

Referencia de Step de  
transformación

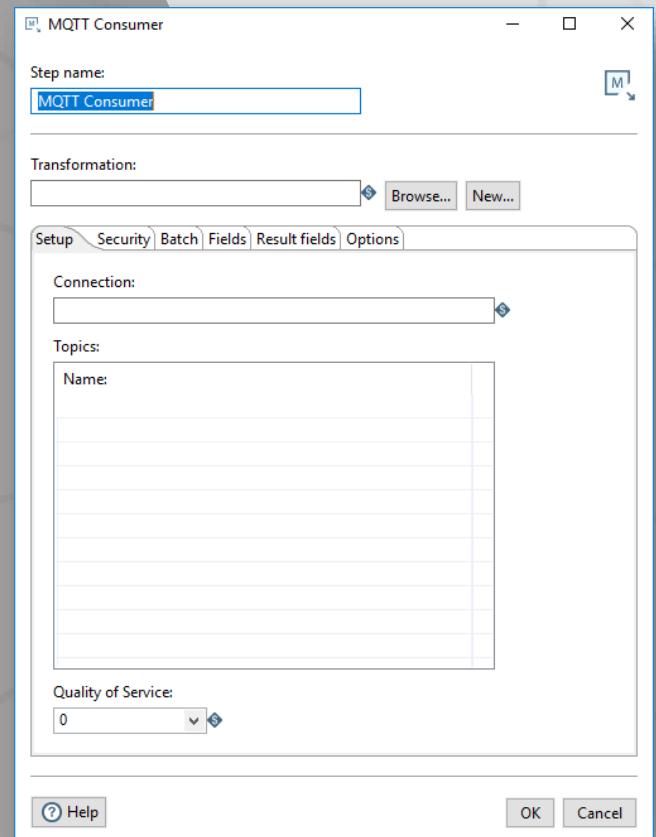


# MQTT Consumer

- El cliente PDI puede extraer datos de transmisión de un agente o clientes MQTT a través de una transformación MQTT. El paso principal de MQTT Consumer ejecuta una transformación secundaria que se ejecuta de acuerdo con el tamaño o la duración del lote del mensaje, lo que le permite procesar un flujo continuo de registros casi en tiempo real. La transformación secundaria debe comenzar con el paso Obtener registros de la secuencia.
- Además, desde el paso de MQTT Consumer, puede seleccionar un paso en la transformación secundaria para transmitir los registros a la transformación principal. Esta capacidad permite que los registros procesados por un paso de MQTT Consumer en una transformación principal se transfieran a otros pasos incluidos en la misma transformación principal.

# General

Opción	Descripción
Step name	Especifica el nombre único del paso en el lienzo. El nombre del paso se establece en "Consumidor MQTT" de forma predeterminada.
Transformation	<p>Especifique la transformación secundaria que se ejecutará realizando cualquiera de las siguientes acciones:</p> <ul style="list-style-type: none"> <li>• Ingresar la ruta.</li> <li>• Clic en <b>Browse</b> para seleccionar una transformación secundaria existente</li> <li>• Al hacer clic en <b>New</b> para crear y guardar una nueva transformación secundaria. (ver <a href="#">Create and Save a New Child Transformation</a> para más detalles).</li> </ul> <p><b>NOTA:</b> La transformación secundaria seleccionada debe comenzar con el paso <b>Get Records from Stream</b>.</p> <p>Si selecciona una transformación que tiene la misma ruta raíz que la transformación actual, la variable \${Internal.Entry.Current.Directory} se inserta automáticamente en lugar de la ruta raíz común. Por ejemplo, si la ruta de la transformación actual es: /home/admin/transformation.ktr y selecciona una transformación en el directorio /home/admin/path/sub.ktr, la ruta se convierte automáticamente a: \${Internal.Entry.Current.Directory}/path/sub.ktr</p> <p>Si está trabajando con un repositorio, debe especificar el nombre de la transformación. Si no está trabajando con un repositorio, debe especificar el nombre de archivo XML de la transformación. Las transformaciones previamente especificadas por referencia se convierten automáticamente para ser especificadas por el nombre de transformación en el repositorio de Pentaho.</p>



# Crea y guarda una nueva transformación hija

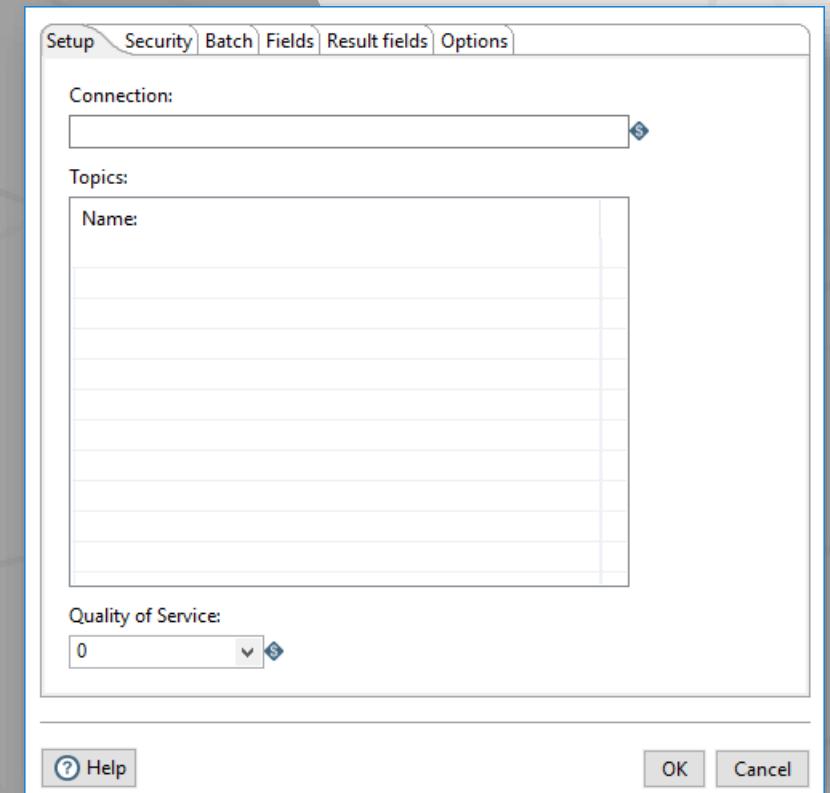
- Si aún no tiene una transformación secundaria, puede crear una al configurar el paso de MQTT Consumer. Al hacer clic en el botón Nuevo, una nueva transformación secundaria generará automáticamente el paso requerido Obtener registros de la transmisión en una nueva pestaña de lienzo. Todos sus campos y tipos se personalizan en el paso Obtener registros de la transmisión secundaria de la transformación secundaria para que coincidan con los campos y tipos especificados en la pestaña Campos del paso del consumidor principal de MQTT.

## Pasos

1. En el paso Consumidor MQTT, haga clic en **New**. Aparece el cuadro de diálogo **Save As**.
2. Navegue hasta la ubicación donde desea guardar su nueva transformación secundaria y luego escriba el nombre del archivo.
3. Clic en **Save**. Aparece un cuadro de notificación que le informa que la transformación secundaria se ha creado y abierto en una nueva pestaña. Si no desea ver esta notificación en el futuro, seleccione la casilla de verificación **Don't show me this again**.
4. Haga clic en la nueva pestaña de transformación para ver y editar la transformación secundaria. Contiene automáticamente el paso Get Records from Stream. Opcionalmente, puede continuar construyendo esta transformación y guardarla.
5. Cuando haya terminado, regrese al paso de MQTT Consumer.

# Pestaña Setup

Opción	Descripción
Connection	Especifique la dirección del servidor MQTT al que se conectará este paso para enviar o recuperar mensajes.
Topics	Especifique los temas de MQTT a los que se suscribirá.
Quality of Service (QoS)	La calidad de servicio (QoS) es un nivel de garantía para la entrega de mensajes. Selecciona una de las siguientes opciones. <ul style="list-style-type: none"> <li>• Como máximo una vez (0) - Predeterminado</li> <li>• Al menos una vez (1)</li> <li>• Exactamente una vez (2)</li> </ul>

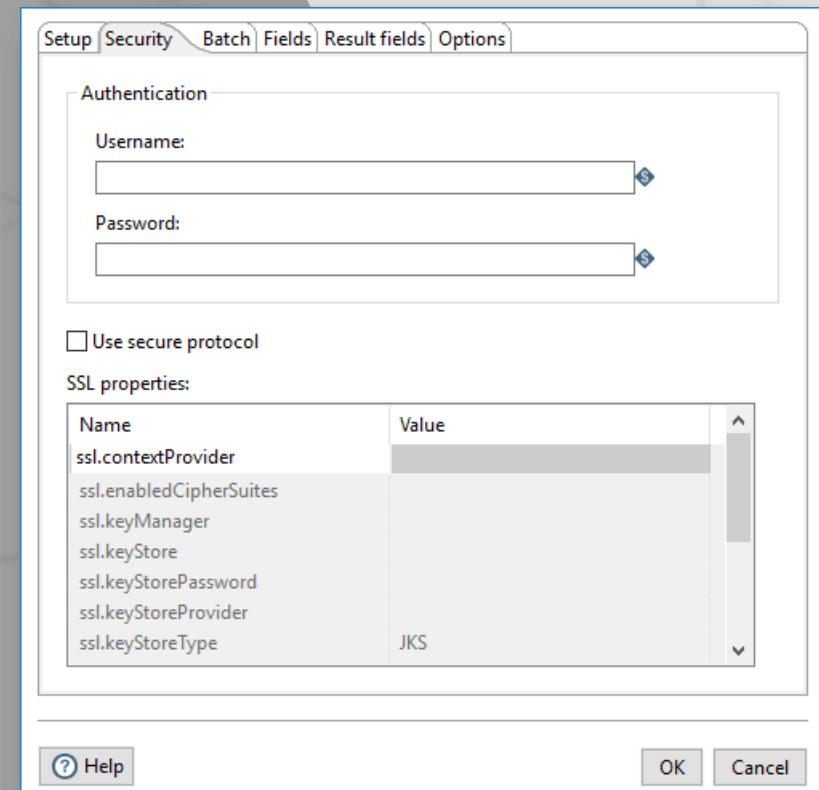


# Pestaña Security

Opción	Descripción
Username	Especifique el nombre de usuario requerido para acceder al servidor MQTT.
Password	Especifique la contraseña asociada con el nombre de usuario.
Use secure protocol	Seleccione esta opción si desea definir las propiedades SSL para la conexión.

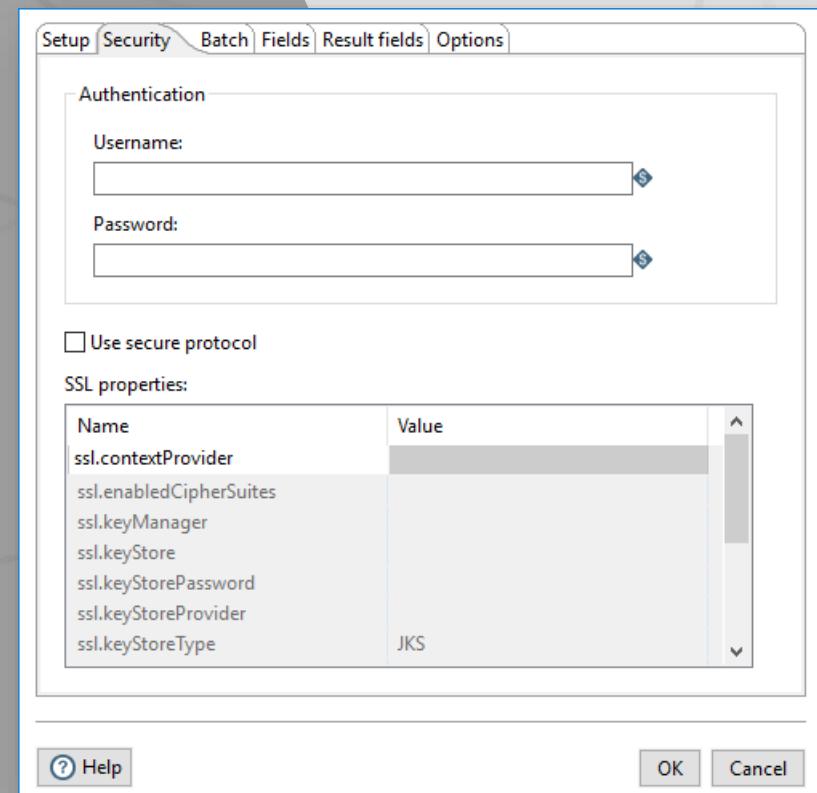
## Nota

Esta configuración del protocolo de seguridad se usa solo en Kettle. No se utiliza en AEL Spark.



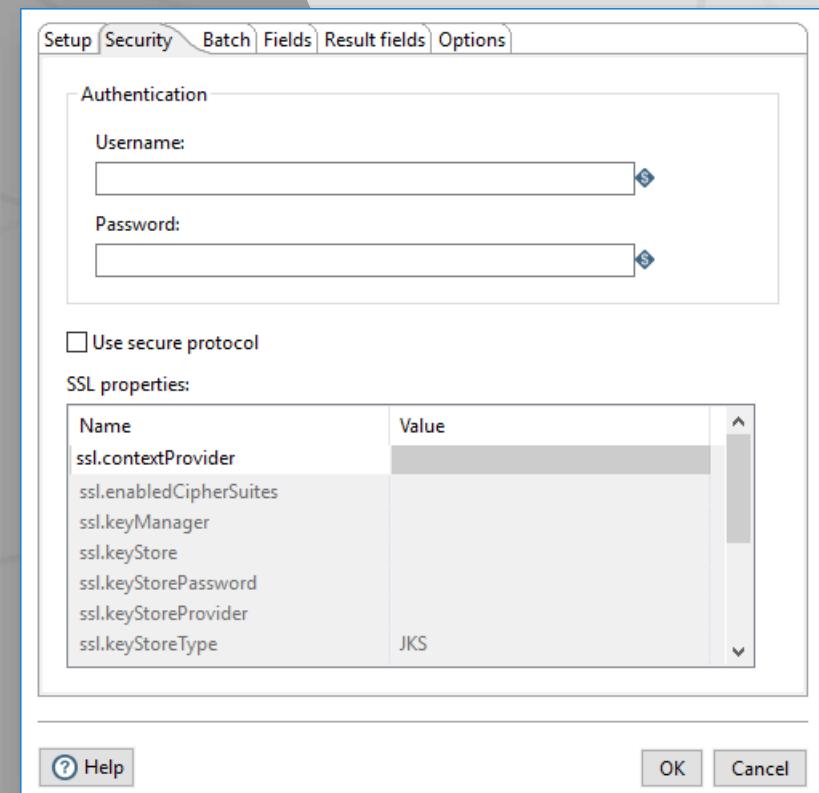
# Pestaña Security

Opción	Descripción
SSL Properties	<p><b>ssl.contextProvider:</b> especifique el proveedor JSSE subyacente.</p> <p><b>ssl.enabledCipherSuites:</b> especifique qué cifrados están habilitados. Los valores dependen del proveedor.</p> <p><b>ssl.keyManager:</b> especifique el algoritmo que se usará para crear un objeto KeyManagerFactory en lugar de usar el algoritmo predeterminado disponible en la plataforma.</p> <p><b>ssl.keyStore:</b> especifique el nombre del archivo que contiene el objeto KeyStore que desea que use el KeyManager.</p> <p><b>ssl.keyStorePassword:</b> especifique la contraseña para el objeto KeyStore que desea que use el KeyManager.</p> <p><b>ssl.keyStoreProvider:</b> especifique el nombre o la cadena de identificación para el proveedor del almacén de claves.</p> <p><b>ssl.keyStoreType:</b> especifique el nombre o la cadena de identificación para el tipo de almacén de claves.</p> <p><b>ssl.protocol</b> - Especifique el tipo de protocolo SSL a usar.</p>



# Pestaña Security

Opción	Descripción
SSL Properties	<p><b>ssl.trustManager:</b> especifique el algoritmo que se usará para crear un objeto TrustManagerFactory, en lugar de usar el algoritmo predeterminado disponible en la plataforma.</p> <p><b>ssl.trustStore:</b> especifique el nombre del archivo que contiene el objeto KeyStore que desea que use el TrustManager.</p> <p><b>ssl.trustStorePassword:</b> especifique la contraseña para el objeto TrustStore que desea que use el TrustManager.</p> <p><b>ssl.trustStoreProvider:</b> especifique el identificador o la cadena para el proveedor del almacén de confianza.</p> <p><b>ssl.trustStoreType:</b> especifique el tipo de objeto KeyStore que desea que utilice el TrustManager.</p>



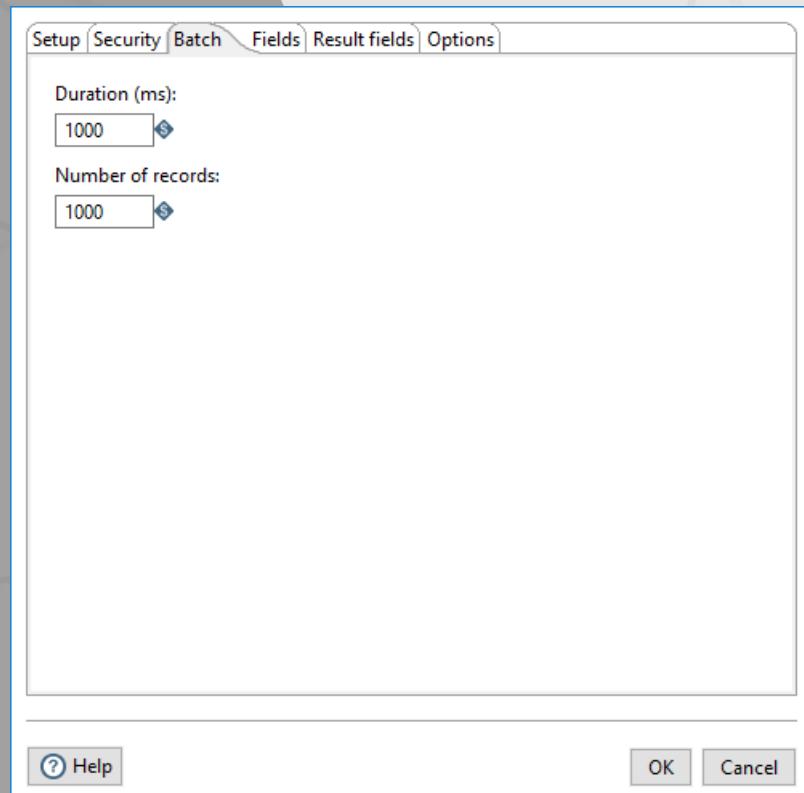
# Pestaña Batch

Opción	Descripción
Duration (ms)	<p>Especifique un tiempo en milisegundos. Este valor es la cantidad de tiempo que el paso dedicará a recopilar registros antes de la ejecución de la transformación.</p> <p><b>NOTA:</b> Debe establecer este campo si está utilizando Spark como su motor de procesamiento.</p> <p>Si se establece en un valor de '0', el número de registros activa el consumo.</p>
Number of records	<p>Especifique un número. Después de cada "X" número de registros, la transformación especificada se ejecutará y estos registros "X" se pasarán a la transformación.</p> <p>Si se establece en un valor de '0', la Duración activa el consumo.</p>



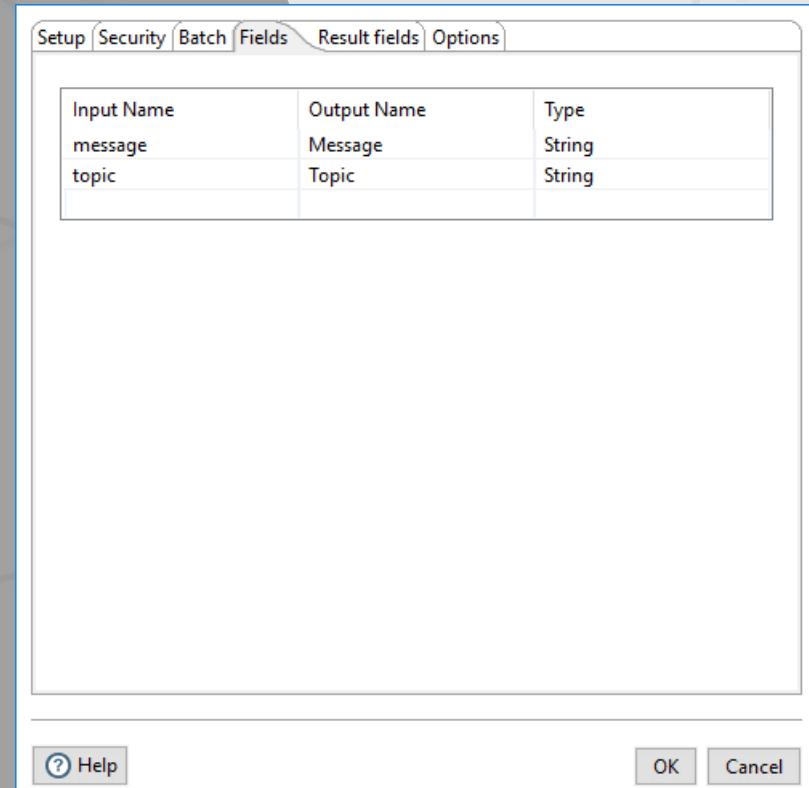
## Nota

El Número de registros o la Duración deben contener un valor mayor que "0" para ejecutar la transformación.



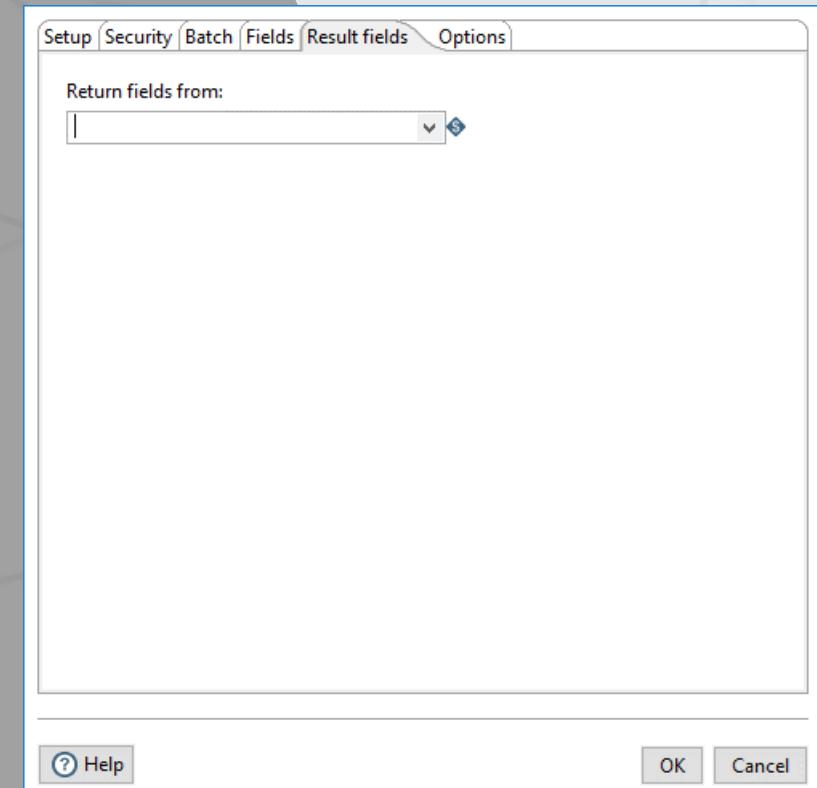
# Pestaña Fields

Opción	Descripción
Input name	<p>El nombre de entrada se recibe de los flujos MQTT. Los siguientes son recibidos por defecto:</p> <ul style="list-style-type: none"> <li>• mensaje: el mensaje individual contenido en un registro.</li> <li>• tema: la categoría a la que se publican los registros.</li> </ul>
Output name	El nombre de salida se puede asignar a los requisitos de suscriptores y miembros.
Type	Esto siempre será una cadena. Este campo se aplica a los nombres de entrada de "mensaje" y "tema".



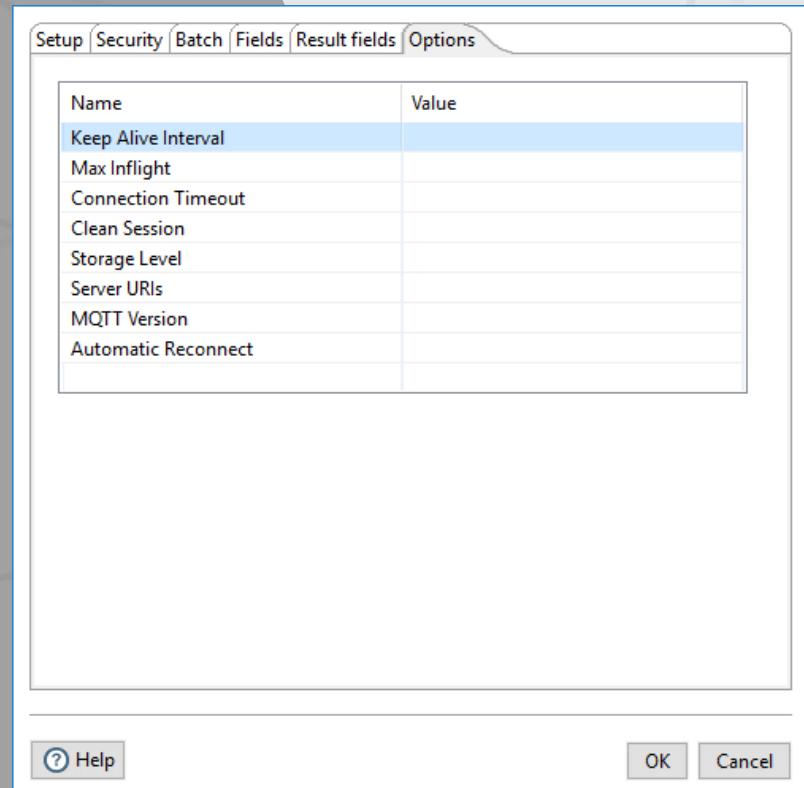
# Pestaña Result fields

Opción	Descripción
Return fields from:	Seleccione el nombre del paso (de la transformación secundaria) que transmitirá los campos a la transformación principal. Los valores de los datos en estos campos devueltos estarán disponibles para cualquier paso posterior en la transformación principal.



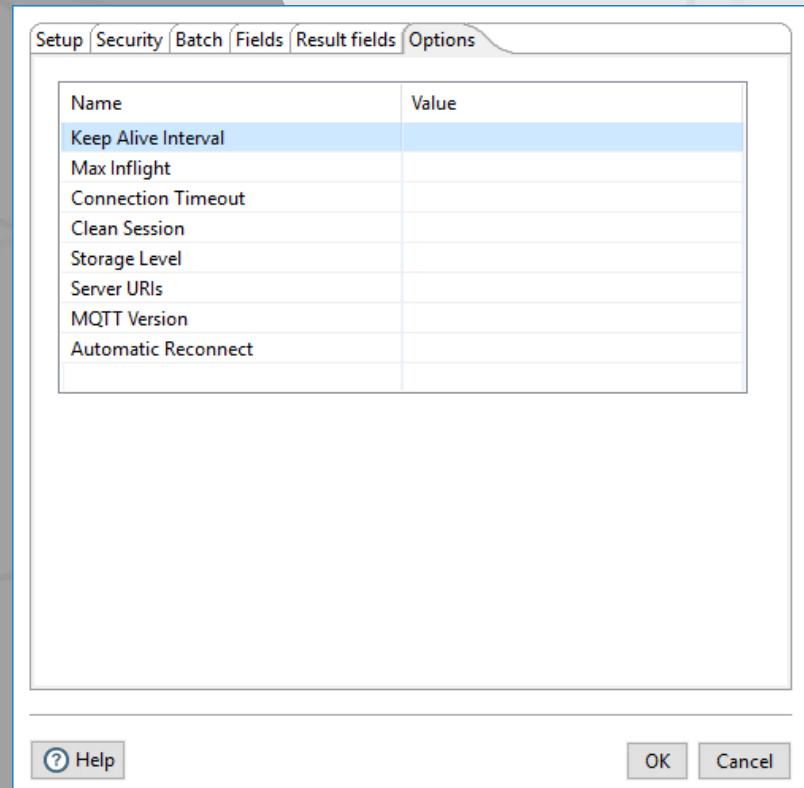
# Pestaña Options

Opción	Descripción
Keep Alive Interval	Especifique un número máximo de segundos de intervalo que se permite que transcurran entre el punto en el que el Cliente termina de transmitir un Paquete de Control y el punto en que comienza a enviar el siguiente.
Max Inflight	Especifique un número para el número máximo de mensajes que se estén procesando en un momento dado.
Connection Timeout	Especifique el tiempo, en segundos, para desconectar si no se recibe un mensaje.
Clean Session	Especifique si el agente almacenará o purgará mensajes para una sesión. Selecciona uno de los siguientes. <ul style="list-style-type: none"> <li>• Verdadero: cuando se establece en Verdadero, el intermediario no almacenará ninguna información para el cliente. Se borrará toda la información de una sesión persistente anterior.</li> <li>• Falso: cuando se establece en Falso, el agente almacenará todas las suscripciones para el cliente. Cuando el parámetro QoS (Calidad de servicio) se establece en "1" o "2", todos los mensajes perdidos se almacenarán. Para obtener más información, consulte el parámetro Calidad de servicio en la pestaña Configuración.</li> </ul>



# Pestaña Options

Opción	Descripción
Storage Level	Indica si los mensajes están almacenados en la memoria o en el disco. <ul style="list-style-type: none"> <li>• El valor predeterminado (en blanco) es la memoria.</li> <li>• Para disco, ingrese una ruta válida.</li> </ul> Nota: Esta configuración solo se utiliza en Kettle. No se utiliza en AEL Spark, que utiliza su propia configuración.
Server URLs	Especifique el identificador universal de recursos (URI) del servidor MQTT.
MQTT Version	Especifique la versión del protocolo MQTT a la que se conecta este paso.
Automatic Reconnect	Permite al cliente intentar una reconexión automática al servidor si se desconecta. Seleccione Verdadero o Falso: <ul style="list-style-type: none"> <li>• Verdadero: Sí: intente volver a conectarse al servidor.</li> <li>• Falso: No: no intente volver a conectarse al servidor.</li> </ul>



# Soporte de inyección de metadatos

Todos los campos de este paso admiten la inyección de metadatos. Puede usar este paso con [ETL Metadata Injection](#) para pasar los metadatos a su transformación en tiempo de ejecución.

## Nota

La inyección de metadatos no es compatible con los pasos que se ejecutan en la capa de ejecución adaptable (AEL).

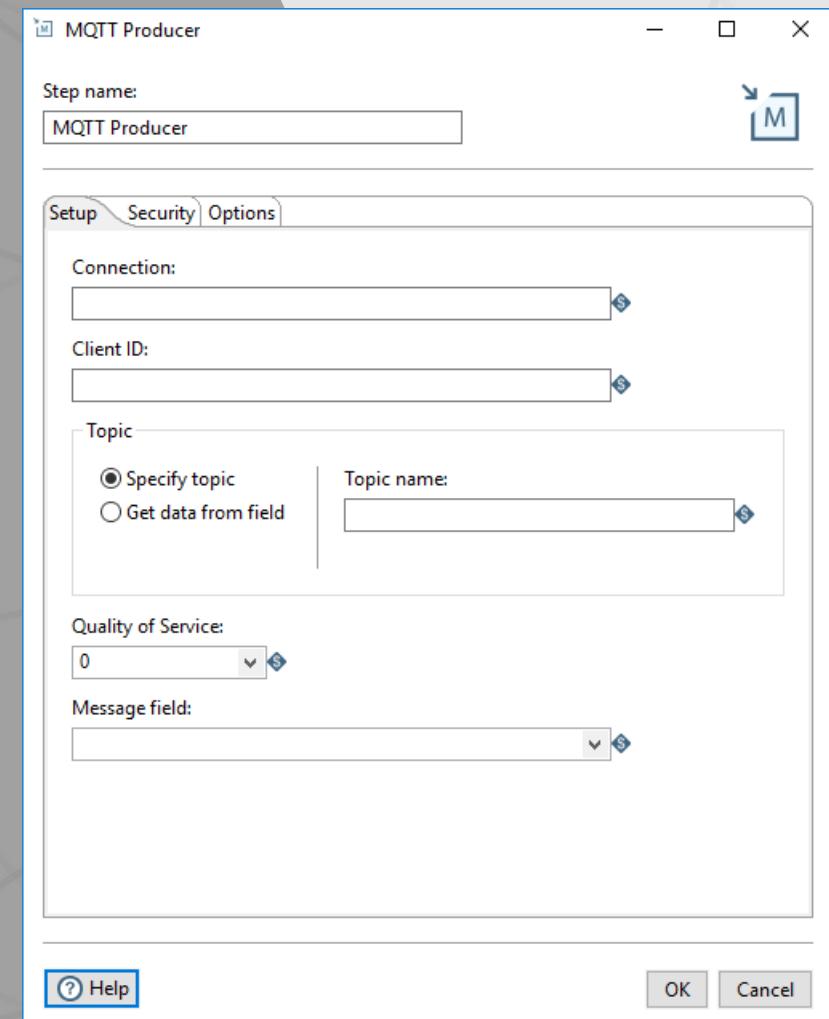
# MQTT Producer

- El paso Productor de MQTT le permite publicar mensajes casi en tiempo real a un agente de MQTT. Dentro de una transformación, el paso Productor MQTT publica una secuencia de registros a un tema MQTT.
- Para obtener más información sobre el protocolo MQTT, consulte:  
<https://www.hivemq.com/mqtt/>

# General

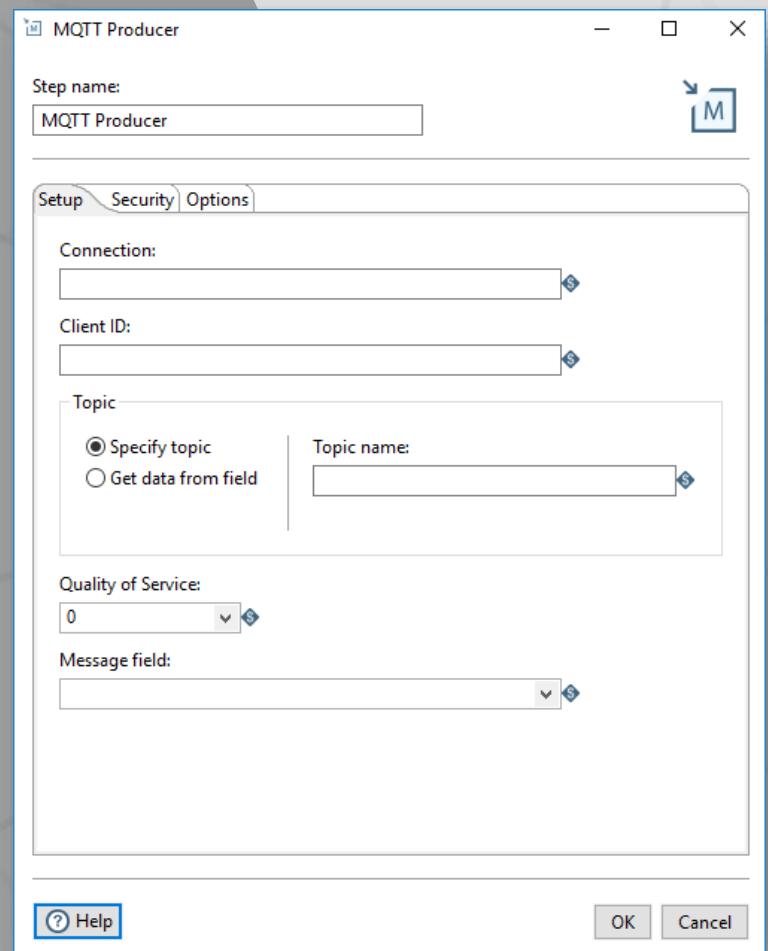
Ingrese la siguiente información en el campo de nombre del paso de transformación.

- Nombre de paso: especifica el nombre único de la transformación en el lienzo. Puede personalizar el nombre o dejar 'MQTT Producer' como predeterminado



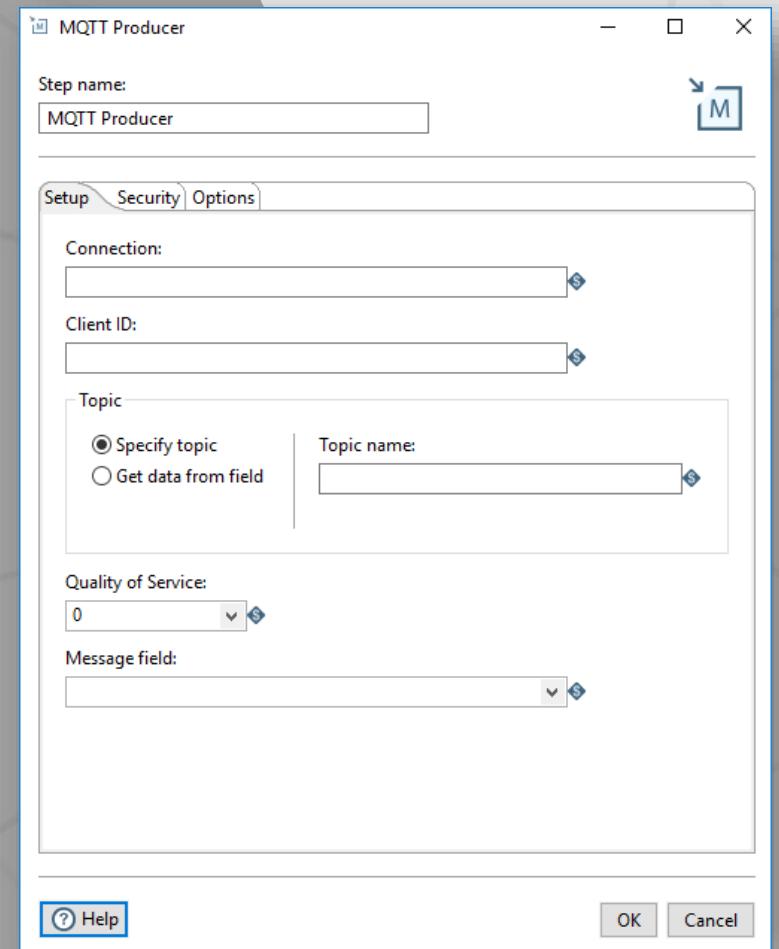
# Pestaña Setup

Opción	Descripción
Connection	Especifique la dirección del servidor MQTT al que se conectará este paso para enviar o recibir mensajes.
Client ID	Especifique una ID única para el cliente MQTT. El servidor MQTT utiliza este ID de cliente para reconocer cada cliente distinto y el estado actual de ese cliente.
Topic	Especifique el nombre del tema usando uno de los siguientes métodos: <ul style="list-style-type: none"> <li>Seleccione <b>Specify topic</b> para ingresar un nombre de tema específico. Luego, en el campo <b>Topic name</b>, ingrese el nombre del tema MQTT en el que desea publicar datos de transmisión (mensajes). Cada paso de MQTT Producer iniciará un solo hilo para la publicación.</li> <li>Seleccione <b>Get data from field</b> para especificar un nombre de Tema basado en un campo de otro paso que genere filas en el mismo flujo de transformación. Usando la lista desplegable, luego seleccione el nombre del campo que desea usar. Puede usar esta opción para controlar dinámicamente la configuración del valor para el nombre del tema. Cada mensaje individual seguirá teniendo solo un tema, pero cada fila que ingrese al paso del Productor de MQTT generará un mensaje nuevo con un tema potencialmente diferente.</li> </ul>



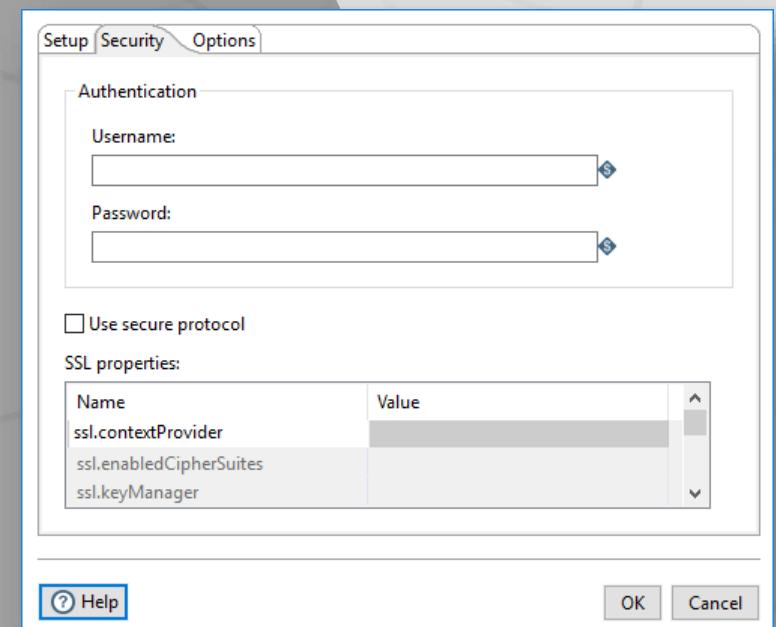
# Pestaña Setup

Opción	Descripción
Quality of Service (QoS)	<p>La calidad de servicio (QoS) es un nivel de garantía para la entrega de mensajes.</p> <p>Selecciona una de las siguientes opciones.</p> <ul style="list-style-type: none"> <li>• Como máximo una vez (0) - Predeterminado</li> <li>• Al menos una vez (1)</li> <li>• Exactamente una vez (2)</li> </ul>
Message Field	Seleccione el registro individual contenido en un tema que desea enviar.



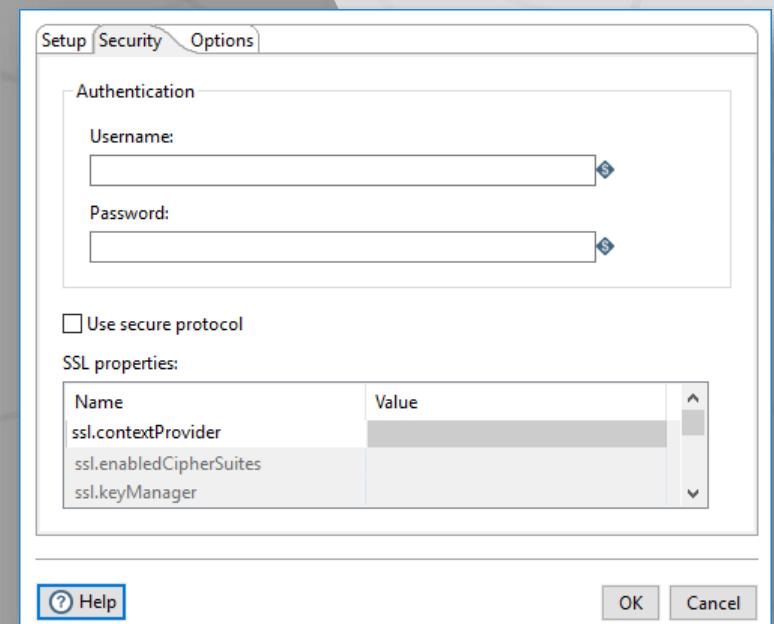
# Pestaña Security

Opción	Descripción
Username	Especifique el nombre de usuario requerido para acceder al servidor MQTT.
Password	Especifique la contraseña asociada con el nombre de usuario.
Use secure protocol	Seleccione esta opción si desea definir las propiedades SSL para la conexión.  <b>Nota:</b> Esta configuración del protocolo de seguridad se usa solo en Kettle. No se utiliza en AEL Spark.



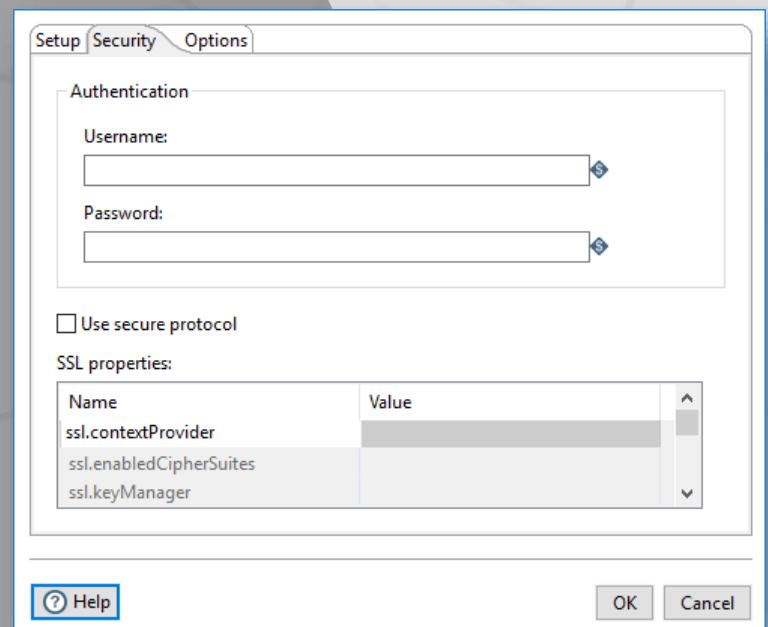
# Pestaña Security

Opción	Descripción
SSL Properties	<p><b>ssl.contextProvider:</b> especifique el proveedor JSSE subyacente.</p> <p><b>ssl.enabledCipherSuites:</b> especifique qué cifrados están habilitados. Los valores dependen del proveedor.</p> <p><b>ssl.keyManager:</b> especifique el algoritmo que se usará para crear un objeto KeyManagerFactory en lugar de usar el algoritmo predeterminado disponible en la plataforma.</p> <p><b>ssl.keyStore:</b> especifique el nombre del archivo que contiene el objeto KeyStore que desea que use el KeyManager.</p> <p><b>ssl.keyStorePassword:</b> especifique la contraseña para el objeto KeyStore que desea que use el KeyManager.</p> <p><b>ssl.keyStoreProvider:</b> especifique el nombre o la cadena de identificación para el proveedor del almacén de claves.</p> <p><b>ssl.keyStoreType:</b> especifique el nombre o la cadena de identificación para el tipo de almacén de claves.</p>



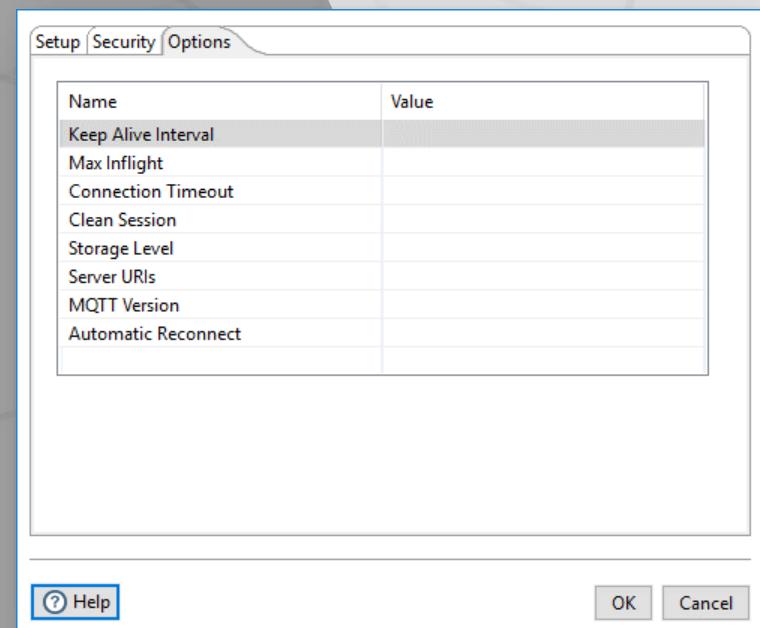
# Pestaña Security

Opción	Descripción
SSL Properties	<p><b>ssl.protocol:</b> Especifique el tipo de protocolo SSL a usar.</p> <p><b>ssl.trustManager:</b> especifique el algoritmo que se usará para crear un objeto TrustManagerFactory, en lugar de usar el algoritmo predeterminado disponible en la plataforma.</p> <p><b>ssl.trustStore:</b> especifique el nombre del archivo que contiene el objeto KeyStore que desea que use el TrustManager.</p> <p><b>ssl.trustStorePassword:</b> especifique la contraseña para el objeto TrustStore que desea que use el TrustManager.</p> <p><b>ssl.trustStoreProvider:</b> especifique el identificador o la cadena para el proveedor del almacén de confianza.</p> <p><b>ssl.trustStoreType:</b> especifique el tipo de objeto KeyStore que desea que utilice el TrustManager.</p>



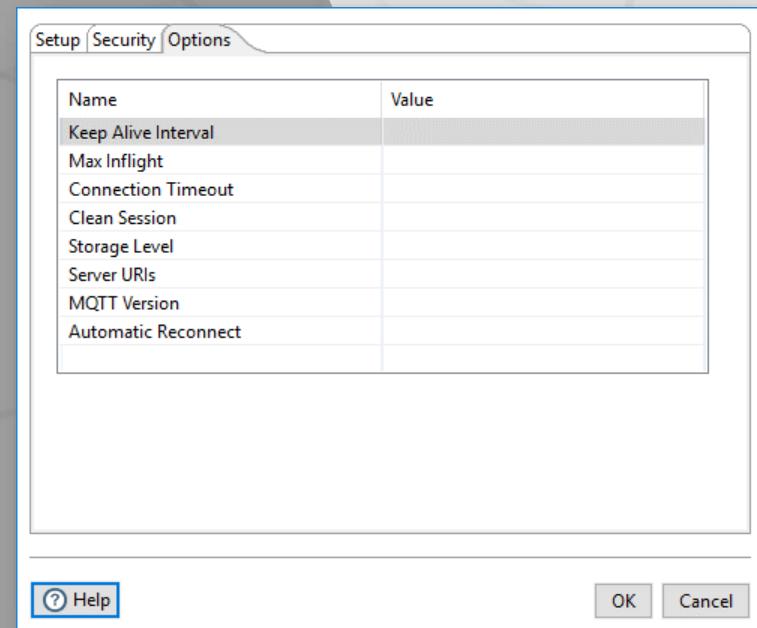
# Pestaña Options

Opción	Descripción
Keep Alive Interval	Especifique un número máximo de segundos de intervalo que se permite que transcurran entre el punto en el que el Cliente termina de transmitir un Paquete de Control y el punto en que comienza a enviar el siguiente.
Max Inflight	Especifique un número para el número máximo de mensajes que se estén procesando en un momento dado.
Connection Timeout	Especifique el tiempo, en segundos, para desconectar si no se recibe un mensaje.
Clean Session	Especifique si el agente almacenará o purgará mensajes para una sesión. Selecciona uno de los siguientes. <b>Verdadero</b> : cuando se establece en Verdadero, el intermediario no almacenará ninguna información para el cliente. Se borrará toda la información de una sesión persistente anterior. <b>Falso</b> : cuando se establece en Falso, el agente almacenará todas las suscripciones para el cliente. Cuando el parámetro QoS (Calidad de servicio) se establece en '1' o '2', todos los mensajes perdidos se almacenarán.



