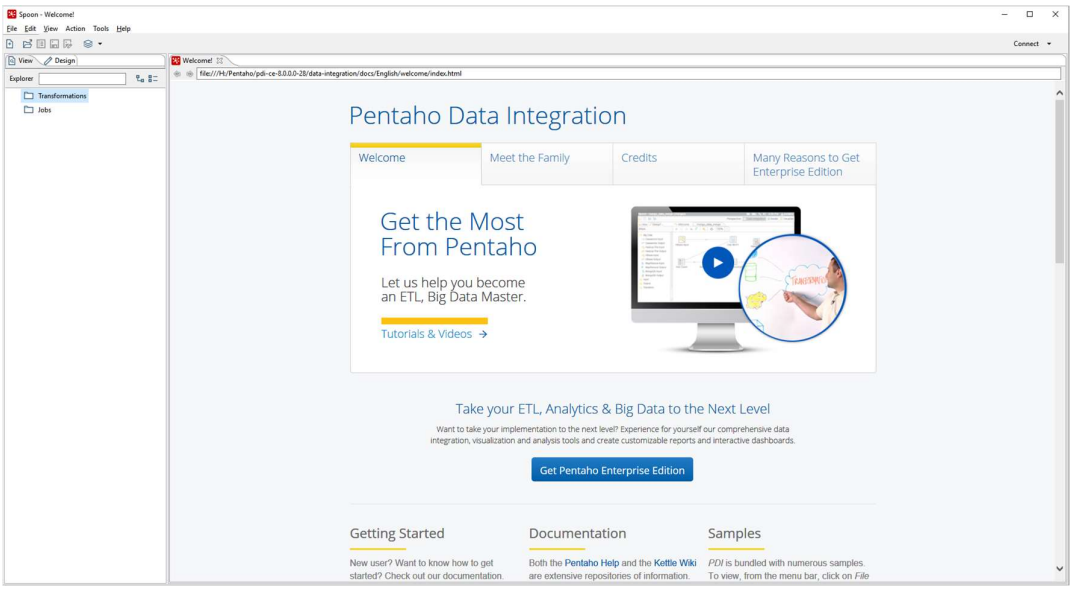


Figura 3. Arranque de Pentaho Server

Para comenzar a trabajar basta con que seleccionemos la opción File->New-> Transformation, y podremos comenzar a diseñar nuestra transformación.

Cuando creamos una transformación nueva, aparecerá una pantalla como la mostrada en la Figura 4, en la que se nos indica que arrastremos al área de diseño elementos procedentes de la paleta de pasos.

Los pasos se encuentran organizados en carpetas temáticas en PDI, siendo las de uso más habitual aquellas de Entrada (Input), Salida (Output), Unión (Joins), y Transformación (Transformation). La potencia de extracción de información se hace evidente en PDI simplemente con abrir la carpeta de Input. En la Figura 5 podemos ver que, de base, PDI soporta decenas de fuentes posibles de datos, incluyendo la generación de valores aleatorios en caso de que necesitemos simular flujos de datos de distintos volúmenes o incluso infinitos.

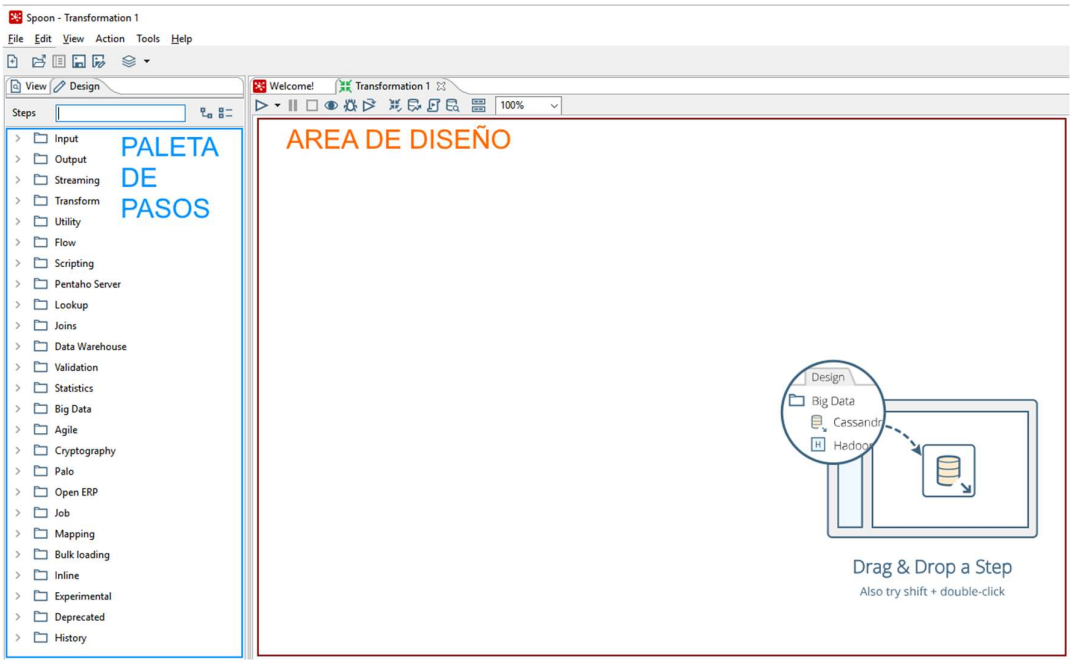


Figura 4. Área de trabajo de las transformaciones en PDI

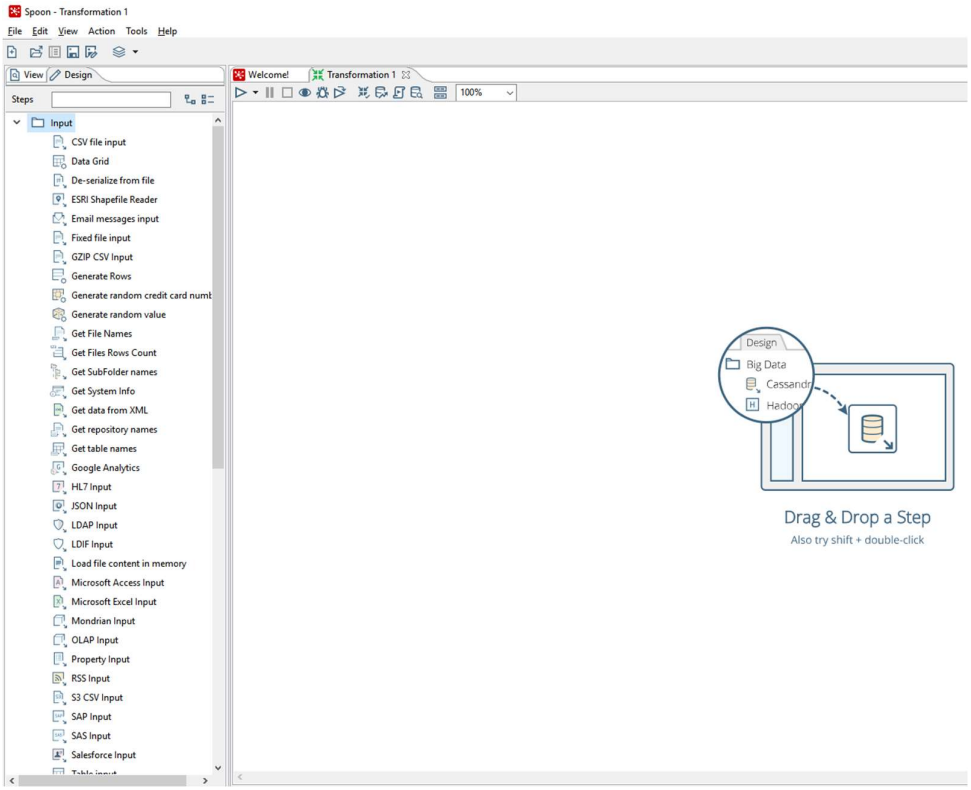


Figura 5. Pasos para la lectura de fuentes de datos en PDI

Para ejemplificar un proceso típico de manipulación de datos, esta guía de introducción vamos a utilizar el archivo de llamadas al departamento de bomberos de San Francisco[1]. Este conjunto de datos es un conjunto de un volumen relativamente elevado (1,5 GB y más de 4 millones de entradas).

El proceso que vamos a realizar va a ser el siguiente: Vamos a extraer del conjunto de datos, sin información a priori de otras fuentes, cuales son los distintos vecindarios que se encuentran en el fichero y el número de llamadas que contiene cada uno de ellos. Una vez obtenidos estos datos los guardaremos como archivos CSV de salida.

Para comenzar, vamos a leer el archivo de llamadas. El archivo es un CSV, por lo que utilizaremos el paso CSV Input. Primero, arrastramos el paso CSV Input desde la paleta de pasos al área de diseño. A continuación, hacemos doble clic sobre él para abrir la ventana de configuración. En la Figura 6 podemos ver la ventana de configuración del paso CSV.

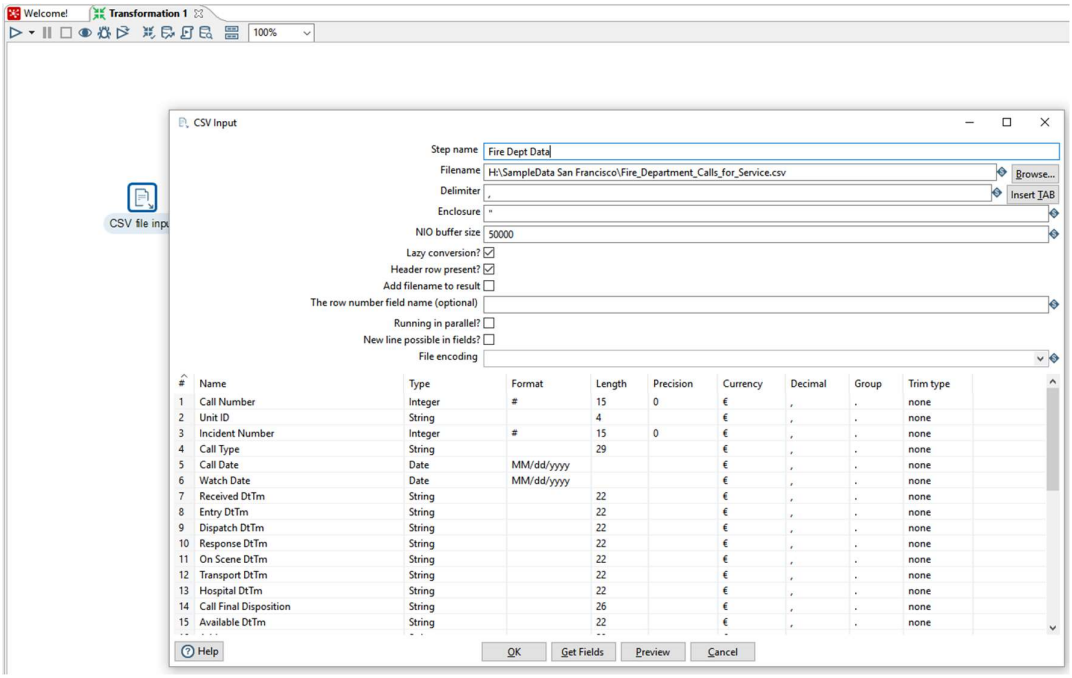


Figura 6. Lectura de datos de CSV

Para poder leer los datos del archivo, primero debemos seleccionar el archivo del cual queremos leer. La selección del archivo se realiza mediante el botón Browse…, que se encuentra a la derecha del nombre del archivo.