# Pestaña Options

Opción	Descripción
Storage Level	<p>Indica si los mensajes están almacenados en la memoria o en el disco.</p> <ul style="list-style-type: none"> <li>• El valor predeterminado (en <i>blanco</i>) es la memoria.</li> <li>• Para disco, ingrese una ruta válida.</li> </ul> <p>Esta configuración solo se utiliza en Kettle. No se utiliza en AEL Spark, que utiliza su propia configuración.</p>
Server URLs	Especifique el identificador universal de recursos (URI) del servidor MQTT.
MQTT Version	Especifique la versión del protocolo MQTT a la que se conecta este paso.
Automatic Reconnect	<p>Permite al cliente intentar una reconexión automática al servidor si se desconecta. Seleccione Verdadero o Falso:</p> <p><b>Verdadero:</b> Sí: intente volver a conectarse al servidor.  <b>Falso :</b> No, no intente volver a conectarse al servidor.</p>



# Soporte de inyección de metadatos

- Todos los campos de este paso admiten la inyección de metadatos. Puede usar este paso con la [inyección de metadatos de ETL](#) para pasar los metadatos a su transformación en tiempo de ejecución.
- La inyección de metadatos no es compatible con los pasos que se ejecutan en [la capa de ejecución adaptable \(AEL\)](#).

# JMS Producer

- El paso de Java Messaging Service (JMS) Producer publica mensajes casi en tiempo real al servidor Apache ActiveMQ JMS o al middleware IBM MQ. Puede utilizar el paso JMS Producer para definir una transformación que se publica en una cola JMS para cada actualización de un almacén. A su vez, esta cola podría iniciar otro trabajo que vacíe el caché de una aplicación.

# Antes de que empieces

Antes de utilizar el paso de JMS Producer, tenga en cuenta las siguientes condiciones:

- Debe estar familiarizado con la mensajería JMS para usar este paso. Además, debe tener un intermediario de mensajes, como Apache ActiveMQ o IBM MQ, disponible antes de configurar este paso.
- Este paso admite JMS 2.0 y requiere [Apache ActiveMQ Artemis](#).
- Si necesita usar JMS 1.1 con ActiveMQ o Artemis, use las versiones anteriores de los pasos de JMS Consumer y JMS Producer, también disponibles en Pentaho versión 8.1 y anteriores.

# Antes de que empieces

Coloque los archivos JAR de cliente de IBM MQ para el middleware de IBM MQ en los siguientes directorios:

- En el cliente PDI: data-integration/system/karaf/deploy
- En el servidor Pentaho: server/pentaho-server/pentaho-solutions/system/karaf/ deploy

Debe ubicar las clases de Websphere MQ para las bibliotecas de JMS Java desde su instalación de IBM Websphere MQ. También puede encontrar estas bibliotecas en su [IBM Websphere MQ Client SupportPac](#). La versión de las bibliotecas Java de Websphere MQ contra la que se construyeron los pasos del complemento PDI es 9.0.0.3. Las bibliotecas que debe tener disponibles para su distribución en el complemento JMS de PDI son:

- ocom.ibm.mq.osgi.allclientprereqs\_9.0.0.3.jar
- ocom.ibm.mq.osgi.allclient\_9.0.0.3.jar
- ocom.ibm.mq.jmqi.jar
- odhbcore.jar

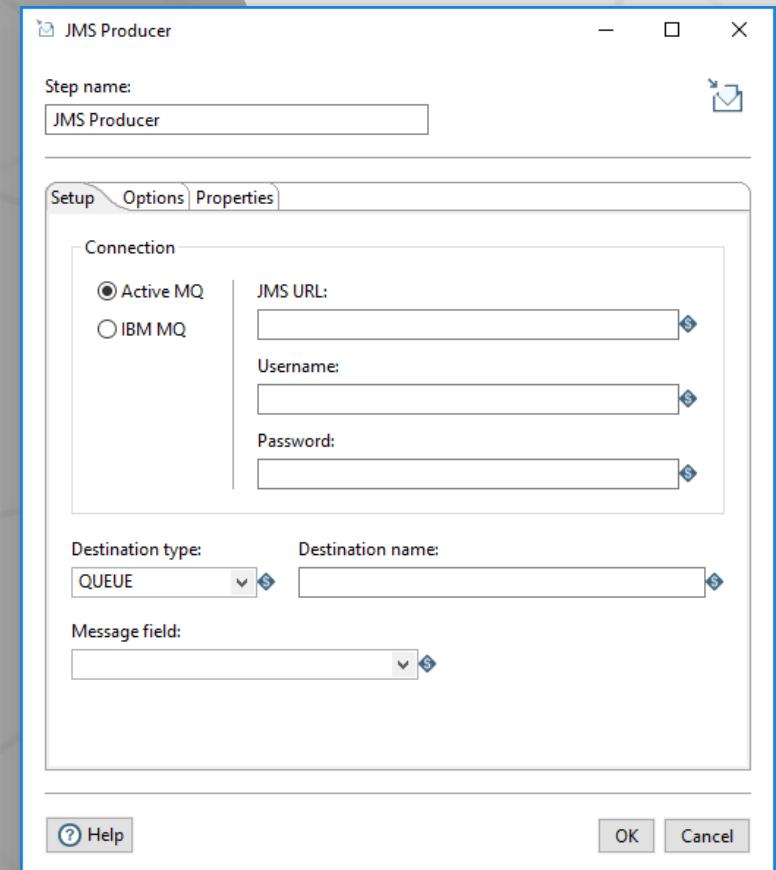
# Antes de que empieces

Debido a que las licencias de IBM nos impiden distribuir estas bibliotecas directamente, deberá agregarlas a sus directorios PDI.

- Coloque los jars de la biblioteca JMS para ConnectionFactory otras clases de soporte en los siguientes directorios:
  - En el cliente PDI: `data-integration/system/karaf/deploy`
  - En el servidor Pentaho: `server/pentaho-server/pentaho-solutions/system/karaf/deploy`

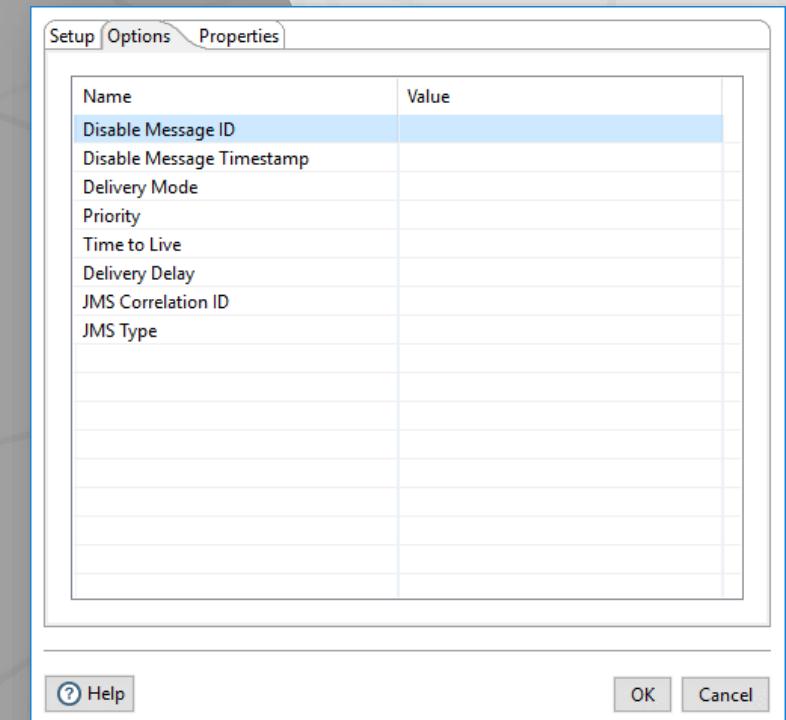
# Pestaña de Setup

Opción	Descripción
IBM MQ	Active este tipo de conexión si está utilizando IBM MQ como su intermediario de mensajes.
Active MQ	Active este tipo de conexión si está utilizando Apache ActiveMQ Artemis y JMS 2.0 como su agente de mensajes.
JMS URL	Ingrese la URL del corredor para el tipo de conexión seleccionado.
Username	Introduzca el nombre de usuario Apache ActiveMQ Artemis o IBM MQ.
Password	Ingrese la contraseña de Apache ActiveMQ Artemis o IBM MQ.



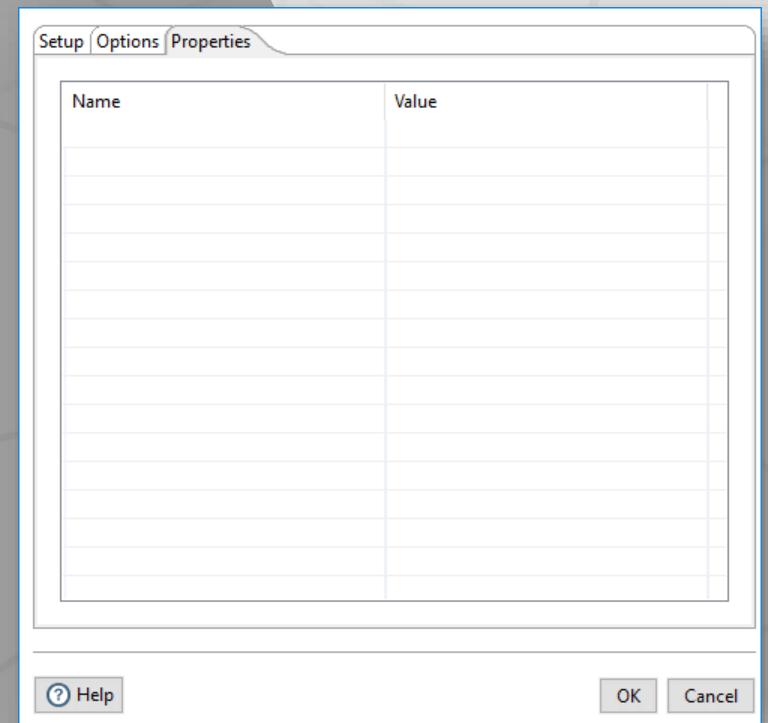
# Pestaña Options

Opción	Descripción
Disable Message ID	<b>verdadero o falso</b>
Disable Message Timestamp	<b>verdadero o falso</b>
Delivery Mode	Entero (ejemplo '1')
Priority	Entero entre 0 y 9 (ejemplo '2')
Time to Live	Entero (ejemplo '4') Este valor debe ser mayor que el valor del <b>Retardo de entrega</b> .
Delivery Delay	Entero (ejemplo '2')
JMS Correlation ID	Cualquier ID (ejemplo 'Vantara')
JMS Type	<b>Active MQ o IBM MQ</b>



# Pestaña Properties

Opción	Descripción
Disable Message ID	<b>verdadero o falso</b>
Disable Message Timestamp	<b>verdadero o falso</b>
Delivery Mode	Entero (ejemplo '1')
Priority	Entero entre 0 y 9 (ejemplo '2')
Time to Live	Entero (ejemplo '4') Este valor debe ser mayor que el valor del <b>Retardo de entrega</b> .
Delivery Delay	Entero (ejemplo '2')
JMS Correlation ID	Cualquier ID (ejemplo 'Vantara')
JMS Type	<b>Active MQ o IBM MQ</b>



# Soporte de inyección de metadatos

Todos los campos de este paso admiten la inyección de metadatos. Puede usar este paso con la [inyección de metadatos de ETL](#) para pasar los metadatos a su transformación en tiempo de ejecución.

# JMS Consumer

- Utilice el paso del consumidor del Servicio de mensajería de Java (JMS) para recibir datos de transmisión desde el servidor JMS de Apache ActiveMQ o el middleware de IBM MQ.
- El paso principal de JMS Consumer ejecuta un proceso secundario (sub-transformación) que se ejecuta de acuerdo con el tamaño o la duración del lote de mensajes, lo que le permite procesar un flujo continuo de registros casi en tiempo real. La transformación secundaria debe comenzar con el paso [Obtener registros de la secuencia](#). Puede configurar el paso de JMS Consumer para ingerir continuamente datos de transmisión desde su servidor JMS.

# JMS Consumer

- En el mismo paso de JMS Consumer, puede definir la cantidad de mensajes que se aceptarán para procesar, así como los formatos de datos específicos para transmitir datos de actividad y métricas del sistema. Puede configurar este paso para recopilar eventos monitoreados, rastrear el consumo de flujos de datos por parte del usuario y monitorear alertas. Además, puede seleccionar un paso en la transformación secundaria para transmitir los registros nuevamente a la transformación principal, que pasa los registros hacia abajo a cualquier otro paso incluido dentro de la misma transformación principal.

## Nota

Dado que el paso de JMS Consumer ingiere datos de transmisión continua, es posible que desee utilizar el paso Abortar en la transformación principal o secundaria para dejar de consumir registros de JMS para flujos de trabajo específicos. Por ejemplo, puede ejecutar la transformación principal en un horario programado, o abortar la transformación secundaria si los datos del sensor exceden un rango preestablecido.



# Antes de que empieces

Antes de utilizar el paso de JMS Consumer, tenga en cuenta las siguientes condiciones:

- Debe estar familiarizado con la mensajería JMS para usar este paso. Además, debe tener un intermediario de mensajes, como Apache ActiveMQ o IBM MQ, disponible antes de configurar este paso.
- Este paso admite JMS 2.0 y requiere [Apache ActiveMQ Artemis](#).
- Si necesita usar JMS 1.1 con ActiveMQ o Artemis, use las versiones anteriores de los pasos de JMS Consumer y JMS Producer, también disponibles en Pentaho versión 8.1 y anteriores.

# Antes de que empieces

Coloque los archivos JAR de cliente de IBM MQ para el middleware de IBM MQ en los siguientes directorios:

- En el cliente PDI: data-integration/system/karaf/deploy
- En el servidor Pentaho: server/pentaho-server/pentaho-solutions/system/karaf/deploy

# Antes de que empieces

Debe ubicar las clases de Websphere MQ para las bibliotecas de JMS Java desde su instalación de IBM Websphere MQ. También puede encontrar estas bibliotecas en su [IBM Websphere MQ Client SupportPac](#). La versión de las bibliotecas Java de WebsphereMQ contra la que se construyeron los pasos del complemento PDI es 9.0.0.3. Las bibliotecas que debe tener disponibles para su distribución en el complemento JMS de PDI son:

- ocom.ibm.mq.osgi.allclientprereqs\_9.0.0.3.jar
- ocom.ibm.mq.osgi.allclient\_9.0.0.3.jar
- ocom.ibm.mq.jmqi.jar
- odhbcore.jar

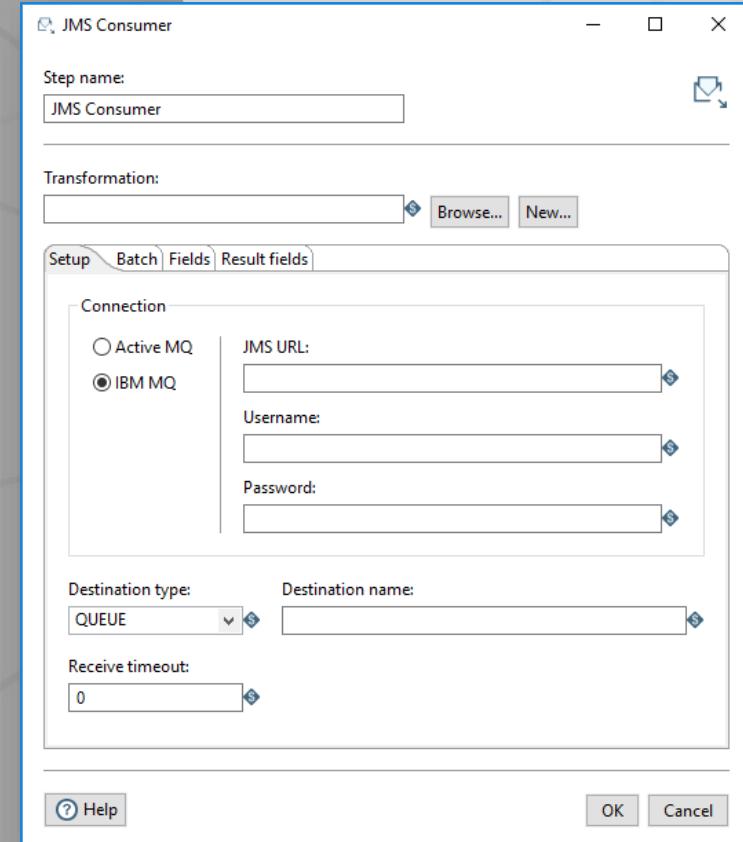
Debido a que las licencias de IBM nos impiden distribuir estas bibliotecas directamente, deberá agregarlas a sus directorios PDI.

# Antes de que empieces

- Coloque los archivos JMS Library para ConnectionFactory y otras clases de soporte en los siguientes directorios:
  - En el cliente PDI: data-integration/system/karaf/deploy
  - En el servidor Pentaho: server/pentaho-server/pentaho-solutions/system/karaf/deploy

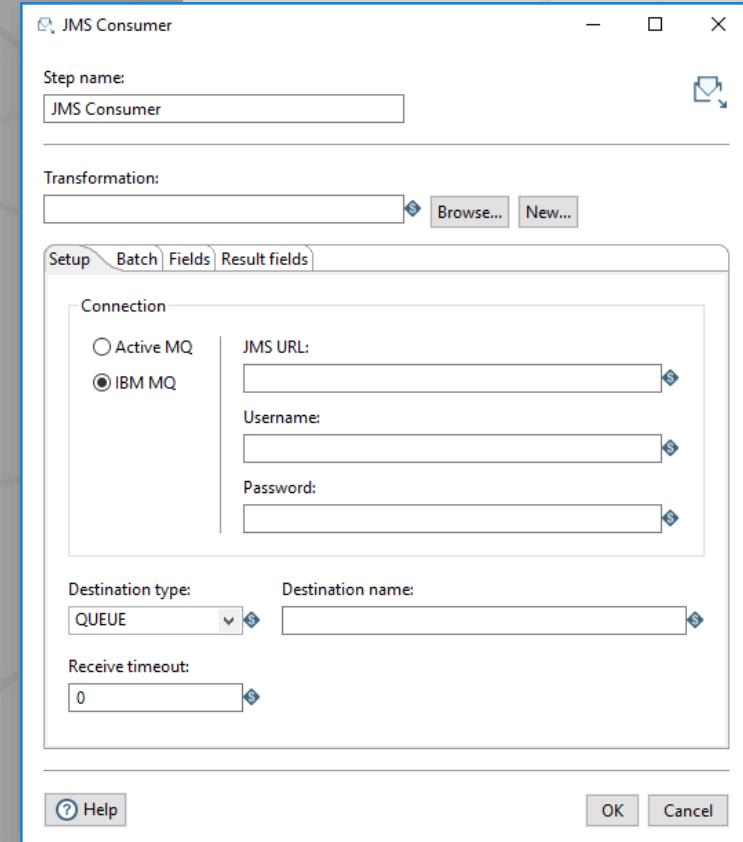
# Pestaña Properties

Opción	Descripción
Step name	Especifica el nombre único del paso en el lienzo. El <b>nombre del paso</b> se establece en <b>JMS Consumer</b> de forma predeterminada.
Transformation	Especifique la transformación secundaria que se ejecutará realizando cualquiera de las siguientes acciones: <ul style="list-style-type: none"> <li>Entrando en su camino</li> <li>Al hacer clic en <b>Browse</b> para seleccionar una transformación secundaria existente</li> <li>Al hacer clic en <b>New</b> para crear y guardar una nueva transformación secundaria. Consulte <a href="#">Crear y guardar una nueva transformación infantil</a> para obtener más detalles.</li> </ul> <p>Nota: La transformación secundaria seleccionada debe comenzar con el paso <b>Obtener registros de la transmisión</b>.</p>



# Pestaña Properties

Opción	Descripción
Transformation	<p>Si selecciona una transformación que tiene la misma ruta raíz que la transformación actual, la variable \${Internal.Entry.Current.Directory} se inserta automáticamente en lugar de la ruta raíz común. Por ejemplo, si la ruta de la transformación actual es /home/admin/transformation.ktr y selecciona una transformación en el directorio /home/admin/path/sub.ktr , la ruta se convierte automáticamente a \${Internal.Entry.Current.Directory}/path/sub.ktr .</p> <p>Si está trabajando con un repositorio, debe especificar el nombre de la transformación. Si no está trabajando con un repositorio, debe especificar el nombre de archivo XML de la transformación.</p> <p>Las transformaciones previamente especificadas por referencia se convierten automáticamente para ser especificadas por el nombre de transformación en el repositorio de Pentaho.</p>



# Crea y guarda una nueva transformación hija

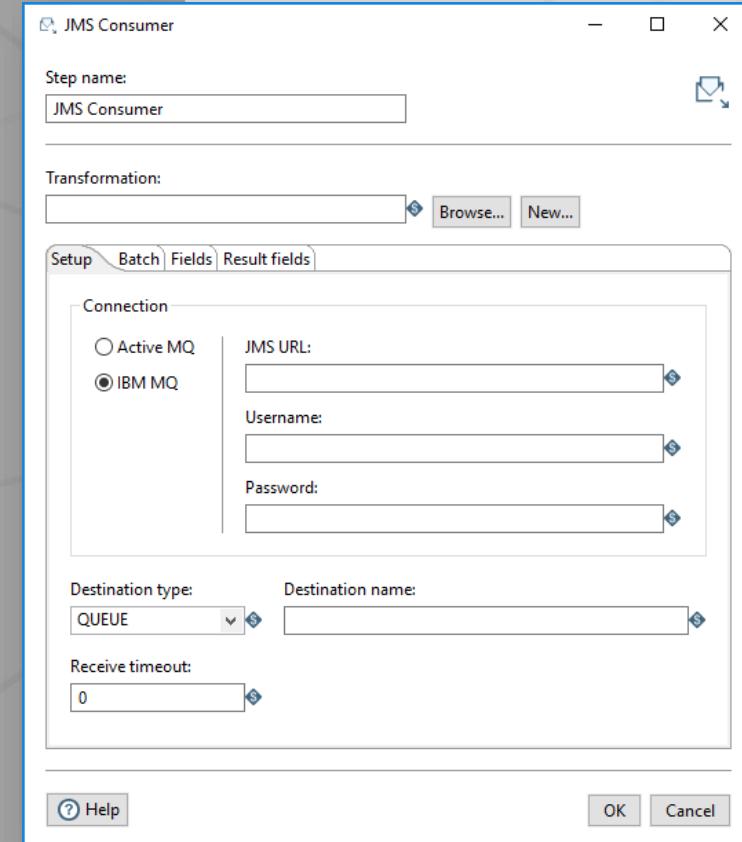
- Si aún no tiene una transformación secundaria, puede crear una al configurar el paso de JMS Consumer. Al hacer clic en el botón **New** , una nueva transformación secundaria generará automáticamente el paso requerido Obtener registros de la transmisión en una nueva pestaña de lienzo. Todos sus campos y tipos se personalizan en el paso Obtener registros de la transmisión secundaria de la transformación secundaria para que coincida con los campos y tipos especificados en la pestaña Campos del paso del consumidor de Kafka principal.

## Pasos

1. En el paso de JMS Consumer, haga clic en **New** . Aparece el cuadro de diálogo **Save As** .
2. Navegue hasta la ubicación donde desea guardar su nueva transformación secundaria y luego escriba el nombre del archivo.
3. Haga clic **Save As** . Aparece un cuadro de notificación que le informa que la transformación secundaria se ha creado y abierto en una nueva pestaña. Si no desea ver esta notificación en el futuro, seleccione la casilla de verificación **Don't show me this again** .
4. Haga clic en la nueva pestaña de transformación para ver y editar la transformación secundaria. Contiene automáticamente el paso Get Records from Stream. Opcionalmente, puede continuar construyendo esta transformación y guardarla.
5. Cuando haya terminado, vuelva al paso JMS Consumer.

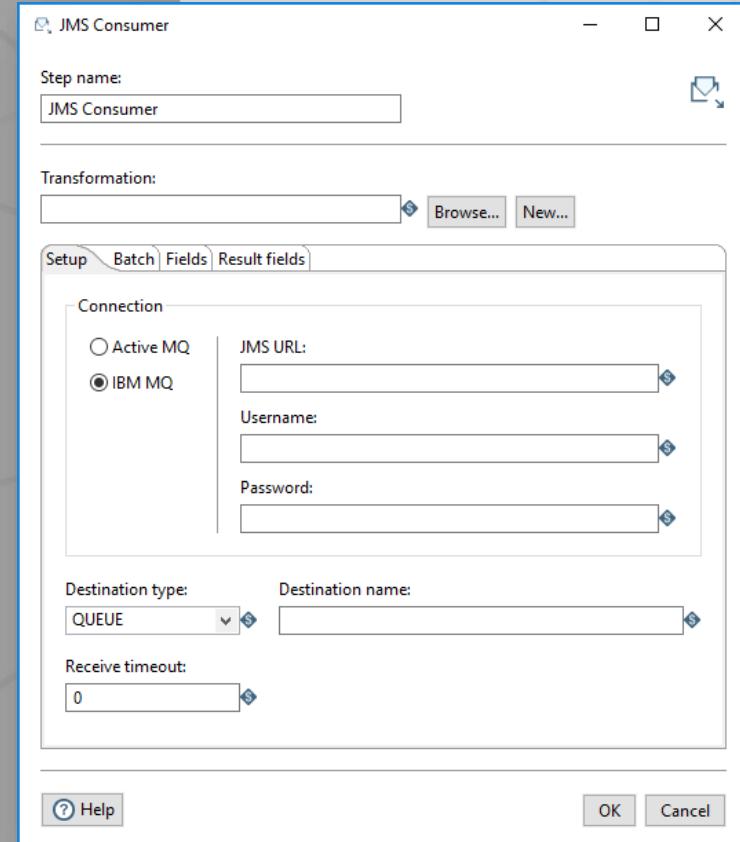
# Pestaña Setup

Opción	Descripción
IBM MQ	Active este tipo de conexión si está utilizando IBM MQ como su intermediario de mensajes.
ActiveMQ	Active este tipo de conexión si está utilizando Apache ActiveMQ Artemis y JMS 2.0 como su agente de mensajes.
JMS URL	Ingrese la URL del corredor para el tipo de conexión seleccionado.
Username	Introduzca el nombre de usuario Apache ActiveMQ Artemis o IBM MQ.
Password	Ingrese la contraseña de Apache ActiveMQ Artemis o IBM MQ.
Destination type	<p>Seleccione <b>Topic</b> o <b>Queue</b> en la lista para especificar el modelo de entrega que desea usar.</p> <ul style="list-style-type: none"> <li>• <b>Topic</b> utiliza un modelo de entrega de publicación y suscripción para que un mensaje se pueda entregar a múltiples consumidores. Los mensajes se envían al destino del tema y, en última instancia, a todos los consumidores activos que son suscriptores del tema.</li> <li>• <b>Queue</b> utiliza un modelo de entrega punto a punto. En este modelo, un mensaje se entrega de un solo productor a un solo consumidor. Los mensajes se envían al destino, que es una cola, y luego se envían a uno de los consumidores registrados para la cola.</li> </ul>



# Pestaña Setup

Opción	Descripción
Destination name	Especifique el nombre del tema o cola.
Receive timeout	Especifique el tiempo para esperar los mensajes entrantes en milisegundos. <b>Nota:</b> Un ajuste de tiempo de espera de cero ('0') nunca caduca.

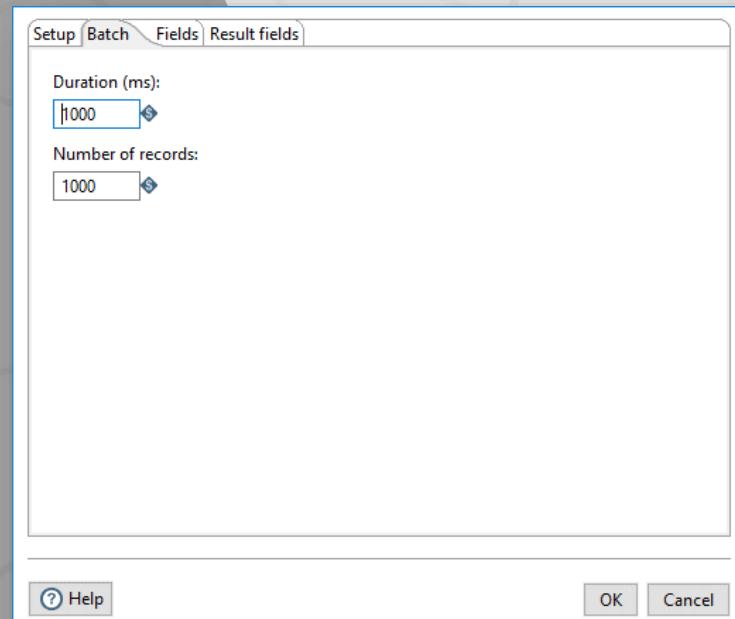


# Pestaña Batch

Opción	Descripción
Duración (ms)	Especifique un tiempo en milisegundos. Este valor es la cantidad de tiempo que el paso dedicará a recopilar registros antes de la ejecución de la transformación. Si se establece en un valor de '0', el <b>Número de registros</b> activa el consumo
Number of records	Especifique un número. Después de cada 'X' número de registros, la transformación especificada se ejecutará y estos registros 'X' se pasarán a la transformación. Si se establece en un valor de '0', la <b>Duración</b> activa el consumo.

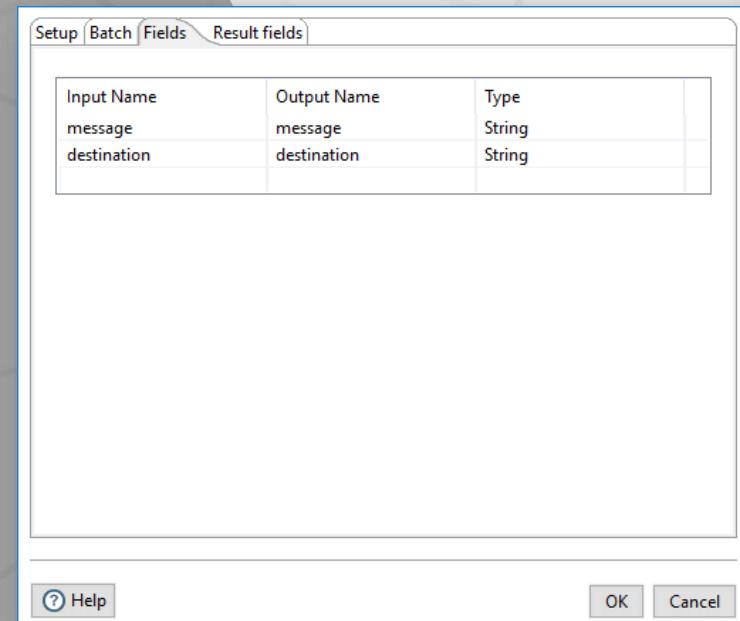
## Nota

El **Número de registros** o la **Duración** deben contener un valor mayor que '0' para ejecutar la transformación.



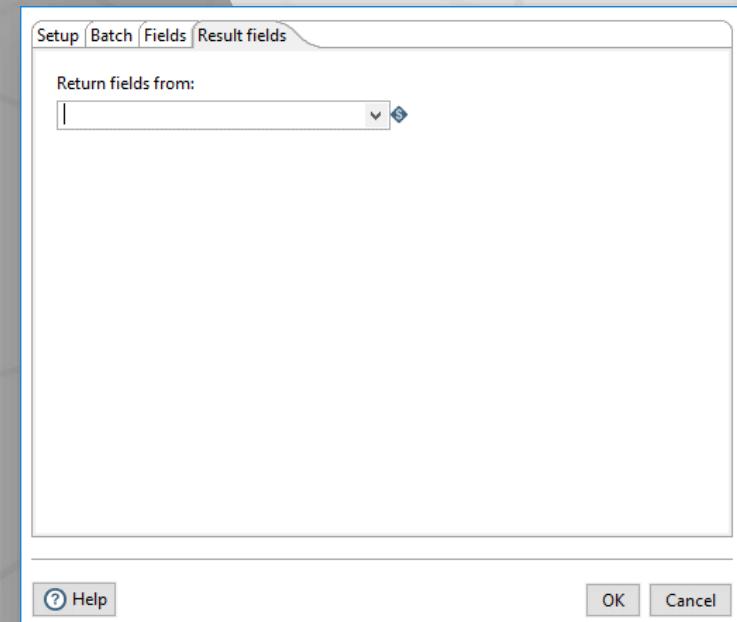
# Pestaña Fields

Opción	Descripción
<b>Input name</b>	El nombre de entrada se recibe de los flujos JMS. Los siguientes son recibidos por defecto: <b>message</b> : El mensaje individual contenido en un registro. Cada registro consta de una clave, un valor y una marca de tiempo. <b>destination</b> : el nombre del tema o cola.
<b>Output name</b>	El nombre de salida se puede asignar a los requisitos de suscriptores y miembros.
<b>Type</b>	El campo Tipo define el formato de datos para transmitir el registro, que es el mismo tipo de datos que produjo los registros. La opción es String.



# Pestaña Result fields

Opción	Descripción
<b>Return fields from</b>	Use la opción <b>Return fields from</b> en esta pestaña para seleccionar el nombre del paso de la transformación secundaria que transmitirá los registros a la transformación principal. Los valores de los datos en estos campos devueltos estarán disponibles para cualquier paso posterior en la transformación principal.



# Soporte de inyección de metadatos

Todos los campos de este paso admiten la inyección de metadatos. Puede usar este paso con la [inyección de metadatos de ETL](#) para pasar los metadatos a su transformación en tiempo de ejecución.

# Group By

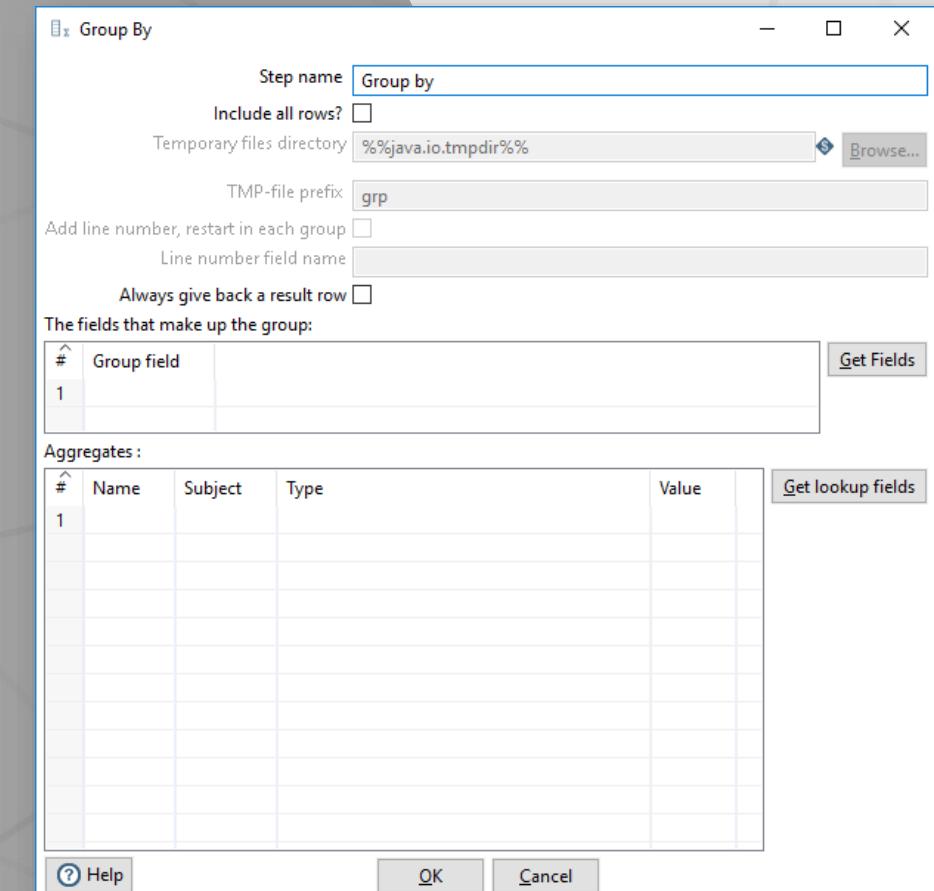
- Este paso agrupa las filas de una fuente, en función de un campo específico o una colección de campos. Se genera una nueva fila para cada grupo. También puede generar uno o más valores agregados para los grupos. Los usos comunes son calcular el promedio de ventas por producto y contar el número de un artículo que tiene en stock.
- El Grupo por paso está diseñado para entradas ordenadas. Si su entrada no está ordenada, solo se agrupan correctamente las filas dobles consecutivas. Si ordena los datos fuera de PDI, la sensibilidad a los casos de los datos en los campos puede producir resultados de agrupamiento inesperados.
- Puede usar el [grupo de memoria por](#) paso para manejar la entrada no ordenada.

# General

Opción	Descripción
Step Name	Especifica el nombre único del grupo por paso en el lienzo. Puede personalizar el nombre o dejarlo como predeterminado.
Include all rows?	Seleccione si desea incluir todas las filas en la salida, en lugar de solo las filas agregadas. Las siguientes opciones no están disponibles a menos que se seleccione la opción Incluir todas las filas: <ul style="list-style-type: none"> <li>• <b>Directorio de archivos temporales</b></li> <li>• <b>Prefijo de archivo TMP</b></li> <li>• <b>Añadir número de línea, reiniciar en cada grupo</b></li> <li>• <b>Número de línea nombre de campo</b></li> </ul>
Temporary files directory	Especifique el directorio donde se almacenan los archivos temporales. El valor predeterminado es el directorio temporal estándar para el sistema. Debe especificar un directorio cuando la opción Incluir todas las filas está seleccionada y el número de filas agrupadas excede las 5000 filas.
TMP-file prefix	Especifica el prefijo de archivo para nombrar archivos temporales.
Add line number, restart in each group	Agrega un número de línea que se reinicia en 1 en cada grupo. Cuando se seleccionan Incluir todas las filas y esta opción, todas las filas se incluyen en la salida y tienen un número de línea para cada fila.
Line number field name	Especifica el nombre del campo donde desea agregar números de línea para cada nuevo grupo.
Always give back a result row	Devuelve una fila de resultados, incluso cuando no hay una fila de entrada. Cuando no hay filas de entrada, esta opción devuelve un conteo de cero (0).

# Aggregates Table

Opción	Descripción
Name	El nombre del campo agregado.
Subject	El tema en el que desea utilizar un método de agregación.
Type	<p>El método agregado.          Los métodos de agregación son:</p> <ul style="list-style-type: none"> <li>Suma</li> <li>(Promedio)</li> <li>Mediana</li> <li>Percentil</li> <li>Mínimo</li> <li>Máximo</li> <li>Número de valores (N)</li> <li>Concatenar cadenas separadas por, (coma)</li> <li>Primer valor no nulo</li> <li>Último valor no nulo</li> <li>Primer valor (incluyendo nulo)</li> <li>Último valor (incluyendo nulo)</li> <li>Suma acumulada (solo opción de todas las filas)</li> <li>Promedio acumulativo (solo opción de todas las filas)</li> <li>Desviación estándar</li> <li>Concatenar cadenas separadas por el carácter especificado en la columna <b>Valor</b></li> <li>Número de valores distintos</li> <li>Número de filas (sin argumento de campo).</li> </ul>
Valor	El valor agregado



# Ejemplos

- Los ejemplos incluidos en el directorio design-tools \ data-integration \ samples \ transformations son:
- Calculate median and percentiles using the group by steps.ktr
- General - Repeat fields - Group by - Denormalize.ktr
- Group By - Calculate standard deviation.ktr
- Group By - include all rows without a grouping.ktr
- Group by - include all rows and calculations .ktr.

# Use Group By con Spark

Las siguientes diferencias se producen cuando se utiliza el paso **por grupo** con Spark:

- Los nombres de los campos no pueden contener espacios, guiones ni caracteres especiales, y deben comenzar con una letra.
- Las filas están ordenadas por los campos de agrupación.
- El paso **Sort** antes del paso **Group By** es opcional. Las transformaciones existentes que contienen un paso de **Sort** antes del paso **Group By** se ejecutarán con éxito.
- Los pasos **Group By** y **Memory Group By** funcionan igual.
- Si selecciona la opción **Include All Rows**, no puede usar el tipo agregado de Número de valores distintos.
- El **Temporary files directory** y las opciones de **TMP-file prefix** no se aplican. Los archivos temporales no se crean cuando se ejecutan con Spark.

## Nota

Antes de poder utilizar el paso por grupo con Spark, debe configurar la [capa de ejecución adaptable \(AEL\)](#)

# Soporte de inyección de metadatos

Todos los campos de este paso admiten la inyección de metadatos. Puede usar este paso con la [inyección de metadatos de ETL](#) para pasar los metadatos a su transformación en tiempo de ejecución.

Los valores de inyección de metadatos para el tipo de agregación son:

SUM  
AVERAGE  
MEDIAN  
PERCENTILE  
MIN  
MAX  
COUNT\_ALL

CONCAT\_COMMMA  
FIRST  
LAST  
FIRST\_INCL\_NULL  
LAST\_INCL\_NULL  
CUM\_SUM  
CUM\_AVG

STD\_DEV  
CONCAT\_STRING  
COUNT\_DISTINCT  
COUNT\_ANY

## Nota

La inyección de metadatos no es compatible con los pasos que se ejecutan en la [capa de ejecución adaptable \(AEL\)](#).

# Transformation Executor

- El paso del Ejecutor de Transformación le permite ejecutar una transformación de Integración de Datos Pentaho (PDI). Es similar al paso [Job Executor](#), pero funciona con transformaciones.
- Dependiendo de sus necesidades de transformación de datos, el paso del Ejecutor de Transformación se puede configurar para funcionar de cualquiera de las siguientes maneras:
- De forma predeterminada, la transformación especificada se ejecutará una vez para cada fila de entrada. Puede utilizar la fila de entrada para establecer parámetros y variables. El paso del ejecutor luego pasa esta fila a la transformación en forma de una fila de resultados.
- También puede pasar un grupo de registros según el valor de un campo, de modo que cuando el valor del campo cambie dinámicamente, se ejecute la transformación especificada. En estos casos, la primera fila del grupo de filas se utiliza para establecer parámetros o variables en la transformación.
- Puede iniciar varias copias de este paso para ayudar en el proceso de transformación paralela.

## Nota

*Esta función no es compatible actualmente con la capa de ejecución adaptable (AEL).*

# Consideraciones de AEL

- El paso del Ejecutor de Transformación se puede usar para ejecutar una sub-transformación con Spark en AEL. Cuando utilice AEL con el paso Ejecutor de transformación, tenga en cuenta las siguientes excepciones:
- El **Field to use** en la pestaña Parámetros no es compatible con AEL.
- Los resultados de la ejecución no muestran el registro detallado de la transformación secundaria en AEL.

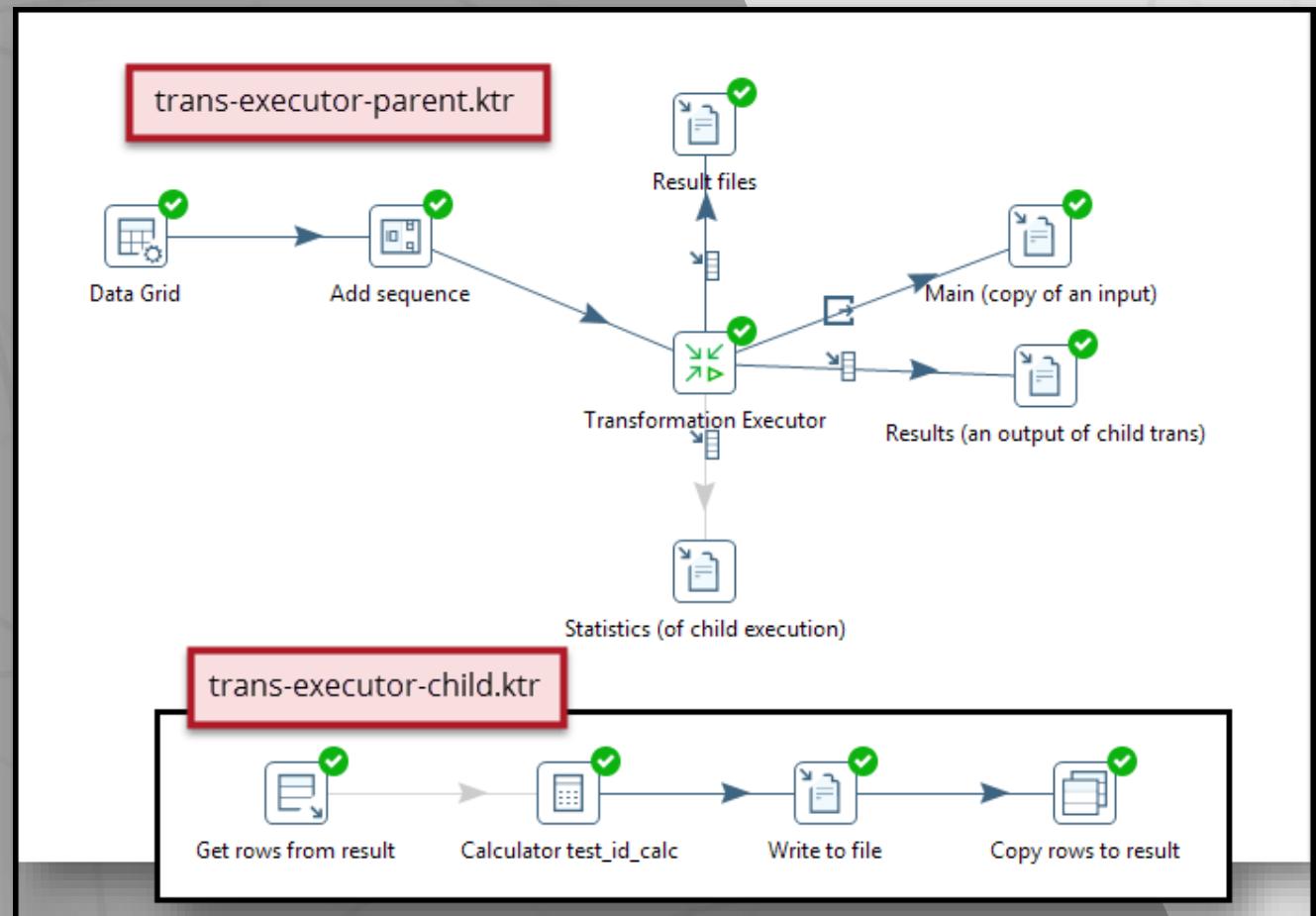
# Notas de manejo de errores y transformación de padres

- Este paso no se cancela cuando se produce un error en la transformación de llamada. Para controlar el flujo o abortar la transformación en caso de errores, especifique los campos y un paso de destino en la pestaña **Execution results** para registrar el número de errores.
- Durante la implementación real, el registro de la transformación principal solo contiene el último lote de datos procesado. Este método disminuye la tensión en el registro de back-end. Puede obtener un registro detallado de la transformación secundaria viendo los resultados de la ejecución. Asegúrese de definir un paso de destino dentro de las pestañas Resultado de ejecución y vea el nombre del campo del texto de registro de ejecución, de manera predeterminada, 'ExecutionLogText'.