Una vez seleccionado el archivo, debemos analizar cuál es su contenido para identificar las columnas y los tipos de datos de las mismas. PDI realiza esta tarea por nosotros haciendo un muestreo limitado de los datos. Para ello, hacemos clic en el botón Get

[1] https://data.sfgov.org/Public-Safety/Fire-Department-Calls-for-Service/nuek-vuh3/data

Fields, localizado en la parte inferior de la ventana, e indicamos el número de filas a analizar. Cuantas más filas analicemos más exactos serán los resultados y peor el tiempo de ejecución. Si estamos satisfechos con los resultados presionamos OK y tendremos configurado el paso de entrada de datos.

Dado que únicamente queremos analizar en este ejemplo los datos referentes a los distritos y al número de llamadas, vamos a reducir cuanto antes nuestro conjunto de datos. Para ello, utilizaremos el paso Select values, que permite que alteremos la estructura de nuestros datos para quedarnos sólo con parte de ellos o alterar su tipo de datos. Igual que en el caso anterior, arrastramos el paso al área de diseño. Una vez añadido el paso debemos conectarlo con el paso anterior para que los metadatos (la información referente a qué columnas tenemos para poder operar) puedan ser leídos por nuestro paso Select values.

Para conectar dos pasos, basta con mantener el botón Shift de teclado mientras se selecciona y arrastra desde el paso origen hasta el paso destino. El resultado debería ser similar a la Figura 7.

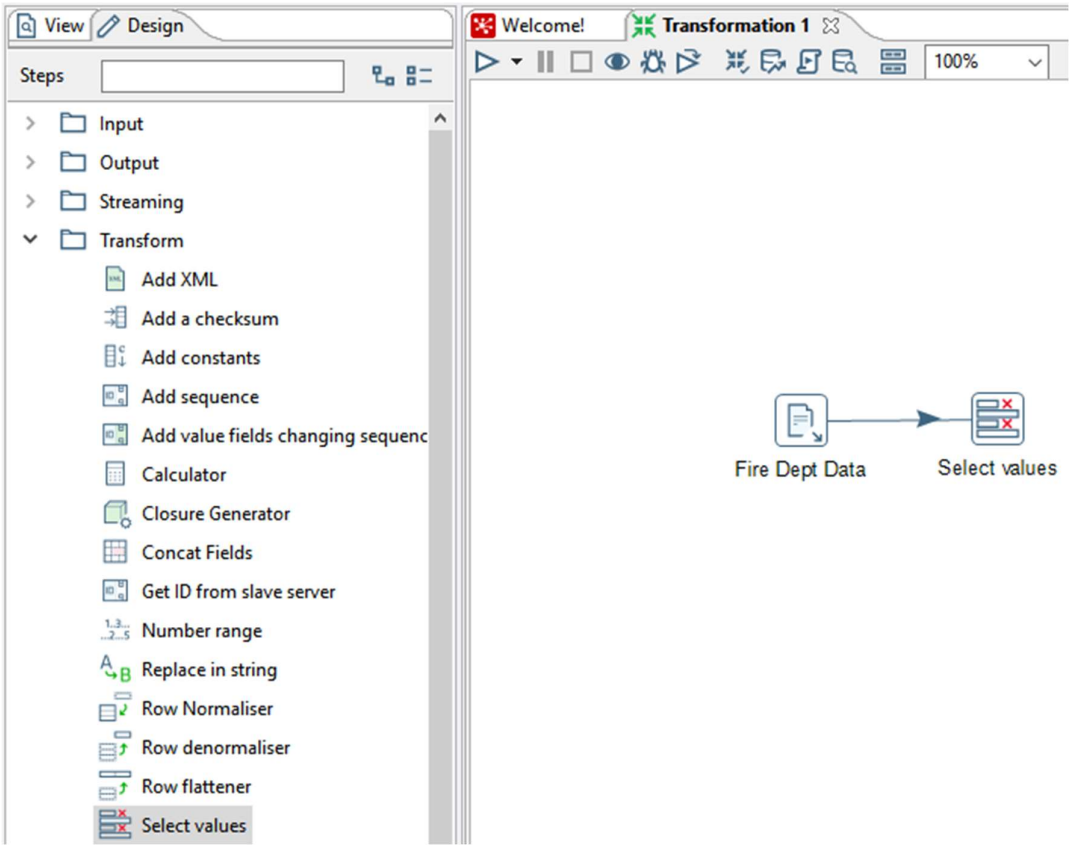


Figura 7. Conexión entre el paso de lectura CSV y el paso de Select values

Una vez conectados los pasos podemos configurar nuestro paso Select values. Al igual que en el caso anterior, basta con que hagamos doble clic sobre el mismo y utilicemos el botón Get fields to select, localizado en la parte derecha del diálogo que nos aparece.

El paso Select values permite operar de distintas formas con los campos disponibles. En nuestro caso, la forma más sencilla será seleccionar únicamente los datos que queremos, en este caso los correspondientes al vecindario, tal y como se muestra en la Figura 8. Seleccionar los vecindarios no eliminará los registros repetidos, simplemente hará que a partir de este momento el flujo sólo contenga las columnas correspondientes a los vecindarios, teniendo así más de 4 millones de registros con vecindarios repetidos, un respetando un registro por llamada.

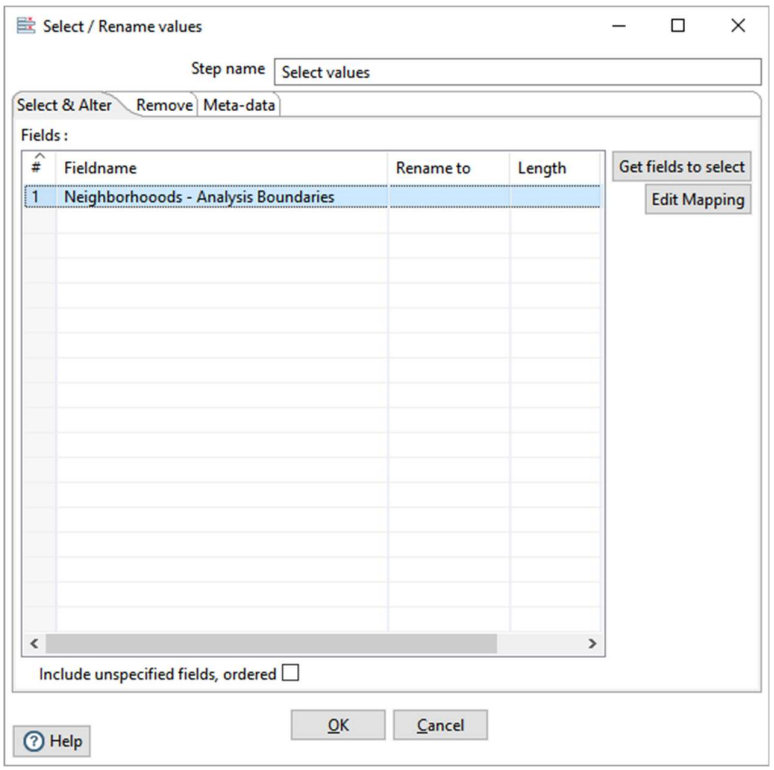


Figura 8. Selección de información de los vecindarios

Una vez tenemos la lista de registros de los vecindarios, vamos a calcular cuántas llamadas hay por vecindario. Para ello, vamos a utilizar primero el paso de Sort rows y después el paso de Group by.

El paso de Sort rows es necesario cada vez que se van a realizar operaciones que dependen del orden de los valores del flujo. Por razones técnicas y de optimización, PDI requiere que las filas estén ordenadas en la entrada de los pasos de Group by y Merge join entre otros. No es necesario conocer de memoria qué pasos requieren de ordenación, pues PDI avisa al usuario si el paso que se va a utilizar asume que la entrada está ordenada.

Repetiremos la misma operación que antes, añadiendo primero el paso Sort rows y conectándolo con nuestro paso Select values. La configuración del paso Sort rows se muestra en la Figura 9.

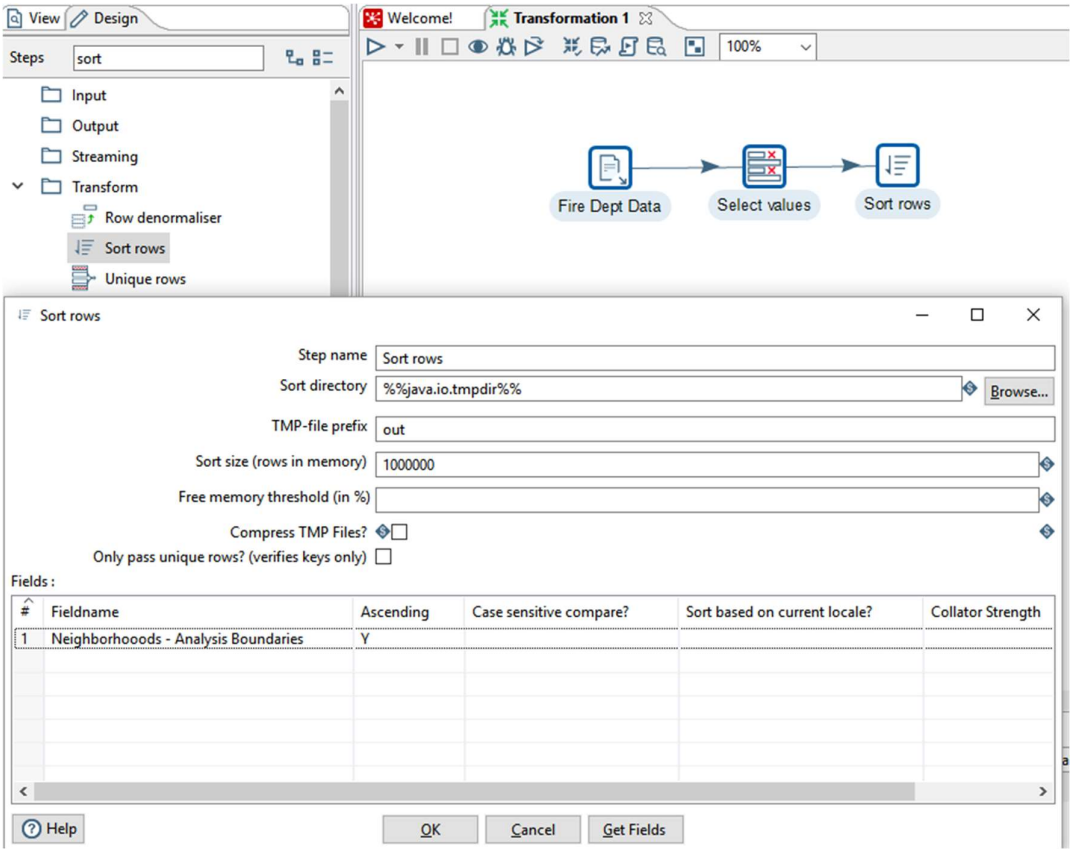


Figura 9. Ordenación de filas de datos de vecindarios

Una vez tenemos los datos ordenados, únicamente necesitamos calcular el número de llamadas por vecindario. Este paso nos dará como resultado a la vez una lista de valores de vecindarios únicos junto con su total de llamadas por cada uno. Para ello, añadimos un paso Group by y lo conectamos al paso Sort rows.