# Ejemplo

Estas transformaciones de muestra demuestran las capacidades de este paso. Las muestras están disponibles en el paquete de distribución y se encuentran en la carpeta de design-tools/data-integration/samples/transformations/transformation-executor .

- trans-executor-child.ktr : agrega una secuencia para ingresar filas.
- trans-executor-parent.ktr : pasa las filas a una transformación que luego se ejecuta tres veces. Puede ver los pasos de resultados, archivos de resultados y filas de resultados para ver el resultado.

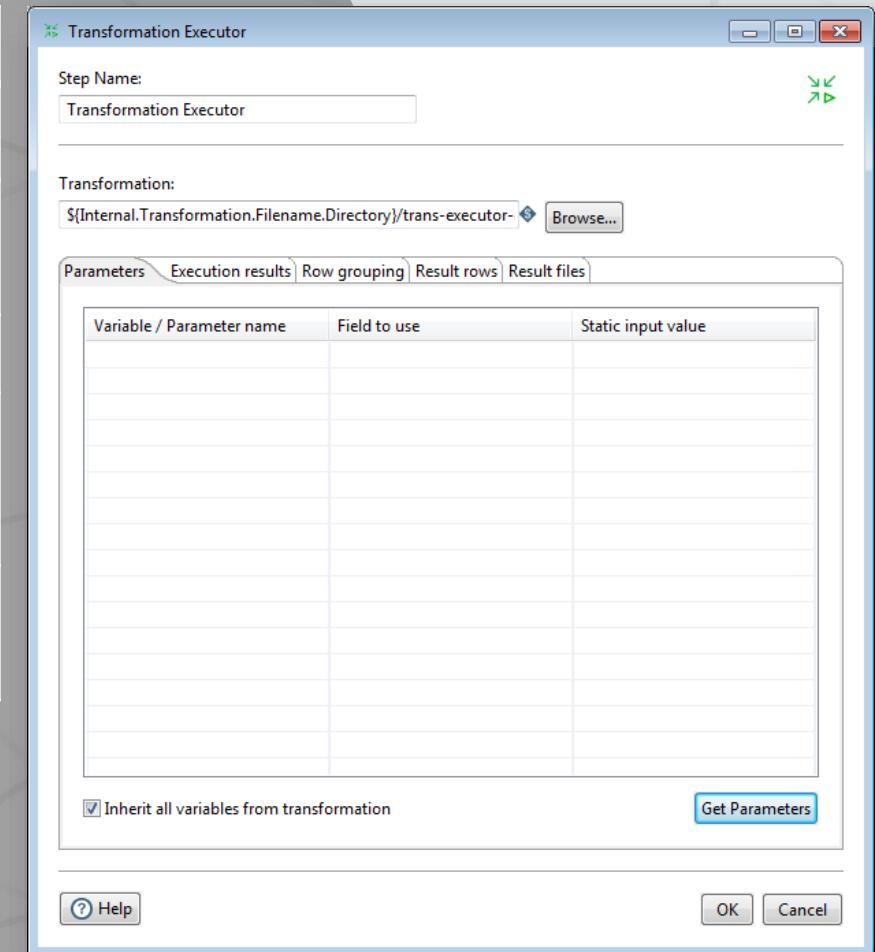


# General

Opciones	Descripción
Step Name	Especifica el nombre único de la transformación en el lienzo. Una transformación se puede colocar en el lienzo varias veces; sin embargo, representa la misma transformación. El <b>Step Name</b> se establece en 'Ejecutor de transformación' de forma predeterminada.
Transformation	Especifique su transformación para ejecutar ingresando su ruta o haciendo clic en <b>Browse</b> . Si selecciona una transformación que tenga la misma ruta de acceso raíz que la transformación actual, la variable \${Internal.Entry.Current.Directory} se insertará automáticamente en lugar de la ruta de acceso raíz común. Por ejemplo, si la ruta de la transformación actual es /home/admin/transformation.ktr y selecciona una transformación en la carpeta /home/admin/path/sub.ktr que la ruta se convertirá automáticamente a \${Internal.Entry.Current .Directorio} /path/sub.ktr. Si está trabajando con un repositorio, especifique el nombre de la transformación. Si no está trabajando con un repositorio, especifique el nombre de archivo XML de la transformación. Las transformaciones previamente especificadas por referencia se convierten automáticamente para ser especificadas por el nombre de transformación dentro del Repositorio Pentaho

# Pestaña Parameters

Opción	Descripción
Variable/Parameter name	Especifique el nombre de la variable o parámetro.
Field to use	Especifique qué campo establecer un parámetro particular o valor de variable. Si especifica un campo de entrada para usar, el valor de entrada estática no se usa.
Static input value	En lugar de un campo, especifique un valor estático para usar.
Inherit all variables from transformation (check box)	Cuando esta casilla de verificación está seleccionada, todas las variables definidas en la transformación actual se pasan a la transformación especificada.
Get Parameters (button)	Haga clic en este botón para insertar todos los parámetros definidos de la transformación especificada. La descripción del parámetro se inserta en el campo de valor de entrada estática.



# Pestaña Execution Results

Opción	Descripción
The target step for the execution results	Use el menú desplegable para seleccionar un paso en la transformación actual como el paso objetivo para recibir los resultados de la transformación especificada.
Execution time (ms)	Especifique el nombre del campo para el tiempo de ejecución de la transformación.
Execution result	Especifique el nombre del campo para el resultado de la ejecución de la transformación.
Number of errors	Especifique el nombre del campo para el número de errores durante la ejecución de la transformación.
Number of rows read	Especifique el nombre del campo para el número total de filas leídas durante la ejecución de la transformación.
Number of rows written	Especifique el nombre del campo para el número total de filas escritas durante la ejecución de la transformación.
Number of rows input	Especifique el nombre del campo para el número total de filas de entrada durante la ejecución de la transformación.
Number of rows output	Especifique el nombre del campo para el número total de filas de salida durante la ejecución de la transformación.

Parameters Execution results Row grouping Result rows Result files

Target step for execution results:  
Statistics (of child execution)

Field description	Field name
Execution time (ms)	ExecutionTime
Execution result	ExecutionResult
Number of errors	ExecutionNrErrors
Number of rows read	ExecutionLinesRead
Number of rows written	ExecutionLinesWritten
Number of rows input	ExecutionLinesInput
Number of rows output	ExecutionLinesOutput
Number of rows rejected	ExecutionLinesRejected
Number of rows updated	ExecutionLinesUpdated
Number of rows deleted	ExecutionLinesDeleted
Number of files retrieved	ExecutionFilesRetrieved
Exit status	ExecutionExitStatus
Execution logging text	ExecutionLogText
Log channel ID	ExecutionLogChannelId

# Pestaña Execution Results

Opción	Descripción
Number of rows rejected	Especifique el nombre del campo para el número total de filas rechazadas durante la ejecución de la transformación.
Number of rows updated	Especifique el nombre del campo para el número total de filas actualizadas durante la ejecución de la transformación.
Number of rows deleted	Especifique el nombre del campo para el número total de filas eliminadas durante la ejecución de la transformación.
Number of files retrieved	Especifique el nombre del campo para el número total de archivos recuperados durante la ejecución de la transformación.
Exit status	Especifique el nombre del campo para el estado de salida de la ejecución de la transformación.
Execution logging text	Especifique el nombre del campo para el texto de registro a partir de la ejecución de la transformación.
Log channel ID	Especifique el nombre del campo para el ID del canal de registro utilizado durante la ejecución de la transformación.

Parameters **Execution results** Row grouping Result rows Result files

Target step for execution results:  
Statistics (of child execution)

Field description	Field name
Execution time (ms)	ExecutionTime
Execution result	ExecutionResult
Number of errors	ExecutionNrErrors
Number of rows read	ExecutionLinesRead
Number of rows written	ExecutionLinesWritten
Number of rows input	ExecutionLinesInput
Number of rows output	ExecutionLinesOutput
Number of rows rejected	ExecutionLinesRejected
Number of rows updated	ExecutionLinesUpdated
Number of rows deleted	ExecutionLinesDeleted
Number of files retrieved	ExecutionFilesRetrieved
Exit status	ExecutionExitStatus
Execution logging text	ExecutionLogText
Log channel ID	ExecutionLogChannelId

# Pestaña Row Grouping

Opción	Descripción
Number of rows to send to transformation	Especifique un número. Después de cada 'X' número de filas, el trabajo se ejecutará y estas filas 'X' pasarán a la transformación.
Field to group rows on	Especifique un campo para agrupar filas. Las filas se recopilarán en un grupo siempre que el valor del campo permanezca igual. Si el valor cambia, la transformación se ejecutará y las filas acumuladas se pasarán a la transformación.
Duration time when collecting rows	Especifique un tiempo en milisegundos. Este valor es la cantidad de tiempo que el paso pasará recolectando filas antes de la ejecución de la transformación.

Parameters Execution results Row grouping Result rows Result files

Number of rows to send to transformation:  
1000

Field to group rows on:

Duration time when collecting rows:

# Pestaña Result Rows

Opción	Descripción
Target step for result rows	Use el menú desplegable para seleccionar un paso en la transformación actual como el paso objetivo.
Field name	Especifique el nombre del campo.
Data type	Use el menú desplegable para especificar el tipo de datos del campo, como el número, la fecha o la cadena.
Length	Si corresponde, especifique la longitud del tipo de datos especificado.
Precision	Si corresponde, especifique la precisión a utilizar.

Parameters Execution results Row grouping **Result rows** Result files

Target step for result rows:  
Results (an output of child trans)

Expected layout for result rows:

Field name	Data type	Length	Precision
test_string	String	50	
test_id	Integer		0
test_id_calc	Integer		0

# Pestaña Result Files

Opción	Descripción
Target step for result files information	Use el menú desplegable para seleccionar un paso en la transformación como el paso objetivo.
Result file name field	Especifique el nombre del campo para los archivos de resultados.

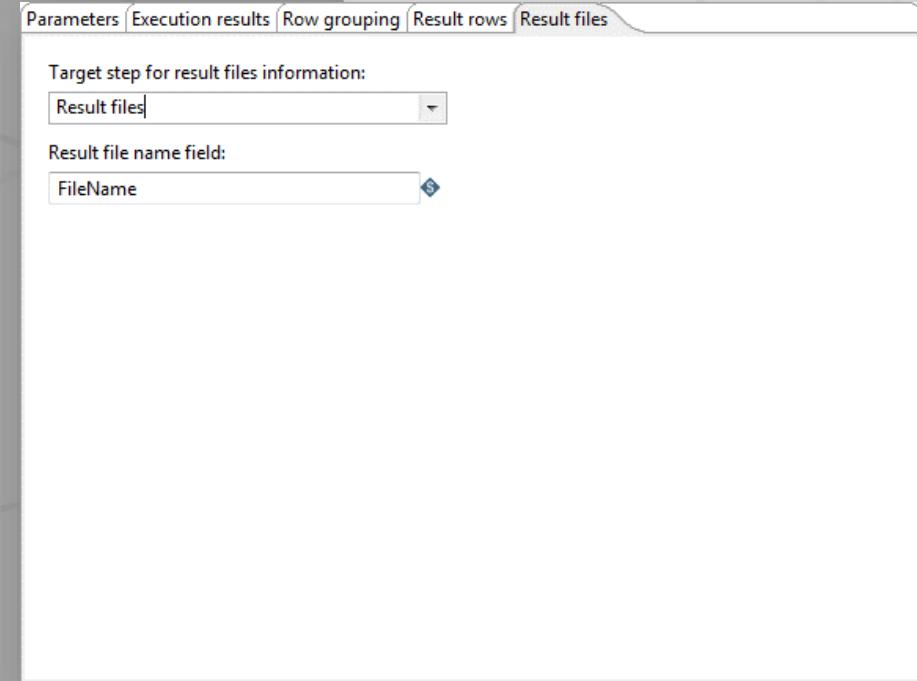
Parameters Execution results Row grouping Result rows Result files

Target step for result files information:

Result files

Result file name field:

FileName

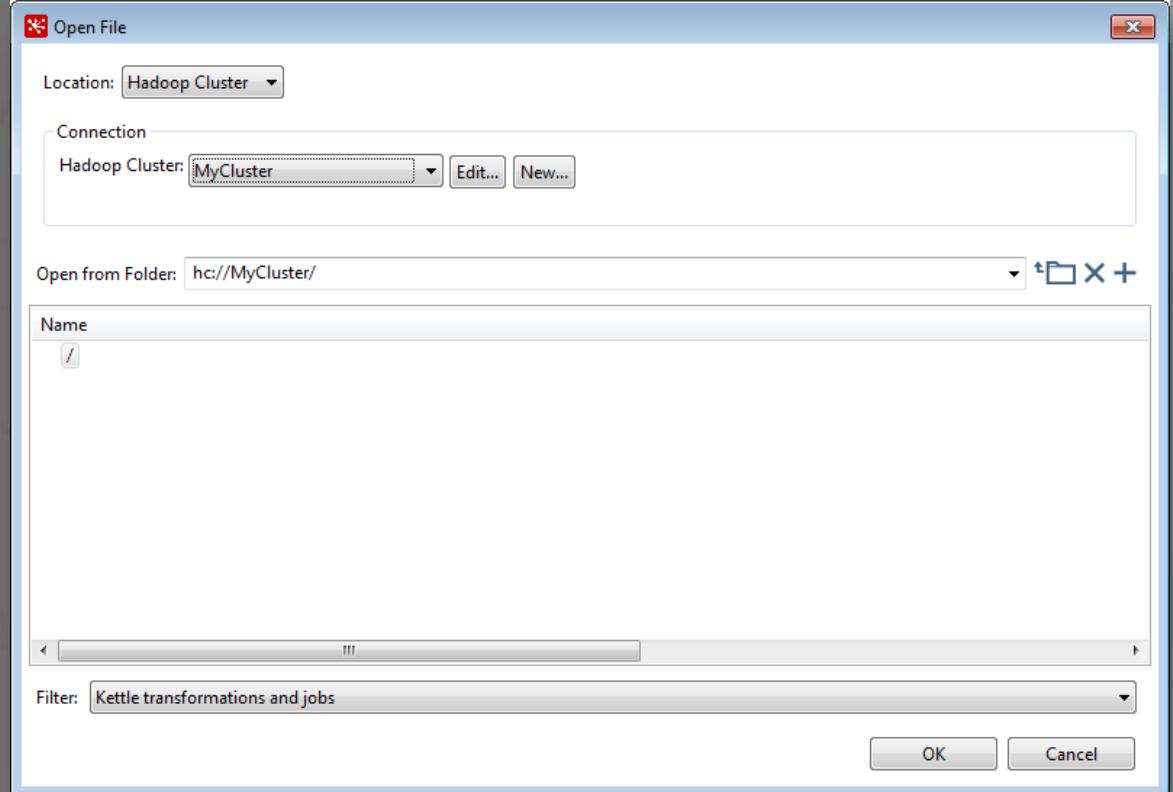


# Virtual File System Browser

- Algunos pasos de transformación y entradas de trabajo utilizan cuadros de diálogo del sistema de archivos virtuales (VFS) en lugar de las ventanas tradicionales del sistema de archivos locales. Con los cuadros de diálogo de archivos VFS, puede especificar una URL VFS en lugar de una ruta local típica. Se accede a los archivos mediante HTTP, con las URL que contienen datos de esquema que identifican un protocolo para usar. Consulte <http://commons.apache.org/vfs/apidocs/index.html> para ver la documentación del esquema VFS. Sus archivos pueden ser locales o remotos. También pueden residir en formatos comprimidos como .tar , .zip u otros formatos comprimidos.



# Virtual File System Browser



## Nota

Las siguientes direcciones son ejemplos de URL de VFS:

**Local:** [ftp://userID:password@ftp.myhost.com/path\\_to/file.txt](ftp://userID:password@ftp.myhost.com/path_to/file.txt)

**S3:** [s3n://s3n/\(s3\\_bucket\\_name\)/\(absolute\\_path\\_to\\_file\)](s3n://s3n/(s3_bucket_name)/(absolute_path_to_file))

**HDFS:** [hdfs://nombre\\_usuario:mypassword@mynamenode:puerto/ruta](hdfs://nombre_usuario:mypassword@mynamenode:puerto/ruta)

## Pasos

1. Seleccione **File > Open URL** en el cliente PDI para abrir el navegador VFS como se muestra en la siguiente figura
2. Elija el tipo de sistema de archivos de su **Location**. Los siguientes sistemas de archivos son compatibles:
  - **Local** : abre archivos en su máquina local. Utilice las carpetas en el panel Nombre del cuadro de diálogo Abrir archivo para seleccionar un recurso.
  - **Hadoop Cluster** : abre archivos en cualquier clúster de Hadoop excepto S3. Haga clic en el cuadro desplegable Clúster de Hadoop para seleccionar el clúster deseado y luego el recurso al que desea acceder.
  - **S3** : (Servicio de almacenamiento simple) accede a los recursos en los servicios web de Amazon. Para obtener instrucciones sobre cómo configurar las credenciales de AWS, consulte Cómo [trabajar con las credenciales de AWS](#) .
  - **HDFS** : abre archivos en cualquier sistema de archivos distribuidos de Hadoop, excepto MapR. Seleccione el clúster que desee para la opción **Hadoop Cluster** y luego seleccione el recurso al que desea acceder.
  - **MapRFS** : abre archivos en el sistema de archivos MapR. Utilice las carpetas en el panel Nombre del cuadro de diálogo Abrir archivo para seleccionar un recurso MapR.
  - **Google Cloud Storage** : abre archivos en el sistema de archivos Google Cloud Storage.
  - **Google Drive** : abre archivos en el sistema de archivos de Google. Debe configurar PDI en el acceso inicial al sistema de archivos de Google. Consulte [Acceso a un Google Drive](#) para obtener más información.

# Acceso a Google Drive

## Pasos

1. Active la API de Google Drive, que da como resultado un archivo credentials.json . Consulte <https://developers.google.com/drive/api/v3/quickstart/java> para obtener más información.
2. Cambie el nombre de su archivo credentials.json a client\_secreat.json y cópielo en el directorio data-integration / plugins / pentaho-googledrive-vfs / credentials , y reinicie PDI. La opción **Google Drive** no aparecerá como una **Location** hasta que copie el archivo client\_secreat.json en el directorio de credenciales y reinicie.
3. Seleccione **Google Drive** como su **Location** . Se le solicitará que inicie sesión en su cuenta de Google.
4. Una vez que haya iniciado sesión, aparecerá la pantalla de permiso de Google Drive.
5. Haga clic en **Allow** para acceder a sus recursos de Google Drive.

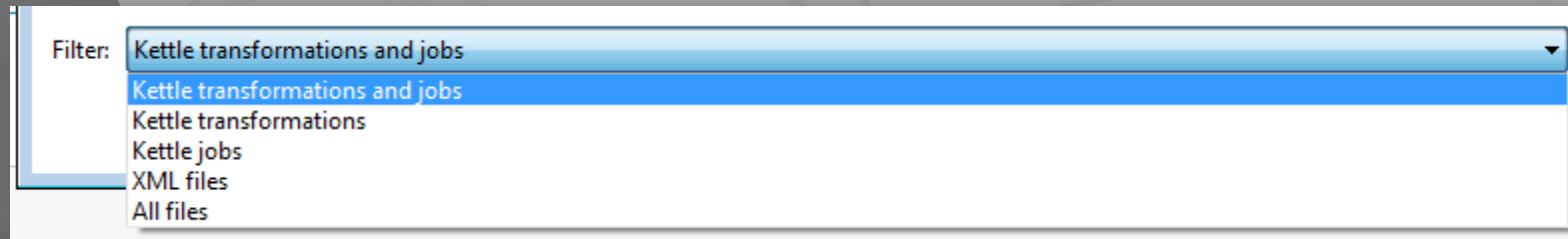
Pentaho luego almacena un token de seguridad llamado StoredCredential en el directorio de integración de datos / plugins / pentaho-googledrive-vfs / credentials . Con este token, puede acceder a sus recursos de Google Drive siempre que inicie sesión en su cuenta de Google. Si alguna vez se elimina este token de seguridad, se le solicitará nuevamente que inicie sesión en su cuenta de Google después de reiniciar el PDI. Si alguna vez cambia los permisos de su cuenta de Google, debe eliminar el token y repetir los pasos anteriores para generar un nuevo token.

# Acceso a Google Drive

Si desea acceder a su Google Drive a través de una transformación que se ejecuta directamente en su servidor Pentaho, copie los archivos `StoredCredential` y `client_secret.json` en el directorio `pentaho-server / pentaho-solutions / system / kettle / plugins / pentaho-googledrive-vfs / credentials` en su servidor Pentaho.

# Agregar y eliminar carpetas o archivos

También puede usar el navegador VFS para eliminar archivos o carpetas en su sistema de archivos. Se aplica un filtro predeterminado para que inicialmente se muestren los archivos de trabajo y la transformación de Kettle. Para ver otros archivos, haga clic en el menú desplegable **Filter** y seleccione el tipo de archivo que desea seleccionar. Una vez que haya seleccionado el archivo o la carpeta que desea eliminar, haga clic en la X en la esquina superior derecha del navegador VFS para eliminar su selección. Si desea crear una nueva carpeta, haga clic en + en la esquina superior derecha del navegador VFS, ingrese el nombre de su nueva carpeta y haga clic en **OK**.



# Pasos y entradas soportados

Los pasos de transformación admitidos y las entradas de trabajo abren el VFS browser en lugar del cuadro de diálogo de abrir archivo tradicional. Con el VFS browser, usted especifica una URL VFS en lugar de una ruta de archivo para acceder a esos recursos.

Los siguientes pasos y entradas son compatibles con el VFS browser:

[File Exists](#)

[Mapping](#) (sub-transformation)

[ETL Metadata Injection](#)

[Hadoop Copy Files](#)

[Hadoop File Input](#)

[Hadoop File Output](#)

[Avro Input](#)

[Avro Output](#)

[ORC Input](#)

[ORC Output](#)

[Parquet Input](#)

[Parquet Output](#)

## Nota

Los cuadros de diálogo VFS se configuran a través de ciertos parámetros de transformación. Consulte [Configurar SFTP VFS](#) para obtener más información sobre cómo configurar las opciones para SFTP.

# Configurar las opciones de VFS

El navegador VFS se puede configurar para establecer variables como parámetros para su uso en tiempo de ejecución. Una transformación de ejemplo VFS Configuration Sample.ktr que contiene algunos ejemplos de los parámetros que puede establecer se encuentra en el directorio data-integration / samples / transformations . Para obtener más información sobre la configuración de variables, consulte [VFS Properties](#). Para ver un ejemplo de la configuración de una conexión SFTP VFS, consulte [Configure SFTP VFS](#).

# Google BigQuery Loader

- La entrada de trabajo de Google BigQuery Loader le permite cargar datos en Google BigQuery desde una cuenta de Google Cloud Storage. Google BigQuery Loader admite los siguientes formatos:
- Valores separados por comas (CSV)
- JSON (delimitado por nueva línea)
- Avro.

# Antes de que empieces

Debe tener una cuenta de Google y debe crear credenciales de cuenta de servicio en forma de archivo clave en formato JSON para utilizar la entrada de trabajo de Google BigQuery Loader. También debe establecer permisos para sus cuentas de BigQuery y Google Cloud. Para crear credenciales de cuenta de servicio, consulte <https://cloud.google.com/storage/docs/authentication>.

Realice los siguientes pasos para configurar su sistema para usar Google BigQuery:

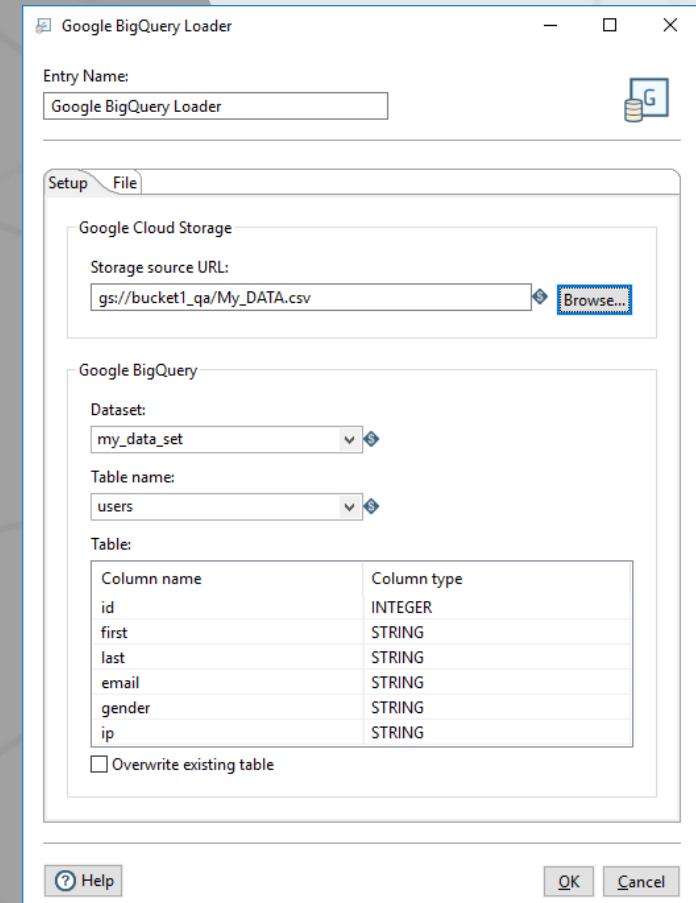
1. Descargue el archivo de credenciales de la cuenta de servicio que creó con la Google API Console en su máquina local.
2. Cree una nueva variable de entorno del sistema en su sistema operativo llamada "GOOGLE\_APPLICATION\_CREDENTIALS".
3. Establezca la ruta al archivo de credenciales de cuenta de servicio JSON descargado como el valor de la variable GOOGLE\_APPLICATION\_CREDENTIALS.
4. Reinicie su máquina local.

## Nota

La variable de entorno y las credenciales deben configurarse en cada máquina que ejecuta el trabajo de BigQuery Loader. El cuadro de diálogo de Google BigQuery Loader no se abrirá para editarlo en el lienzo del trabajo hasta que se complete este procedimiento.

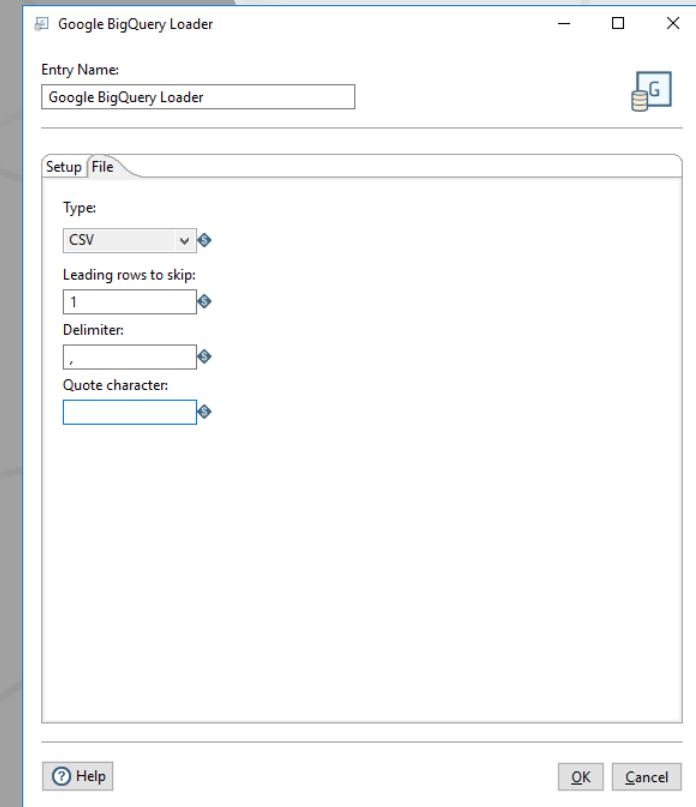
# Pestaña Setup

Campo	Descripción
Storage source URL	Especifique la URL de almacenamiento en la nube de Google de los datos para importar. La URL puede apuntar a un archivo o una carpeta en Google Cloud Storage. La URL debe comenzar con gs:// y debe especificar el grupo y el objeto que desea cargar. Debe especificar el tipo de archivo en la pestaña <b>File</b> .
Dataset	Especifique el conjunto de datos donde desea importar una tabla. El menú desplegable del conjunto de datos se completa automáticamente cuando selecciona la <b>Storage source URL</b> , pero puede ingresar un nuevo nombre de conjunto de datos en el campo. Si el conjunto de datos no existe, se creará en tiempo de ejecución.
Table name	Especifique el nombre de la tabla en el conjunto de datos donde desea importar datos.
Column name	Especifique el nombre de la columna en la tabla de conjunto de datos donde desea importar datos.
Column type	Especifique el tipo de columna en la tabla de conjunto de datos donde desea importar datos.
Overwrite existing table	Seleccione para sobrescribir los datos existentes con datos importados.



# Pestaña File

Campo	Descripción
Type	Especifique el tipo de archivo. Los valores son CSV (predeterminado), JSON y Avro. Debe especificar el tipo de archivo correcto que está asociado con el campo <b>Storage source URL</b> en la pestaña <b>Setup</b> .
Leading rows to skip	Especifique cuántas filas del archivo CSV se omitirán.
Delimitador	Especifique el carácter delimitador utilizado por el archivo CSV.
Quote character	Especifique el carácter de escape (cita) utilizado para los valores que tienen el carácter delimitador en ellos. Por ejemplo, cuando el carácter delimitador es una coma y un campo contiene una coma, y el carácter de cita es una barra invertida, al insertar una barra diagonal inversa antes de la coma en el campo, se evitará que ese campo se evalúe como el comienzo de un nuevo valor de campo.



# Configurar la capa de ejecución adaptable (AEL)

Pentaho utiliza la [Adaptive Execution Layer \(AEL\)](#) para ejecutar transformaciones en diferentes motores. AEL adapta los pasos de una transformación desarrollada en PDI a operadores nativos en el motor que selecciona para su entorno, como Spark en un clúster de Hadoop. El demonio AEL construye una definición de transformación en Spark, que mueve la ejecución directamente al clúster.

Su instalación de Pentaho 8.1 incluye el demonio AEL que puede configurar para que la producción se ejecute en sus clústeres. Después de configurar el demonio AEL, el cliente PDI se comunica tanto con su clúster Spark como con el demonio AEL, que vive en un nodo de su clúster para iniciar y ejecutar transformaciones.

# Configurar la capa de ejecución adaptable (AEL)

Antes de que pueda seleccionar el motor Spark a través de las configuraciones de ejecución, deberá configurar AEL para su sistema y su flujo de trabajo. Dependiendo de su implementación, es posible que deba realizar tareas de configuración adicionales, como configurar AEL en un clúster seguro.

## Advertencia

La capa de ejecución adaptable (AEL) admite la mayoría de los pasos estándar de PDI; Sin embargo, hay algunos pasos que no son compatibles. Por ejemplo, la inyección de metadatos (MDI) no se admite actualmente para los pasos que se ejecutan en AEL.

# Antes de que empieces

Debe cumplir con los siguientes requisitos para usar el demonio AEL y operar el motor Spark para transformaciones:

- Pentaho 8.1 o posterior instalación. Ver la [instalación de Pentaho](#).
- [Cloudera](#) versión 5.10 o posterior o [Hortonworks](#) 2.5 o posterior distribución de Hadoop.
- [Spark Client 2.0, 2.1 y 2.2](#).
- [Aplicación Pentaho Spark](#).
- Si está configurando AEL para su uso con Cloudera, Hortonworks, MapR o Amazon EMR, revise [los Clientes suministrados por el proveedor](#).

## Advertencia

La dependencia de Zookeeper se ha eliminado de Pentaho 8.0. Si instaló AEL para Pentaho 7.1, debe eliminar la carpeta de adaptive-execution y seguir las instrucciones de instalación de Pentaho 8.0 para utilizar AEL con Pentaho 8.0.

# Instalación de Pentaho 8.1

Cuando instala el servidor Pentaho, AEL daemon se instala en la carpeta de integración de datos / ejecución adaptativa . Esta carpeta se denominará 'PDI \_AEL\_DAEMON\_HOME'.

## Spark Client

El cliente Spark es necesario para el funcionamiento del demonio AEL. Las versiones recomendadas del cliente Apache Spark son 2.0, 2.1 y 2.2. Realice los siguientes pasos para instalar el cliente Spark.

- Descargue el cliente Spark, spark-2.1.0-bin-hadoop2.7.tgz, desde <http://spark.apache.org/downloads.html> .
- Extráigalo a una carpeta en el clúster donde el daemon pueda acceder a él. Esta carpeta se denominará la variable 'SPARK\_HOME' .

# Aplicación Pentaho Spark

La aplicación Pentaho Spark se basa en el motor Kettle de PDI, que permite que las transformaciones se ejecuten sin alteraciones dentro de un clúster de Hadoop. Es posible que algunos complementos de terceros, como los complementos disponibles en Pentaho Marketplace, no se incluyan de forma predeterminada dentro de la aplicación Pentaho Spark. Para solucionar este problema, incluimos la herramienta de creación de aplicaciones Spark para que pueda personalizar la aplicación Pentaho Spark agregando o eliminando componentes que se ajusten a sus necesidades.

# Aplicación Pentaho Spark

Después de ejecutar la herramienta de creación de aplicaciones Spark, copie y descomprima el archivo `pdi-spark-driver.zip` resultante en un nodo de su clúster Hadoop. El contenido desempaquetado consiste en la carpeta `data-integration` y el archivo `pdi-spark-executor.zip`, que incluye solo las bibliotecas requeridas por los propios nodos Spark para ejecutar una transformación cuando el demonio AEL está configurado para ejecutarse en modo YARN. Dado que todos los nodos del clúster deben poder acceder al archivo `pdi-spark-executor.zip`, debe copiarse en HDFS. Spark distribuye este archivo `.zip` a otros nodos y luego lo extrae automáticamente.

# Aplicación Pentaho Spark

Realice los siguientes pasos para ejecutar la herramienta de compilación de la aplicación Spark y administrar los archivos resultantes.

1. Asegúrese de haber configurado su cliente PDI con todos los complementos que utilizará.
2. Vaya a la carpeta de herramientas de diseño / integración de datos y localice s -park-app-builder.bat (Windows) o la chispa-app-constructor.sh (Linux).
3. Ejecute el script de la herramienta del generador de aplicaciones Spark. Aparecerá una ventana de la consola y se creará el archivo pdi-spark-driver.zip en la carpeta de integración de datos (a menos que se especifique lo contrario en el parámetro -outputLocation descrito a continuación).

# Aplicación Pentaho Spark

Los siguientes parámetros se pueden usar al ejecutar el script para compilar el pdi-spark-driver.zip .

Parámetro	Acción
<code>-h or --help</code>	Muestra la ayuda.
<code>-e or --exclude-plugins</code>	Especifica complementos de la carpeta <code>data-integration/plugins</code> que no se excluyen del ensamblaje.
<code>-o or --outputLocation</code>	Especifica la ubicación de salida.

# Aplicación Pentaho Spark

4. El archivo pdi-spark-driver.zip contiene una carpeta de integración de datos y el archivo pdi-spark-executor.zip . Copie la carpeta de integración de datos en el nodo de borde donde desea ejecutar el daemon de AEL.
5. Copie el archivo pdi-spark-executor.zip en el nodo HDFS donde ejecutará Spark. Esta carpeta se denominará 'HDFS\_SPARK\_EXECUTOR\_LOCATION' .

## Nota

Para que los nodos del clúster utilicen la funcionalidad proporcionada por los complementos PDI al ejecutar una transformación, deben instalarse en el cliente PDI antes de generar la aplicación Pentaho Spark. Si instala otros complementos más tarde, debe volver a generar la aplicación Pentaho Spark.



# Configurando el demonio AEL para el modo local

Puede configurar el demonio AEL para que se ejecute en el modo local de Spark con fines de desarrollo o demostración. Esto le permitirá compilar y probar una aplicación Spark en su escritorio con datos de muestra y luego reconfigurar la aplicación para que se ejecute en sus clústeres. Para configurar el demonio AEL para un modo local, complete los siguientes pasos:

## Pasos

1. Navegue hasta el directorio ... / data-integration/adaptive-execution/config y abra el archivo application.properties.
2. Establezca las siguientes propiedades para su entorno:
  - Establezca la propiedad sparkHome la sparkHome de acceso de Spark 2 en su máquina local.
  - Establezca la propiedad sparkApp en el directorio de data-integration.
  - Establezca la propiedad hadoopConfDir en el directorio que contiene los archivos \*site.xml .
3. Guarde y cierre el archivo.
4. Vaya a la carpeta de integración de datos / ejecución adaptativa y ejecute el comando daemon.sh desde la interfaz de línea de comandos.

## Nota

La configuración del demonio AEL para que se ejecute en el modo local de Spark no es compatible, pero puede ser útil para el desarrollo y la depuración..

# Configurando el demonio AEL en modo YARN

Normalmente, el demonio AEL se ejecuta en modo YARN para fines de producción. En el modo YARN, la aplicación del controlador se inicia y los delegados trabajan en el clúster YARN. La aplicación pdi-spark-executor debe estar instalada en cada uno de los nodos YARN.

Para configurar el demonio AEL para un entorno de producción YARN, complete los siguientes pasos.

1. Navegue hasta el directorio `.../data-integration/adaptive-execution/config` y abra el archivo `application.properties`.
2. Establezca las siguientes propiedades para su entorno:

## Nota

El script `daemon.sh` solo se admite en entornos basados en UNIX.

# Configurando el demonio AEL en modo YARN

Normalmente, el demonio AEL se ejecuta en modo YARN para fines de producción. En el modo YARN, la aplicación del controlador se inicia y los delegados trabajan en el clúster YARN. La aplicación pdi-spark-executor debe estar instalada en cada uno de los nodos YARN.

Para configurar el demonio AEL para un entorno de producción YARN, complete los siguientes pasos.

1. Navegue hasta el directorio `.../data-integration/adaptive-execution/config` y abra el archivo `application.properties`.
2. Establezca las siguientes propiedades para su entorno:

# Configurando el demonio AEL en modo YARN

Propiedad	Valor
websocketURL	El nombre de dominio completo del nodo donde está instalado el demonio AEL. Por ejemplo, websocketURL = <a href="ws://localhost:\$ael.unencrypted.port">ws://localhost:\$ael.unencrypted.port</a>
sparkHome	La ruta a la carpeta del cliente Spark en su clúster
sparkApp	El directorio de data-integration
hadoopConfDir	El directorio que contiene los archivos *site.xml . Este valor de propiedad le dice a Spark que clúster de Hadoop / YARN debe usar. Puede descargar el directorio que contiene los archivos *site.xml utilizando la herramienta de administración de clústeres, o puede establecer la propiedad hadoopConfDir en la ubicación del clúster.
hadoopUser	La ID de usuario que usará la aplicación Spark, si no está usando seguridad.
sparkMaster	Yarn
assemblyZip	hdfs:\$HDFS_SPARK_EXECUTOR_LOCATION

# Configurando el demonio AEL en modo YARN

3. Guarde y cierre el archivo.
4. Copie el archivo pdi-spark-executor.zip en su clúster HDFS, como se muestra en el siguiente ejemplo.

```
$ hdfs dfs put pdi-spark-executor.zip /opt/pentaho/pdi-spark-executor.zip
```

5. Ejecute el script de inicio pdi-daemon, daemon.sh desde la interfaz de línea de comandos.

Puede iniciar manualmente el daemon de AEL ejecutando daemon.sh . De forma predeterminada, esta secuencia de comandos de inicio se instala en la carpeta de integración de datos / ejecución adaptativa , que se conoce como la variable 'PDI\_AEL\_DAEMON\_HOME'.

# Configurando el demonio AEL en modo YARN

Realice los siguientes pasos para iniciar manualmente el daemon de AEL.

1. Vaya al directorio de data-integration/adaptive-execution .
2. Ejecute el script daemon.sh

El script daemon.sh soporta los siguientes comandos:

Comando	Acción
daemon.sh	Inicia el daemon como un proceso de primer plano.
daemon.sh start	Inicia el daemon como un proceso en segundo plano. Los registros se escriben en el archivo PDI_AEL_DAEMON_HOME/daemon.log .
daemon.sh stop	Detiene el demonio.
daemon.sh status	Informa el estado del demonio.

# Configurar el registro de eventos

Los eventos de Spark se pueden capturar en un registro de eventos que se puede ver con el servidor de historial de Spark. El Spark History Server es una interfaz de usuario basada en navegador web para el registro de eventos. Puede ver las transformaciones de Spark en ejecución o completadas utilizando el servidor de historial de chispas. Antes de poder utilizar el servidor de historial de Spark, debe configurar AEL para registrar los eventos.

# Configurar el registro de eventos

Realice las siguientes tareas para configurar AEL para registrar eventos:

1. Navegue hasta el directorio de integración de datos / ejecución-adaptable / config y abra el archivo application.properties .
2. Establezca la propiedad sparkEventLogEnabled en true. Si este campo falta o está configurado como falso, Spark no registra eventos.
3. Establezca la propiedad sparkEventLogDir en un directorio donde desea almacenar el registro. Puede ser un directorio del sistema de archivos (por ejemplo, [file: /// users / home / spark-events](#) ), o un directorio HDFS (por ejemplo, hdfs: / usrs / home / spark-events ).
4. Configure la propiedad spark.history.fs.logDirectory para que apunte al mismo directorio de registro de eventos que configuró en el paso anterior.

# Configurar el registro de eventos

Ahora puede ver las transformaciones de PDI utilizando el servidor de historial de Spark. Consulte los siguientes documentos para obtener más información sobre cómo ejecutar el servidor de historial de Spark:

<https://spark.apache.org/docs/latest/monitoring.html>

[https://www.cloudera.com/documentation/enterprise/5-9-x/topics/operation\\_spark\\_applications.html](https://www.cloudera.com/documentation/enterprise/5-9-x/topics/operation_spark_applications.html)

<https://jaceklaskowski.gitbooks.io/mastering-apache-spark/spark-history-server.html>

# Cientes suministrados por el vendedor

Es posible que se requieran pasos de configuración adicionales al utilizar AEL con la versión de un proveedor del cliente Spark.

## Cloudera

Si su cliente Cloudera Spark no contiene las bibliotecas de Hadoop, debe agregar las bibliotecas de Hadoop a la ruta de clase usando la variable de entorno SPARK\_DIST\_CLASSPATH. Para ello pueden utilizar el siguiente comando:

```
exportar SPARK_DIST_CLASSPATH = $ (hadoop classpath)
```

# Hortonworks

La versión de la plataforma de datos Hortonworks (HDP) en su nodo perimetral donde reside su servidor Pentaho debe ser la misma versión utilizada en su clúster o el demonio AEL y el cliente PDI dejarán de funcionar. Para evitar que esto suceda, debe exportar la variable HDP\_VERSION. Por ejemplo:

```
exportar HDP_VERSION = ${HDP_VERSION:-2.6.0.3-8}
```

Puede verificar la versión de HDP en su clúster con el siguiente comando:

```
hdp-select status hadoop-client
```

# MapaR

Para configurar el demonio AEL para que se ejecute en un entorno de producción MapR Spark 2.1, complete los siguientes pasos.

1. Navegue hasta el directorio ... / data-integration / adaptive-execute / config y abra el archivo application.properties .
2. Establezca la siguiente propiedad para su entorno MapR Spark 2.1:

# MapaR

Propiedad	Valor
hadoopConfDir	<p>Esta propiedad identifica el clúster de Hadoop que Spark utilizará.</p> <p>Debido a que MapR identifica el clúster de Hadoop de forma predeterminada, establezca el valor de la propiedad en vacío, como se muestra aquí: <code>hadoopConfDir = ""</code></p>
-Dhadoop.login	<p>Esta propiedad identifica el entorno de seguridad que utilizará el clúster de Hadoop.</p> <p><i>Si habilita la seguridad</i>, el valor de la variable de entorno <code>MAPR_ECOSYSTEM_LOGIN_OPTS</code> incluirá la opción 'JVM híbrida' para la propiedad <code>hadoop.login</code>.</p> <p>Establezca el valor de la propiedad en 'hybrid' para especificar un entorno de seguridad mixto utilizando Kerberos y tecnologías de seguridad internas de MapR como se muestra aquí:</p> <p><code>-Dhadoop.login=hybrid</code></p>
-Djava.security.auth.login.config	<p>Esta propiedad identifica el archivo de configuración a usar cuando habilita la seguridad.</p> <p>La distribución de MapR para Hadoop utiliza el Servicio de Autorización y Autenticación de Java (JAAS) para controlar las características de seguridad. El archivo <code>/opt/mapr/conf/mapr.login.conf</code> especifica los parámetros de configuración para JAAS.</p> <p>Establezca el valor de la propiedad en <code>/opt/mapr/conf/mapr.login.conf</code> como se muestra aquí:</p> <p><code>-Djava.security.auth.login.config=/opt/mapr/conf/mapr.login.conf</code></p>

# MapaR

3. Guarde y cierre el archivo.
4. Antes de ejecutar el demonio, agregue las bibliotecas de Hadoop a la ruta de clase ejecutando el siguiente comando desde el símbolo del sistema (ventana de terminal) en el clúster:

```
exportar SPARK_DIST_CLASSPATH = $(hadoop classpath)
```

Ahora puede probar su configuración de AEL creando una configuración de ejecución utilizando el motor Spark. Consulte [Ejecutar configuraciones](#) para más detalles.

# Amazon EMR

Cuando se ejecuta AEL en Amazon EMR, la compresión LZO y Oracle JDK 8 son componentes necesarios.

## **Soporte LZO**

LZO es un formato de compresión compatible con Amazon EMR. Es necesario para ejecutar AEL en EMR. Para configurar la compresión LZO, deberá agregar varias propiedades.

# Amazon EMR

1. Siga las instrucciones disponibles aquí para instalar la biblioteca de compresión LZO de Linux desde la línea de comandos: [https://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.6.1/bk\\_command-line-installation/content/install\\_compression\\_libraries.html](https://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.6.1/bk_command-line-installation/content/install_compression_libraries.html)
2. Navegue hasta el directorio ... /data-integration/adaptive-execution/config/ y abra el archivo application.properties .
3. Agregue las siguientes propiedades:
  - spark.executor.extraClassPath= /usr/lib/hadoop-lzo/lib/hadoop-lzo.jar
  - spark.driver.extraClassPath = /usr/lib/hadoop-lzo/lib/hadoop-lzo.jar
4. Agregue las siguientes propiedades para incluir -Djava.library.path = / usr / lib / hadoop-lzo / lib / native al final de cada línea:
  - sparkExecutorExtraJavaOptions
  - sparkDriverExtraJavaOptions
5. Guarde y cierre el archivo.

# Oracle JDK 8

Amazon EMR usa Open JDK 8, mientras que Pentaho AEL solo es compatible con Oracle JDK 8. Por lo tanto, los usuarios deben instalar Oracle JDK 8 para ejecutar correctamente sus instancias de EMR en AEL para que sean compatibles. Para acceder a un script de ejemplo para instalar Oracle JDK 8, consulte:

<https://github.com/pentaho/pentaho-engineering-samples/blob/master/Supplementary%20Files/pentaho-EMR/install-oracle-java-8.sh>

## Nota

El contenido que se incluye en este enlace es un ejemplo y no tiene garantía. Hitachi Vantara no será responsable en caso de daños incidentales o consecuentes relacionados con el suministro, rendimiento o uso del contenido proporcionado aquí. El usuario es responsable de aceptar el acuerdo de licencia con Oracle.