La configuración del paso Group by puede resultar un poco confusa para los desarrolladores nóveles. En la parte superior se deben colocar los campos que corresponden con la agregación que se quiere realizar. En nuestro caso aquí colocaremos los vecindarios. Cualquier campo de entrada no incluido en este apartado será eliminado de la salida.

Por otra parte, en la zona inferior se deben de indicar los cálculos a realizar. Aquí la primera columna representa los nuevos campos a crear, la segunda columna el campo sobre el que realizar operaciones (para sumas, medias, etc.), y la tercera columna el tipo de operación a realizar. Para nuestros cálculos configuraremos el paso Group by tal y como se muestra en la Figura 10.

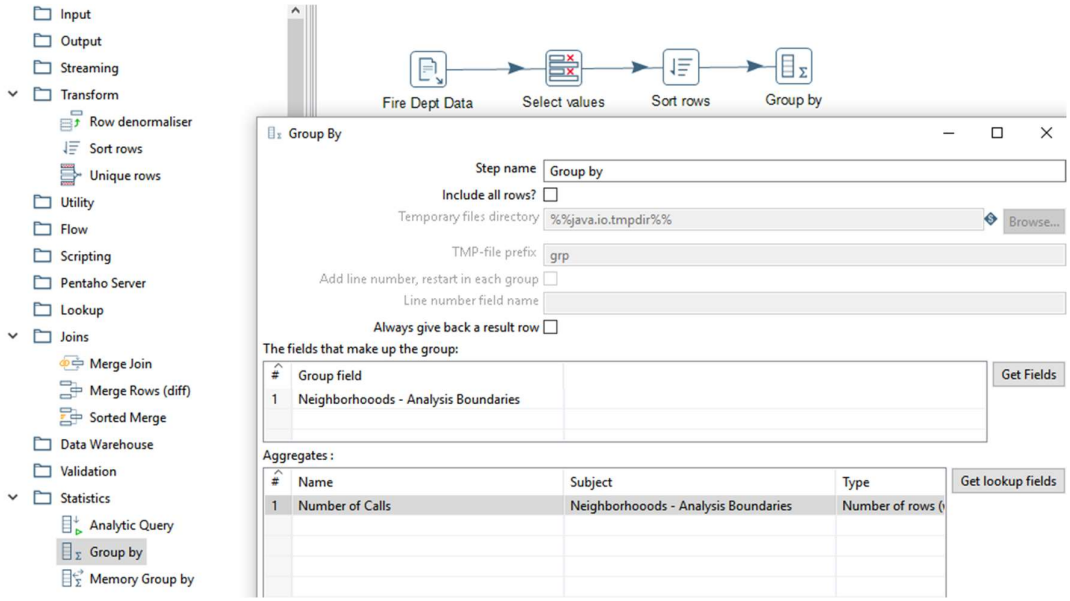


Figura 10. Cálculo de llamadas por vecindario

Para terminar nuestra transformación guardaremos los resultados que hemos obtenido en un archivo de salida, que llamaremos FireDeptResults.csv. La gestión de archivos de salida se lleva a cabo mediante el paso Text File Output. Los pasos Text File Output permiten guardar en formato de texto o CSV e indicar distintos elementos separadores para separar la información de las distintas columnas. La configuración del paso Text File Output puede verse en las Figuras 11 y 12. De forma similar al paso Text File Input, el paso Text File Output requiere que recuperemos los campos del flujo, pulsando el botón Get Fields en la pestaña Fields. Una vez realizada esta operación podremos configurar los metadatos de los campos a guardar por si queremos modificarlos.

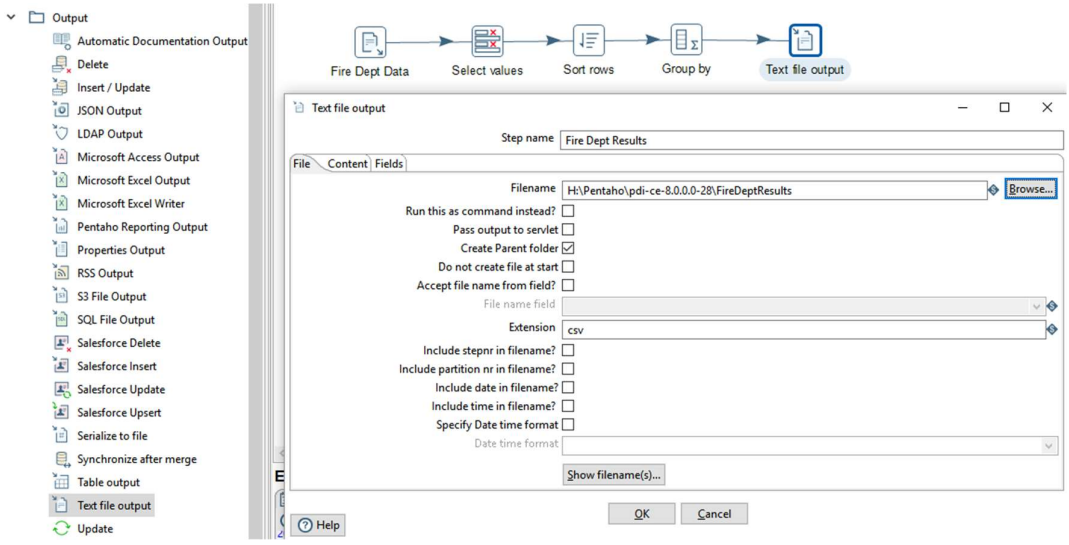


Figura 11. Guardado de resultados en fichero CSV (1)

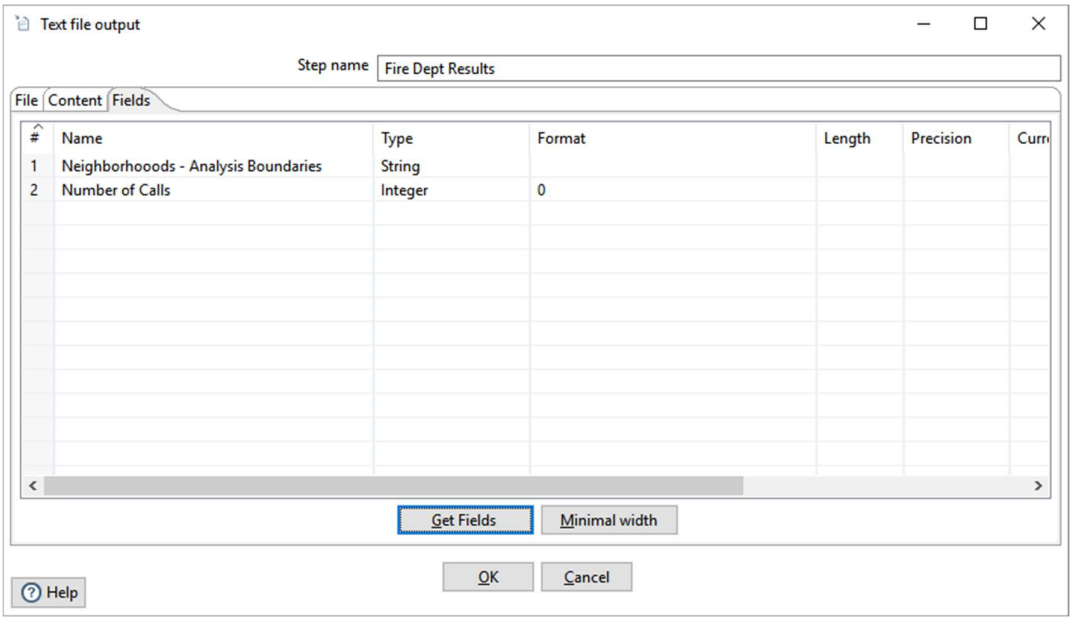


Figura 12. Guardado de resultados en fichero CSV (2)

Si estamos satisfechos con nuestra transformación, podemos proceder a probarla utilizando el botón de flecha "Play" en la parte superior de la interfaz, tal y como se muestra en la Figura 13.

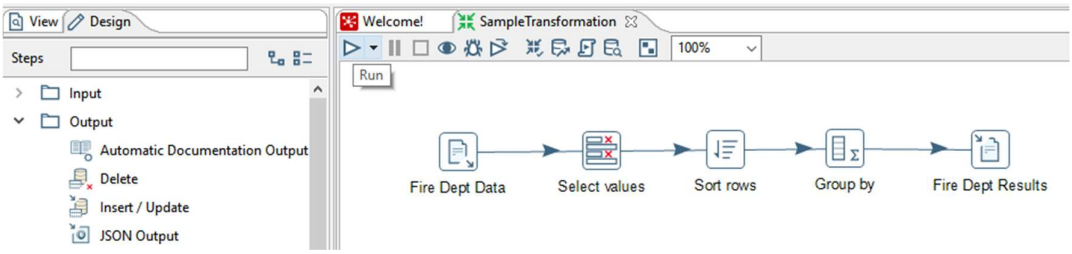


Figura 13. Ejecución de la transformación

Al ejecutar la transformación podremos ver la velocidad de procesamiento de las filas en cada paso, aunque tendremos que tener en cuenta que un paso puede ralentizarse por los pasos posterior o anterior al mismo ya que el buffer de datos es limitado.

Si todo ha ido bien tendremos un resultado similar al mostrado en la Figura 14.