# Temas avanzados

Los siguientes temas ayudan a ampliar su conocimiento de la capa de ejecución adaptable más allá de la configuración y el uso básicos:

- [Especificar propiedades de chispa adicionales](#)

Puede definir propiedades Spark adicionales dentro del archivo application.properties o como parámetros de modificación de ejecución dentro de una transformación.

- [Configurando AEL con Spark en un cluster seguro](#)

Si su servidor de daemon AEL y sus máquinas de clúster están en un entorno seguro como un centro de datos, es posible que solo desee configurar una conexión segura entre el cliente PDI y el servidor de daemon AEL.

# Temas avanzados

Los siguientes temas ayudan a ampliar su conocimiento de la capa de ejecución adaptable más allá de la configuración y el uso básicos:

- [Especificar propiedades de chispa adicionales](#)

Puede definir propiedades Spark adicionales dentro del archivo application.properties o como parámetros de modificación de ejecución dentro de una transformación.

- [Configurando AEL con Spark en un cluster seguro](#)

Si su servidor de daemon AEL y sus máquinas de clúster están en un entorno seguro como un centro de datos, es posible que solo desee configurar una conexión segura entre el cliente PDI y el servidor de daemon AEL.

# Otros pasos recomendados

[S3 CSV Input](#)

[S3 File Output](#)

[ORC Input](#)

[ORC Output](#)

[Avro Input](#)

[Avro Output](#)

[Parquet Input](#)

[Parquet Output](#)

[Cassandra Input](#)

[Cassandra Output](#)

[SSTable Output](#)

[HBase Input](#)

[HBase Output](#)

[MongoDB Input](#)

[MongoDB Output](#)

[Splunk Input](#)

[Splunk Output](#)

[Salesforce Input](#)

Manipulación de  
datos y metadatos de  
PDI





# TEMAS

- Nuevas formas de transformar datos
- Entendiendo el paso de los valores de selección y sus diferentes usos.
- Obtención de información del sistema y variables predefinidas.
- Manipulación de estructuras XML y JSON.

# Manipulación de datos y metadatos de PDI

- En los capítulos anteriores, aprendió los conceptos básicos de la transformación de datos. Este capítulo expande sus conocimientos al enseñarle una variedad de características esenciales. Además de explorar nuevos pasos PDI para la manipulación de datos, este capítulo explica en detalle un recurso clave en cada proceso: el paso **Select values**. El capítulo también le enseña cómo obtener información del sistema y variables predefinidas para utilizarlas como parte del flujo de datos. Una sección especial está dedicada a leer y escribir estructuras XML y JSON.

# Manipulación de campos simples.

Como ya se dijo, las transformaciones tratan con el flujo de datos. A medida que los datos pasan por los pasos PDI de una transformación, se manipulan o transforman de diferentes maneras. Ya has experimentado con algunas posibilidades. En esta sección, aprenderá en detalle cómo transformar diferentes datos de acuerdo con sus necesidades.

# Trabajando con Strings

Para cualquier cadena, hay muchas transformaciones que puedes aplicar a ella. PDI ofrece varias formas de hacer esto. También es común encontrar más de un paso que puede usarse para el mismo propósito. La siguiente tabla resume los pasos de PDI más comunes utilizados para operar en los campos de String:

# Trabajando con Strings

PDI step	Descripción
<b>Calculator</b>	<p>Con el paso de la <b>Calculator</b>, es posible aplicar varias transformaciones en cadenas, incluida la conversión a mayúsculas y minúsculas y la eliminación de caracteres especiales.</p> <p>También puede usar la <b>Calculator</b> para concatenar dos o tres cadenas usando las operaciones A + B y A + B + C respectivamente.</p>
<b>String Operations</b>	Este paso, como su nombre lo indica, está dedicado a trabajar exclusivamente con cuerdas. Le permite realizar varias operaciones, como recortar espacios en blanco iniciales o eliminar caracteres especiales.
<b>Replace in string</b>	<b>Replace in string</b> es un paso que se utiliza para buscar y reemplazar cadenas dentro de un campo string. La cadena de búsqueda puede ser texto plano o una expresión regular, como se explica en las siguientes secciones.
<b>String Cut</b>	Este paso te permite cortar una parte de una cuerda.
<b>Split Fields</b>	Utilice este paso para dividir un campo de Cadena en dos o más campos según la información del delimitador.
<b>Formula</b>	Este paso le permite definir nuevos campos basados en fórmulas. Para construir estas fórmulas, hay una lista de funciones disponibles que incluyen funciones para extraer partes de una cadena o concatenar cadenas, entre otras. Cada función tiene documentación correspondiente, por lo que es fácil entender cómo usarlas.
<b>UDJE</b>	Ya ha utilizado este paso para algunas operaciones de cadena, por ejemplo, para concatenar valores. El paso UDJE es adecuado para implementar casi cualquier operación de cadena, ya que puede escribirse usando una única expresión Java.

## Nota

Como se deduce de la tabla, no existe una forma única de operar con cadenas. Depende de usted elegir el paso que mejor se adapte a los requisitos en cada caso o el paso con el que se sienta más cómodo.

## Nota

Además de esta lista de pasos, los pasos **Modified Java Script Value** y **User Defined Java Class** le permiten implementar estas y otras expresiones más complicadas, como más adelante en el **Control del flujo de datos**.

# Extraer partes de cadenas usando expresiones regulares

Algunos de los pasos de la tabla anterior se pueden usar para extraer partes estáticas de una cadena, por ejemplo, los primeros N caracteres o los caracteres de la posición X a la posición Y.

Ahora suponga que tiene un campo que contiene un código postal. Este código puede estar en cualquier parte del campo y puede reconocerlo porque está precedido por ZIP CODE: o ZC :. El código postal en sí tiene exactamente cinco dígitos. No hay manera de saber de antemano en qué posición de la cadena se encuentra el código postal, por lo que ninguna de estas opciones le sirve en este caso. Para extraer la parte del campo en situaciones como esta, puede usar **RegEx Evaluation step**. Este paso, clasificado en la categoría de Scripting, verifica si una cadena coincide con una expresión regular o no. Además, le permite extraer subcadenas particulares de un campo de entrada y usarlas para llenar nuevos campos.

# Extraer partes de cadenas usando expresiones regulares

Antes de implementar la solución, debemos construir la expresión regular que representa nuestro campo. En este caso, la expresión sería:

```
* (CÓDIGO ZIP | ZP) : [0-9] {5} . *
```

Lo que significa que cualquier cantidad de caracteres va seguida de CÓDIGO ZIP o ZP, seguido de dos puntos, luego cinco dígitos (el código postal), y opcionalmente, más caracteres al final.

# Extraer partes de cadenas usando expresiones regulares

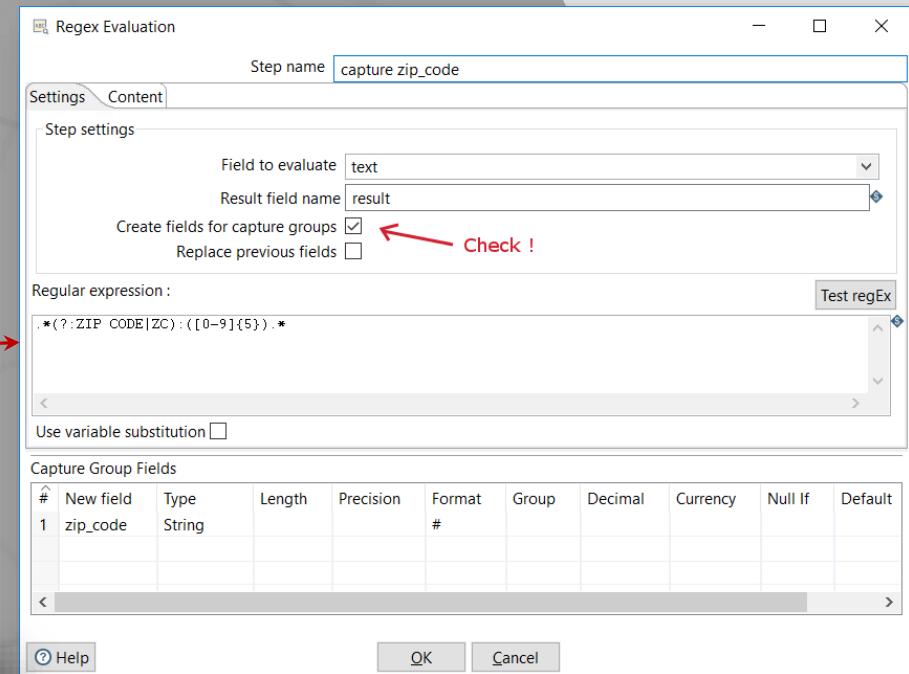
Para poder extraer el código postal, necesitamos capturarlo como un grupo. Hacemos esto encerrando la parte del código postal de la expresión regular—`[0-9] {5}`—entre paréntesis. Cualquier cosa entre paréntesis será considerada como un grupo. Como no estamos interesados en capturar (ZIP CODE | ZP) un grupo, convertiremos esa parte de la expresión en un grupo que no captura. Lo hacemos escribiendo?: Justo después del corchete de apertura. Nuestra expresión final será.

```
* (? : CÓDIGO ZIP | ZP) : ([0-9] {5}) . *
```

# Extraer partes de cadenas usando expresiones regulares

## Nota

1. Crea una nueva transformación.
2. Cree un conjunto de datos con un solo campo y una lista de códigos válidos e inválidos.
3. Al final de la transmisión, agregue un paso **RegEx Evaluation**.
4. Edita el paso y rellénalo de la siguiente manera (Ver imagen)
5. Cerramos la ventana
6. Ejecutar una vista previa de este último paso. Debería ver algo similar a esto (dependiendo de sus datos):



**Examine preview data**

Rows of step: capture zip\_code (6 rows)

#	text	result	zip_code
1	asdfg	N	<null>
2	AAAAA ZC:39391 ----	Y	39391
3	/// ZIP CODE:92000	Y	92000
4	ZC: 1234	N	<null>
5	ZIP CODE UNAVAILABLE	N	<null>
6	101010	N	<null>

**Close**



# Buscando y reemplazando usando expresiones regulares

Como puede deducir de la tabla anterior, si necesita buscar y reemplazar un texto, una opción es usar el paso **Replace in string**. En el escenario más simple, busca texto fijo y lo reemplaza con otro texto fijo. Sin embargo, este paso admite búsquedas y reemplazos más elaborados mediante el uso de expresiones regulares y referencias de grupo. Vamos a explicar esto con algunos ejemplos:

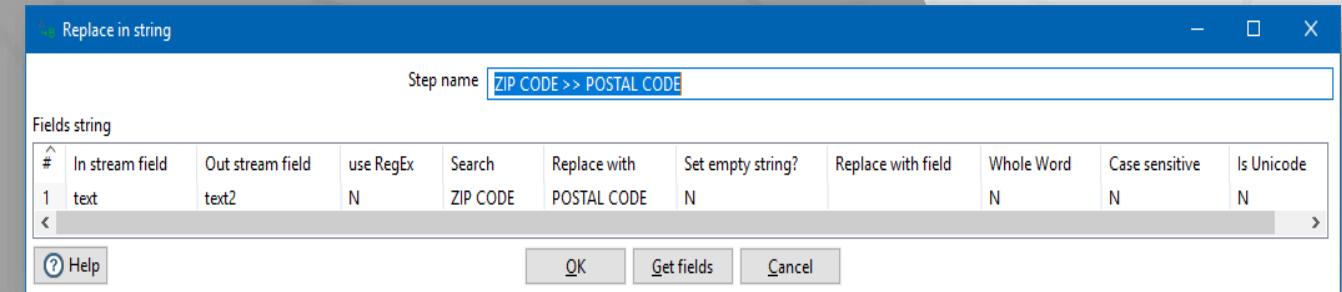
# Buscando y reemplazando usando expresiones regulares

## Nota

1. Cree una nueva transformación.
2. Cree un conjunto de datos con códigos postales, similar al creado en el ejemplo anterior.
3. Al final del flujo, agregue un paso **Replace in string**.
4. En **In stream field**, escriba o seleccione el campo que contiene el texto con los códigos postales.
5. En **Out stream field**, escriba un nombre para un nuevo campo.

## Nota

Supongamos que el escenario más simple en el que desea reemplazar el texto **ZIP CODE** con el texto **POSTAL CODE**. Bajo **use RegEx** escriba N, como **Search**, escriba **ZIP CODE**



Examine preview data

Rows of step: ZIP CODE >> POSTAL CODE (6 rows)

#	text	text2
1	asdfg	asdfg
2	AAAA ZC:39391 ----	AAAA ZC:39391 ----
3	/// ZIP CODE:92000	/// POSTAL CODE:92000
4	ZC: 1234	ZC: 1234
5	ZIP CODE UNAVAILABLE	POSTAL CODE UNAVAILABLE
6	101010	101010

Close

# Buscando y reemplazando usando expresiones regulares

Spoon - using\_regexp

File Edit View Action Tools Help

View Design

Search

informations

using\_regexp

- Run configurations
- Database connections
- Steps
  - Data Grid
  - Searching and Replace
  - ZIP CODE >> POSTAL CODE**
  - ZIP CODE | ZC >> POSTAL CODE**
  - ZIP CODE | ZC >> POSTAL CODE (uses group reference)**
  - capture zip\_code**
- Hops
- Data Grid --> capture
- Data Grid --> ZIP COI
- Data Grid --> Search
- Partition schemas
- Slave server
- Kettle cluster schemas
- Data Services
- Hadoop clusters

Welcome! date\_range\_with\_delay Spark submit using\_regexp

100%

Preview each step to see the result of the different "search and replace" options

capture zip\_code

A B

ZIP CODE >> POSTAL CODE

A B

ZIP CODE | ZC >> POSTAL CODE

A B

ZIP CODE | ZC >> POSTAL CODE (uses group reference)

Data Grid

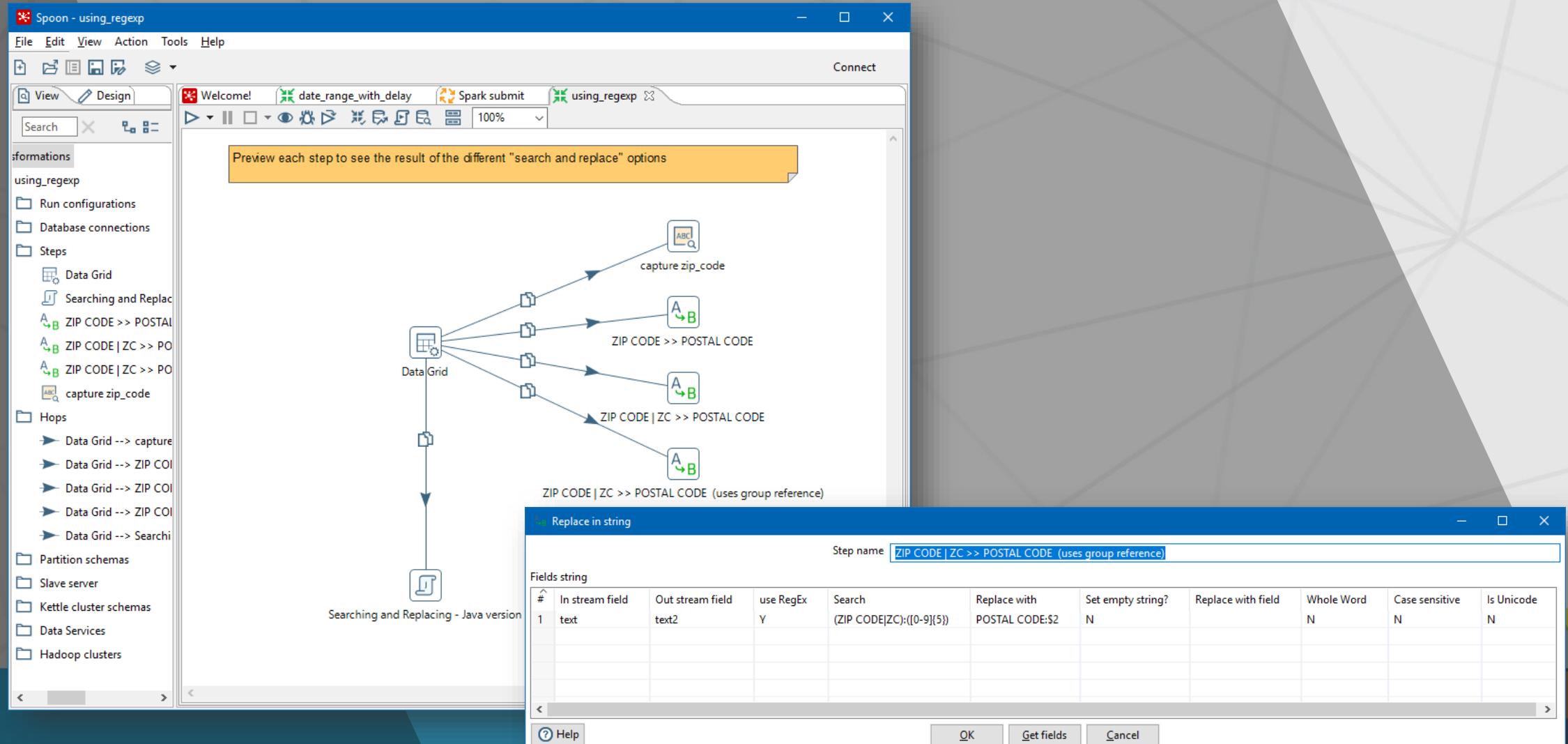
Searching and Replacing - Java version

Replace in string

Step name: ZIP CODE | ZC >> POSTAL CODE (uses group reference)

#	In stream field	Out stream field	use RegEx	Search	Replace with	Set empty string?	Replace with field	Whole Word	Case sensitive	Is Unicode
1	text	text2	Y	(ZIP CODE ZC):(0-9){5}	POSTAL CODE:\$2	N		N	N	N

Help OK Get fields Cancel



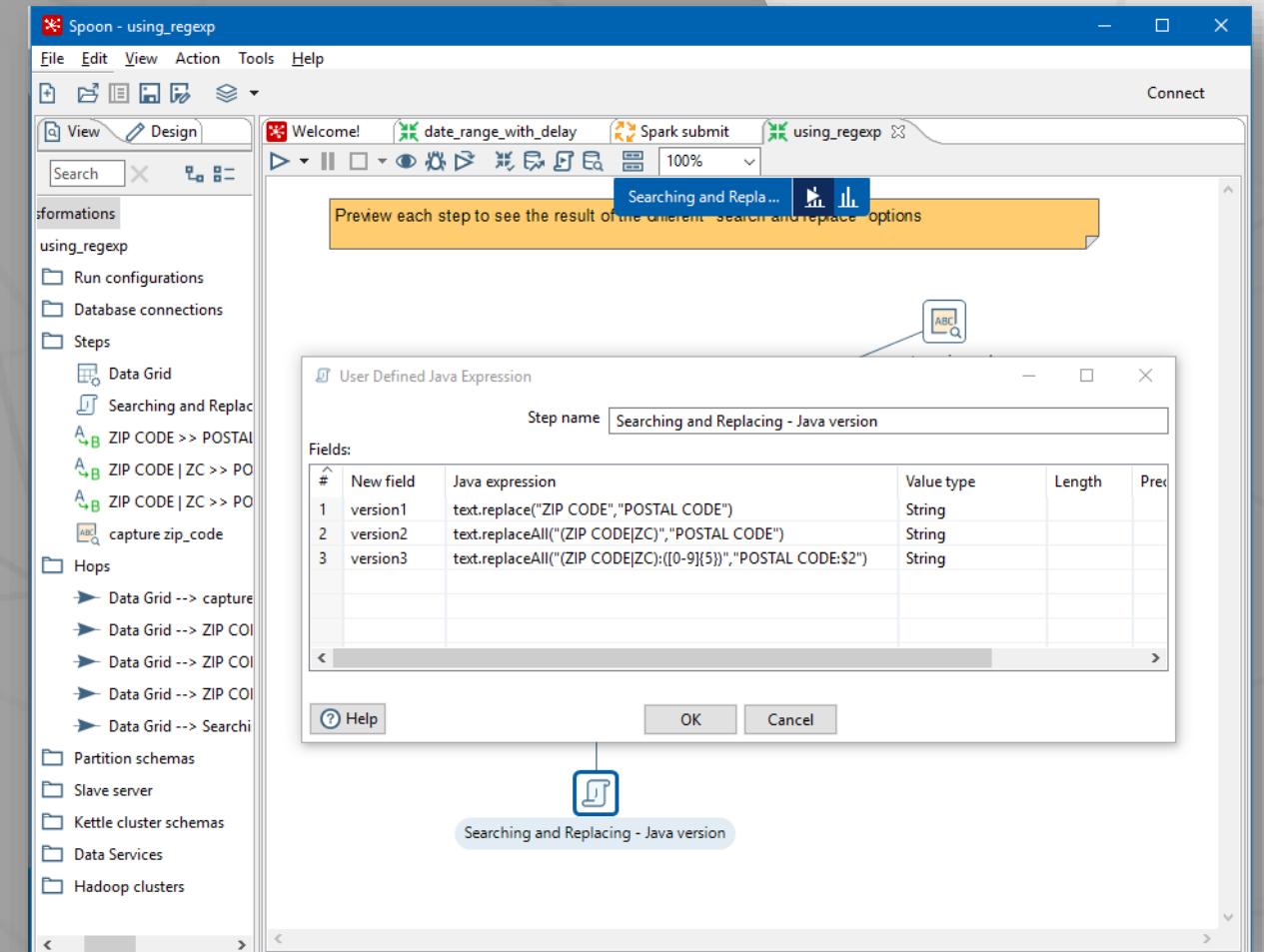
# Buscando y reemplazando usando expresiones regulares

También puede implementar todas estas operaciones con un paso UDJE utilizando los métodos Java **Replace** y **ReplaceAll**. Las expresiones que debe usar para los ejemplos anteriores son:

```
text.replace ("ZIP CODE", "POSTAL CODE")
```

```
text.replaceAll ("(ZIP CODE | ZC)", "POSTAL
CODE")
```

```
text.replaceAll ( "(CÓDIGO POSTAL | ZC):
([0-9] {5})", "CÓDIGO POSTAL: $ 2")
```



# Haciendo un poco de matemáticas con campos numéricos

Algunos de los pasos presentados para las operaciones de string también son adecuados para algunas matemáticas. Calculator es el primer paso que debe buscar si pretende realizar operaciones simples, por ejemplo, suma, resta, división, porcentaje, entre otros.

Como ejemplo de cómo usar un paso de Fórmula, suponga que tiene tres campos de enteros denominados a, b y c, y desea crear un nuevo campo de la siguiente manera: Si todos los valores (a, b y c) son positivos, El nuevo campo será el valor mínimo entre ellos.

En otro caso, el nuevo campo será cero. La fórmula se puede indicar como:

```
IF (AND ([a]> 0; [b]> 0; [c]> 0) ; Min ([a]; [b]; [c]) ; 0) .
```

## Nota

Tenga en cuenta que en un paso de Fórmula, los nombres de los campos están entre corchetes, y los parámetros de las funciones están separados por un punto y coma.

# Operando con fechas

Las fechas son uno de los tipos de datos más comunes y hay un amplio conjunto de operaciones que puede que necesite aplicar a los campos de Fecha:

- Extraer partes de una fecha, por ejemplo, el año.
- Dada una fecha, obteniendo algunas descripciones, por ejemplo, el día laborable.
- Añadiendo fechas
- Comparando fechas

PDI ofrece muchas posibilidades para manipular campos de fecha. Sin embargo, estas características no están organizadas en un solo lugar dentro de Spoon. Hay varios pasos en diferentes categorías de pasos que te permiten manipular fechas. Además, dada una operación particular, generalmente hay un par de formas de realizarla. Esta sección pretende presentarte las diferentes opciones.

# Realización de operaciones en fechas.

La forma más sencilla de operar con fechas en PDI es con **Calculator**. La lista de operaciones disponibles en un paso **Calculator** es realmente larga y puede ser difícil encontrar una operación particular entre el conjunto de opciones. Sin embargo, si está buscando una función relacionada con la fecha, puede aplicar un filtro:



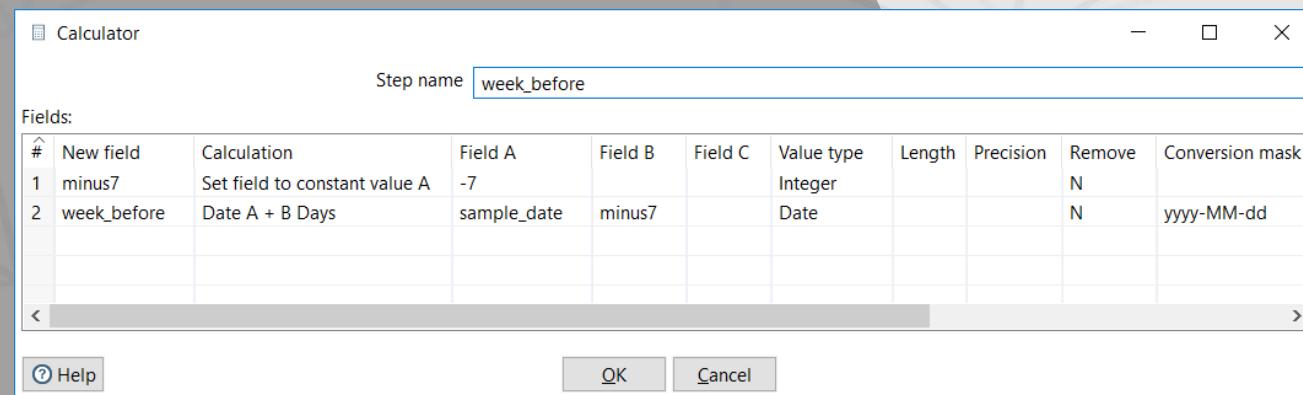
# Restar fechas con el paso de la calculadora

Veamos un ejemplo para demostrar cómo aplicar una operación de fecha con el paso de la Calculadora. En este caso, tenemos una lista de fechas. Para cada fecha, queremos que la fecha coincidente sea una semana antes, es decir, nuestra fecha menos siete días.

Si explora las opciones de la Calculadora, no hay ninguna función para restar fechas, pero sí una para agregar fechas. Entonces, el truco es sumar un número negativo.

## Pasos

1. Cree una nueva transformación y agregue un **Data Grid**.
2. Utilice **Data Grid** para definir un nuevo campo de tipo Fecha y llenar la cuadrícula con algunos valores aleatorios.
3. Después de **Data Grid**, agregue un paso de **Calculator**. Lo usaremos para las matemáticas.
4. Llenamos **calculator** de la siguiente manera:



# Restar fechas con el paso de la calculadora

## Pasos

- Cierre la ventana de configuración, seleccione la Calculadora y ejecute una vista previa. Debería ver algo como esto:

Examine preview data

Rows of step: week\_before (10 rows)

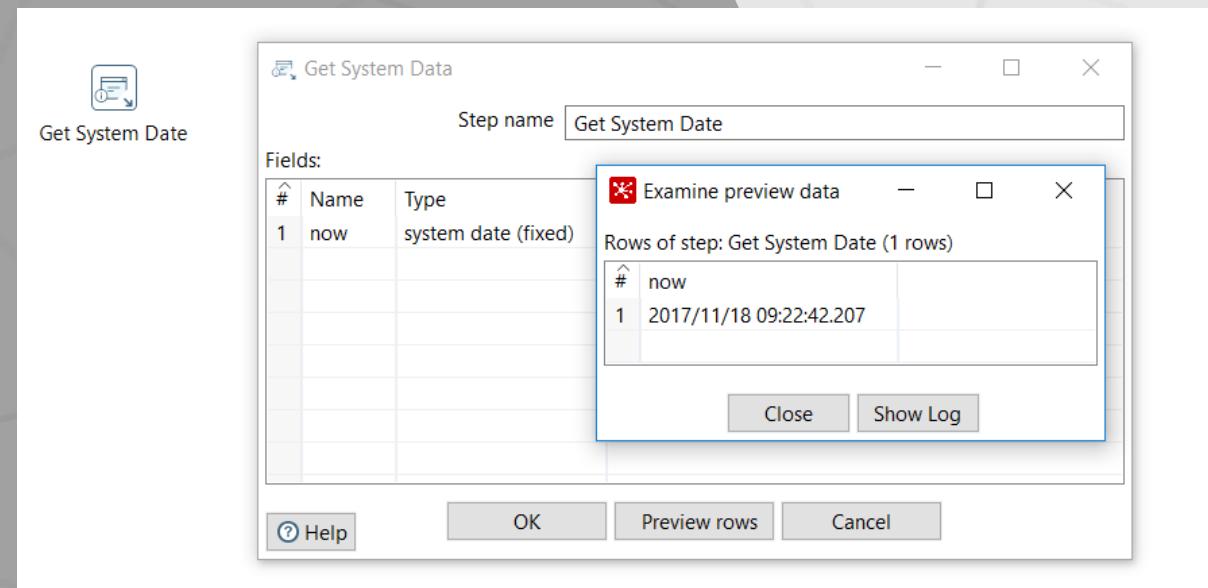
#	sample_date	minus7	week_before
1	2017-01-02	-7	2016-12-26
2	2017-01-14	-7	2017-01-07
3	2017-02-21	-7	2017-02-14
4	2017-03-29	-7	2017-03-22
5	2017-04-30	-7	2017-04-23
6	2017-05-13	-7	2017-05-06
7	2017-05-14	-7	2017-05-07
8	2017-05-28	-7	2017-05-21
9	2017-06-13	-7	2017-06-06
10	2017-06-28	-7	2017-06-21

Close

# Obtención de información relativa a la fecha actual.

Un requisito muy común es obtener la fecha del sistema o alguna fecha u hora relativa a él. PDI ofrece un par de maneras de hacer esto. La forma más sencilla de obtener información del sistema en relación con la fecha actual es utilizar el paso **Get System Info**. El paso ofrece una larga lista de opciones, la mayoría de ellas relacionadas con la fecha actual. Algunos ejemplos son **system date (fixed)**, **Yesterday 00:00:00**, **Last day of this month 23:59:59**, y **First day of this quarter 00:00:00**, entre otros.

El paso no proporciona ninguna opción de formato, por lo tanto, cuando usa este paso para agregar información del sistema de fecha a su conjunto de datos, el campo se agrega con el formato de fecha predeterminado, como puede ver en la siguiente captura de pantalla:



# Realizando otras operaciones útiles en fechas.

A pesar de la lista completa de opciones, existe la posibilidad de que una función que está buscando no esté disponible en ningún paso de PDI. Supongamos que, entre los campos descriptivos de una fecha, desea el nombre del mes en su idioma local o la fecha actual en una zona horaria diferente. Ninguno de los pasos presentados proporciona esa información.

Afortunadamente, siempre existe el recurso de código. En la categoría de pasos Scripting, hay varios pasos que se pueden usar para este propósito: **Modified Java Script Value**, **User Defined Java Class** y **User Defined Java Expression**. Estos pasos le permiten utilizar las bibliotecas de Java o JavaScript. La siguiente subsección ofrece una breve introducción a la **User Defined Java Class** para que esté listo para implementar más funciones de fecha en sus transformaciones.

# Realizando otras operaciones útiles en fechas.



## Pasos

1. Cree una transformación con un conjunto de fechas, tal como lo hizo antes.
2. Desde la categoría de pasos de **Scripting**, agregue un paso **User Defined Java Class** (o **UDJC** para abreviar) y vincúlelo a la secuencia de datos.
3. Haga doble clic en el paso. Aparecerá una ventana de configuración con un panel llamado **Class code**. Dentro de este panel, escriba el siguiente fragmento de código
4. En el medio del código, donde dice **// AQUÍ VA SU CÓDIGO**, se inserta el código que hará la operación

```
import java.util.Calendar;
import java.util.Locale;
public boolean processRow(StepMetaInterface smi, StepDataInterface sdi) throws
KettleException
{
    Object[] r = getRow();
    if (r == null)
    {
        setOutputDone();
        return false;
    }

    Object[] outputRow = createOutputRow(r, data.outputRowMeta.size());

    // AQUÍ VA SU CÓDIGO

    Calendar cal = Calendar.getInstance();
    cal.setTime(get(Fields.In, "mydate").getDate(r));
    String monthName = cal.getDisplayName(cal.MONTH,Calendar.LONG, new
Locale("FR"));
    get(Fields.Out, "monthName").setValue(outputRow, monthName);
    putRow(data.outputRowMeta, outputRow);

    return true;
}
```



## Nota

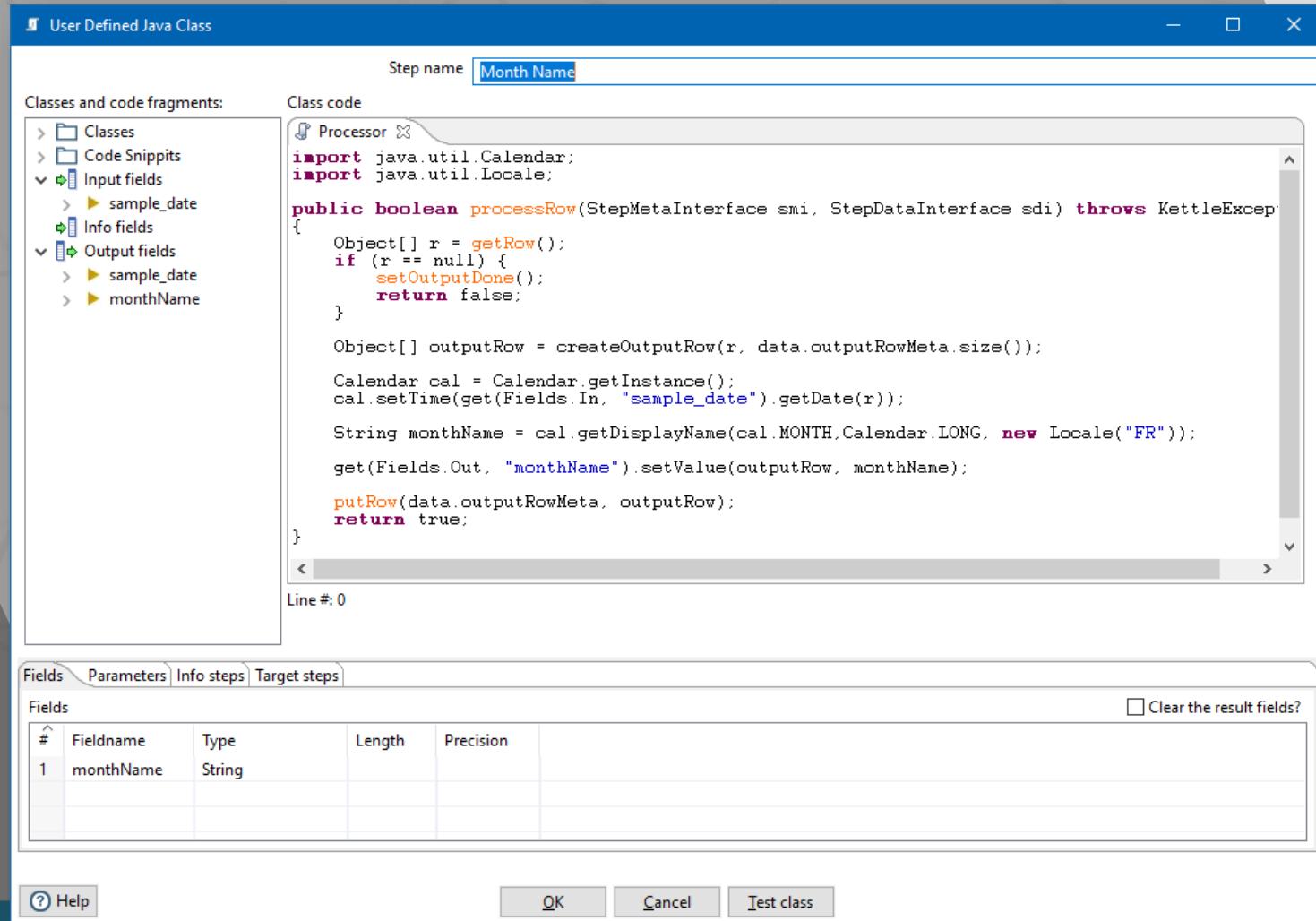
No te preocupes por los detalles; Centrarse en el código específico para las fechas. Aprenderá a usar este paso en detalle en el Capítulo **Manipulación de datos por codificación**.



## Nota

Para obtener una referencia completa a la clase de Calendario, consulte el siguiente enlace: <https://docs.oracle.com/javase/8/docs/api/java/util/Calendar.html>.

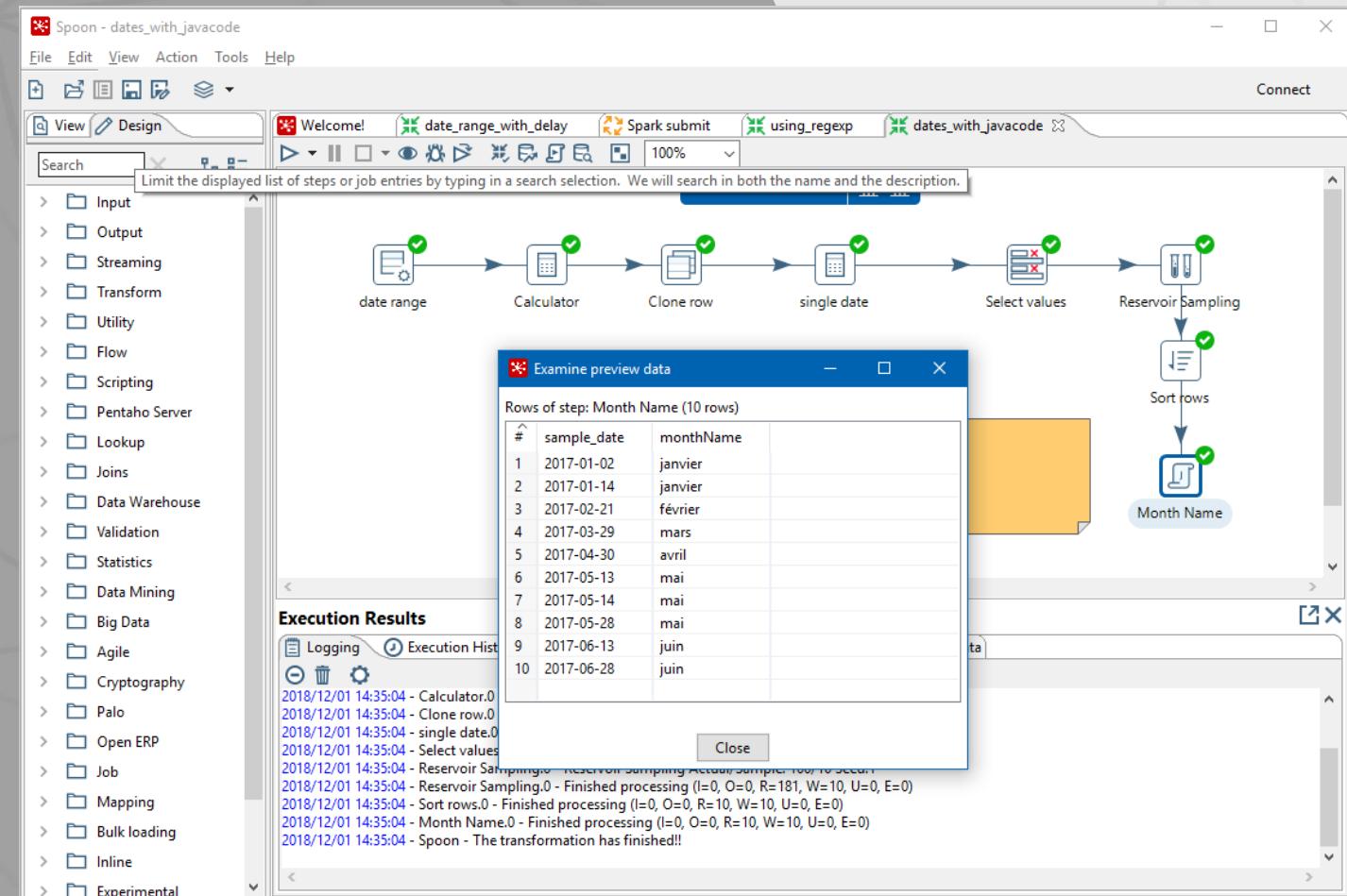
# Realizando otras operaciones útiles en fechas.



# Realizando otras operaciones útiles en fechas.

## Pasos

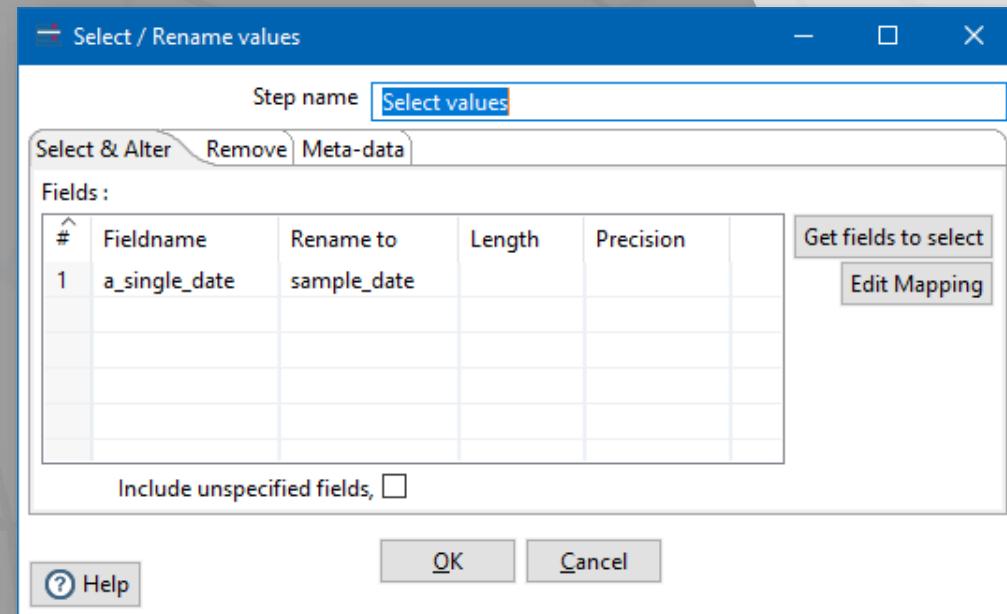
1. En la cuadrícula inferior del paso Java Class, complete la primera fila con monthName debajo de la columna Fieldname y String debajo de la columna Type.
2. Cerrar la ventana.
3. Con el paso Java Class seleccionado, ejecute una vista previa. Verá algo como lo siguiente (dependiendo del conjunto de fechas en su Transformación):



# Modificando los metadatos de los streams.

En las secciones anteriores, aprendió varias formas de transformar datos usando diferentes pasos de PDI. Ya hay un paso en particular que le es familiar, que está disponible en todas las funcionalidades: el paso **Select values**. A pesar de estar clasificado como un paso de transformación, el paso **Select values** hace más que simplemente transformar datos. Le permite seleccionar, reordenar, renombrar y eliminar campos, o cambiar los metadatos de un campo. La ventana de configuración del paso tiene tres pestañas:

Select & Alter, Remove, Meta-data



## Nota

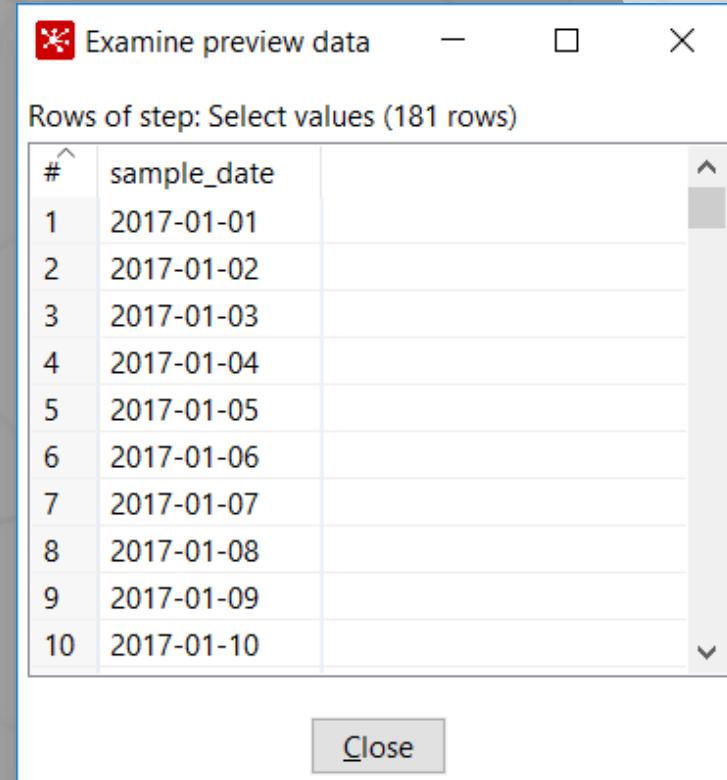
Puede usar solo una de las pestañas del paso **Select values** a la vez. PDI no le impedirá que complete más de una pestaña, pero eso podría llevar a un comportamiento inesperado.

# Tab Select & Alter

La pestaña Select & Alter, que aparece seleccionada de forma predeterminada, le permite especificar los campos que desea conservar. Probémoslo con un ejemplo simple:

## Pasos

1. Abre la Transformación que genera la lista de fechas.
2. Desde la rama **Transform** del árbol de Pasos, agregue un paso **Select values** y cree un salto desde el último paso de la Calculadora hacia este.
3. Editar el paso **Select values**. En la pestaña **Select & Alter** (que aparece seleccionada de forma predeterminada), escriba **a\_single\_date** en **Fieldname**.
4. Cierra la ventana y guarda tu trabajo.
5. Seleccione el paso **Select values** y ejecute una vista previa. Solo deberías ver la última columna, **a\_single\_date**

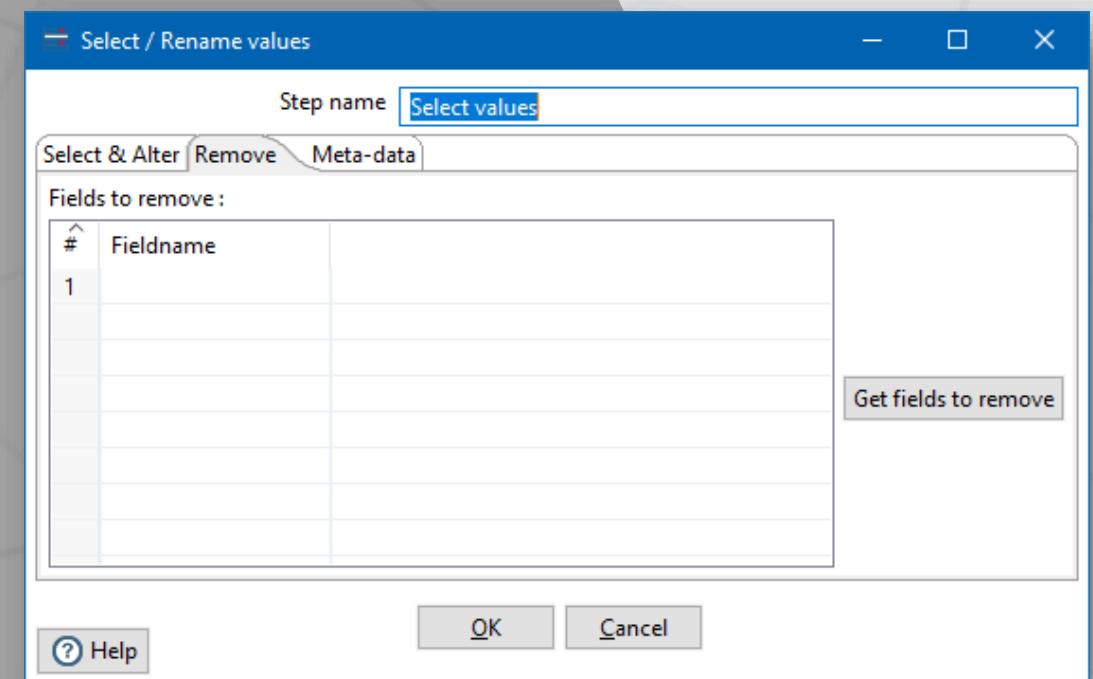


The screenshot shows a modal window titled "Examine preview data" with the sub-tittle "Rows of step: Select values (181 rows)". The window displays a table with two columns: a row number column labeled "#" and a date column labeled "sample\_date". The data starts at row 1 with the date 2017-01-01 and continues sequentially up to row 10 with the date 2017-01-10. A vertical scroll bar is visible on the right side of the table. At the bottom of the window is a "Close" button.

#	sample_date
1	2017-01-01
2	2017-01-02
3	2017-01-03
4	2017-01-04
5	2017-01-05
6	2017-01-06
7	2017-01-07
8	2017-01-08
9	2017-01-09
10	2017-01-10

# Tab Remove

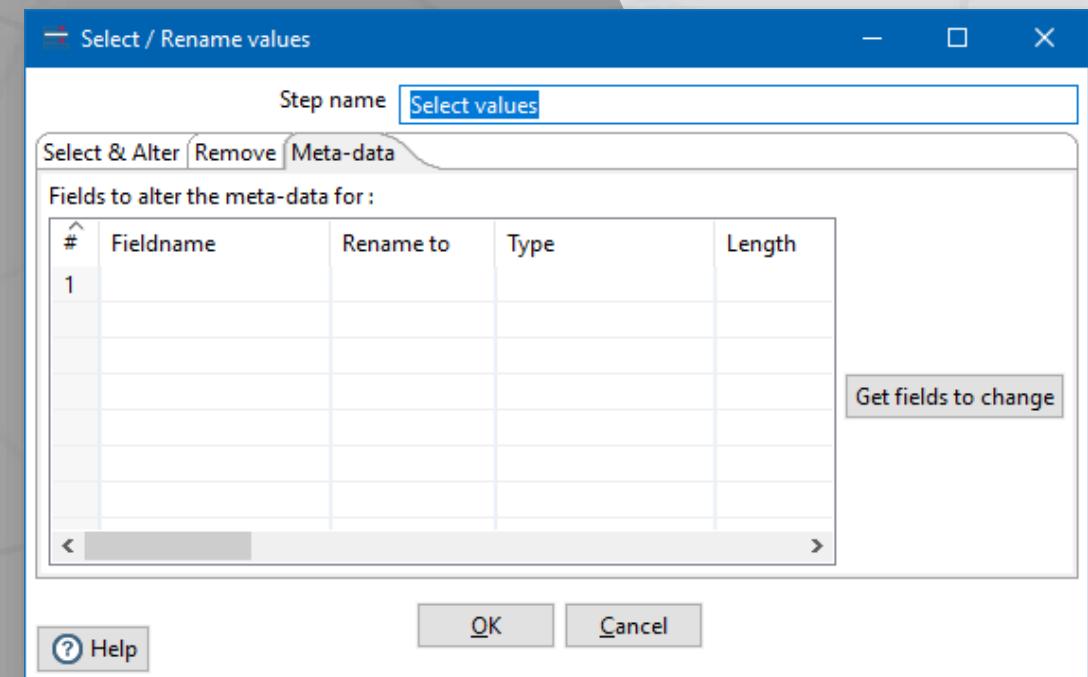
La pestaña Remove es útil para descartar campos no deseados. Esta pestaña es útil si desea eliminar algunos campos. Para eliminar muchos, es más fácil usar la pestaña **Select & Alter** y no especifique los campos para eliminar, sino los campos para mantener.



# Tab Meta-data

Finalmente, la pestaña Meta-data se utiliza cuando desea cambiar la definición de un campo. Puede cambiar el tipo, nombre o formato. En el caso de números o fechas, puede aplicar cualquiera de las máscaras ya explicadas.

El paso **Select values** es solo uno de los varios pasos que contienen información de campo. En estos casos, el botón Get Fields no es necesario.



# Trabajando con estructuras complejas.

La primera sección del capítulo explica las diferentes formas de trabajar con campos simples. Sin embargo, es común tener estructuras complejas con las que trabajar. Una de las situaciones más comunes es tener que analizar los resultados de un Web Service o una llamada REST, que devuelve datos en formato XML o JSON. Esta sección explica cómo analizar este tipo de datos complejos.

# Trabajando con XML

XML significa Extensible Markup Language. Es básicamente un lenguaje diseñado para describir datos y se usa ampliamente no solo para almacenar datos, sino también para intercambiar datos entre sistemas heterogéneos a través de Internet. En esta sección, describiremos cómo se ve una estructura XML y cómo analizarla con PDI.

# Introducción a la terminología XML

Antes de comenzar a trabajar con XML, veamos una breve introducción a la estructura y la terminología básica. Mira esta pieza de XML que muestra información sobre los países:

```
<world>
...
<country>
  <name>Japan</name>
    <capital>Tokyo</capital>
    <language isofficial="T">
      <name>Japanese</name>
      <percentage>99.1</percentage>
    </language>
    <language isofficial="F">
      <name>Korean</name>
      <percentage>0.5</percentage>
    </language>
    <language isofficial="F">
      <name>Chinese</name>
      <percentage>0.2</percentage>
    </language>
  ...
</country>
```

## Nota

Esta es solo la terminología básica relacionada con los archivos XML. Para una referencia completa, puede visitar

<http://www.w3schools.com/xml/>

# Familiarizándose con la notación XPath

XPath es un conjunto de reglas que se utiliza para obtener información de un documento XML. En XPath, los documentos XML se tratan como árboles de nodos. Hay varios tipos de nodos: elementos, atributos y textos son algunos de ellos. **world**, **country** e **isofficial** son algunos de los nodos en el archivo de muestra.

En el archivo de **country** de muestra, país es el padre de los elementos **name**, **capital** y **language**. Estos tres elementos son los hijos del país.

Para seleccionar un nodo en un documento XML, debe usar una expresión de ruta relativa a un nodo actual.

La siguiente tabla tiene algunos ejemplos de expresiones de ruta que puede usar para especificar campos. Los ejemplos asumen que el nodo actual es el **language** :

## Nota

Entre los nodos hay relaciones. Un nodo tiene un parent, cero o más hijos, hermanos, ancestros y descendientes, dependiendo de dónde estén los otros nodos en la jerarquía.

# Familiarizándose con la notación XPath

Path expression	Descripción	Ejemplo
node_name	Selecciona todos los nodos secundarios del nodo <b>node_name</b> .	Un nombre de nodo de muestra para el <b>language</b> del nodo actual sería <b>percentage</b> .
.	Selecciona el nodo actual.	<b>language</b>
..	Selecciona el padre del nodo actual.	.. se refiere a <b>country</b> , el nodo padre del <b>language</b> del nodo actual . . . / <b>capital</b> se refiere al nodo de <b>capital</b> dentro del <b>country</b> .
@	Selecciona un atributo.	<b>@isofficial</b> obtiene el atributo <b>isofficial</b> en el <b>language</b> del nodo actual.

## Nota

Tenga en cuenta que **name** de las expresiones y **./name** no son lo mismo. La primera expresión selecciona el nombre del idioma, mientras que la segunda selecciona el nombre del país.



## Nota

Para obtener más información sobre XPath, visite [https://www.w3schools.com/xml/xpath\\_intro.asp](https://www.w3schools.com/xml/xpath_intro.asp)



# Análisis de estructuras XML con PDI

Con PDI, es posible leer un archivo XML, así como analizar una estructura XML que se encuentra en un campo en nuestro conjunto de datos. En ambos casos, para analizar esa estructura, use el paso **Get data from XML**. Para indicar a PDI qué información obtener de la estructura, se requiere que use la notación XPath explicada anteriormente.

## Pasos

1. Cree una nueva transformación, asígnele un nombre y guárdela.
2. Desde los pasos **Input**, arrastre al lienzo un paso **Get data from XML**.
3. Abra la ventana de configuración para este paso haciendo doble clic en ella.



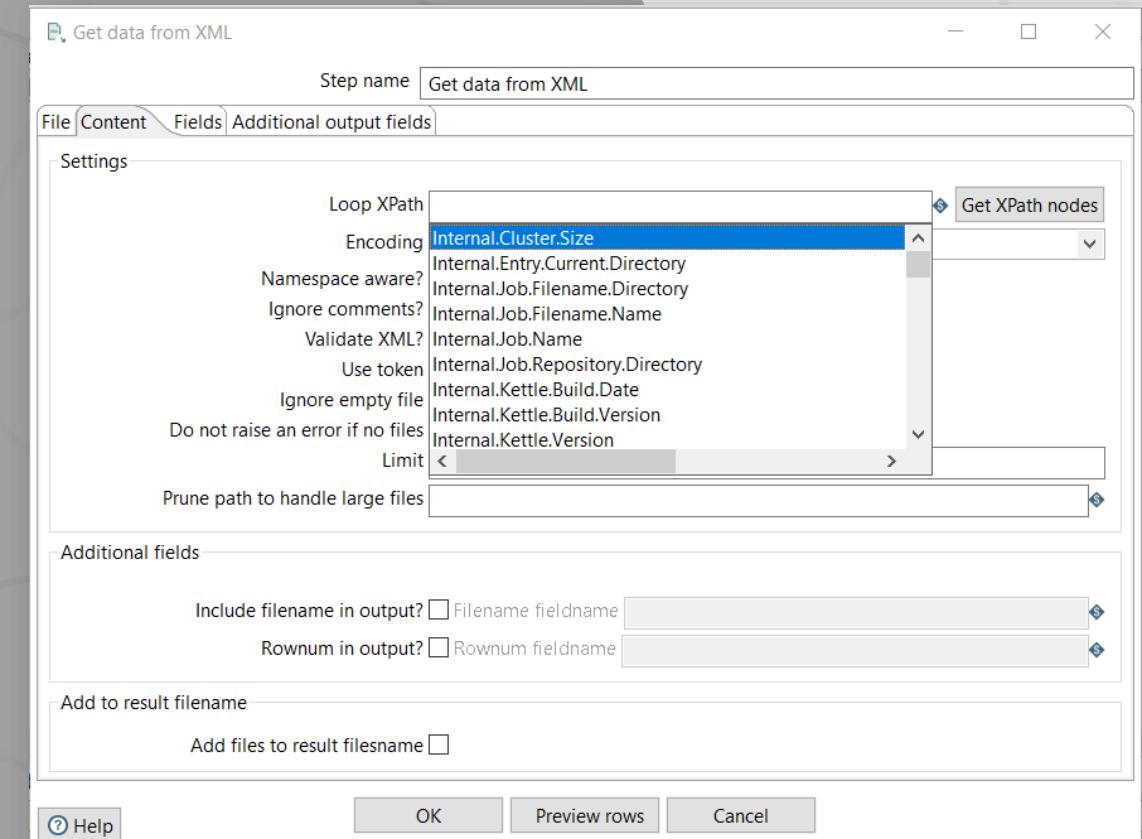
Para especificar el nombre y la ubicación de un archivo XML, debe completar la pestaña **File** tal como lo hace en cualquier paso de entrada de archivo. En este caso, leeremos un archivo de una carpeta relativa a la carpeta donde almacenó la transformación

# Análisis de estructuras XML con PDI

## Pasos

1. En el cuadro de texto **File or directory**, presione Ctrl + barra espaciadora o Shift + cmd + espacio en una Mac. Aparece una lista desplegable que contiene una lista de variables definidas:
2. Seleccione `$ {Internal.Entry.Current.Directory}`. El cuadro de texto se rellena con este texto.
3. Complete el texto para que pueda leer esto: `$ {Internal.Entry.Current.Directory} /resources/countries.xml`
4. Haga clic en el botón **Add**. La ruta completa se mueve a la cuadrícula.
5. Seleccione la pestaña **Content** y haga clic en **Get XPath nodes** para seleccionar **Loop XPath**. En la lista que aparece, seleccione `/world/country/language`, de modo que PDI genere una fila para cada elemento `/world/country/language` en el archivo.

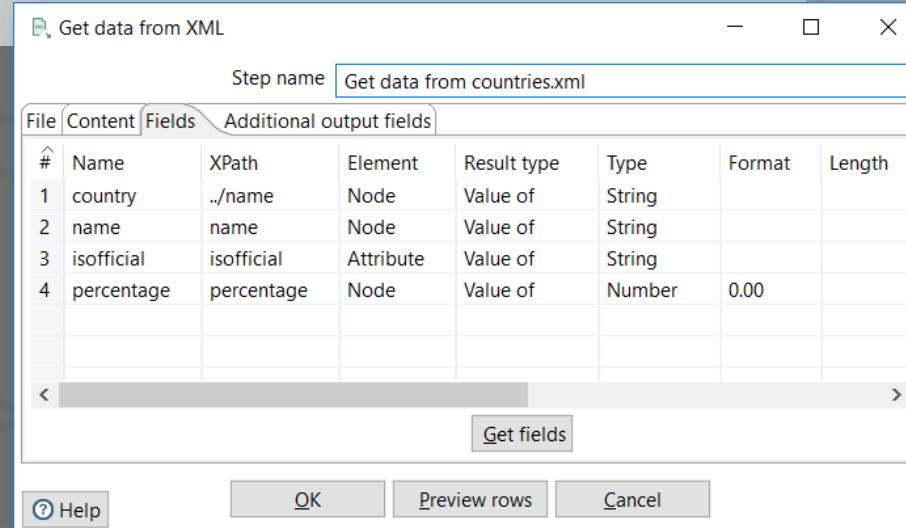
Después de seleccionar el loop XPath, debe especificar los campos a obtener. Para hacer esto, debe completar la cuadrícula en la pestaña **Fields** usando la notación XPath.



# Análisis de estructuras XML con PDI

## Pasos

6. Seleccione la ficha Campo y complete la cuadrícula, como se muestra en la siguiente captura de pantalla



## Pasos

7. Haga clic en **Preview rows**, y debería ver algo como la siguiente captura de pantalla

Rows of step: Get data from countries.xml (983 rows)				
#	country	name	isofficial	percentage
432	Italy	Romani	F	0.20
433	Italy	Albaniana	F	0.20
434	Jamaica	Creole English	F	94.20
435	Jamaica	Hindi	F	1.90
436	Japan	Japanese	T	99.10
437	Japan	Korean	F	0.50
438	Japan	Chinese	F	0.20
439	Japan	Philippine Languages	F	0.10
440	Japan	English	F	0.10
441	Japan	Ainu	F	0.00

## Nota

Si presiona el botón Obtener campos, PDI llenará la cuadrícula con los nodos secundarios del nodo actual. Si desea obtener algún otro nodo, debe escribir su XPath a mano.

## Nota

Si tiene limitaciones al leer sus archivos XML debido al tamaño o complejidad de su estructura, hay un paso alternativo: el paso [XML Input Stream \(StAX\)](#).

# Analizar una estructura XML almacenada en un campo

Si, como parte de su conjunto de datos, tiene un campo que contiene una estructura XML, puede analizarlo de la misma manera que lo hizo antes. Mira la siguiente muestra que contiene paletas de colores:

```
BlackAndWhite,<color-palette>
<color>#ffffffff</color><color>#b9b9b9</color><color>#8e8e8e</color><color>#727272</color><color>#000000</color>
</color-palette>

primary,<color-palette>
<color>#f40707</color><color>#1b00c7</color><color>#e7e035</color><color>#17d640</color><color>#f6f3f3</color>
</color-palette>

Blue,<color-palette>
<color>#a8cafe</color><color>#224dce</color><color>#05146e</color><color>#0c226f</color><color>#ede7fd</color>
</color-palette>

Generic,<color-palette>
<color>#C52F0D</color><color>#123D82</color><color>#4A0866</color><color>#445500</color><color>#FFAA00</color>
</color-palette>
```

# Analizar una estructura XML almacenada en un campo

Cada línea contiene el nombre de una paleta de colores y una lista de colores representados como una estructura XML. Supongamos que tiene la muestra anterior en un archivo llamado colors.txt, y para cada paleta, desea obtener la lista de colores. Lo haces de la siguiente manera:

## Nota

1. Lea el archivo con **Text file input**, establezca una coma como separador y configure dos campos de strings: **palette\_name** y **colors**.
2. Despues del paso **Text file input**, agregue un paso **Get data from XML**.
3. Haga doble clic en el paso y compruebe que la opción **XML source is defined in a field?**
4. Bajo **get XML source from a field**, select the name of the field containing the XML structure colors
5. Seleccione la pestaña **Content**. Haga clic **Get XPath nodes**. Aparecerá una ventana para que proporcione un fragmento XML de muestra. Copie y pegue un fragmento XML representativo de su archivo en esta ventana, por ejemplo:

```
<color-palette>
<color>#fffffff</color>
<color>#b9b9b9</color>
<color>#8e8e8e</color>
<color>#727272</color>
<color>#000000</color>
</color-palette>
```

6. Cerrar la ventana. PDI listará los nodos encontrados en ese fragmento de XML, en este caso **/color-palette** y **/color-palette/color**.
7. Como nodo actual, seleccione **/color-palette/color**.
8. Configure la pestaña **Fields**. Crea un campo llamado color. Para **XPath**, simplemente escriba un punto.
9. Cierre la ventana y ejecute una vista previa. Debería ver los nuevos campos extraídos de la estructura XML, así como los campos de los pasos anteriores, en este caso, el nombre de la paleta. El resultado se verá como el siguiente:



```
<color-palette>
<color>#fffffff</color>
<color>#b9b9b9</color>
<color>#8e8e8e</color>
<color>#727272</color>
<color>#000000</color>
</color-palette>
```

**Examine preview data**

Rows of step: colors (20 rows)

#	palette_name	color
1	BlackAndWhite	#fffffff
2	BlackAndWhite	#b9b9b9
3	BlackAndWhite	#8e8e8e
4	BlackAndWhite	#727272
5	BlackAndWhite	#000000
6	primary	#f40707
7	primary	#1b00c7
8	primary	#e7e035
9	primary	#17d640
10	primary	#f6f3f3

**Close**

# Analizando estructuras JSON

Una estructura JSON (**JavaScript Object Notation**) puede contener tipos de datos escalares, como números, booleanos o cadenas, y también datos estructurados como matrices u objetos. De manera similar a XML, JSON también puede almacenar datos jerárquicos.

Los objetos se representan como una colección de pares `name_of_field:value_of_field` y puede tener una matriz de estos elementos representados por una lista, encerrada por `[]`.

# Analizando estructuras JSON

En la estructura JSON de muestra, puede identificar fácilmente una serie de libros, cada uno con cierta información, por ejemplo, el autor y el nombre del libro.

```
{ "store": {  
    "book": [  
        { "category": "reference",  
          "author": "Nigel Rees",  
          "title": "Sayings of the Century",  
          "price": 8.95  
        },  
        { "category": "fiction",  
          "author": "Evelyn Waugh",  
          "title": "Sword of Honour",  
          "price": 12.99  
        },  
        { "category": "fiction",  
          "author": "Herman Melville",  
          "title": "Moby Dick",  
          "isbn": "0-553-21311-3",  
          "price": 8.99  
        },  
        ...  
    ]  
}
```

## Nota

Encontrará el archivo original como samples / transformations / files / jsonfile.js dentro del directorio de instalación de PDI.



# Familiarizándose con la notación JSONPath

Para analizar las estructuras JSON, usamos JSONPath. JSONPath es el XPath para JSON; se usa para referirse a elementos JSON de la misma manera que XPath se usa para elementos XML.

Las expresiones JSONPath usan notación de puntos, como en `$ .store.book [0] .title`.

La siguiente tabla muestra una descripción básica de los elementos de sintaxis de JSONPath. Los ejemplos se refieren a la estructura del libro presentada anteriormente:

# Familiarizándose con la notación JSONPath

Para analizar las estructuras JSON, usamos JSONPath. JSONPath es el XPath para JSON; se usa para referirse a elementos JSON de la misma manera que XPath se usa para elementos XML.

Las expresiones JSONPath usan notación de puntos, como en `$ .store.book [0] .title`.

La siguiente tabla muestra una descripción básica de los elementos de sintaxis de JSONPath. Los ejemplos se refieren a la estructura del libro presentada anteriormente:

# Familiarizándose con la notación JSONPath

JSONPath expression	Descripción	Ejemplo
\$	Root object	\$ devuelve toda la estructura JSON
.	Operador Child; Se usa para acceder a diferentes niveles de la estructura JSON.	\$ .. title devuelve los títulos de los libros
*	Wildcard para referirse a todos los elementos.	\$ .store.book. * devuelve todos los libros
[]	Operador de matriz	\$ .store.book [0] devuelve los datos sobre el primer libro en la matriz

## Nota



Tenga en cuenta la diferencia entre `$ .store.book` que devuelve un solo elemento, que es un conjunto de libros, y `$ .store.book. *`, Que devuelve cinco elementos, cada uno correspondiente a un libro diferente.

# Análisis de estructuras JSON con PDI

Con PDI, es posible leer un archivo con una estructura JSON, así como analizar una estructura JSON que se encuentra en un campo de nuestro conjunto de datos. En ambos casos, para analizar esa estructura, utilice el paso de entrada JSON. Para indicar a PDI qué información obtener de la estructura, se requiere que use la notación JSONPath como se explicó anteriormente.

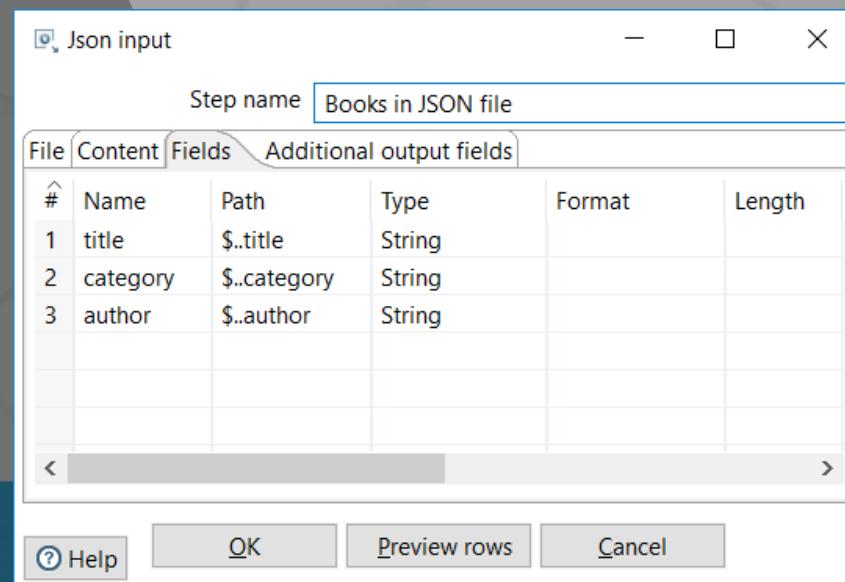
## Leyendo un archivo JSON con el paso de entrada JSON

Vamos a crear una transformación muy simple para demostrar cómo leer un archivo JSON. En esta Transformación, leeremos el archivo de muestra con libros:

# Análisis de estructuras JSON con PDI

## Nota

1. Crear una transformación.
2. Desde la categoría **Input**, arrastre al área de trabajo un paso **JSON input**.
3. Haga doble clic en el paso y configúrelo para leer el archivo JSON. Agregue la ruta completa del archivo tal como lo haría para leer cualquier otro tipo de archivo.
4. Una vez que especifique el nombre de archivo, haga clic en la pestaña **Fields** y configúrelo de la siguiente manera
5. Haga clic en **Preview rows**. Deberías ver lo siguiente:



**Examine preview data**

Rows of step: Books in JSON file (5 rows)

#	title	category	author
1	Sayings of the Century	reference	Nigel Rees
2	Sword of Honour	fiction	Evelyn Waugh
3	Moby Dick	fiction	Herman Melville
4	The Lord of the Rings	fiction	J. R. R. Tolkien
5	La belle vie	fiction	Samatar

**Close** **Show Log**

# Analizar una estructura JSON almacenada en un campo

Existe la posibilidad de que tenga una estructura JSON en un campo de su conjunto de datos. En este caso, la forma de analizar la estructura no difiere mucho de lo que se explicó anteriormente.

Si tiene un campo que contiene una estructura JSON y necesita analizarlo para extraer algunos valores, proceda de la siguiente manera:

## Nota

- 
1. Al final de su flujo de datos, agregue un paso **JSON input**.
  2. Haga doble clic en el paso y marque la opción **Source is from a previous step**.
  3. Bajo **Select field:**, seleccione el nombre del campo que contiene el valor JSON.
  4. Configure la pestaña **Fields** como lo hizo antes.



Controlando el flujo  
de datos.





# TEMAS

- **Filtrando datos.**
- **Copiando, distribuyendo, y particionando datos.**
- **Dividiendo un Stream en función de condiciones.**
- **Fusionando Stream.**
- **Buscando y obteniendo datos de un Stream secundario.**

# Filtrando datos

Ya hemos aprendido a realizar varios tipos de cálculos que enriquecen el conjunto de datos. Todavía hay otro tipo de operación que se usa frecuentemente; no tiene que ver con enriquecer los datos, sino con descartar o filtrar información no deseada. Ese es el núcleo de esta sección.

## Filtrado de filas según condiciones.

Supongamos que tiene un conjunto de datos y solo desea mantener las filas que coinciden con una condición. Para demostrar cómo implementar este tipo de filtrado, leeremos un archivo, crearemos una lista de palabras que se encuentran en el archivo y luego filtraremos los nulos o las palabras no deseadas. Dividiremos el ejercicio en dos partes:

- En la primera parte, leeremos el archivo y prepararemos los datos para el filtrado.
- En la segunda parte, filtraremos efectivamente los datos.

# Leyendo un archivo y obteniendo la lista de palabras que se encuentran en él.

## Pasos

1. Crea una nueva transformación.
2. Al utilizar el paso **Text file input**, lea su archivo. El truco aquí es poner como Separador un signo que no espera en el archivo, como | Al hacerlo, cada línea será reconocida como un solo campo. Configure la pestaña **Fields** con un solo campo string llamado línea.
3. Este archivo en particular tiene un encabezado grande que describe el contenido y el origen de los datos. No estamos interesados en esas líneas, por lo que en la pestaña **Content**, como **Header**, escriba 378, que es el número de líneas que preceden al contenido específico en el que estamos interesados.
4. Desde la categoría de pasos Transform, arrastre al lienzo **Split field to rows** y cree un salto desde el paso **Text file input** hasta este.
5. Haga doble clic en el paso. **Field to split**, seleccione **line**. En **New field name**, escriba **word**. Cerrar la ventana.
6. Con este último paso seleccionado, ejecute una vista previa. Su ventana de vista previa debe verse como sigue:

Examine preview data

Rows of step: Split field to rows (1000 rows)

#	line	n	word
1	deposit, complicated dislocations of.--Relations between ancient orifices	1	deposit,
2	deposit, complicated dislocations of.--Relations between ancient orifices	1	complicated
3	deposit, complicated dislocations of.--Relations between ancient orifices	1	dislocations
4	deposit, complicated dislocations of.--Relations between ancient orifices	1	of.--Relations
5	deposit, complicated dislocations of.--Relations between ancient orifices	1	between
6	deposit, complicated dislocations of.--Relations between ancient orifices	1	ancient
7	deposit, complicated dislocations of.--Relations between ancient orifices	1	orifices
8	of eruption and subsequent axes of injection.--Iquique, Peru, fossils of,	2	of
9	of eruption and subsequent axes of injection.--Iquique, Peru, fossils of,	2	eruption
1..	of eruption and subsequent axes of injection.--Iquique, Peru, fossils of,	2	and

**Close**   **Stop**   **Get more rows**

## Nota

Antes de comenzar, necesitarás al menos un archivo de texto para jugar. El archivo de texto utilizado en este tutorial se llama smcng10.txt. Su contenido es sobre observaciones geológicas en América del Sur por Darwin, Charles, 1809-1882 y puede descargarlo desde:

<https://archive.org/download/geologicalobserv03620gut/smcng10.txt>

## Pasos

7. Cierre la ventana de vista previa.
8. Agregue el paso **Select values** para eliminar el campo **line**.

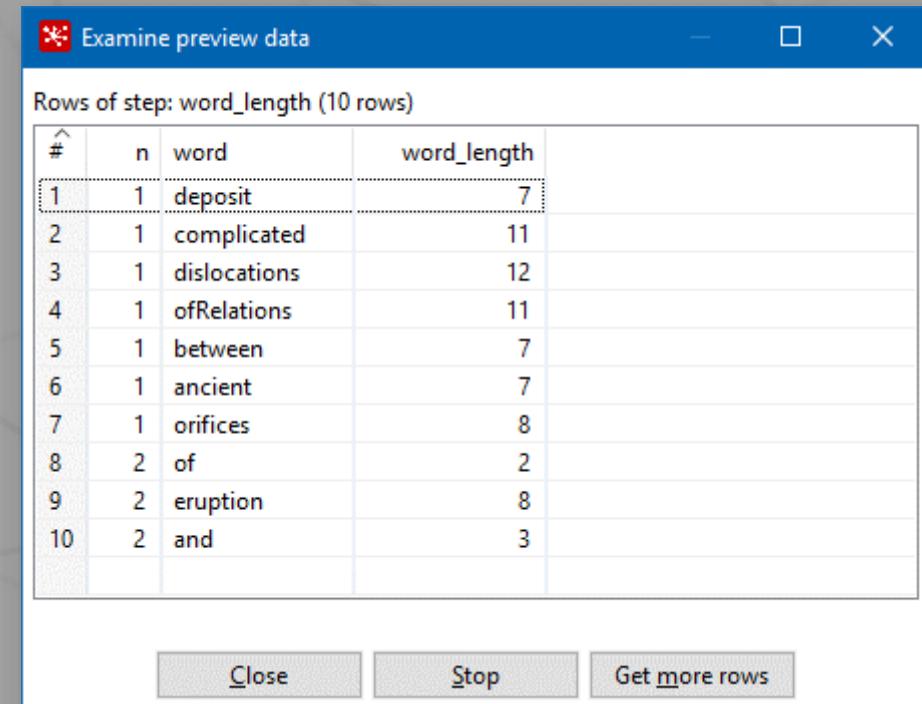
## Nota

No es obligatorio eliminar este campo, pero como ya no se utilizará más, eliminarlo hará que las futuras vistas previas sean más claras.

# Leyendo un archivo y obteniendo la lista de palabras que se encuentran en él.

## Pasos

9. Despues del último paso, agregue el paso **Replace in String**. Úsalo para eliminar todos los caracteres excepto letras y números. Para hacer eso, use la expresión regular [^ A-Z0-9].
10. Ahora crea un nuevo campo con la longitud de la palabra.
11. Ejecutar una vista previa de este último paso. Deberías ver esto:



The screenshot shows a 'Examine preview data' window with the title 'Rows of step: word\_length (10 rows)'. The table has three columns: '#', 'n', and 'word\_length'. The data is as follows:

#	n	word	word_length
1	1	deposit	7
2	1	complicated	11
3	1	dislocations	12
4	1	ofRelations	11
5	1	between	7
6	1	ancient	7
7	1	orifices	8
8	2	of	2
9	2	eruption	8
10	2	and	3

[Close](#) [Stop](#) [Get more rows](#)

## Nota

En lugar de leer el archivo con **Text file input**, podría haber implementado una solución similar utilizando **Load file content in memory**. Este paso lee todo el texto en la memoria, en un solo campo.

# Filtrado de filas no deseadas con un paso

## Filtrar filas

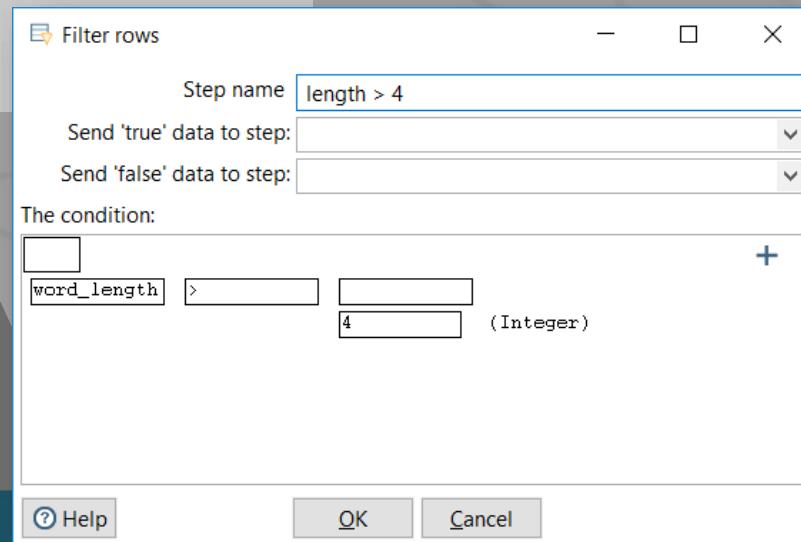
Simplemente crea una lista de palabras que provienen de un archivo de muestra. Como vería, hay filas en blanco y también muchas filas con pronombres, artículos y otras palabras muy pequeñas. Si desea conservar solo las palabras relevantes para el asunto del archivo de texto, querrá descartar todo esto. Para hacer esto, usaremos el paso Filtrar filas, un paso dedicado a filtrar filas según las condiciones y las comparaciones. En su versión más simple, para cada fila, el paso verifica una condición dada y solo pasamos aquellas filas para las cuales la condición es verdadera. Las otras filas se pierden. Las siguientes son las instrucciones para implementar esta solución:

# Filtrado de filas no deseadas con un paso

## Filtrar filas

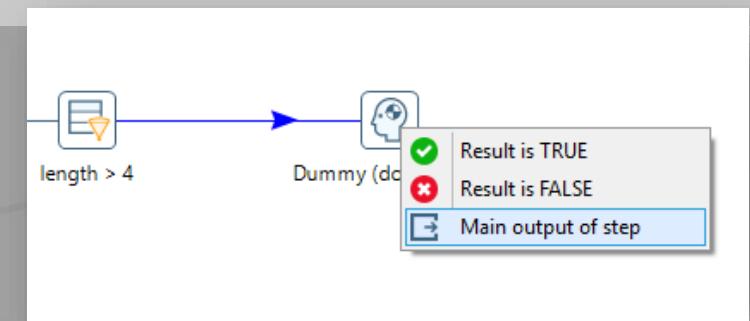
### Pasos

1. Expanda la categoría **Flow** de pasos y arrastre el paso **Filter row** al área de trabajo.
2. Cree un salto desde el último paso hasta el paso **Filter rows**.
3. Edite el paso **Filter rows** haciendo doble clic en él.
4. Haga clic en el cuadro de texto **<field>** a la izquierda del signo **=**. Aparece la lista de campos. Seleccione **word\_length**.
5. Haga clic en el signo **=**. Aparece una lista de operaciones. Seleccione **>**.
6. En el cuadro de texto **<valor>**, escriba **4**. La ventana tiene el siguiente aspecto:



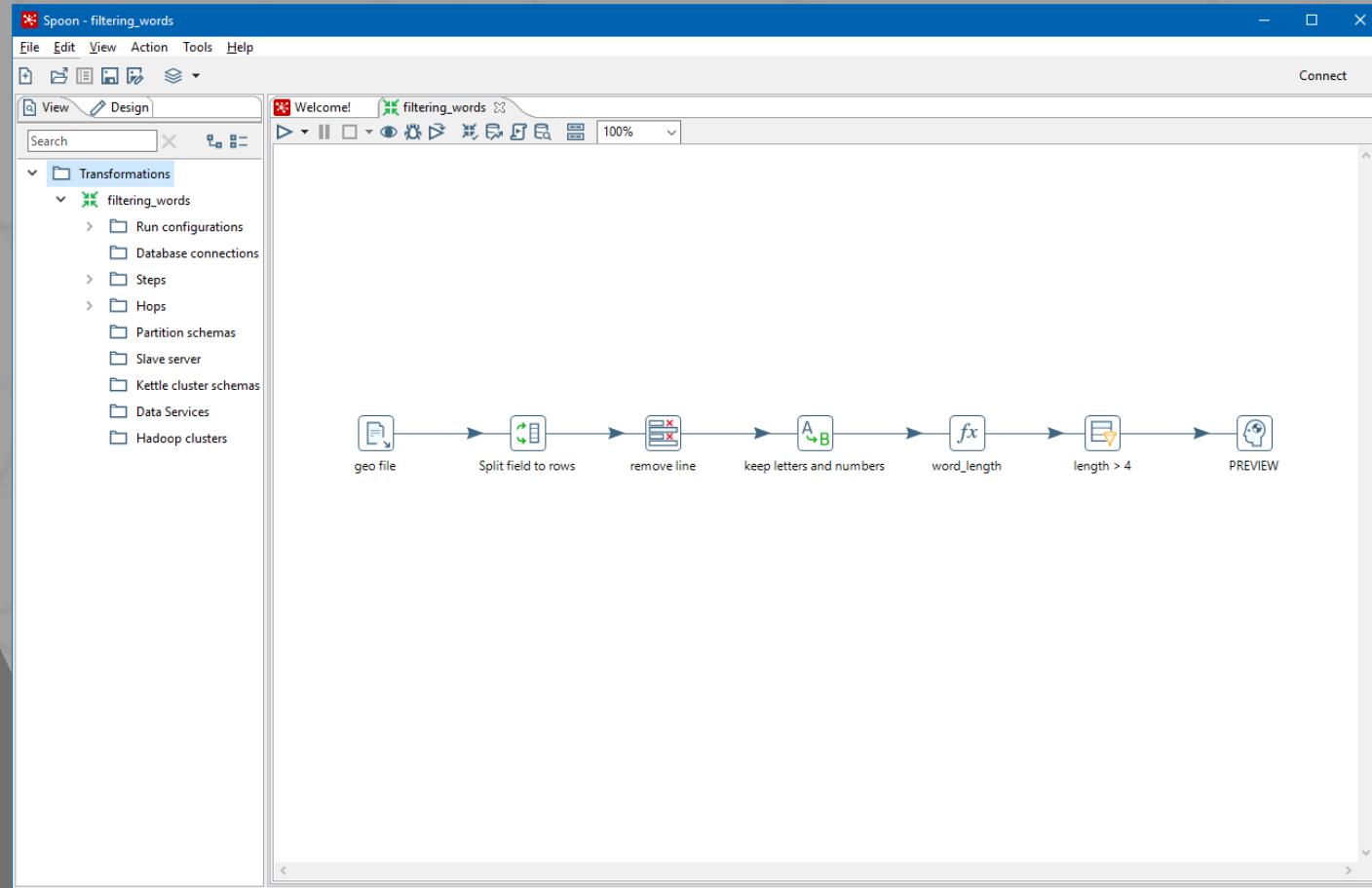
### Pasos

7. Clic en **OK**
8. En la categoría **Flow**, agregue el paso **Dummy**.
9. Crea un salto desde **Filter rows step** al paso **Dummy**. Cuando se le pregunte por el tipo de salto, **Main output of step**, como se muestra en la siguiente captura de pantalla



# Filtrado de filas no deseadas con un paso

## Filtrar filas

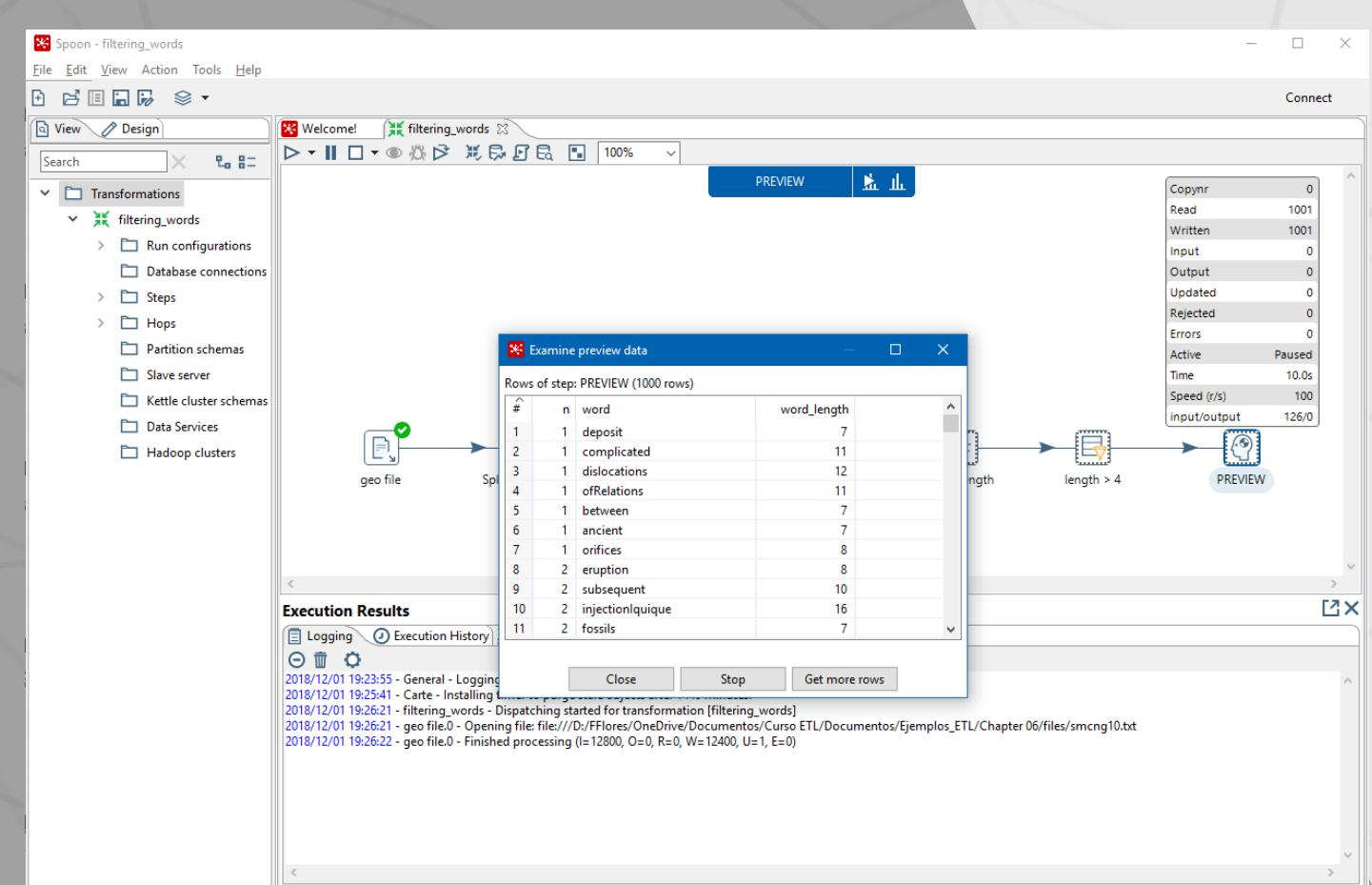


# Filtrado de filas no deseadas con un paso

## Filtrar filas

### Pasos

10. Con el paso Dummy seleccionado, ejecute una vista previa.  
Verás algo como lo siguiente:



# Filtrado de filas no deseadas con un paso

## Filtrar filas

- **word IS NOT NULL** en este caso, solo pasan las filas donde las palabras no son nulas ni con valores vacíos
- **line STARTS WITH word** en este caso, la fila pasa solo si el campo de **word** coincide con los primeros caracteres del campo de **line**. Tenga en cuenta que en este ejemplo el filtro incluye dos campos.
- **word REGEXP (g|e).+** este filtro permite pasar solo las palabras que comienzan con “g” o con “e”
- **word in list geology; sun** en este caso, solo las filas con palabras **geology** y **sun** pasan el filtro

# Filtrado de filas mediante el paso de filtro de Java

- Como alternativa al paso **Filter rows**, hay otro paso para el mismo propósito: el paso **Java Filter**. Este paso es útil cuando sus condiciones son demasiado complicadas y resulta difícil o imposible crearlas en un paso **Filter rows** regular.
- Con el paso **Java Filter**, en lugar de crear la condición de forma interactiva, escribe una expresión de Java que se evalúa como verdadera o falsa.
- La misma condición se puede expresar con una expresión Java como:

```
(word.matches(".*Geo.*") && word_length>8) || word.equals ("earth")  
|| word.equals ("rocks").
```

# Filtrado de filas mediante el paso de filtro de Java

## Pasos

1. Abra la transformación creada en el ejercicio anterior y guárdela con un nombre diferente.
2. Eliminar Filter rows y el paso **Dummy**.
3. Desde la categoría de **Flow**, agregue el paso **Java Filter**.
4. Haga doble clic en el paso **Java Filter**.
5. En el cuadro de texto **Condition (Java expression)**, Escribimos:

```
(word.matches(".*Geo.*") &&
word_length>8) || word.equals ("tierra") ||
word.equals("rocas") .
```

6. Cerrar la ventana.
7. Con Java Filter seleccionado, ejecute una vista previa



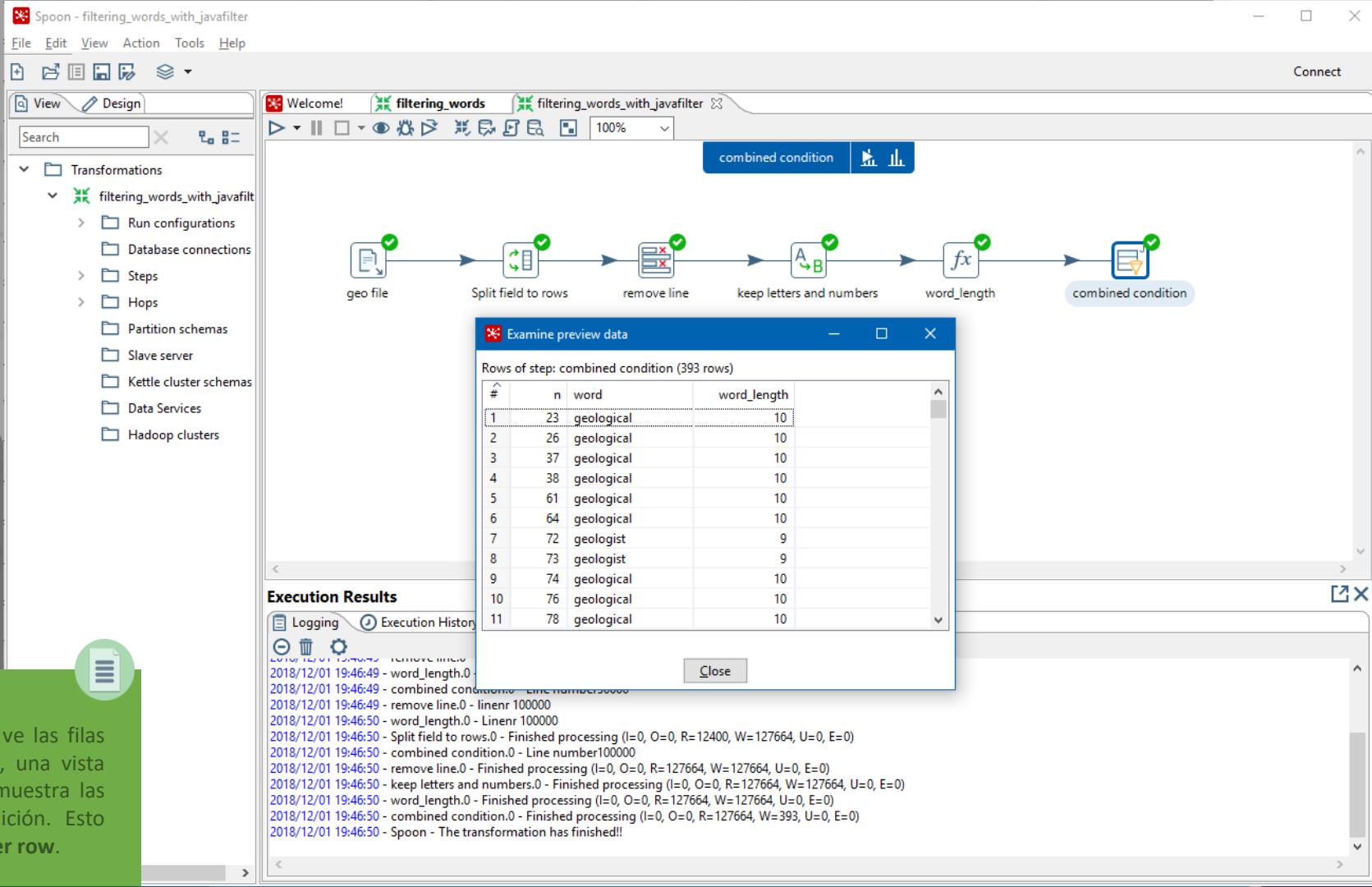
Examine preview data

Rows of step: combined condition (393 rows)

#	n	word	word_length
1	23	geological	10
2	26	geological	10
3	37	geological	10
4	38	geological	10
5	61	geological	10
6	64	geological	10
7	72	geologist	9
8	73	geologist	9
9	74	geological	10
10	76	geological	10
11	78	geological	10

Close

# Filtrado de filas mediante el paso de filtro de Java



The screenshot shows the Apache Kettle Spoon interface. On the left, the 'Transformations' tree view shows a single transformation named 'filtering\_words\_with\_javafilter'. The main workspace displays a flow starting with a 'geo file' input, followed by a 'Split field to rows' step, a 'remove line' step, a 'keep letters and numbers' step (which includes a 'A-B' range configuration), a 'word\_length' step (a Java Filter step), and finally a 'combined condition' step. A preview window titled 'Examine preview data' shows 393 rows of data with columns 'n' and 'word', and a 'word\_length' column. The data includes multiple entries for the word 'geological'. Below the preview is an 'Execution Results' panel displaying the log output of the transformation's execution.

```

Rows of step: combined condition (393 rows)
# n word word_length
1 23 geological 10
2 26 geological 10
3 37 geological 10
4 38 geological 10
5 61 geological 10
6 64 geological 10
7 72 geologist 9
8 73 geologist 9
9 74 geological 10
10 76 geological 10
11 78 geological 10

```

Execution Results

```

2018/12/01 19:46:49 - word_length.0
2018/12/01 19:46:49 - combined condition.0 - Line number100000
2018/12/01 19:46:50 - remove line.0 - linenr 100000
2018/12/01 19:46:50 - word_length.0 - Linenr 100000
2018/12/01 19:46:50 - Split field to rows.0 - Finished processing (I=0, O=0, R=12400, W=127664, U=0, E=0)
2018/12/01 19:46:50 - combined condition.0 - Line number100000
2018/12/01 19:46:50 - remove line.0 - Finished processing (I=0, O=0, R=127664, W=127664, U=0, E=0)
2018/12/01 19:46:50 - keep letters and numbers.0 - Finished processing (I=0, O=0, R=127664, W=127664, U=0, E=0)
2018/12/01 19:46:50 - word_length.0 - Finished processing (I=0, O=0, R=127664, W=127664, U=0, E=0)
2018/12/01 19:46:50 - combined condition.0 - Finished processing (I=0, O=0, R=127664, W=393, U=0, E=0)
2018/12/01 19:46:50 - Spoon - The transformation has finished!