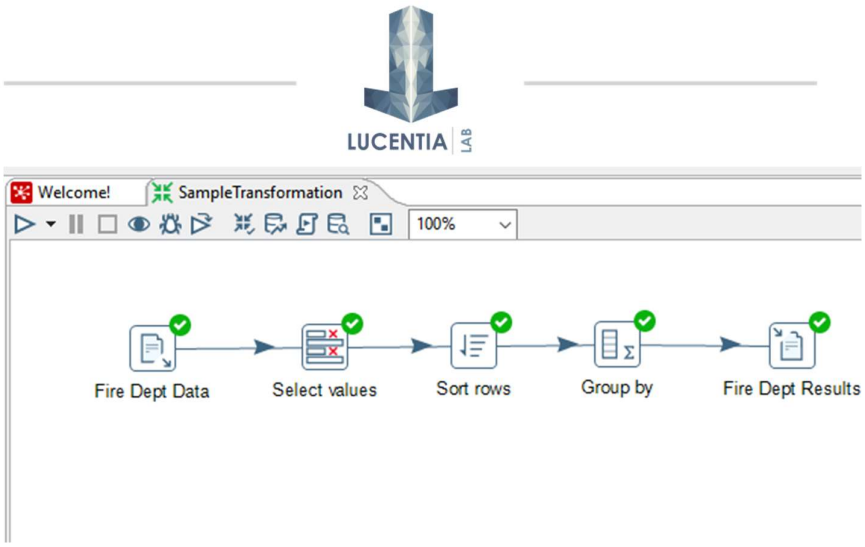
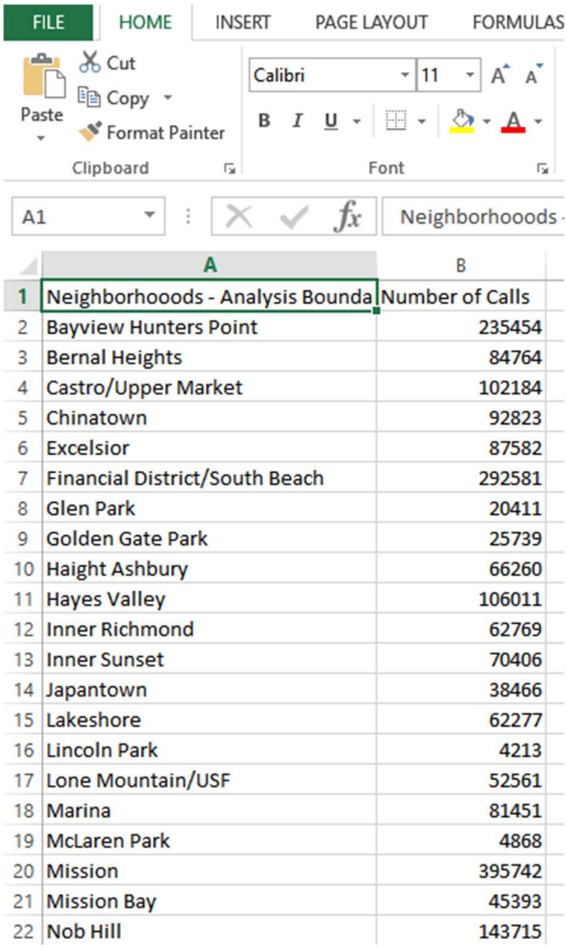


Figura 14. Éxito en la ejecución de la transformación

Al abrir el fichero CSV con Excel o un programa similar (Google Sheets, Open Office, etc.) podremos ver los resultados obtenidos, tal y como se muestran en la Figura 15. Acabamos de procesar más de 4 millones de filas de forma automática, pudiendo repetir el proceso con datos nuevos tantas veces como queramos, siempre que sigan el mismo formato.



	A	B
1	Neighborhooods - Analysis Bounda	Number of Calls
2	Bayview Hunters Point	235454
3	Bernal Heights	84764
4	Castro/Upper Market	102184
5	Chinatown	92823
6	Excelsior	87582
7	Financial District/South Beach	292581
8	Glen Park	20411
9	Golden Gate Park	25739
10	Haight Ashbury	66260
11	Hayes Valley	106011
12	Inner Richmond	62769
13	Inner Sunset	70406
14	Japantown	38466
15	Lakeshore	62277
16	Lincoln Park	4213
17	Lone Mountain/USF	52561
18	Marina	81451
19	McLaren Park	4868
20	Mission	395742
21	Mission Bay	45393
22	Nob Hill	143715

Figura 15. Extracto de los resultados obtenidos

Con esta transformación concluimos nuestro ejemplo de introducción a Pentaho Data Integration. A continuación veremos algunos aspectos que permitirán profundizar en el uso habitual de PDI para la transformación de datos de forma general.

a. Transformaciones y Trabajos

Hasta el momento hemos visto las transformaciones de Pentaho Data Integration, que están compuestas por pasos que nos permiten llevar a cabo procesos ETL. El concepto de Trabajos en PDI resulta de organizar jerárquicamente la ejecución de procesos ETL. Los trabajos son "transformaciones de transformaciones". Un trabajo no contiene pasos de lectura, manipulación de datos, o escritura, sino que contiene un punto de comienzo (paso "Start") y un paso de finalización ("Success"). Entre estos pasos se encontrarán, o bien llamadas a transformaciones y otros trabajos que hemos diseñado previamente, o bien pasos auxiliares como envío de correos para notificar que ha habido problemas, gestionar directorios y archivos, eliminar temporales, etc. Podemos ver un ejemplo de un trabajo en la Figura 16.