```

## Nota

Cuando ejecuta una vista previa, ve las filas que salen de un paso. Como tal, una vista previa del paso **Java Filter** solo muestra las filas que coinciden con la condición. Esto también es cierto para el paso **Filter row**.

# Filtrado de datos basados en números de fila

- Hasta ahora ha estado filtrando según las condiciones de los valores de los campos. También puede filtrar filas en función de los números de fila. Hay un par de pasos que nos permiten hacer eso. Aquí hay un breve resumen de ellos:

# Filtrado de datos basados en números de fila

Paso	Descripción
<b>Sample rows (Statistics category)</b>	Este paso muestra las filas basándose en una lista de números de filas o rangos de números de filas. Por ejemplo, 1,5,10..20 filtrará la fila 1, la fila 5 y todas las filas desde 10 hasta 20 (10 y 20 incluidas).
<b>Reservoir Sampling (Statistics category)</b>	Este paso le permite muestrear un número fijo de filas. El paso utiliza un muestreo uniforme, lo que significa que todas las filas entrantes tienen la misma posibilidad de ser seleccionadas.
<b>Top / Bottom / First / Last filter (Transform category)</b>	Este complemento le permite filtrar las primeras N filas o las últimas N filas de su conjunto de datos.
<b>Filter rows / Java Filter steps (Flow category)</b>	Estos pasos ofrecen una alternativa general para filtrar una o más filas según sus números. La única condición previa para hacerlo es tener un campo en su conjunto de datos que contenga el número de fila.
<b>Identify last row in a stream + Filter rows or Java Filter</b>	<b>Identify last row in a stream</b> genera un campo booleano que es verdadero solo para la última fila en el conjunto de datos. En función de ese campo, puede filtrar la última fila con <b>Filter row</b> o con el paso <b>Java Filter</b> .

## Nota

 **Top / Bottom / First / Last filter step** es un plugin que puedes instalar a través del Marketplace.

# División de streams incondicionalmente

## Copiando filas

En cualquier lugar de una transformación, puede decidir dividir la transmisión principal en dos o más transmisiones. Cuando lo haga, debe decidir qué hacer con los datos que dejan el último paso: **copy** o **distribute**.

Copiar significa que todo el conjunto de datos se copia en cada uno de los pasos de destino. ¿Por qué copiarías todo el conjunto de datos? Principalmente porque desea aplicar diferentes tratamientos al mismo conjunto de datos.

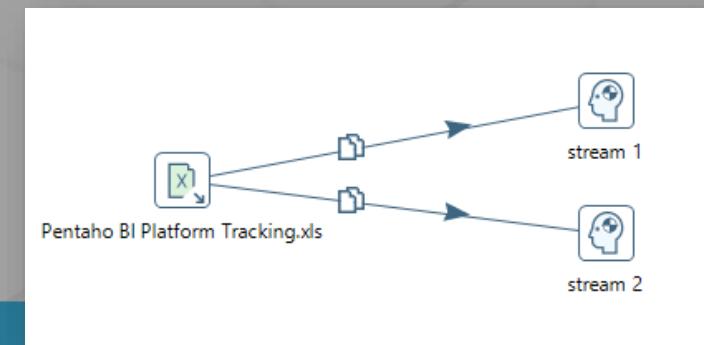
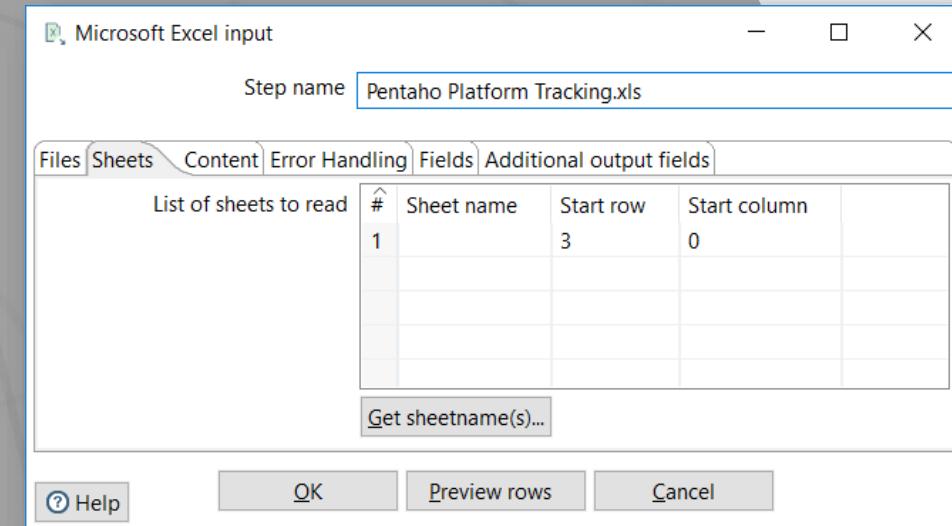
# División de streams incondicionalmente

## Pasos

1. Crear una transformación.
2. Lea el archivo exportado desde JIRA utilizando el paso de entrada de Microsoft Excel. Después de proporcionar el nombre del archivo, haga clic en la pestaña Hojas y rellénelo como se muestra en la siguiente captura de pantalla, para que omita las filas del encabezado y la primera columna.
3. Haga clic en la pestaña **Fields** y complete la cuadrícula haciendo clic en el botón **Get fields from header row ...**
4. Haga clic en **Preview rows** para asegurarse de que está leyendo el archivo correctamente. Debería ver todos los contenidos del archivo de Excel, excepto la primera columna y las tres líneas de encabezado.
5. Haga clic en OK.
6. En la categoría **Flow**, agregue dos pasos **Dummy**.
7. Cree un salto desde el paso de **Excel Input** hacia uno de los pasos **Dummy**.
8. Ahora cree un segundo salto desde el paso **Excel Input** hacia el segundo paso Dummy.
9. Guarde la transformación y ejecútelo.

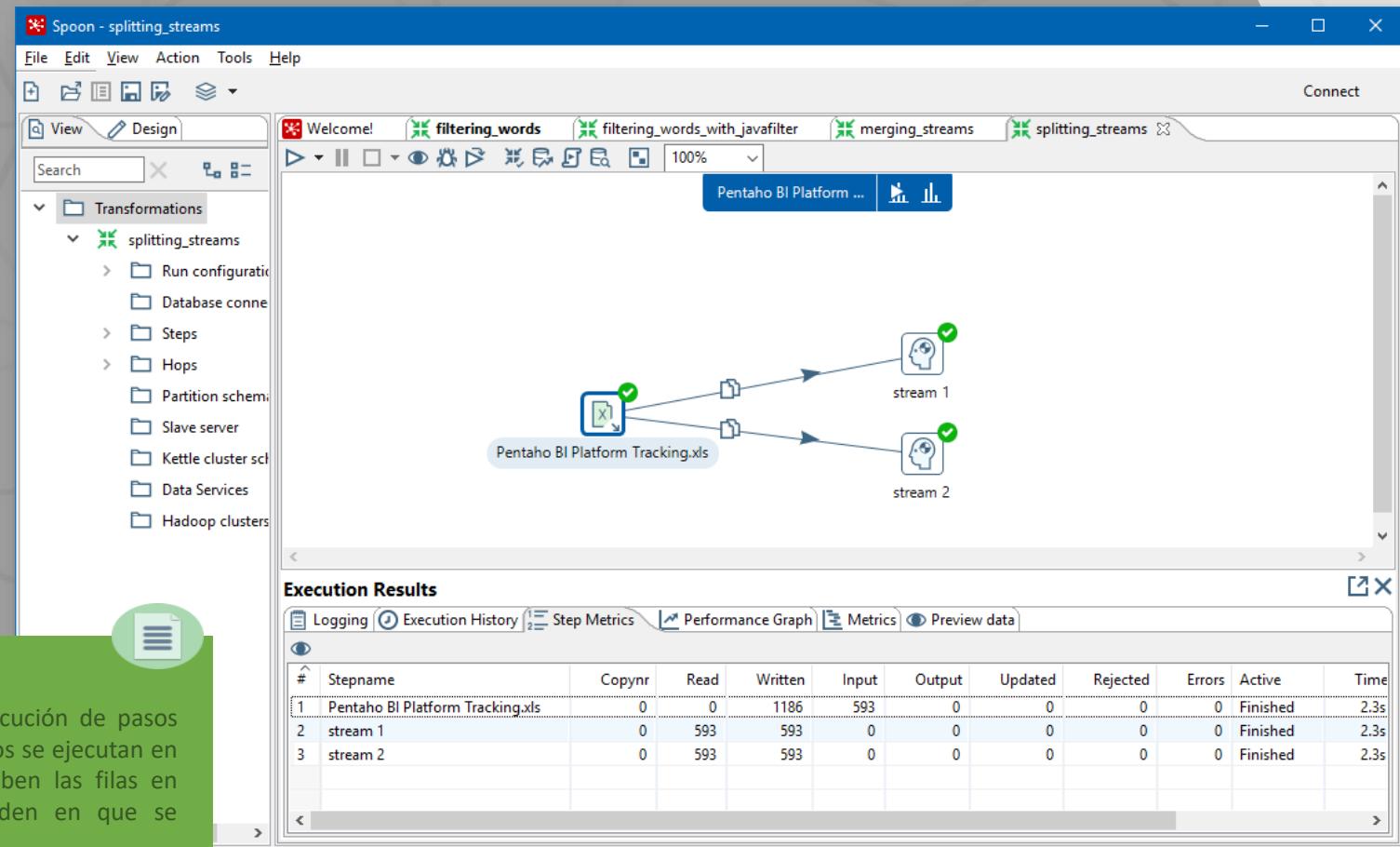
### Advertencia

Aparecerá una ventana de advertencia que le pedirá que decida si desea copiar o distribuir filas. Haz clic en Copiar ...



# División de streams incondicionalmente

Miramos la pestaña de métricas de pasos en la ventana Ejecución:

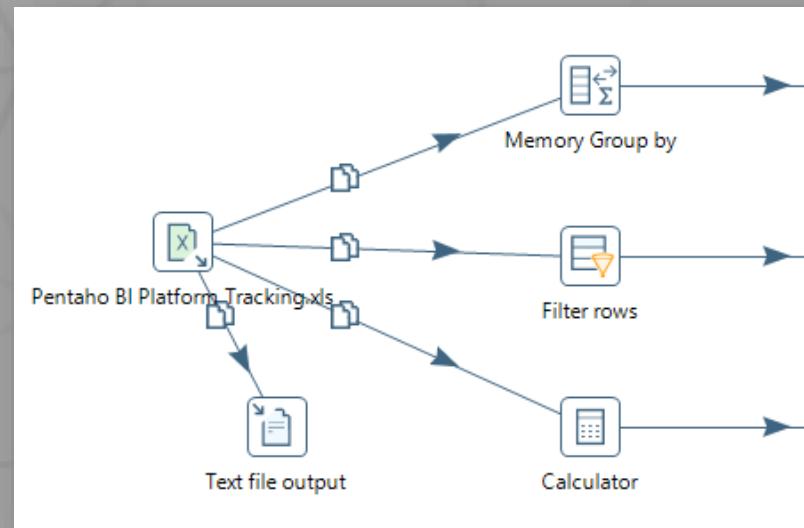


## Nota

No debe asumir un orden particular en la ejecución de pasos debido a su naturaleza asíncrona. Como los pasos se ejecutan en paralelo y todas las secuencias de salida reciben las filas en sincronización, no tiene control sobre el orden en que se ejecutan.

# División de streams incondicionalmente

Miramos la pestaña de métricas de pasos en la ventana Ejecución:



## Nota

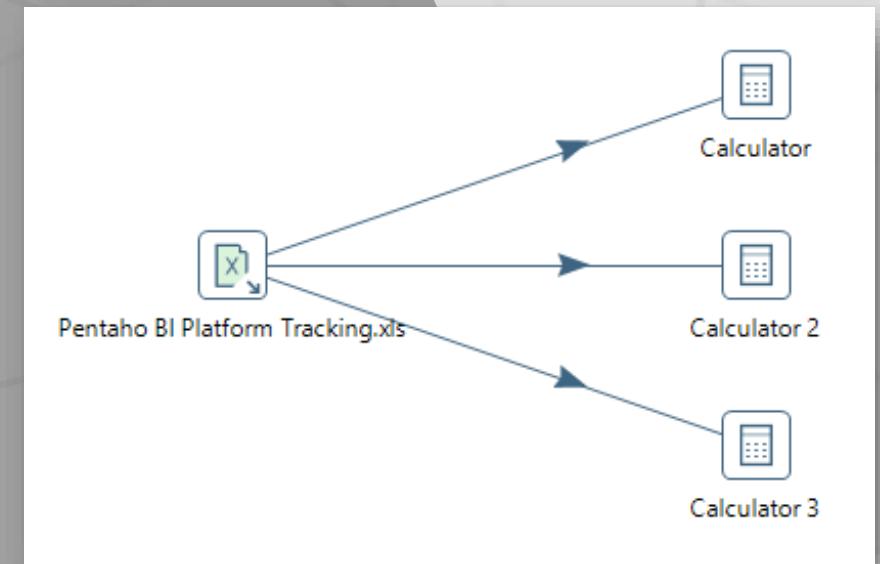
No debe asumir un orden particular en la ejecución de pasos debido a su naturaleza asíncrona. Como los pasos se ejecutan en paralelo y todas las secuencias de salida reciben las filas en sincronización, no tiene control sobre el orden en que se ejecutan.

# Filas de distribución

Como se dijo, cuando se divide una secuencia, puede copiar o distribuir las filas. Copiar es crear copias de todo el conjunto de datos y enviar cada una de ellas a cada flujo de salida. Distribuir significa que las filas del conjunto de datos se distribuyen entre los pasos de destino. Esos pasos se ejecutan en subprocessos separados, por lo que la distribución es una forma de implementar el procesamiento paralelo.

Cuando distribuyes, los pasos de destino reciben las filas en forma de round-robin. Por ejemplo, si tiene tres pasos de destino, como por ejemplo, las tres calculadoras en la siguiente captura de pantalla, la primera fila de datos va al primer paso de destino, la segunda fila va al segundo paso, la tercera fila va al tercer paso , la cuarta fila va al cuarto paso, y así sucesivamente.

Visualmente, cuando las filas se distribuyen, los saltos que salen de los pasos desde los que se distribuyen son simples; no cambian su apariencia, como se muestra a continuación:



# Filas de distribución

Para evitar que se le solicite la acción de copiar cada vez que cree más de un salto al salir de un paso, puede configurar la opción Copiar de manera predeterminada. Para hacerlo, abra la ventana de opciones de PDI (**Tools | Options...** en el menú principal) y deseleccione la opción **Show Copy or Distribute dialog**. Recuerde que para ver el cambio aplicado, deberá reiniciar Spoon.

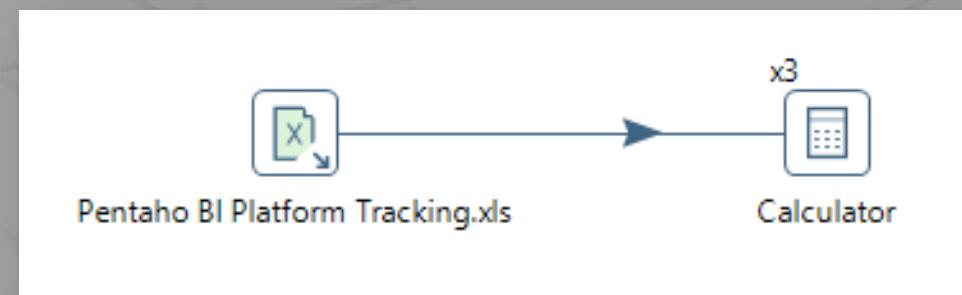
# Filas de distribución

Una vez que haya cambiado esta opción, el método predeterminado es Copiar filas. Si desea distribuir filas, puede cambiar la acción haciendo clic con el botón derecho en el paso desde el que desea copiar o distribuir, seleccionando Movimiento de datos ... en el menú contextual que aparece y luego seleccionando la opción deseada.

Otra forma de distribuir es cambiar el número de copias de un paso. Las siguientes instrucciones paso a paso crean tres copias del paso de la Calculadora. El resultado es técnicamente equivalente al ejemplo anterior:

## Pasos

1. Haga clic derecho en un paso.
2. En el menú contextual, seleccione **Change Number of Copies to Start ....**
3. En **Number of copies**, especifique 3. Cierre la ventana. El look and feel cambia de la siguiente manera:



# División de un flujo basado en una condición simple

Para esta subsección, trabajaremos una vez más con el archivo de Excel exportado desde JIRA, que contiene todas las nuevas características propuestas para las próximas versiones de PDI. Esta vez, suponga que desea implementar diferentes tratamientos a los datos, dependiendo de la gravedad de los problemas. Por ejemplo:

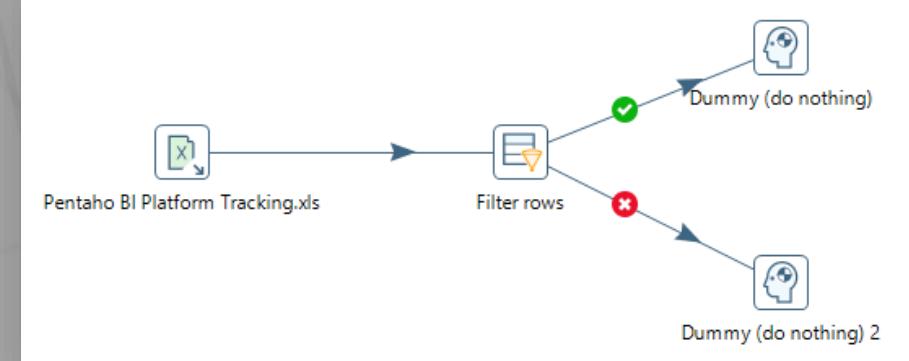
- Con los problemas que tienen una gravedad Urgente o Alta, desea crear una lista y enviarla por correo electrónico
- Con el resto de los problemas, desea crear un orden de archivos de Excel por estado y gravedad, y copiar esta salida en una carpeta compartida para ser revisada

La fuente de ambas salidas es la misma, pero dependiendo de las características de los problemas, las filas deben tomar una u otra forma.

# División de un flujo basado en una condición simple

## Pasos

1. Crea una nueva transformación.
2. Lea el archivo de Excel tal como lo hizo en la sección anterior.
3. Agregue el paso **Filter rows** y cree un salto desde el paso de entrada de Excel hacia este.
4. Agrega dos pasos **Dummy**. Los usaremos como destino de las filas que provienen del paso **Filter rows**.
5. Crea un salto desde la **Filter rows** a uno de esos pasos. Como el tipo de salto, seleccione **Result is TRUE**.
6. Cree un salto desde el paso **Filter row** al otro paso **Dummy**. Esta vez, como el tipo de salto, seleccione **Result is FALSE**. La transformación se ve como sigue
7. Haga doble clic en el paso **Filter row** para editarlo.
8. Ingrese la condición **Severity = [High] OR Severity = [Urgent]**. Luego cierra la ventana.



**Examine preview data**

Rows of step: High and Urgent Issues (113 rows)

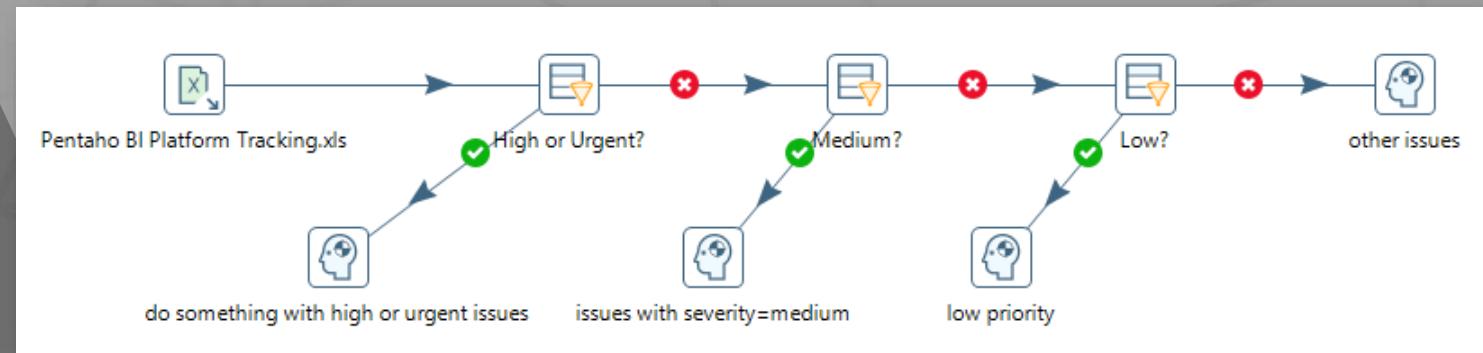
#	Issue Type	Summary	Story ...	Assignee	Sub...	Severity	Status	Resolution
1	New Feature	HBase Output step does not work with Spark AEL	0.0	Unassigned	<n...	Urgent	Open	Unresolved
2	New Feature	XML Output step incorrectly encodes 'less-than' character	0.0	Unassigned	<n...	Urgent	Open	Unresolved
3	New Feature	Provide an option/ability to read hyperlink URL's from an Excel...	0.0	Unassigned	<n...	High	Open	Unresolved
4	New Feature	Add Metadata Injection (MDI) support to the Database Looku...	0.0	Unassigned	<n...	High	Open	Unresolved
5	New Feature	Data Services can't handle SQL "CASE" in ORDER BY	0.0	Unassigned	<n...	High	Open	Unresolved
6	New Feature	Data Services - Can't do operations between fields (example: ...	0.0	Unassigned	<n...	High	Open	Unresolved
7	New Feature	Include Microsoft JDBC driver by default	0.0	Unassigned	<n...	High	Open	Unresolved
8	New Feature	Metadata injection on metadata injection should support dyna...	0.0	Unassigned	<n...	High	Open	Unresolved
9	New Feature	When we output the same field type (date) multiple times onl...	0.0	Unassigned	<n...	High	Open	Unresolved
1..	New Feature	Expand Remote Job via API call	0.0	Unassigned	<n...	High	Open	Unresolved

## Nota

Alternativamente, puedes usar una sola condición:  
**Severity IN LIST High;Urgent**

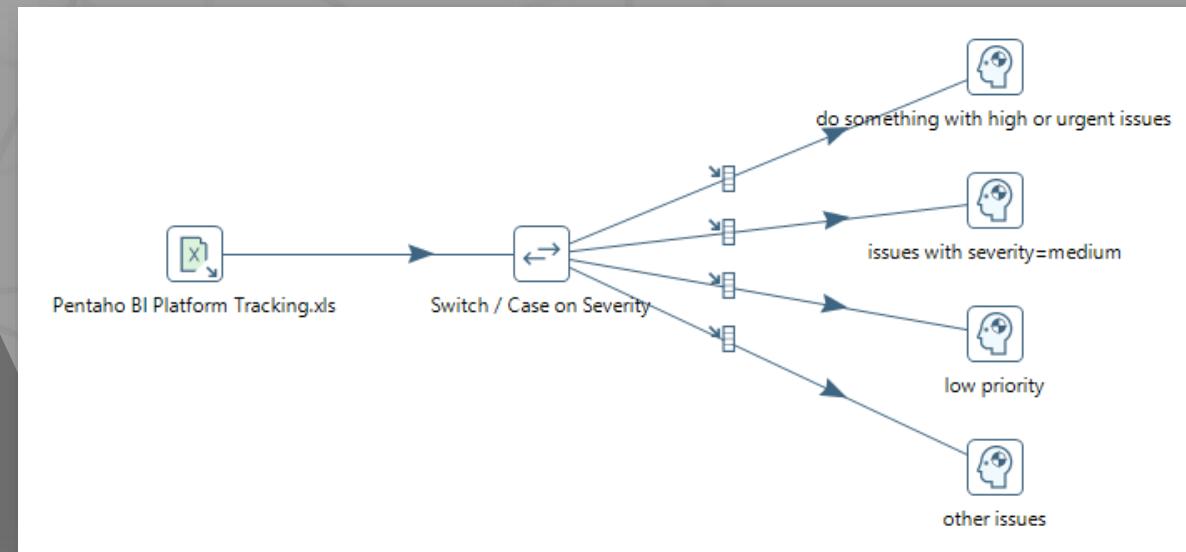
# Explorando pasos PDI para dividir un flujo basado en condiciones

Cuando tiene que tomar una decisión y, tras esa decisión, dividir el flujo en dos, puede usar el paso **Filter rows**, como lo hizo en este último ejercicio. Alternativamente, puede utilizar el paso **Java Filter**. Como se dijo en el Manipulación de datos y metadatos de PDI, el propósito de ambos pasos, Filtrar filas y Java, es el mismo; La principal diferencia es la forma en que escribe o ingresa las condiciones. A veces hay que tomar decisiones anidadas; por ejemplo:



# Transformación de la muestra con condiciones anidadas.

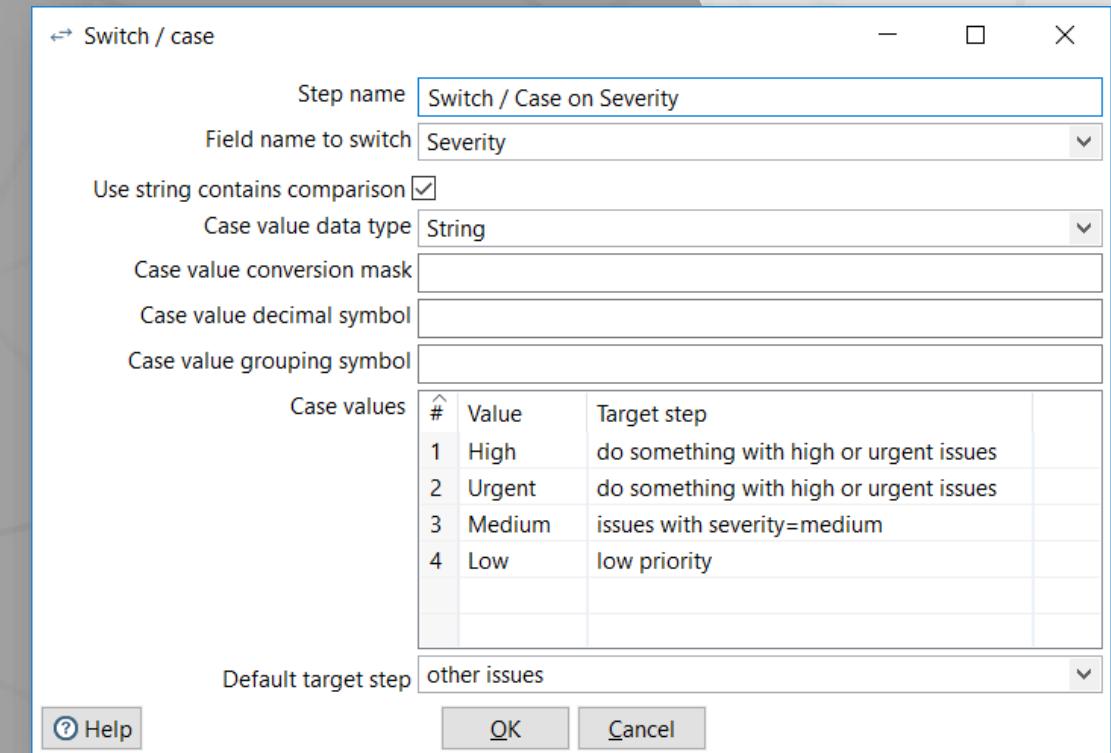
Cuando las condiciones son tan simples como comprobar si un campo es igual a un valor, tiene una solución alternativa más simple utilizando el paso Switch / case. Este paso, agrupado en la categoría Flow, dirige las filas de datos a uno o más pasos de destino, según el valor encontrado en un campo determinado. Si tomamos el ejemplo anterior y reemplazamos los filtros con el paso **Switch / case**, tendríamos algo como lo siguiente:



# Usando el paso Switch / Case

La configuración para el paso Switch / case se vería de la siguiente manera:

Con esta solución, el campo a usar para las comparaciones es la **Severity** y los valores con los que comparamos son **Urgent**, **High**, **Medium** y **Low**. Dependiendo de los valores del campo **Severity**, las filas se enviarán a cualquiera de los pasos de destino.



# Fusionando Streams de varias maneras.

Acaba de ver cómo las filas de un conjunto de datos pueden tomar diferentes rutas. Aquí aprenderá lo contrario, cómo los datos procedentes de diferentes lugares se fusionan en una sola secuencia. Vamos a dividir esta sección en dos partes:

Fusionando dos o más Streams

Personalizando la forma de fusionar Streams.

## Nota

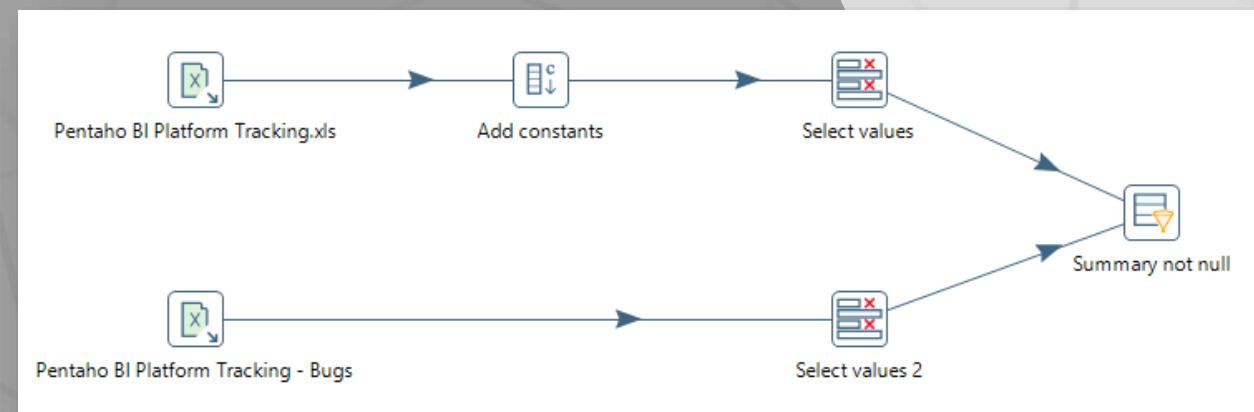
El concepto más importante que debe conocer antes de fusionar dos o más flujos es que los conjuntos de datos tienen que compartir los mismos metadatos. Entonces, en nuestro ejemplo, además de leer los dos archivos de Excel, tenemos que transformar los conjuntos de datos de manera que ambos se vean iguales.



# Fusionando Streams de varias maneras.

## Pasos

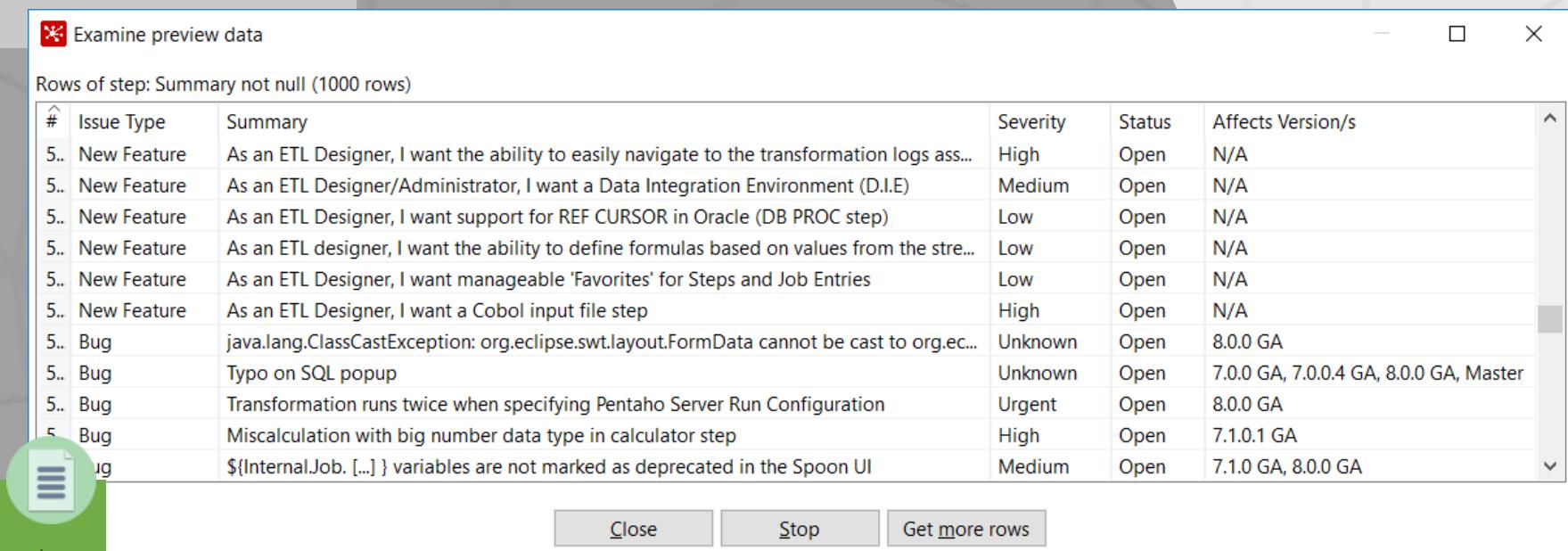
1. Crea una nueva transformación.
2. Arrastra al lienzo el paso **Microsoft Excel Input** y lea el archivo de Excel con las nuevas funciones.
3. Como el archivo no tiene el campo **Affects Version/s**, lo crearemos con un valor predeterminado. Entonces, desde la categoría **Transform** de pasos, agregue el paso **Add constants** y cree un salto desde el primer paso de **Excel input** hacia este paso.
4. Use el paso **Add constants** para agregar un campo de tipo String, con el nombre **Affects Version/s** y el valor **N / A**.
5. Finalmente, agregue el paso **Select values** para mantener solo los campos propuestos: **Issue Type, Summary, Severity, Status, and Affects Version/s**.
6. Agregue otro paso de **Microsoft Excel Input** y utilícelo para leer el nuevo archivo.
7. Despues del paso **Filter rows**, agregue otro paso **Select values** y configúrelo para mantener la misma lista de campos que mantuvo con el otro paso **Select values**.
8. En la categoría **Flow**, agregue el paso **Filter rows**.
9. Cree un salto desde cada **Select values** hacia este paso. Cuando se le pregunte por el tipo de salto, seleccione **Main output of step**. Tu transformación debe verse como la siguiente:



# Fusionando Streams de varias maneras.

## Pasos

10. Utilice el paso **Filter rows** para filtrar las filas con **Summary** nulo..
11. Con el paso **Filter rows** seleccionado, ejecute una vista previa. Verás lo siguiente:



The screenshot shows a 'Examine preview data' window from PDI. The title bar says 'Examine preview data'. The main area is titled 'Rows of step: Summary not null (1000 rows)'. It contains a table with columns: #, Issue Type, Summary, Severity, Status, and Affects Version/s. The table lists various bugs and new features, such as 'New Feature' entries for ETL Designer features and 'Bug' entries for Java exceptions and UI issues. At the bottom right of the window are buttons for Close, Stop, and Get more rows.

#	Issue Type	Summary	Severity	Status	Affects Version/s
5..	New Feature	As an ETL Designer, I want the ability to easily navigate to the transformation logs ass...	High	Open	N/A
5..	New Feature	As an ETL Designer/Administrator, I want a Data Integration Environment (D.I.E)	Medium	Open	N/A
5..	New Feature	As an ETL Designer, I want support for REF CURSOR in Oracle (DB PROC step)	Low	Open	N/A
5..	New Feature	As an ETL designer, I want the ability to define formulas based on values from the stre...	Low	Open	N/A
5..	New Feature	As an ETL Designer, I want manageable 'Favorites' for Steps and Job Entries	Low	Open	N/A
5..	New Feature	As an ETL Designer, I want a Cobol input file step	High	Open	N/A
5..	Bug	java.lang.ClassCastException: org.eclipse.swt.layout.FormData cannot be cast to org.ec...	Unknown	Open	8.0.0 GA
5..	Bug	Typo on SQL popup	Unknown	Open	7.0.0 GA, 7.0.0.4 GA, 8.0.0 GA, Master
5..	Bug	Transformation runs twice when specifying Pentaho Server Run Configuration	Urgent	Open	8.0.0 GA
5..	Bug	Miscalculation with big number data type in calculator step	High	Open	7.1.0.1 GA
5..	Bug	\${Internal.Job. [...] } variables are not marked as deprecated in the Spoon UI	Medium	Open	7.1.0 GA, 8.0.0 GA

## Nota

Tenga en cuenta que el PDI le advierte pero no le impide mezclar diseños de filas al crear la Transformación. Si desea que PDI le impida ejecutar Transformaciones con diseños de fila mixtos, puede marcar la opción **Enable safe mode** en la ventana que aparece cuando envía la transformación. Tenga en cuenta que hacer esto causará una caída en el rendimiento.

# Personalizando la forma de fusionar Streams.

Si le importa el orden en que se realiza la unión, hay algunos pasos que pueden ayudarlo. Aquí están las opciones que tienes

Si quieres ...	Puedes hacerlo ...
Añade dos o más Streams y no te importa el orden.	Usa cualquier paso. El paso seleccionado tomará todas las secuencias entrantes en cualquier orden y luego continuará con la tarea específica.
Anexar dos o más flujos en un orden dado	Para dos flujos, use el paso <b>Append streams</b> de la categoría <b>Flow</b> . Te permite decidir qué secuencia va primero.  Para dos o más, use el paso <b>Prioritize streams</b> de la categoría <b>Flow</b> . Te permite decidir el orden de todas las secuencias entrantes.
Fusionar dos Streams ordenadas por uno o más campos	Utilice el paso Combinado ordenado de la categoría Uniones. Este paso le permite decidir en qué campo (s) ordenar las filas entrantes antes de enviarlos a los pasos de destino. Ambos flujos de entrada deben ser ordenado en ese campo (s)

# Personalizando la forma de fusionar Streams.

Si quieres ...	Puedes hacerlo ...
Fusionar dos flujos manteniendo el más reciente cuando hay duplicados	<p>Use el paso Merge Rows (diff) de la categoría Joins.</p> <p>Le dice a PDI los campos clave, es decir, los campos que le dicen que una fila es la misma en ambas secuencias. También le da a PDI los campos para comparar cuando la fila se encuentra en ambos flujos.</p> <p>PDI intenta hacer coincidir las filas de ambas secuencias en función de los campos clave. Luego crea un campo que actuará como una bandera y lo rellena de la siguiente manera:</p> <ul style="list-style-type: none"><li>• Si solo se encontró una fila en la primera secuencia, la bandera se establece en <b>deleted</b></li><li>• Si solo se encontró una fila en la segunda secuencia, el indicador se establece en <b>nuevo</b></li><li>• Si la fila se encontró en ambos flujos y los campos a comparar son los mismos, la bandera se establece en <b>identical</b></li><li>• Si la fila se encontró en ambas secuencias, y al menos uno de los campos para comparar es diferente, el indicador se establece en <b>changed</b></li></ul>

## Nota

Ya sea que utilice pasos arbitrarios o algunos de los pasos especiales mencionados aquí para combinar flujos, no olvide verificar el diseño de los flujos que está fusionando. Preste atención a las advertencias del detector de trampas y evite mezclar diseños de filas.



# Buscando datos con un paso de búsqueda de Stream

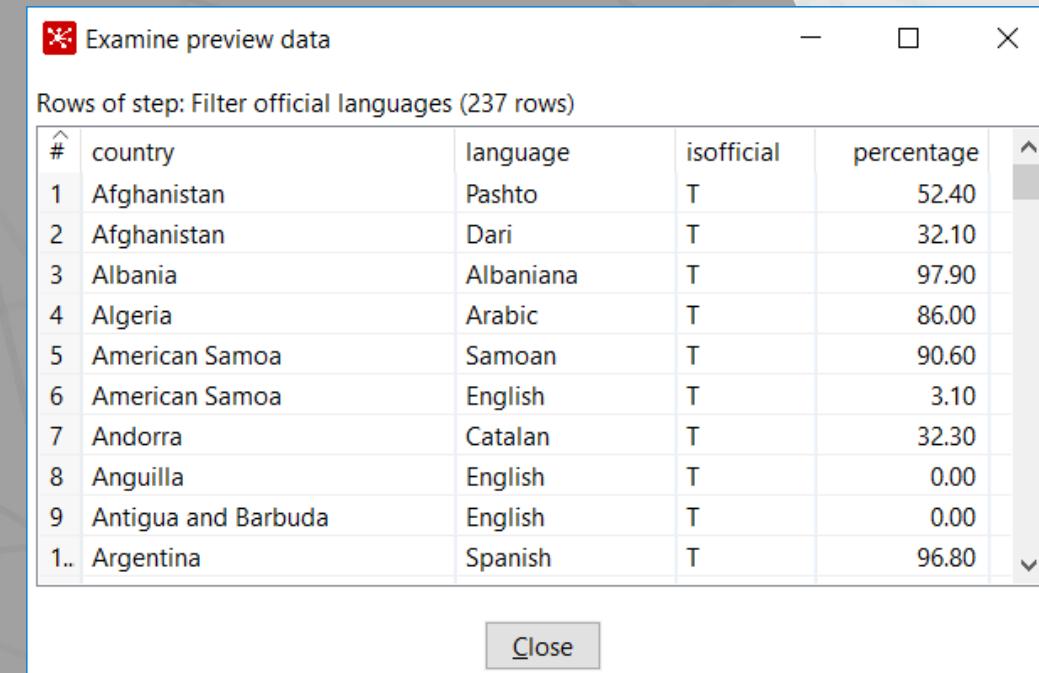
Supongamos que tiene una lista de personas junto con sus nacionalidades y desea saber el idioma que hablan. Este es un caso típico en el que tiene que buscar información en otra fuente de datos, su flujo principal es el conjunto de datos con información de personas y necesita un flujo secundario con información sobre los idiomas. Este flujo secundario es donde buscaremos nueva información. En PDI, lo hacemos con el paso de búsqueda de Stream.

Para explicar cómo usar este paso, implementaremos el ejercicio propuesto donde leeremos una lista de personas y descubriremos los idiomas que hablan las personas en la lista:

# Buscando datos con un paso de búsqueda de Stream

## Pasos

1. Crea una nueva transformación.
2. Al usar el paso **Get data from XML**, lea el archivo con información sobre los países que utilizó en el “Manipulación de datos y metadatos de PDI”, country.xml.
3. Arrastre al lienzo el paso **Filter rows**.
4. Cree un salto desde el paso **Get data from XML** hasta el paso **Filter rows**.
5. Edite el paso **Filter rows** y cree la condición **isofficial = T**.
6. Haga clic en **Filter rows** y ejecute una vista previa. La lista de filas previsualizadas mostrará los países junto con los idiomas oficiales, como se muestra en la siguiente captura de pantalla:



The screenshot shows a modal window titled "Examine preview data" with the sub-tittle "Rows of step: Filter official languages (237 rows)". The window contains a table with columns: #, country, language, isofficial, and percentage. The data includes:

#	country	language	isofficial	percentage
1	Afghanistan	Pashto	T	52.40
2	Afghanistan	Dari	T	32.10
3	Albania	Albaniana	T	97.90
4	Algeria	Arabic	T	86.00
5	American Samoa	Samoan	T	90.60
6	American Samoa	English	T	3.10
7	Andorra	Catalan	T	32.30
8	Anguilla	English	T	0.00
9	Antigua and Barbuda	English	T	0.00
1..	Argentina	Spanish	T	96.80

**Close**

# Buscando datos con un paso de búsqueda de Stream

Ahora que tenemos la ruta donde buscaremos datos, vamos a crear el flujo principal de datos:

## Pasos

1. Utilizamos el siguiente archivo (ver imagen).
2. En la misma transformación, arrastre al lienzo Text file input.



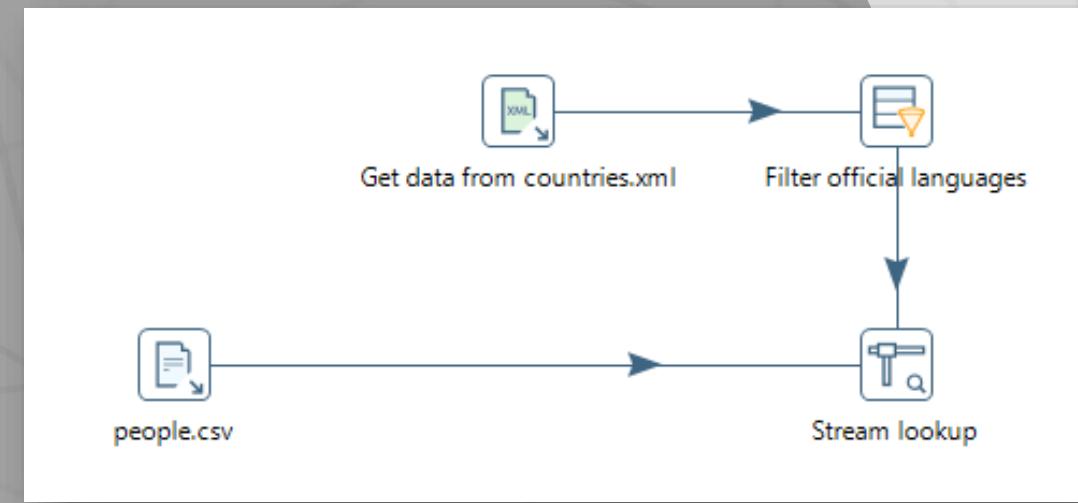
```
ID;Country Name;Name
1;Russia;Mikhail Davydova
2;;Anastasia Davydova
3;Spain;Carmen Rodriguez
4;;Francisco Delgado
5;Japan;Natsuki Harada
6;;Emiko Suzuki
7;China;Lin Jiang
8;;Wei Chiu
9;United States;Chelsea Thompson
10;;Cassandra Sullivan
11;Canada;Mackenzie Martin
12;;Nathan Gauthier
13;Italy;Giovanni Lombardi
14;;Federica Lombardi
```

Tenemos dos **streams**, la principal con la lista de personas, y la segunda con la lista de países y sus idiomas oficiales. Ahora vamos a hacer la tarea principal.

# Buscando datos con un paso de búsqueda de Stream

## Pasos

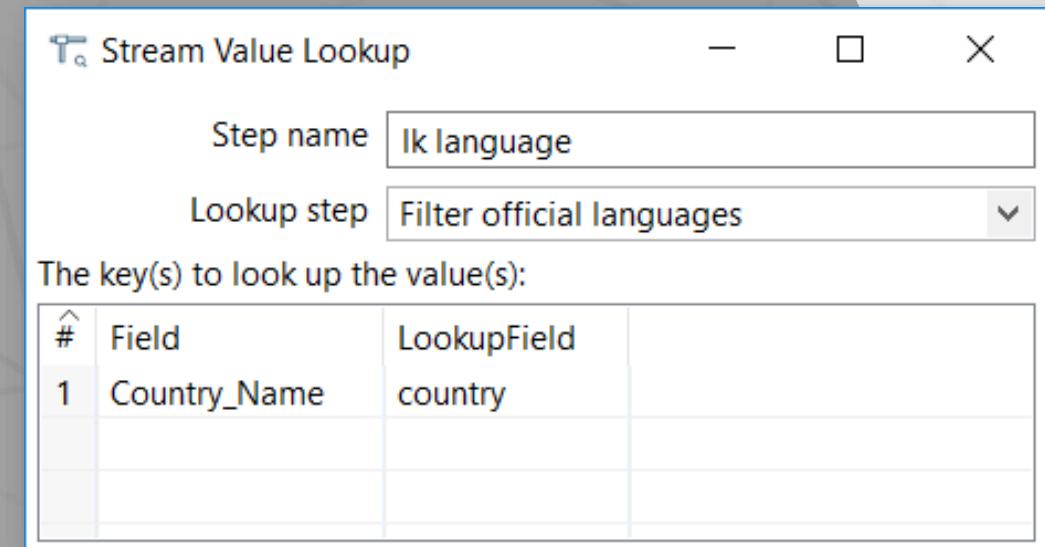
1. La categoría de pasos **Lookup** y arrastra el lienzo El paso **Stream lookup**.
2. Cree un salto desde el paso **Text file input** que acaba de crear, al paso **Stream lookup**.
3. Cree otro salto desde el paso **Filter rows** hasta el paso **Stream lookup**. Cuando se le pregunte por el tipo de salto, elija **Main output of step**. Hasta ahora, tienes lo siguiente:



# Buscando datos con un paso de búsqueda de Stream

## Pasos

4. Editar el paso **Stream lookup** haciendo doble clic en él.
5. En la lista desplegable **Lookup step**, seleccione **Filter official languages**. Esta selección es para indicar a PDI cuál de las transmisiones entrantes es la que se utiliza para buscar.
6. Ahora vamos a llenar la cuadrícula superior en la ventana de configuración. Esta cuadrícula le permite especificar los nombres de los campos que se utilizan para buscar. En la columna de la izquierda, **Field**, indica el campo de su ruta principal: **Country Name**. En la columna derecha, **LookupField**, indica el campo de la transmisión secundaria: idioma. La cuadrícula configurada se ve como sigue:



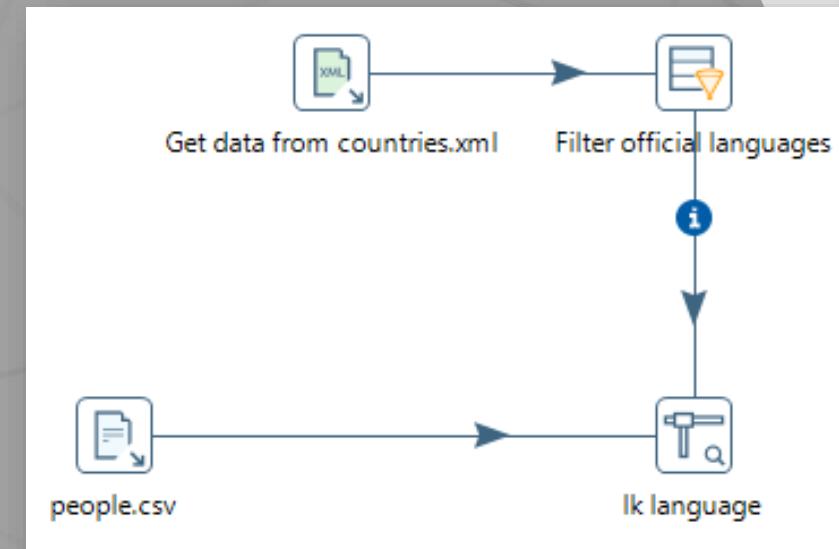
## Nota

Puede completar la primera columna en la cuadrícula superior usando el botón Obtener campos y eliminando todos los campos que no desea usar.

# Buscando datos con un paso de búsqueda de Stream

## Pasos

7. En la cuadrícula inferior, especifique los nombres de los campos que desea recuperar como resultado de la búsqueda. Esos campos provendrán de la corriente secundaria. En este caso, queremos el idioma del campo. Llene la cuadrícula como se muestra
8. Haga clic en **OK**.
9. El salto que va desde el paso **Filter rows** al paso de **Stream lookup** cambia su apariencia. El ícono que aparece sobre el salto muestra que esta es la ruta en la que buscará el paso de **Stream lookup**, como se muestra en la siguiente captura de pantalla:



# Buscando datos con un paso de búsqueda de Stream

Spoon - looking\_for\_data

File Edit View Action Tools Help

View Design

Search

Transformations

- looking\_for\_data
  - Run configurations
  - Database connections
  - Steps
  - Hops
  - Partition schemas
  - Slave server
  - Kettle cluster schemas
  - Data Services
  - Hadoop clusters

splitting\_streams\_ba filtering\_words\_with partitioning nested\_condition distributing\_rows looking\_for\_data

100% Connect

Ik language

Get data from countries.xml Filter official languages

people.csv

Examine preview data

Rows of step: Ik language (14 rows)

#	ID	Country_Name	Name	language
1	1	Russia	Mikhail Davydova	<null>
2	2	Russia	Anastasia Davydova	<null>
3	3	Spain	Carmen Rodriguez	Spanish
4	4	Spain	Francisco Delgado	Spanish
5	5	Japan	Natsuki Harada	Japanese
6	6	Japan	Emiko Suzuki	Japanese
7	7	China	Lin Jiang	Chinese
8	8	China	Wei Chiu	Chinese
9	9	United States	Chelsea Thompson	English
10	10	United States	Cassandra Sullivan	English
11	11	Canada	Mackenzie Martin	French
12	12	Canada	Nathan Gauthier	French
13	13	Italy	Giovanni Lombardi	Italian
14	14	Italy	Federica Lombardi	Italian

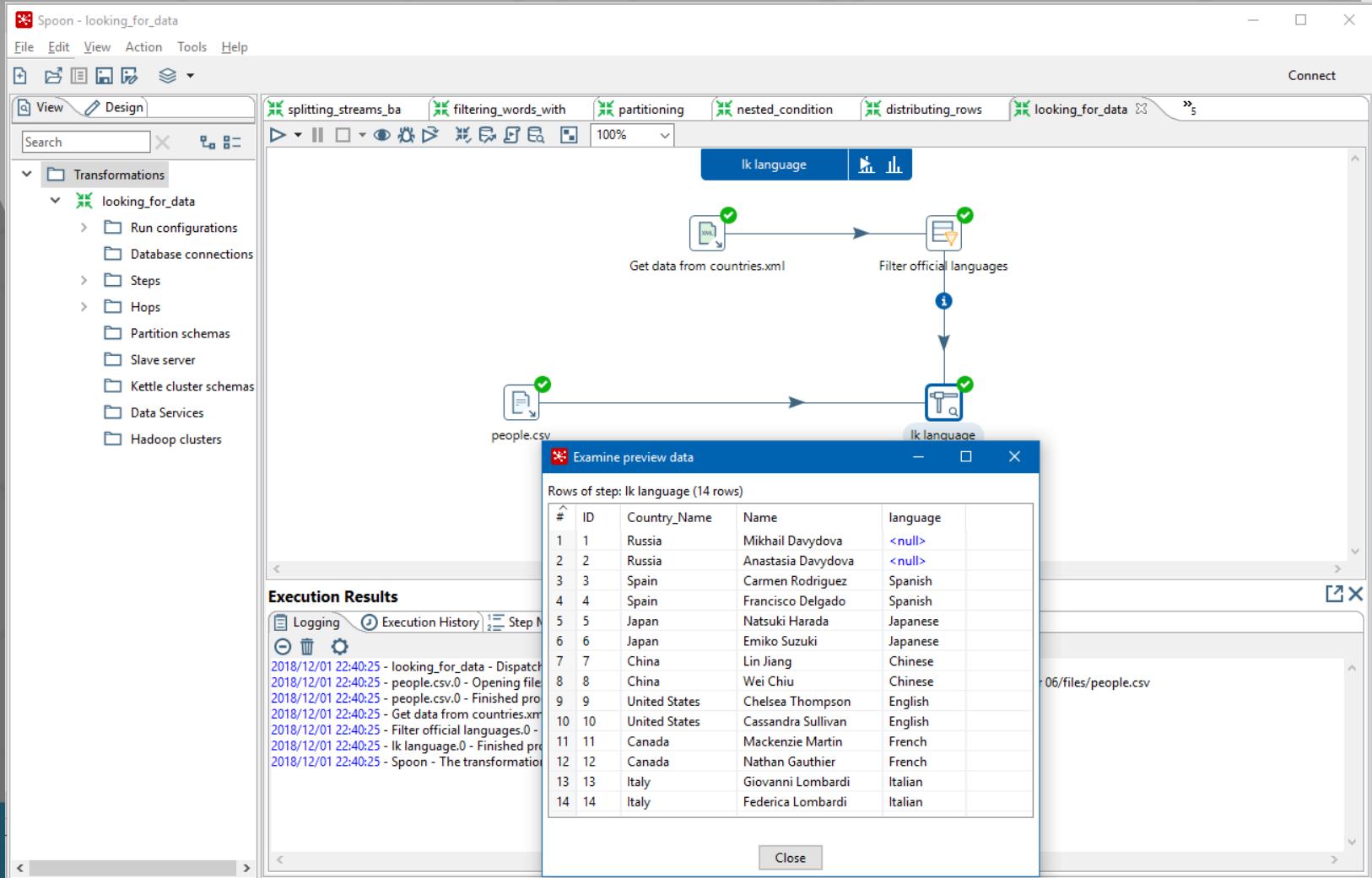
Execution Results

Logging Execution History Step N

2018/12/01 22:40:25 - looking\_for\_data - Dispatch  
 2018/12/01 22:40:25 - people.csv - Opening file  
 2018/12/01 22:40:25 - people.csv - Finished pro  
 2018/12/01 22:40:25 - Get data from countries.xml  
 2018/12/01 22:40:25 - Filter official languages - 0 -  
 2018/12/01 22:40:25 - Ik language - 0 - Finished pro  
 2018/12/01 22:40:25 - Spoon - The transformation

06/files/people.csv

Close



# Limpieza de datos

Los datos del mundo real no siempre son tan perfectos como nos gustaría. Por un lado, hay casos en que los errores en los datos son tan críticos que la única solución es informarlos o incluso abortar un proceso.

Sin embargo, hay un tipo diferente de problema con los datos: problemas menores que pueden solucionarse de alguna manera, como en los siguientes ejemplos:

# Estandarizando la información

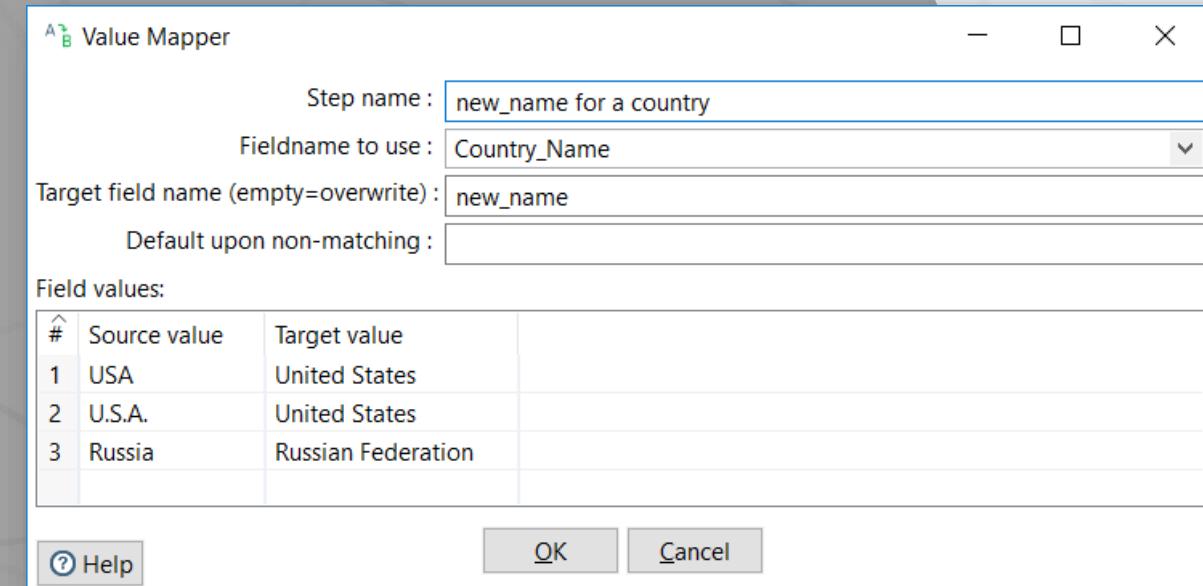
En la sección Buscando datos en el Capítulo, Control del flujo de datos, faltaban países en el archivo countries.xml. De hecho, los países estaban allí, pero con nombres diferentes. Por ejemplo, Rusia en nuestro archivo es Federación de Rusia en el archivo XML. Lo que deberíamos hacer para mejorar la solución es estandarizar la información, manteniendo un nombre único para los países.

Modifique la transformación que busca el idioma de la siguiente manera:

# Estandarizando la información

## Nota

1. Usando el ejemplo de Control del flujo de datos.
2. Elimine el salto que vincula su ruta principal con el paso **Lookup Stream**.
3. Después de leer sus datos, agregue un paso **Value Mapper** y utilícelo para obtener el nombre estándar para los países. Solo los países para los que sepan diferentes notaciones estarán aquí. Mira el siguiente ejemplo:

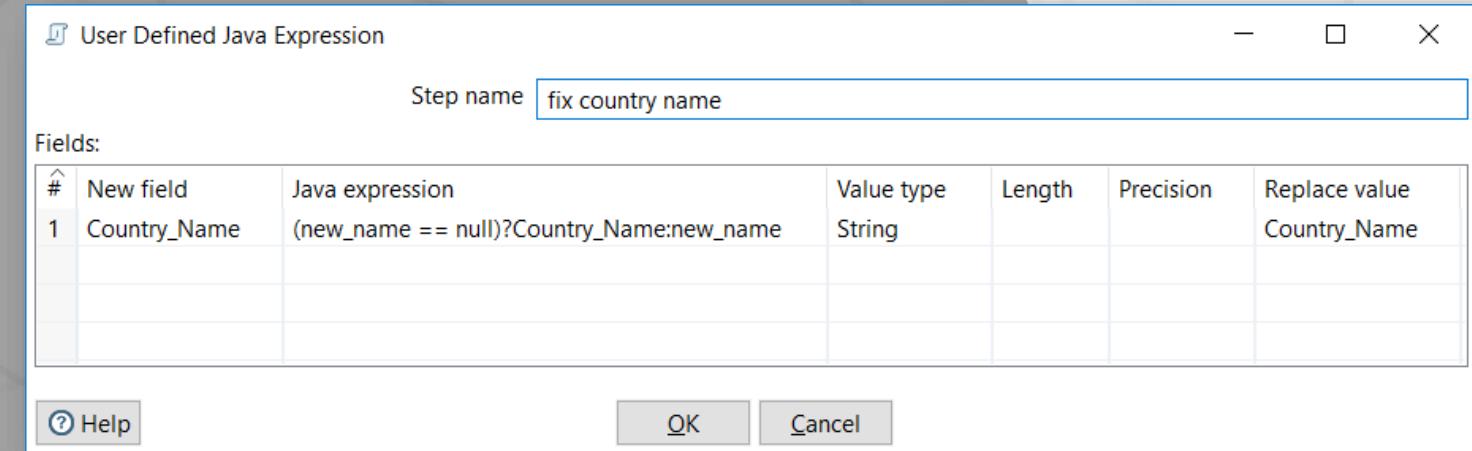


# Estandarizando la información

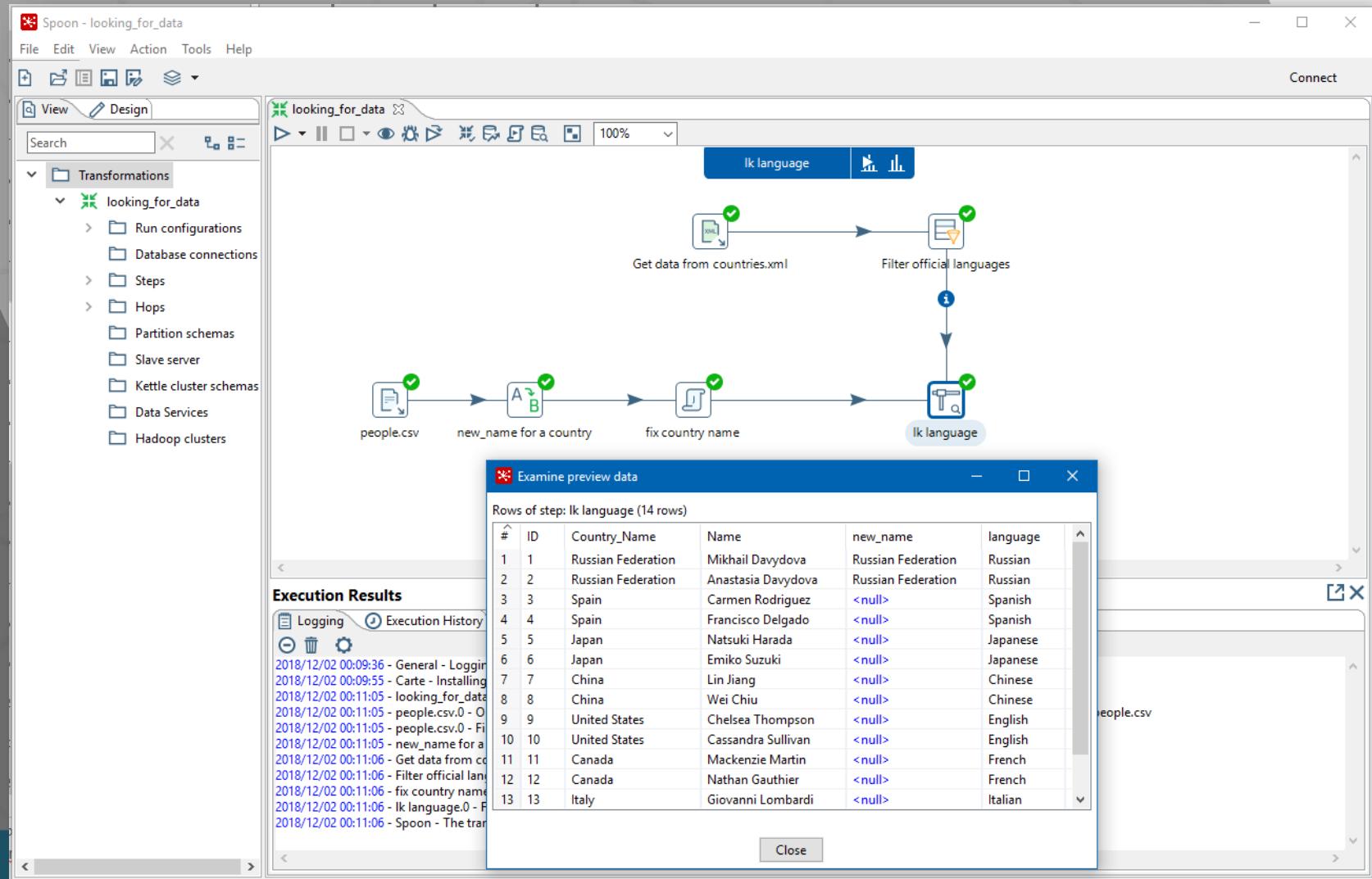
## Nota

4. Con un paso **User Defined Java Expression** (UDJE), sobrescriba los nombres de países que no tienen el valor estándar. En nuestro ejemplo, el valor de Rusia será reemplazado por la Federación Rusa. Así es como configuras el paso:  

5. Cree un salto desde este último paso hacia el paso **Lookup Stream**.
6. Ejecutar una vista previa de este paso. Ahora verá que tiene una búsqueda exitosa de todas las filas.



# Estandarizando la información



# Mejora de la calidad de los datos.

En la sección Filtrado de datos en el Capítulo **Control del flujo de datos**, identificó las palabras encontradas en un archivo de texto. En esa ocasión, ya hizo una limpieza al eliminar del texto todos los caracteres que no formaban parte de palabras legales, por ejemplo, paréntesis, guiones, etc. Recuerde que usó el paso **Replace in String** para esto.

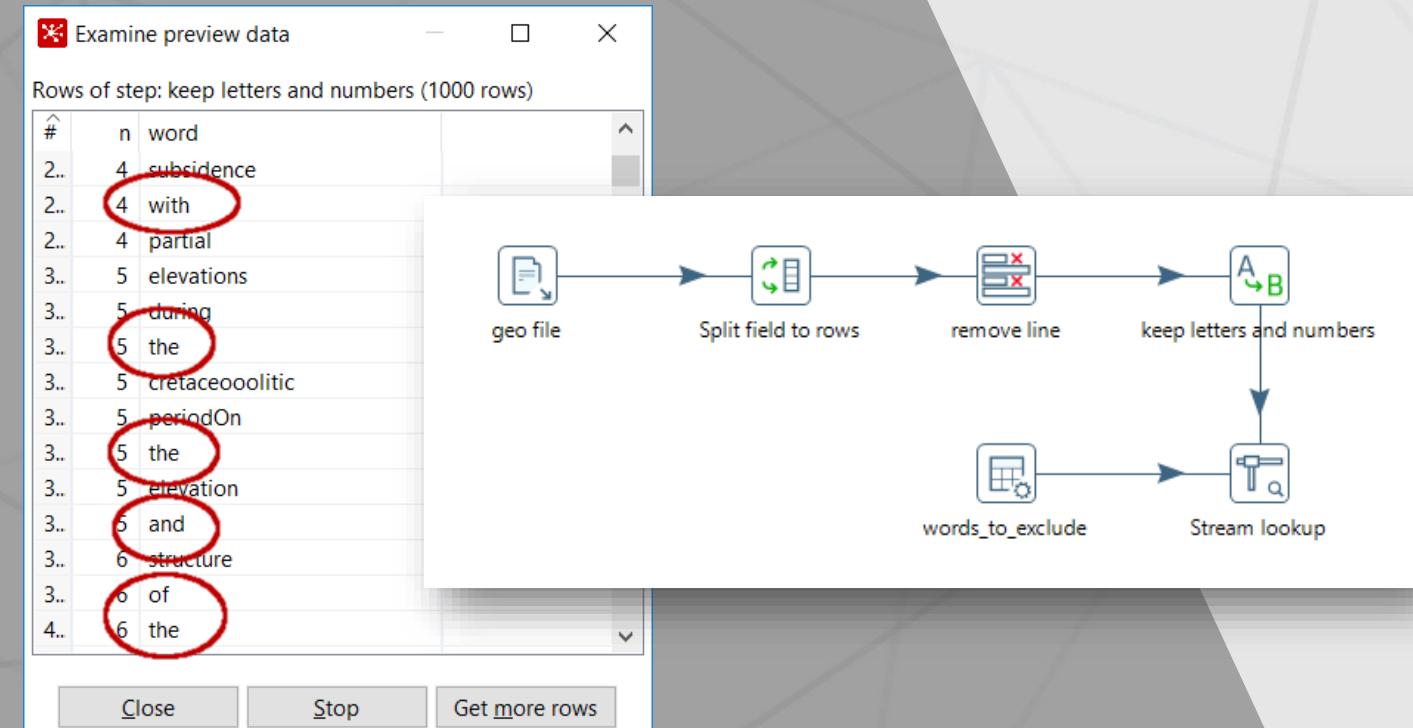
Hay más limpieza que podemos hacer en este texto. Por ejemplo, si su intención es calcular algunas estadísticas con palabras relacionadas con la geología, es posible que prefiera descartar muchas palabras que sean válidas en el idioma inglés pero que sean inútiles para su trabajo. Veamos una manera de deshacerse de estos:

# Mejora de la calidad de los datos.

## Nota

1. Abra la Transformación del Capítulo, **Control del flujo de datos**.
2. Elimine todos los pasos después del paso **Replace in String**.
3. Ejecutar una vista previa de este último paso. Verá palabras que no le interesan, por ejemplo, las resaltadas en la siguiente captura de pantalla:
4. Cree una nueva secuencia de datos que contenga las palabras que desea excluir. Deberías tener una palabra por fila. Algunas palabras candidatas son:

a, and, as, at, by, from, it, in, of, on, that, the, this, to, which, with, is, are, have, and been



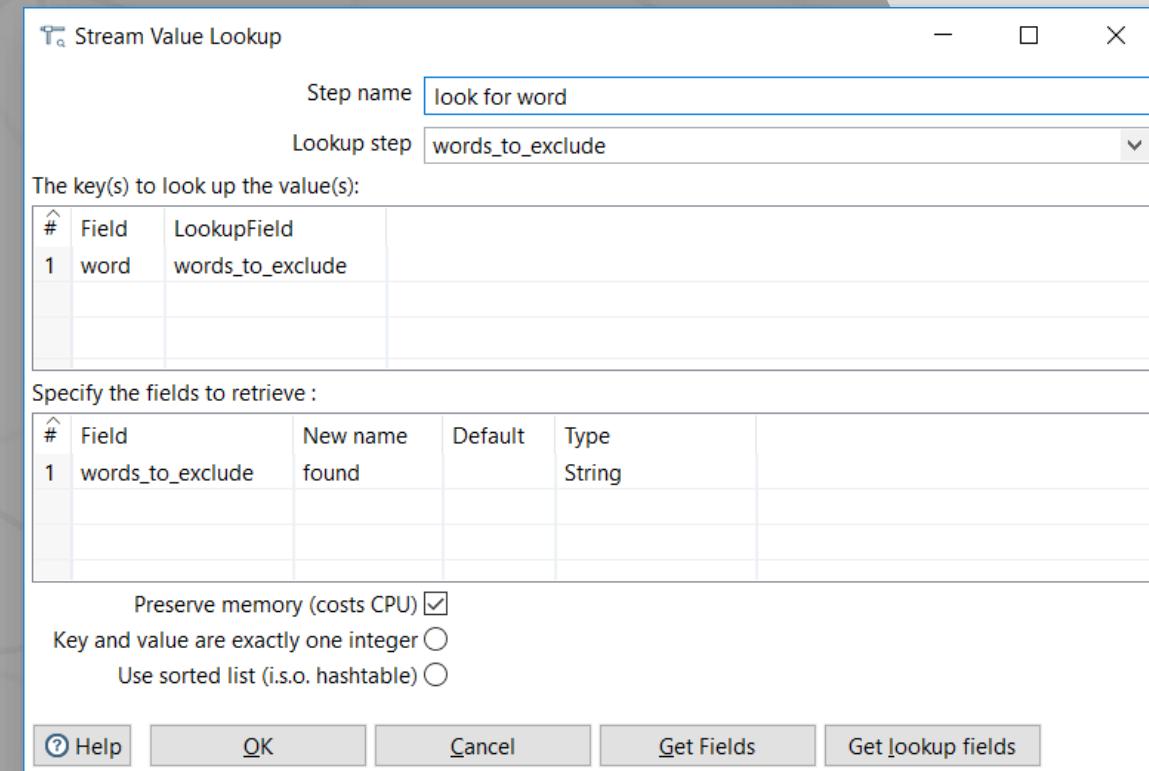
## Nota

Puede crear la lista de palabras en un **Data Grid**, o puede leer la lista desde un archivo simple.

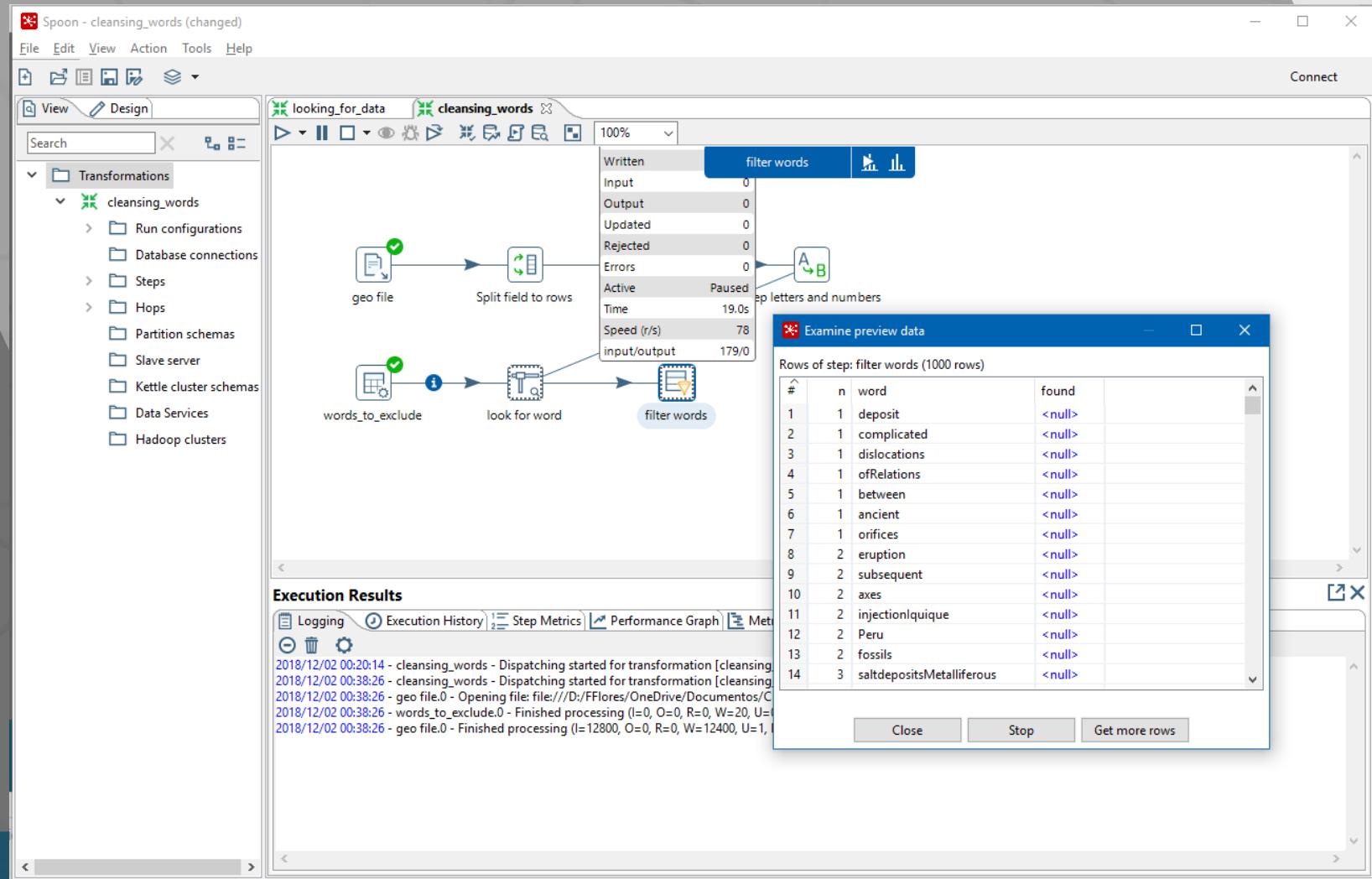
# Mejora de la calidad de los datos.

## Nota

6. Configure **Stream Value Lookup** de la siguiente manera
7. Después de este paso, agregue un paso **Filter rows** y utilícelo para descartar las palabras comunes, es decir, las palabras no encontradas. Como filtro, use **found\_word IS NULL**.
8. Ejecute una vista previa del paso **Filter rows**. Verás que las palabras no deseadas han sido descartadas.



# Mejora de la calidad de los datos.



# Introducción de pasos PDI útiles para la limpieza de datos

La limpieza de datos, también conocida como **data cleaning** o **data scrubbing**, se puede hacer de forma manual o automática, dependiendo de la complejidad de la limpieza. Al conocer de antemano las reglas que se aplican, puede realizar la limpieza automática utilizando cualquier paso de PDI que más le convenga.

Los siguientes son algunos pasos particularmente útiles, incluidos los que usamos en los ejemplos anteriores:

Paso	Propósito
If field value is null	Si un campo es nulo, cambia su valor a una constante. Se puede aplicar a todos los campos del mismo tipo de datos, por ejemplo, a todos los campos de enteros o a campos particulares.
Null if...	Establece un valor de campo en nulo si es igual a un valor constante dado.
Number range	Crea rangos basados en un campo numérico. Un ejemplo de su uso es convertir números flotantes a una escala discreta, como 0, 0.25, 0.50, etc.
Value Mapper	Asigna los valores de un campo de un valor a otro. Por ejemplo, puede usar este paso para convertir los valores yes/no, true/false, or 1/0 a una notación única como Y / N.
Replace in string	Reemplaza todas las apariciones de una cadena dentro de un campo con una cadena diferente. Permite el uso de expresiones regulares como se explica en el Capítulo Manipulación de datos y metadatos de PDI.
String operations	Útil para recortar y eliminar caracteres especiales y más.
Calculator	Le permite eliminar caracteres especiales, convertir a mayúsculas y minúsculas, y recuperar solo dígitos de una cadena, entre otras operaciones.
Stream lookup	Busca valores que vienen de otra corriente. En la limpieza de datos, puede usarlo para establecer un valor predeterminado si su campo no está en una lista determinada.
Database lookup	Lo mismo que Stream Value Lookup, pero busca en una tabla de base de datos.
Unique rows	Elimina filas consecutivas dobles y deja solo ocurrencias únicas.

# Tratar con coincidencias no exactas

La limpieza de datos, también conocida como **data cleaning** o **data scrubbing**, se puede hacer de forma manual o automática, dependiendo de la complejidad de la limpieza. Al conocer de antemano las reglas que se aplican, puede realizar la limpieza automática utilizando cualquier paso de PDI que más le convenga.

Los siguientes son algunos pasos particularmente útiles, incluidos los que usamos en los ejemplos anteriores:

# Limpiar haciendo una búsqueda difusa.

Con una búsqueda difusa, no buscamos coincidencias exactas sino valores similares. PDI le permite realizar búsquedas difusas con el paso especial Fuzzy match. Con este paso, puede encontrar coincidencias aproximadas a una cadena usando algoritmos de coincidencia.

Para ver cómo usar este paso, volvamos a nuestro ejemplo. Supongamos que tenemos una lista de estados válidos junto con sus códigos, de la siguiente manera:

```
State;Abbreviation
Alabama;AL
Alaska;AK
Arizona;AZ
...
West Virginia;WV
Wisconsin;WI
Wyoming;WY
```

# Limpiar haciendo una búsqueda difusa.

Por otro lado, tenemos un flujo de datos y, entre los campos, un campo que representa a los estados. El problema es que no todos los valores son correctos.

Lo siguiente podría ser una lista de valores entrantes:



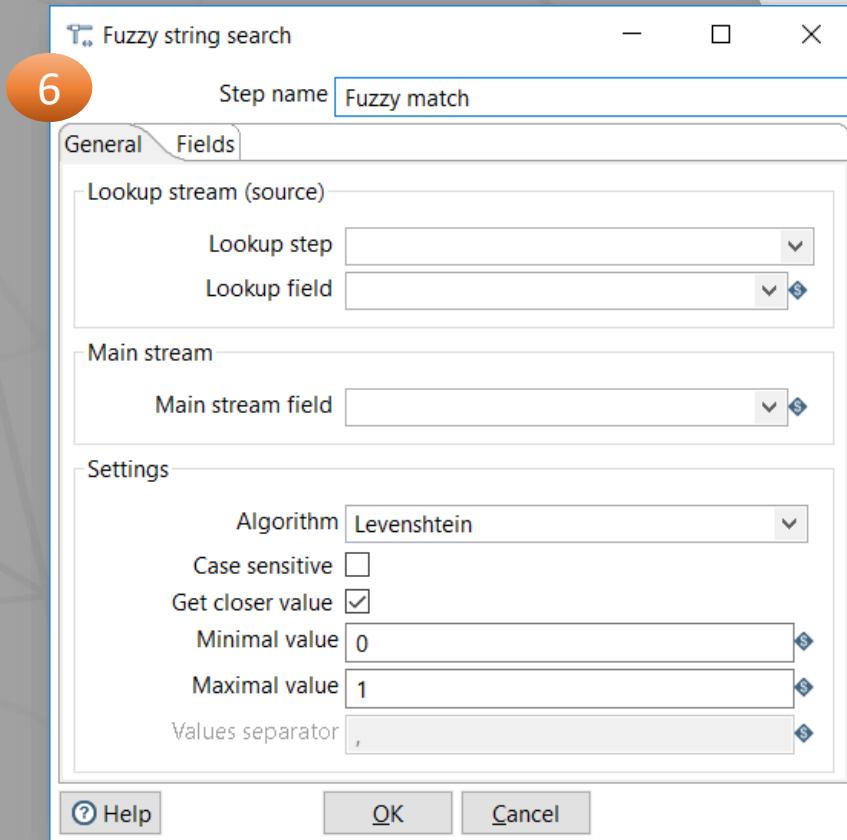
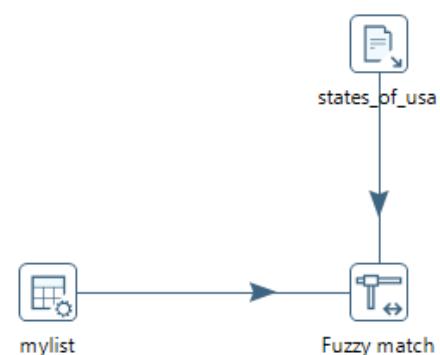
A list of state names enclosed in a white box with a thin black border. The states listed are: California, Colorado, Washington, Massachusetts, Alaska, Connecticut, Road Island, Hawaii, Ohio, and Kentucky. The first four states have a blue vertical bar to their left, while the others do not.

- California
- Colorado
- Washington
- Masachusetts
- Alsaka
- Conneticut
- Road Island
- Hawai
- Ohio
- Kentuky

# Limpiar haciendo una búsqueda difusa.

## Pasos

1. Crear una transformación.
2. Cree un flujo de datos con la lista de estados en los Estados Unidos.
3. Use un **Data Grid** para crear una lista de valores adecuados e incorrectos para los estados. Puede escribir los que se enumeraron anteriormente o crear su propia lista.
4. Desde la categoría **Lookup**, arrastre un paso **Fuzzy match** al área de trabajo.
5. Enlace los pasos como se muestra:
6. Ahora configuraremos el paso, cuyas ventanas de configuración se ven de la siguiente manera:



# Limpiar haciendo una búsqueda difusa.

## Pasos

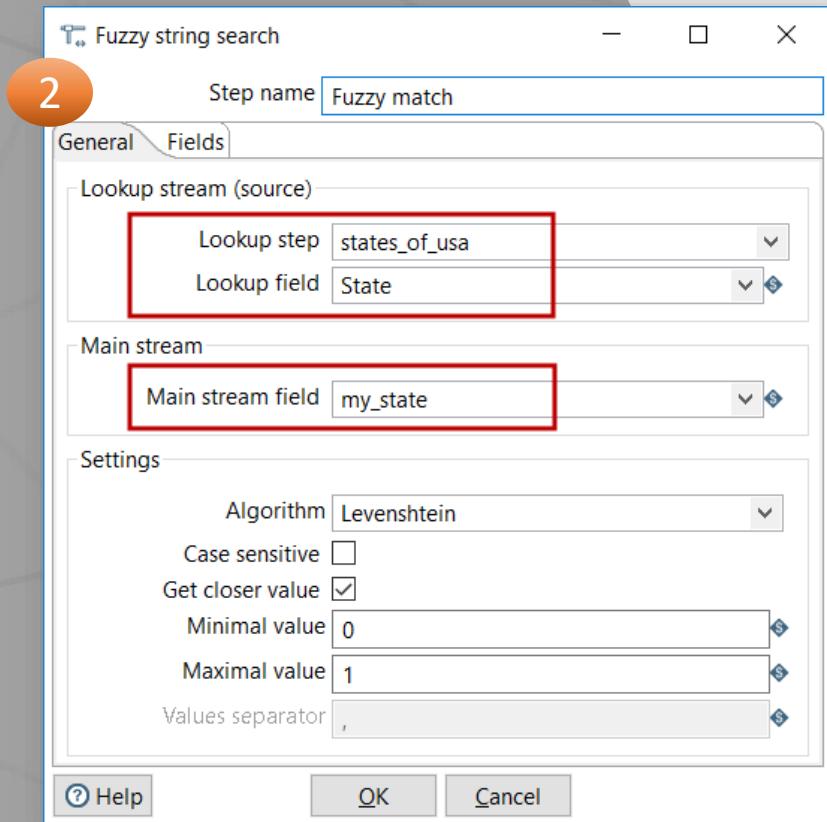
1. Haga doble clic en el paso Fuzzy match.
  2. Rellene la ventana de configuración de la siguiente manera:
  3. Configuración de un paso de coincidencia difusa
  4. Cerrar la ventana.
  5. Con este paso seleccionado, ejecute una vista previa.
- Deberías ver esto:

**Examine preview data**

Rows of step: Fuzzy match (17 rows)

#	my_state	match	distance
1	Califronia	<null>	<null>
2	Calorodo	Colorado	1
3	Washington	Washington	0
4	Masachusetts	Massachusetts	1
5	Alsaka	<null>	<null>
6	Conneticut	Connecticut	1
7	Road Island	<null>	<null>
8	Hawai	Hawaii	1
Ohio	Ohio	0	
Kentuky	Kentucky	1	
.. Pensylvania	Pennsylvania	1	
.. Louisiana	Louisiana	0	
.. Arizonia	Arizona	1	
.. Hawaii	Hawaii	0	
.. Mississipi	Mississippi	1	
.. California	California	0	
.. Howaii	Hawaii	1	

**Close**



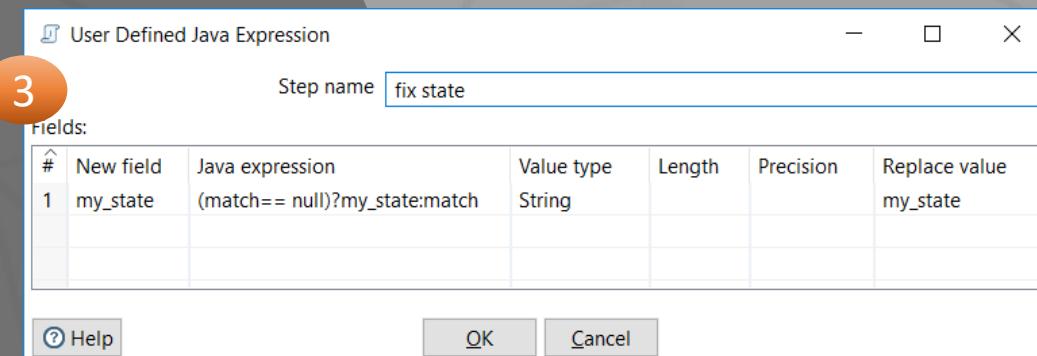
# Deduplicación de coincidencias no exactas

Además, supongamos que tenemos los mismos datos que antes y queremos crear una lista de los estados que aparecen en nuestro conjunto de datos. Entre los valores, tenemos Hawai, Hawai y Howaui. No queremos los tres valores en nuestra lista final. Sólo queremos un estado único: Hawai. Si intentamos deduplicar los datos con el paso de **Unique rows**, seguiremos teniendo tres valores. La única solución es intentar corregir los valores con un algoritmo de búsqueda difusa, y solo después de eso hacer la deduplicación. Esto no difiere mucho de la solución anterior:

# Deduplicación de coincidencias no exactas

## Pasos

1. Abra la transformación que acaba de crear y guárdela con un nombre diferente.
2. Ejecutar una vista previa del paso **Fuzzy match**. En la ventana de vista previa, haga clic en el título de la columna de coincidencia para ordenar las filas por ese campo. Verás que hay valores duplicados
3. Después del paso **Fuzzy match**, agregue un paso UDJE y configúrelo para mantener el estado correcto:



2

Examine preview data

Rows of step: Fuzzy match (17 rows)

my_state	match	distance
1 California	<null>	<null>
2 Alsaka	<null>	<null>
3 Road Island	<null>	<null>
4 Arizona	Arizona	1
5 California	California	0
6 Calorado	Colorado	1
7 Conneticut	Connecticut	1
8 Hawai	Hawaii	1
9 Hawaii	Hawaii	0
1.. Howaii	Hawaii	1
1.. Kentucky	Kentucky	1
1.. Louisiana	Louisiana	0
1.. Masachusetts	Massachusetts	1
1.. Mississipi	Mississippi	1
1.. Ohio	Ohio	0
1.. Pensylvania	Pennsylvania	1
1.. Washington	Washington	0

Close

## Nota

La función de clasificación que acaba de aplicar solo tiene efecto en la ventana de vista previa. Las filas en su conjunto de datos permanecen en el orden original.

# Deduplicación de coincidencias no exactas

## Pasos

4. Use el paso Seleccionar valores para mantener solo la columna con el estado.
5. Desde la categoría Transformar, seleccione y arrastre un paso Ordenar filas y cree un salto desde el paso Seleccionar valores hacia este.
6. Haga doble clic en el paso Ordenar filas. En la cuadrícula, agregue una fila y seleccione el único campo disponible: my\_state. Además, marque la opción ¿Solo pasar filas únicas ?.
7. Con el paso Ordenar filas seleccionado, ejecute una vista previa. Verá la siguiente lista, donde solo hay una aparición de Hawaii y está correctamente escrita. Además, el estado de Hawaii, que también era un valor duplicado, aparece una vez:



#	my_state
1	Alaska
2	Arizona
3	California
4	Colorado
5	Connecticut
6	Hawaii
7	Kentucky
8	Louisiana
9	Massachusetts
1..	Mississippi
1..	Ohio
1..	Pennsylvania
1..	Road Island
1..	Washington

**Close**

## Nota

Deduplicación de coincidencias no exactas. En este ejercicio, eliminamos filas duplicadas con el paso Ordenar filas. Si los datos estuvieran ordenados por estado, podríamos haber usado un paso de filas Únicas.