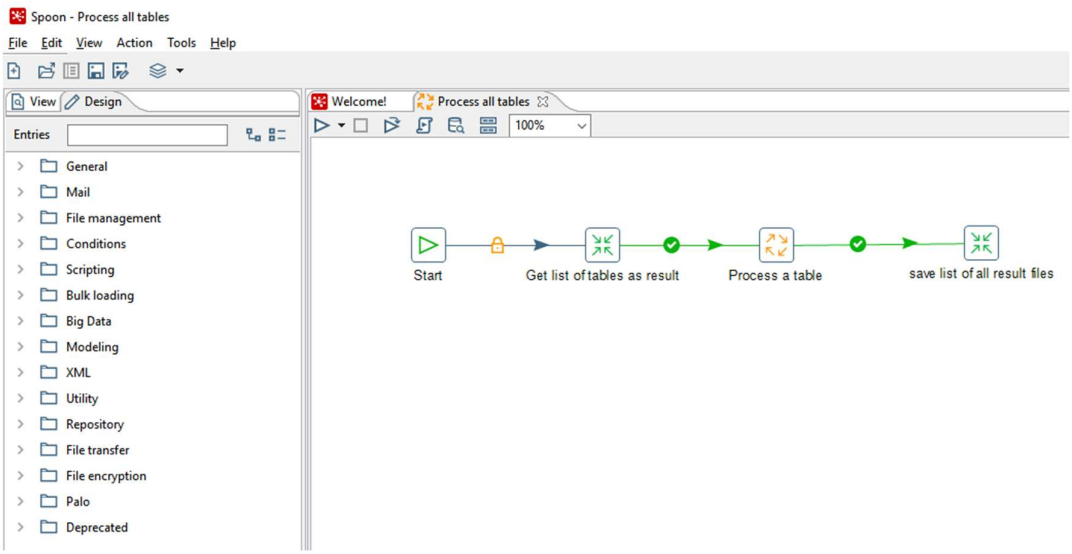


Figura 16. Ejemplo de un trabajo en PDI

b. Ejemplos

Dado el volumen de pasos incluidos (sin contar extensiones) en Pentaho Data Integration, resulta imposible cubrir detalladamente en una guía cada uno de ellos. Para poder buscar pasos que resulten adecuados para una variedad de situaciones a las que un ingeniero de datos debe hacer frente se puede o bien consultar la Wiki de PDI (http://wiki.pentaho.com/display/EAI/Latest+Pentaho+Data+Integration+%28aka+Kettl

e%29+Documentation) o bien abrir las transformaciones y trabajos contenidos en la carpeta "samples", dentro del directorio data-integration:

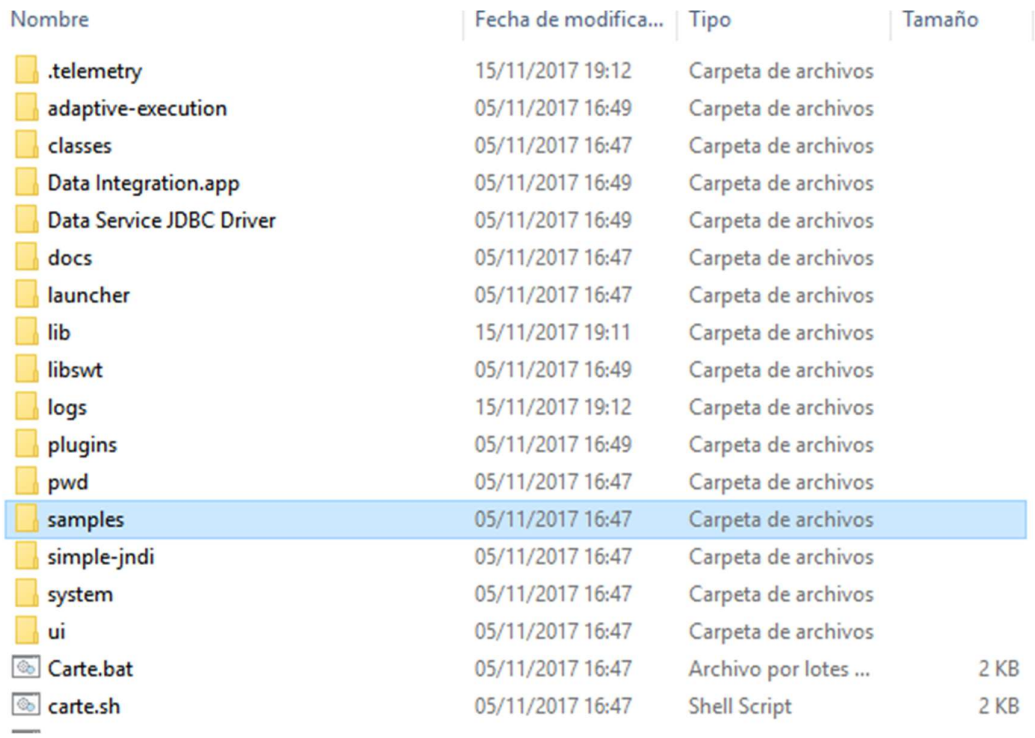


Figura 17. Carpeta samples con ejemplos variados de utilización de pasos de PDI

4. Más información

En esta guía hemos cubierto los aspectos fundamentales de Pentaho Data Integration. El diseño de las transformaciones y trabajos ocupa la amplia mayoría del trabajo de un ingeniero de datos dedicado a procesos ETL (en muchos casos el 95% del tiempo). No obstante, PDI incluye otras características que, si bien no se detallan en esta guía, pueden ser de interés.

Por un lado, PDI dispone de un Marketplace (Tools->Marketplace), de forma similar a Pentaho Server, donde se pueden descargar plugins y pasos adicionales que pueden simplificar el procesamiento de datos.

Por otro, PDI permite organizar las transformaciones y trabajos en repositorios, para facilitar su organización y evitar duplicados y errores en despliegues profesionales.

Finalmente, las características de conectividad con sistemas de ficheros distribuidos (Hadoop FS) requieren para la conexión la instalación de Hadoop Shims, que abstraen la problemática del cambio de versiones de Hadoop y permiten a PDI acceder a datos que se encuentren distribuidos en el sistema de ficheros distribuido.

Cabe destacar que existen además características adicionales en la versión Enterprise, como son el procesamiento de colas o el análisis gráfico directamente sobre los flujos de datos.

Todas estas características son aspectos avanzados, que requieren de un dominio previo de la herramienta. Por ello, se dejan para su profundización mediante otros recursos, si el lector en algún momento se dedica tan en profundidad a la preparación de datos con PDI como para necesitar de ellas.