# Validating and reporting errors to the log

Las siguientes son líneas de muestra de nuestro archivo:

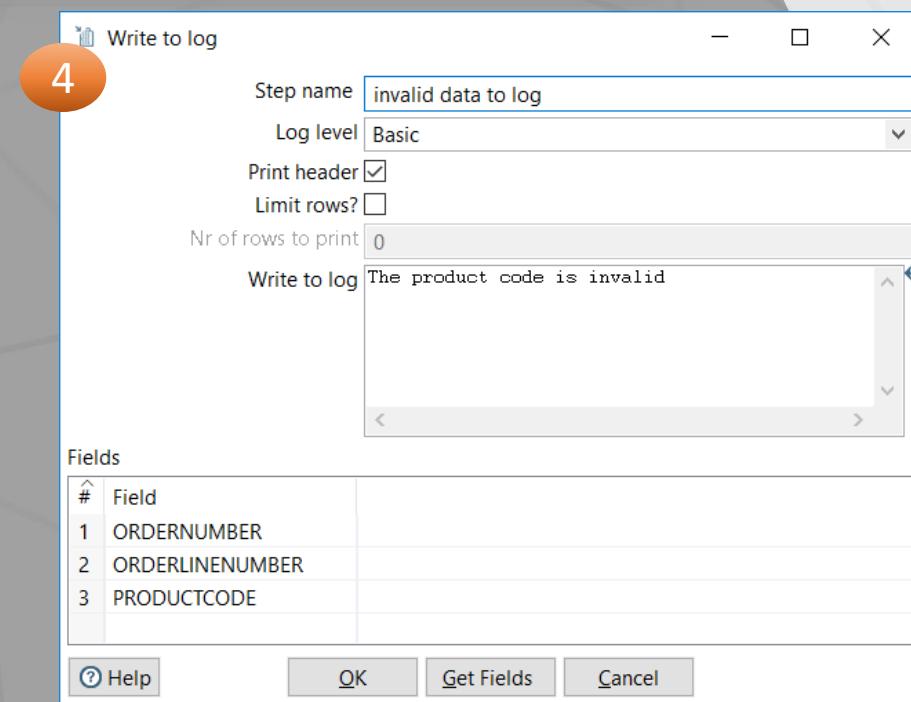
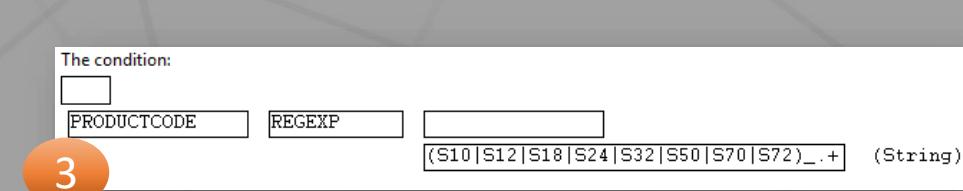
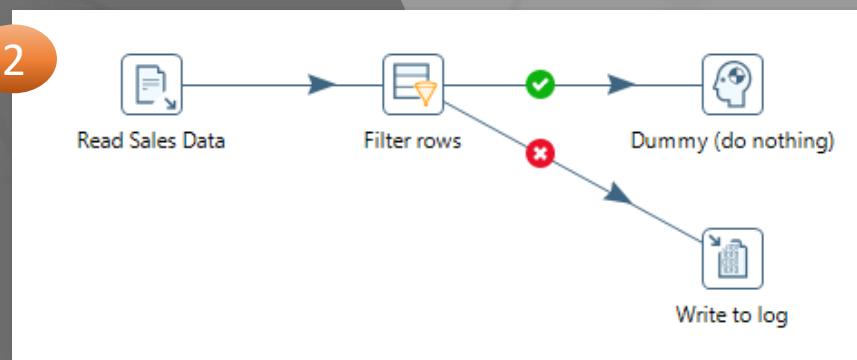
```
ORDERDATE,ORDERNUMBER,ORDERLINENUMBER,PRODUCTCODE,PRODUCTLINE,QUANTITYORDERED,PRICEEACH,SALES  
2/20/2004 0:00 ,10223,10,S24_4278 ,Planes ,23,74.62,1716.26  
11/21/2004 0:00,10337,3,S18_4027 ,Classic Cars ,36,100 ,5679.36  
6/16/2003 0:00 ,10131,2,S700_4002,Planes ,26,85.13,2213.38  
7/6/2004 0:00 ,10266,5,S18_1984 ,Classic Cars ,49,100 ,6203.4  
10/16/2004 0:00,10310,4,S24_2972 ,Classic Cars ,33,41.91,1383.03  
12/4/2004 0:00 ,10353,4,S700_2834,Planes ,48,68.8 ,3302.4  
1/20/2005 0:00 ,10370,8,S12_1666 ,Trucks and Buses,49,100 ,8470.14
```

Entre las columnas, hay un código de producto compuesto de dos partes: un prefijo y un número. Esperamos que el prefijo sea uno de los siguientes: S10, S12, S18, S24, S32, S50, S70 o S72. Implementaremos esta validación. Si el valor no pertenece a esta lista, la fila se informará como un error y la fila de datos se descartará:

# Validating and reporting errors to the log

## Pasos

1. Cree una nueva transformación y lea el archivo sales\_data.csv.
2. Después del paso que lee el archivo, agregue un paso **Filter rows**, un Paso **Dummy** y un paso **Write to log**. Enlace los pasos de la siguiente manera:
3. Haga doble clic en el paso **Filter rows** y configúrelo con la siguiente condición:
4. Haga doble clic en el paso Escribir en el registro y configúrelo de la siguiente manera:



# Validating and reporting errors to the log

## Pasos

5. Cerrar la ventana.
6. Ejecutar la transformación. Verá que todas las filas para las que el código del producto no es válido se informan al registro, como en las siguientes líneas de ejemplo del registro:

```
... - invalid data to log.0 - -----> LineNr 1-----  
... - invalid data to log.0 - The product code is invalid  
... - invalid data to log.0 -  
... - invalid data to log.0 - ORDERNUMBER = 10131  
... - invalid data to log.0 - ORDERLINENUMBER = 2  
... - invalid data to log.0 - PRODUCTCODE = S700_4002  
... - invalid data to log.0 -  
... - invalid data to log.0 - ======  
... - invalid data to log.0 -
```

# Introducción de validaciones comunes y su implementación con PDI.

La validación en el ejemplo anterior fue muy simple. Dependiendo del tipo de restricción, es posible que necesitemos un solo paso de PDI como en este caso, o una combinación de pasos. Por ejemplo, si queremos verificar que la longitud de un valor de cadena sea menor que 50, primero debemos calcular la longitud y luego compararla con el límite de 50 caracteres.

La siguiente tabla describe los tipos más comunes de restricciones y los pasos de PDI que podemos usar para evaluar las reglas esperadas:

# Introducción de validaciones comunes y su implementación con PDI.

Restricción	Implementación
Valor debe tener un tipo de datos dado tales como String o Date	Usted puede leer la campo como <b>String</b> y usar el <b>Select Values</b> a convertir a la Tipo de datos esperado . Implementar el manejo de errores. a identificar datos que no se puede convertir .
El valor debe tener una longitud dada	Utilice cualquiera de los pasos conocidos para calcular la longitud más un paso de filtro, como las <b>Filter rows</b> o el <b>Java Filter</b> .
El valor no puede ser nulo	Utilice <b>Filter rows</b> (función IS NULL) o un paso <b>Java Filter</b>
Los números o fechas deben estar dentro de un rango esperado	Utilice <b>Filter rows</b> (>, <o = funciones) o un paso de <b>Java Filter</b>
Los valores deben pertenecer a una lista discreta conocida	Use <b>Filter rows</b> (IN LIST, o funciones REGEXP), un paso <b>Java Filter</b> o un paso <b>RegExp Evaluation</b> , más un paso de filtro.
Los valores deben pertenecer a la lista que se encuentra en una fuente externa, como un archivo o una base de datos	Use un paso <b>Stream lookup</b> o un paso <b>Database lookup</b> dependiendo del tipo de fuente, luego un filtro para determinar si se encontró el valor o no.
Un campo o una combinación de campos debe ser único en todo el conjunto de datos	Use <b>Group by</b> o <b>Memory Group by</b> para contar las ocurrencias, luego use un filtro para determinar si hay duplicados.
Los valores deben adherirse a un patrón o máscara requerido	Para los campos genéricos, use las filas de filtros (LIKE, CONTAINS, STARTS WITH, ENDS WITH o REGEXP), un paso <b>Java Filter</b> o un paso <b>RegExp Evaluation</b> , más un paso de filtro. Para el tipo específico de datos, como números de tarjetas de crédito o direcciones de correo electrónico, puede utilizar el <b>Credit card validator</b> o el <b>Mail validator</b> , respectivamente.

# Tratamiento de datos no válidos mediante la división y la fusión de flujos

Cuando está transformando datos, no es infrecuente que detecte inexactitudes o errores. A veces, los problemas que encuentre pueden no ser lo suficientemente graves como para descartar las filas. Tal vez pueda adivinar qué datos se supone que deben estar allí en lugar de los valores actuales, o puede suceder que tenga valores predeterminados para los valores no válidos. Veamos algunos ejemplos:

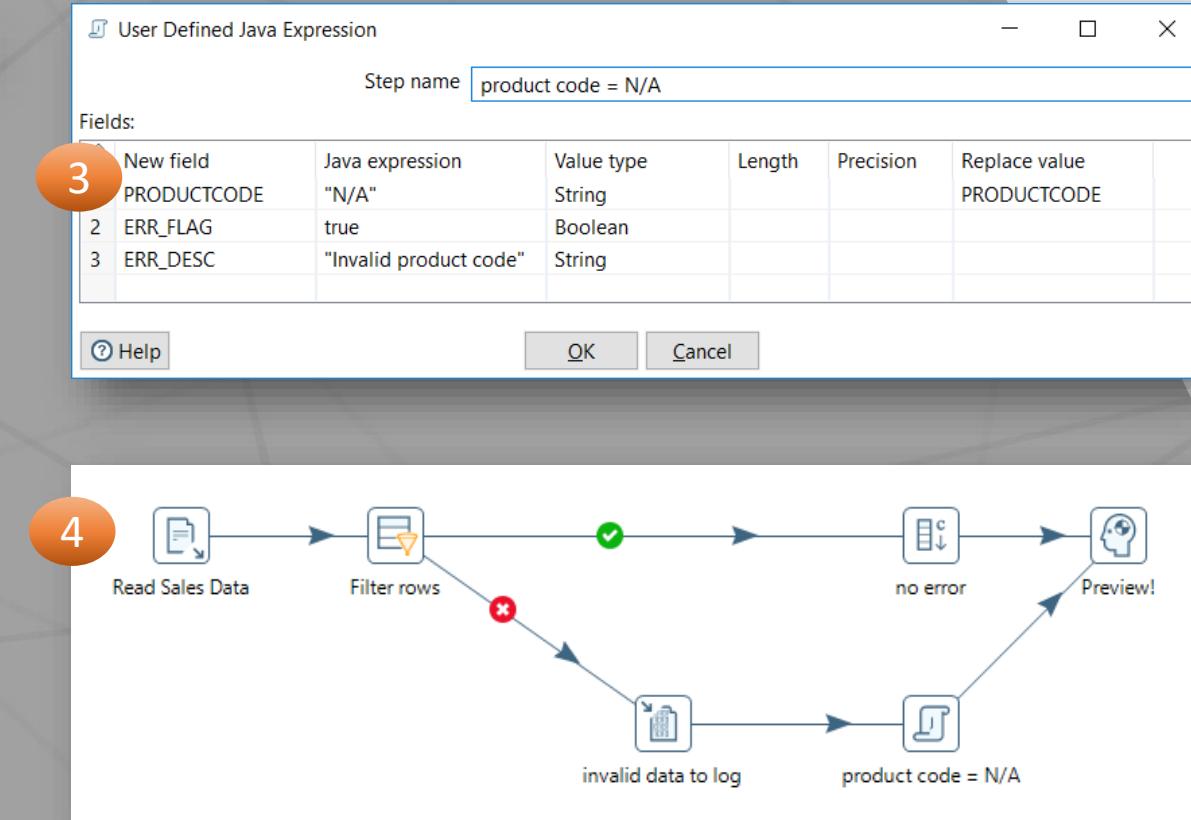
# Arreglando datos que no coinciden con las reglas

Al comienzo de la sección de **Validating Data** en este capítulo, aprendió a validar un campo descartando las filas con valores no válidos. Ahora aprenderás cómo evitar descartar la fila. Para solucionar el problema, proponga un código de producto igual a <invalid>. Después de hacerlo, enviará las filas con valores no válidos a la secuencia principal:

# Arreglando datos que no coinciden con las reglas

## Pasos

1. Abra la transformación que creó en la sección **Validating Data** y guárdela con un nombre diferente.
2. Después del paso **Write to log**, agregue un paso UDJE. Utilice el paso para reemplazar el código de producto no válido con el texto <invalid> y también para agregar dos cadenas, una bandera para indicar que hay un error y un nuevo campo llamado ERR\_DESC con la descripción del problema:
3. Como desea combinar las filas incorrectas con las buenas, tambien debe agregar los campos en la otra secuencia. Así como la secuencia real del paso **Filter rows**, agregue un paso **Add constant** para agregar los mismos campos, pero con valores diferentes: ERR\_FLAG = falso y ERR\_DESC vacío.
4. Utilice un paso Dummy para unirse a los flujos, de la siguiente manera:
5. Ejecutar una vista previa en este último paso. Verá todas las filas, las filas con los códigos de producto correctos y las filas con los valores fijos:



# Arreglando datos que no coinciden con las reglas

## Pasos

- Ejecutar una vista previa en este último paso. Verá todas las filas, las filas con los códigos de producto correctos y las filas con los valores fijos:

Examine preview data

Rows of step: Preview! (200 rows)

#	ORDERDATE	ORDERN...	ORDE...	PRODUCTCODE	PRODUCTLINE	QUA...	PRICEEACH	SALES	ERR_FLAG	ERR_DESC
1..	08/21/2004		10284	1 S18_2581	Planes	31	71.81	2226.11	N	
1..	05/09/2005		10415	3 S72_1253	Planes	42	57.61	2419.62	N	
1..	02/17/2005		10381	2 S18_1097	Trucks and Buses	48	98	4704	N	
1..	03/19/2004		10231	2 S12_1108	Classic Cars	42	100	8378.58	N	
1..	09/09/2004		10293	6 S18_3259	Trains	22	100	2418.24	N	
1..	06/16/2003		10131	2 N/A	Planes	26	85.13	2213.38	Y	Invalid product code
1..	12/04/2004		10353	4 N/A	Planes	48	68.8	3302.4	Y	Invalid product code
1..	03/30/2005		10398	7 N/A	Ships	36	100	3910.32	Y	Invalid product code
1..	10/06/2003		10155	3 N/A	Planes	44	85.87	3778.28	Y	Invalid product code
1..	06/15/2004		10258	1 N/A	Classic Cars	45	80.92	3641.4	Y	Invalid product code
1..	12/17/2004		10361	11 N/A	Planes	35	100	4277.35	Y	Invalid product code
1..	12/04/2004		10353	9 N/A	Planes	39	100	5043.87	Y	Invalid product code

[Close](#)

# Manipulación de datos por codificación

Independientemente de la transformación que necesite hacer con sus datos, tiene muchas posibilidades de encontrar pasos de integración de datos de Pentaho (PDI) capaces de realizar el trabajo. A pesar de eso, puede ser que no haya pasos adecuados que cumplan con sus requisitos o que una Transformación aparentemente menor consuma muchos pasos vinculados en un arreglo muy confuso que sea difícil de probar o entender. Dejar caer iconos coloridos aquí y allá y hacer notas para aclarar una Transformación puede ser práctico hasta cierto punto, pero hay algunas situaciones como las que se describen aquí donde inevitablemente tendrá que codificar.

# Realizando tareas sencillas con el paso de JavaScript.

En las primeras versiones de PDI, la codificación en JavaScript era la única forma en que los usuarios tenían que realizar muchas tareas. En las últimas versiones, hay muchas otras formas de realizar estas tareas, pero JavaScript sigue siendo una opción. Existe el paso de JavaScript que le permite insertar código en una transformación de PDI.

# Usando el lenguaje JavaScript en PDI

JavaScript es un lenguaje de script utilizado principalmente en el desarrollo de sitios web. Sin embargo, dentro de PDI, usted usa solo el lenguaje central: no ejecuta un navegador web y no le importa el HTML. Hay muchos motores de JavaScript disponibles. PDI utiliza el motor Rhino de Mozilla. Rhino es una implementación de código abierto del lenguaje JavaScript principal; No contiene objetos o métodos relacionados con la manipulación de páginas web.

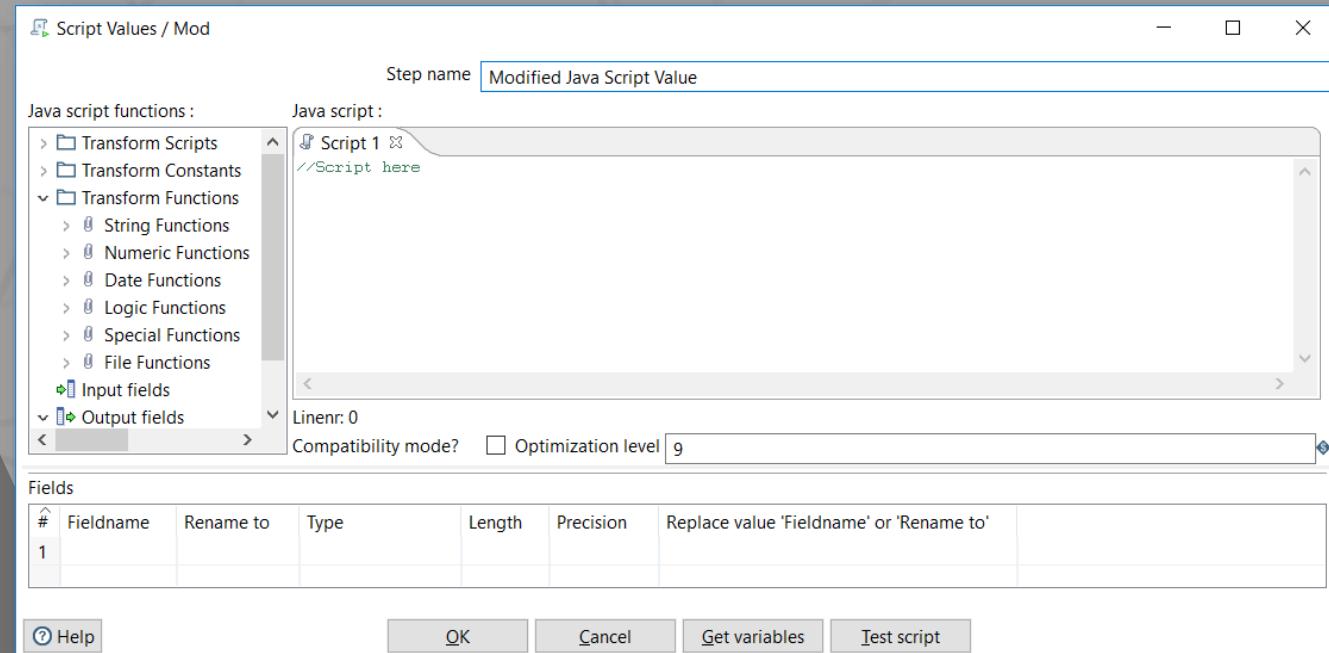
## Nota

Si está interesado en saber más acerca de Rhino, siga este enlace:  
[https://developer.mozilla.org/en/Rhino\\_Overview](https://developer.mozilla.org/en/Rhino_Overview)



# Insertando código JavaScript usando el paso JavaScript

El Modified JavaScript Value (paso de JavaScript para abreviar) le permite insertar código JavaScript dentro de su Transformación. El código que escribe en el área de script principal se ejecuta una vez por fila que llega al paso. Vamos a explorar su ventana de diálogo:



# Insertando código JavaScript usando el paso JavaScript

## Ventana de diálogo de JavaScript

La mayor parte de la ventana está ocupada por el área de edición. Es allí donde se escribe el código JavaScript utilizando la sintaxis estándar del idioma y las funciones y campos del árbol en el lado izquierdo de la ventana.

La rama Transformar funciones del árbol contiene una lista rica de funciones que están listas para usar. Las funciones están agrupadas por categoría:

Las categorías String, Numeric, Date y Logic contienen las funciones habituales de JavaScript.

### Nota

Esta no es una lista completa de funciones de JavaScript. Se le permite usar funciones de JavaScript incluso si no están en esta lista.



# Insertando código JavaScript usando el paso JavaScript

- La categoría **Special** contiene una mezcla de funciones de utilidad. La mayoría de ellos no son funciones de JavaScript sino funciones de PDI. Una de estas funciones es `writeToLog ()`, muy útil para mostrar datos en el registro PDI.
- Finalmente, la categoría **File**, como su nombre lo indica, contiene una lista de funciones que realizan una verificación simple o acciones relacionadas con archivos y carpetas, por ejemplo, `fileExist ()` o `createFolder ()`.

Para agregar una función a su secuencia de comandos, simplemente haga doble clic en ella o arrástrela a la ubicación de la secuencia de comandos donde desee usarla o simplemente escríbala.

## Nota

Si no está seguro de cómo usar una función en particular o qué hace una función, simplemente haga clic derecho en la función y seleccione **Sample**. Aparece una nueva ventana de secuencia de comandos con una descripción de la función y el código de muestra que muestra cómo usarla.



# Insertando código JavaScript usando el paso JavaScript

La rama **Input fields** contiene la lista de los campos procedentes de los pasos anteriores. Para ver y usar el valor de un campo para la fila actual, haga doble clic en él o arrástrelo al área de código. También puedes escribirlo a mano.

Cuando utiliza uno de los campos de entrada en el código, se trata como una variable de JavaScript. Como tal, el nombre del campo debe seguir las convenciones para un nombre de variable, por ejemplo, no puede contener puntos o comenzar con símbolos que no sean caracteres. Como el PDI es bastante permisivo con los nombres, puede tener campos en su transmisión cuyos nombres no son válidos para su uso dentro del código JavaScript.

## Nota

Si no está seguro de cómo usar una función en particular o qué hace una función, simplemente haga clic derecho en la función y seleccione **Sample**. Aparece una nueva ventana de secuencia de comandos con una descripción de la función y el código de muestra que muestra cómo usarla.



## Nota

Si pretende usar un campo con un nombre que no sigue las reglas de nombre, cámbiele el nombre justo antes del paso de JavaScript con el paso **Select Values**. Si usa ese campo sin cambiar su nombre, no se le advertirá cuando codifique, pero obtendrá un error o resultados inesperados cuando ejecute la transformación.



# Insertando código JavaScript usando el paso JavaScript

Finalmente, los **Output fields** son una lista de los campos que dejarán el paso.

En las siguientes subsecciones, volverá a crear una Transformación del Capítulo Control del flujo de datos, y reemplazar parte de su funcionalidad con JavaScript. Así que antes de continuar, realice lo siguiente:

## Pasos

1. Abra la Transformación que lee un archivo y filtra las palabras del Control del flujo de datos.
2. Seleccione los dos primeros pasos, Text file input y Split field to rows, y cópielos en una nueva transformación.
3. Desde la categoría de pasos **Scripting**, seleccione y arrastre un paso de **Modified Java Script Value** al área de trabajo. Crea un salto desde **Split field to rows step toward this** hacia este.

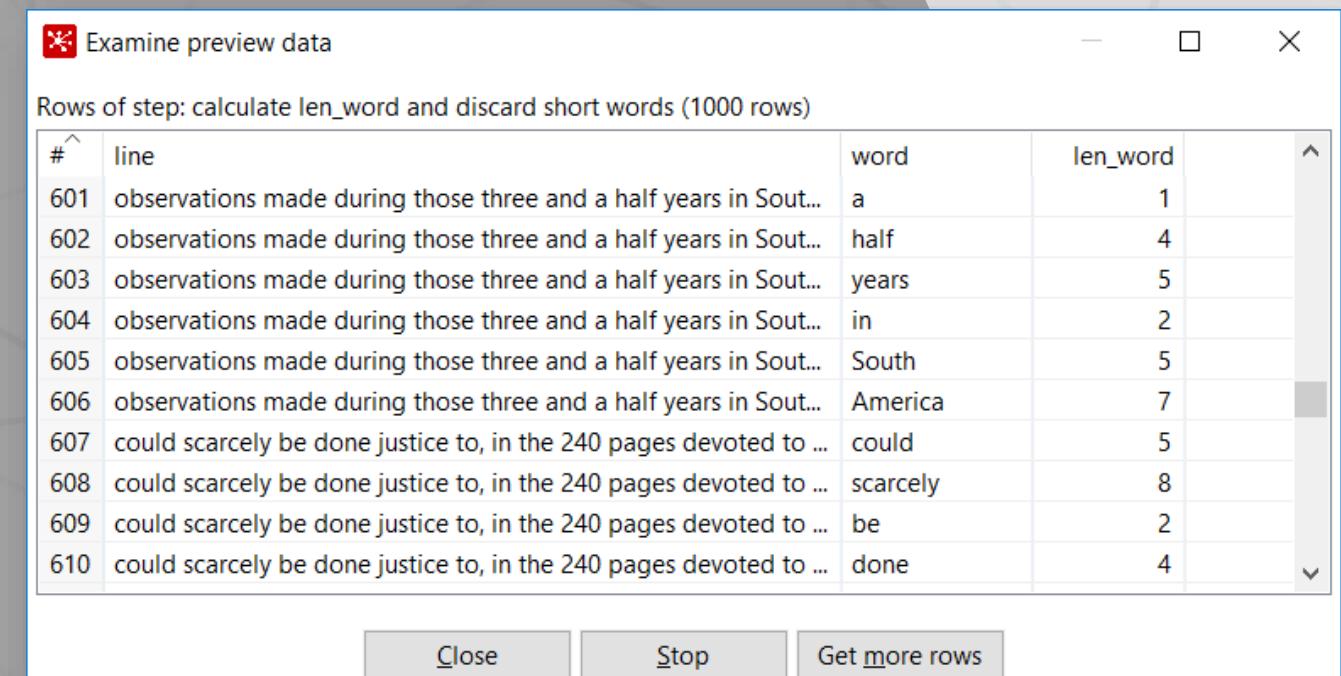
# Añadiendo campos

La tarea más sencilla que puedes hacer con JavaScript agrega un campo. Veamos cómo crear un campo simple que contenga la longitud del campo de word:

## Pasos

1. Haga doble clic en el paso **Modified Java Script Value**: paso de JavaScript de ahora en adelante y debajo del texto // Script aquí, escriba lo siguiente:  

```
var len_word = word.length;
```
2. Haga clic en el botón **Get variables**. La cuadrícula inferior se llenará con la variable definida, **len\_word**.
3. Cierre la ventana de configuración y guarde la transformación.
4. Asegúrate de que el paso de JavaScript esté seleccionado y haz una vista previa. Deberías ver lo siguiente:



#	line	word	len_word
601	observations made during those three and a half years in Sout...	a	1
602	observations made during those three and a half years in Sout...	half	4
603	observations made during those three and a half years in Sout...	years	5
604	observations made during those three and a half years in Sout...	in	2
605	observations made during those three and a half years in Sout...	South	5
606	observations made during those three and a half years in Sout...	America	7
607	could scarcely be done justice to, in the 240 pages devoted to ...	could	5
608	could scarcely be done justice to, in the 240 pages devoted to ...	scarcely	8
609	could scarcely be done justice to, in the 240 pages devoted to ...	be	2
610	could scarcely be done justice to, in the 240 pages devoted to ...	done	4

**Examine preview data**

Rows of step: calculate len\_word and discard short words (1000 rows)

**Note**

Tenga en cuenta que ve las primeras 1000 filas. Si desea ver más, simplemente haga clic en Obtener más filas.

# Modificando campos

Con el paso de JavaScript, también puede modificar un campo existente. Esto no difiere mucho de la forma en que agrega nuevos campos. En este ejercicio, modificaremos el campo de palabra convirtiéndolo en mayúsculas:

## Pasos

1. Haga doble clic en el paso de JavaScript y después del código, escribió en el ejercicio anterior, escriba lo siguiente: `var u_word = upper(word);`
2. En la cuadrícula inferior, agregue la nueva variable `u_word`, que sustituirá el campo de `word` de la siguiente manera
3. Cierre la ventana y ejecute una vista previa. Verás esto

2

Fields						
#	Fieldname	Rename to	Type	Length	Precision	Replace value 'Fieldname' or 'Rename to'
1	len_word		Integer		0	N
2	u_word	word	String			Y

3

Examine preview data

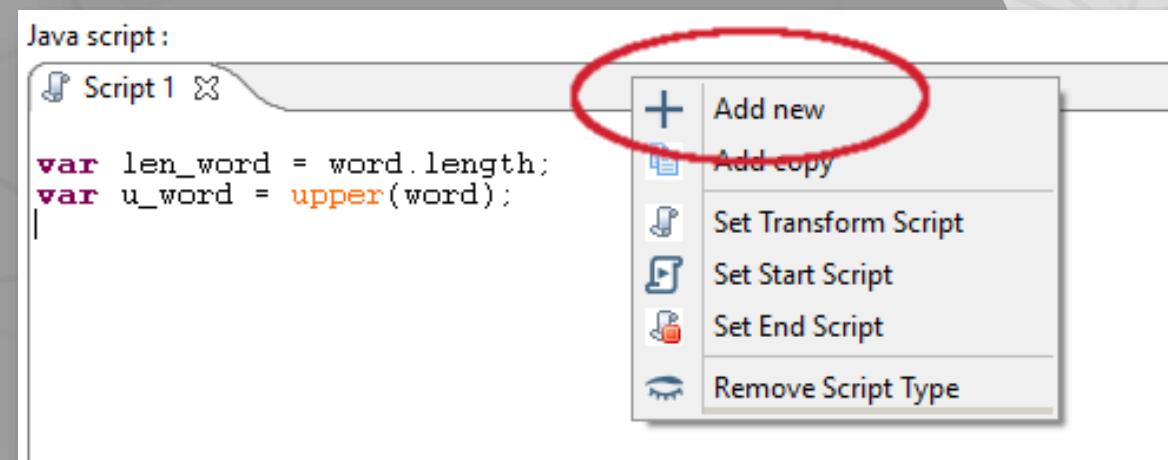
Rows of step: calculate len\_word and discard short words (1000 rows)

#	line	word	len_word
601	observations made during those three and a half years in South America	A	1
602	observations made during those three and a half years in South America	HALF	4
603	observations made during those three and a half years in South America	YEARS	5
604	observations made during those three and a half years in South America	IN	2
605	observations made during those three and a half years in South America	SOUTH	5
606	observations made during those three and a half years in South America	AMERICA	7
607	could scarcely be done justice to, in the 240 pages devoted to their	COULD	5
608	could scarcely be done justice to, in the 240 pages devoted to their	SCARCELY	8
609	could scarcely be done justice to, in the 240 pages devoted to their	BE	2
610	could scarcely be done justice to, in the 240 pages devoted to their	DONE	4

[Close](#) [Stop](#) [Get more rows](#)

# Organizando tu código

Como se dijo, el código que escribe en el área de script se ejecuta para cada fila entrante. Si sucede que necesita inicializar los valores que se aplican a todas las filas, puede y debe hacerlo en un script separado. Si no lo hace, todo el código de inicialización se ejecutará para cada fila en su conjunto de datos. Así es como creas un nuevo script de inicio:



## Pasos

1. Haga clic derecho en la parte superior del área de código y seleccione Add new:

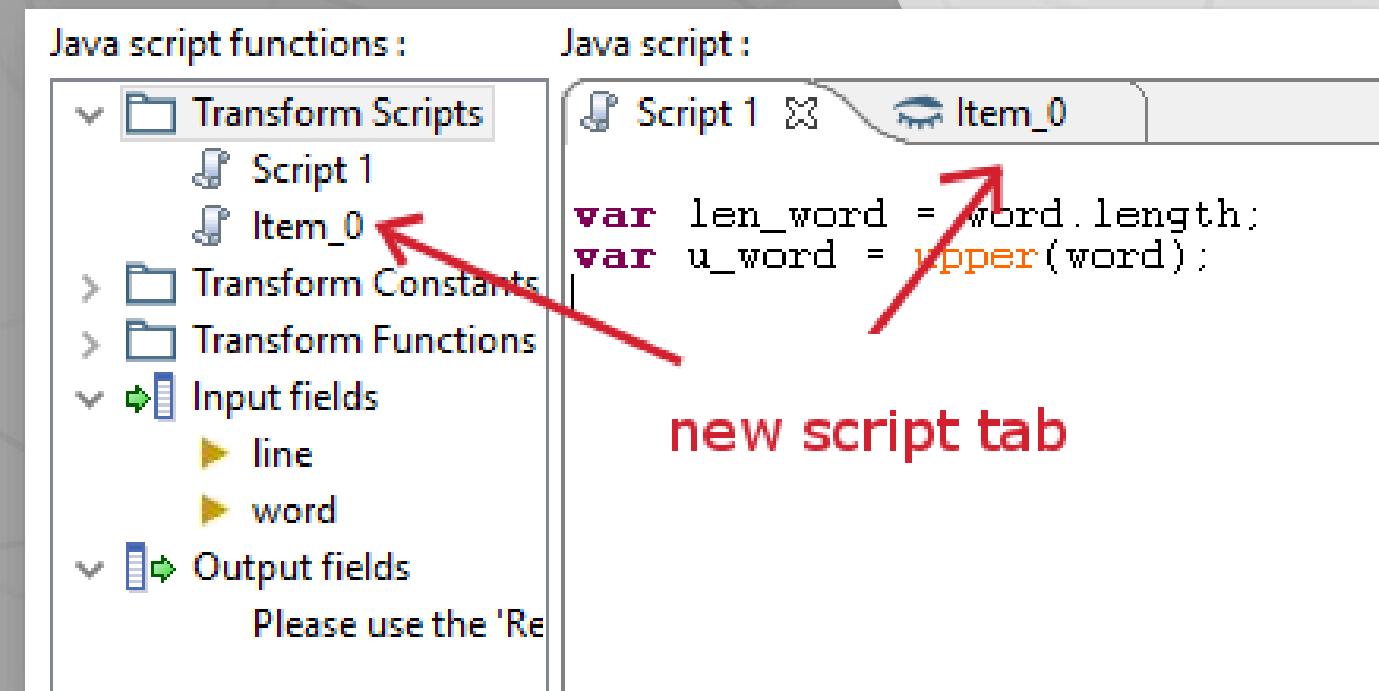
# Organizando tu código

## Pasos

2. Aparecerá una nueva pestaña Script en el área principal y en la lista de scripts a la izquierda:
3. En el árbol Transformar secuencias de comandos, haga clic con el botón derecho en el nombre de la nueva secuencia de comandos y seleccione Cambiar nombre para darle un nombre significativo.
4. Haga clic con el botón derecho en la parte superior de la pestaña Script y seleccione Set Start Script. El icono al lado del título cambiará para que pueda reconocer este script como el de inicio. Esta acción hará que todos los scripts que se escribieron aquí se ejecuten antes de que alguna fila entre en el paso.

## Nota

Todas las variables dentro de JavaScript mantienen sus valores entre filas. En particular, todas las variables que defina en el script de inicio mantendrán sus valores.



# Organizando tu código

De la misma manera que creas un script de inicio, puedes crear un script de finalización para que se ejecute después de que la última fila abandone el paso. En este caso, selecciona Establecer guión final en su lugar.

Además de estos scripts en particular, puede agregar más pestañas para organizar su código. Por ejemplo, puede agregar un script con definiciones de funciones. Luego, en la pestaña Script principal, simplemente llame a las funciones. Así es como lo haces:

## Pasos

1. Haga clic derecho en la parte superior del área de código y seleccione **Add new**. Por defecto, el script no tiene asociado un tipo de script. Déjalo así.
2. Asigne un nombre significativo a la pestaña Script y escriba todo el código que necesita.
3. En la pestaña **Start Scripting**, llame a `LoadScriptFromTab(<your script>);` función. Esto cargará el código de la pestaña creada recientemente.



## Nota

Tenga en cuenta que si carga un script en el script principal, el código se cargará en cada fila de procesamiento.



# Controlando el flujo usando constantes predefinidas

JavaScript tiene una forma de controlar el flujo de datos, es decir, decidir qué filas siguen el flujo normal y cuáles se descartan. Actúa como un filtro. Para controlar el flujo, juegas con una variable especial llamada `trans_Status`. Esta variable se evalúa para cada fila en su conjunto de datos y, dependiendo del valor, el resultado es diferente, como se muestra en la siguiente tabla:

Si el valor <code>trans_Status</code> se establece en ...	La fila actual ...
<code>SKIP_TRANSFORMATION</code>	se elimina del conjunto de datos.
<code>CONTINUE_TRANSFORMATION</code>	se mantiene. No le pasa nada.
<code>ERROR_TRANSFORMATION</code>	Causa el aborto de la transformación.

# Controlando el flujo usando constantes predefinidas

La forma en que establece los valores es tan simple como la siguiente:

```
trans_Status = SKIP_TRANSFORMATION
```

Los tres valores posibles son constantes predefinidas que puedes encontrar en el árbol en el lado izquierdo de la ventana de JavaScript en Transformation Constants. Puede escribir los valores a mano o puede hacer doble clic en ellos en el árbol de la izquierda.

Para demostrar el uso de estos conceptos, agregaremos más funcionalidad a nuestro ejemplo. Ahora mantendremos solo las palabras con longitud mayor a 3:

# Controlando el flujo usando constantes predefinidas

## Pasos

- Haga doble clic en el paso de JavaScript y, de acuerdo con su código, agregue lo siguiente:

```
if (len_word > 3)
    trans_Status =
CONTINUE_TRANSFORMATION;
else
    trans_Status = SKIP_TRANSFORMATION;
```

- Cierre la ventana y ejecute una vista previa. Verás esto:

Examine preview data

Rows of step: calculate len\_word and discard short words (1000 rows)

#	line	word	len_word
601	been written by him. But apart from those geological questions, which have	APART	5
602	been written by him. But apart from those geological questions, which have	FROM	4
603	been written by him. But apart from those geological questions, which have	THOSE	5
604	been written by him. But apart from those geological questions, which have	GEOLOGICAL	10
605	been written by him. But apart from those geological questions, which have	QUESTIONS,	10
606	been written by him. But apart from those geological questions, which have	WHICH	5
607	been written by him. But apart from those geological questions, which have	HAVE	4
608	an important bearing on biological thought and speculation, such as the	IMPORTANT	9
609	an important bearing on biological thought and speculation, such as the	BEARING	7
610	an important bearing on biological thought and speculation, such as the	BIOLOGICAL	10

# Vista previa de datos

Este fragmento de código está destinado a mantener solo las palabras cuya longitud es mayor que 3. Lo logra al establecer el valor de la variable PDI predefinida `trans_Status` a `CONTINUE_TRANSFORMATION` para las filas que desea mantener y a `SKIP_TRANSFORMATION` para las filas que desea descartar . Si presta atención a la última vista previa, observará que todas las palabras tienen al menos una longitud de tres caracteres.

Si ejecuta la Transformación, verá que hay una gran diferencia entre el número de filas que ingresan al paso de JavaScript y las filas que salen del paso. Esto se debe claramente al filtro que se ha aplicado.

# Probando el script usando el botón Test script

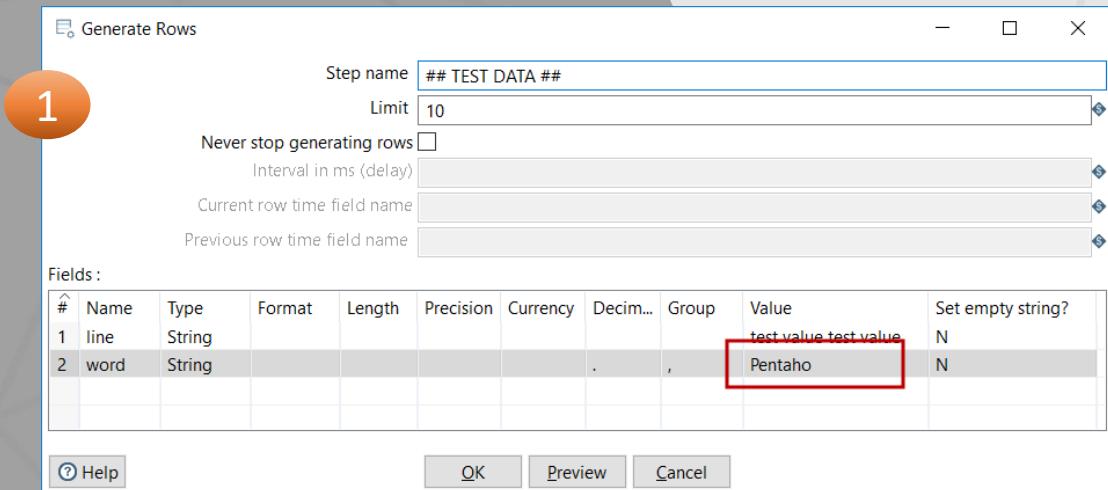
En el ejemplo que acabamos de construir, el código era muy simple y no requería mucha atención. Sin embargo, cuando tenga más código o quizás un algoritmo complicado para ejecutar en sus datos, probar el código de manera efectiva antes de ejecutarlo con sus datos podría ser una gran idea. El botón Test script le permite hacer esto: verificar si el script hace lo que se pretende que haga. En realidad, genera una Transformación en la parte posterior con dos pasos: un paso Generate Rows con datos de muestra y una copia del paso de JavaScript que funciona en esa muestra.

Así es como pruebas tu código JavaScript:

# Probando el script usando el botón Test script

## Pasos

1. Haga clic en el botón **Test script** aparecerá una ventana para crear un conjunto de filas para probar. Rellénelo como se muestra en la siguiente captura de pantalla:
2. Haga clic en **Preview** y aparecerá una ventana que muestra diez filas idénticas con los valores de muestra proporcionados.
3. Cierre la ventana **Preview** y haga clic en **OK** para probar el código. Aparece una ventana con el resultado de haber ejecutado el script en los datos de prueba:



3

Examine preview data

Rows of step: calculate len\_word and discard short words (10 rows)

#	line	word	len_word
1	test value test value	PENTAHO	7
2	test value test value	PENTAHO	7
3	test value test value	PENTAHO	7
4	test value test value	PENTAHO	7
5	test value test value	PENTAHO	7
6	test value test value	PENTAHO	7
7	test value test value	PENTAHO	7
8	test value test value	PENTAHO	7
9	test value test value	PENTAHO	7
10	test value test value	PENTAHO	7

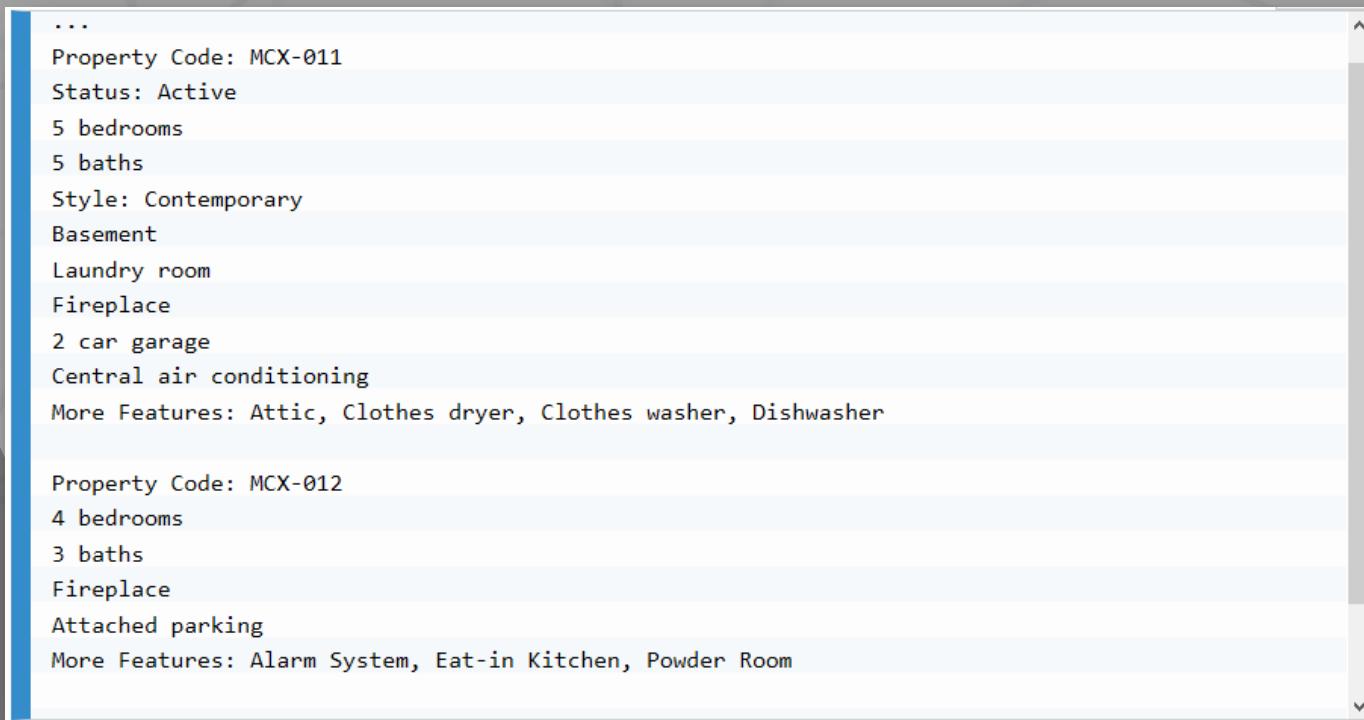
**Close** **Show Log**

# Analizar archivos no estructurados con JavaScript

Es ideal tener archivos de entrada donde la información esté bien formada, es decir, el número de columnas y el tipo de sus datos sean precisos, todas las filas sigan el mismo patrón, y así sucesivamente. Sin embargo, es muy común encontrar archivos de entrada donde la información tiene poca o ninguna estructura o la estructura no sigue la matriz (n filas por m columnas) que espera. Esta es una de las situaciones donde JavaScript puede ayudar.

# Analizar archivos no estructurados con JavaScript

Supongamos que tiene un archivo con una descripción de las casas, que se parece a lo siguiente:



The screenshot shows a terminal window with two sets of house descriptions. The first set, for property MCX-011, includes details like 5 bedrooms, 5 baths, and a 2-car garage. The second set, for property MCX-012, includes details like 4 bedrooms, 3 baths, and attached parking. Both sets also mention various features such as a fireplace, central air conditioning, and laundry rooms.

```
...
Property Code: MCX-011
Status: Active
5 bedrooms
5 baths
Style: Contemporary
Basement
Laundry room
Fireplace
2 car garage
Central air conditioning
More Features: Attic, Clothes dryer, Clothes washer, Dishwasher

Property Code: MCX-012
4 bedrooms
3 baths
Fireplace
Attached parking
More Features: Alarm System, Eat-in Kitchen, Powder Room
```

# Analizar archivos no estructurados con JavaScript

Desea comparar las propiedades entre ellas, pero sería más fácil si el archivo tuviera una estructura precisa. El paso de JavaScript puede ayudarte con esto.

El primer intento de estructurar los datos será agregar a cada fila el código de la casa a la que pertenece esa fila. El propósito es tener lo siguiente:

#	text	prop_code
9	Kitchen	MCX-001
10	Basement	MCX-001
11	Bathroom on main floor	MCX-001
12	2 car garage	MCX-001
13	Attached parking	MCX-001
14	More Features: Eat-In Kitchen Area, Kitchen Pantry, Deck, Fence	MCX-001
15	Property Code: MCX-002	MCX-002
16	5 bedrooms	MCX-002
17	5 baths	MCX-002
18	Style: Colonial	MCX-002

# Vista previa de algunos datos



## Pasos

1. Crea una nueva transformación.
2. Obtenga el archivo de muestra del sitio del libro y léalo con un paso **Text file input step**. Desmarque la casilla de verificación Header y cree un único campo denominado **text**.
3. Ejecutar una vista previa. Debería ver el contenido del archivo en una sola columna llamada **text**.
4. Después del paso de entrada, agregue un paso de JavaScript y haga doble clic en él para editarlo.
5. En el área de edición, escriba el siguiente código JavaScript para crear un campo con el código de la propiedad:

```
var prop_code;
posCod = indexOf(text, 'Property Code:');
if (posCod>=0)
    prop_code = trim(substr(text, posCod+15));
```

6. Haga clic en **Get variables** para agregar la variable **prop\_code** a la cuadrícula debajo del código. La variable contendrá para cada fila, el código de la casa a la que pertenece.
7. Haga clic en **OK** y, con el paso de JavaScript seleccionado, ejecute una vista previa. Debería ver los datos transformados como se esperaba.

## Nota

La función `indexOf` identifica la columna donde se encuentra el código de propiedad en el texto. La función `substr` corta el `Property Code : text`, manteniendo solo el código en sí.

# Vista previa de algunos datos

El código que escribiste puede parecer un poco extraño al principio, pero en realidad no es tan complejo. La idea general es simular un bucle sobre las filas del conjunto de datos.

El código crea una variable llamada `prod_code`, que se usará para crear un nuevo campo para identificar las casas. Cuando el código JavaScript detecta una fila de encabezado de propiedad como por ejemplo: **Property Code: MCX-002**

Establece la variable `prop_code` al código que encuentra en esa línea, en este caso, MCX-002.

Aquí viene el truco: hasta que aparece una nueva fila de encabezado, la variable `prop_code` mantiene ese valor. Por lo tanto, todas las filas que siguen a una fila como la que se mostró anteriormente tendrán el mismo valor para la variable `prop_code`.

Este es un ejemplo en el que puede mantener los valores de las filas anteriores en el conjunto de datos que se utilizarán en la fila actual.

## Nota

Tenga en cuenta que aquí usa JavaScript para ver y usar los valores de las filas anteriores, ipero no puede modificarlos! JavaScript siempre funciona en la fila actual.

# Realizando tareas sencillas con el paso Java Class.

Al igual que el paso de JavaScript, el paso User Defined Java Class también está destinado a insertar código en sus transformaciones, pero en este caso, es el código de Java. Si necesita implementar una funcionalidad que no se proporciona en los pasos integrados o si desea reutilizar algún código Java externo, o para acceder a las bibliotecas de Java o para aumentar el rendimiento, este paso es lo que necesita.

# Realizando tareas sencillas con el paso Java Class.

Para permitir la programación Java dentro de PDI, la herramienta utiliza las bibliotecas de proyectos de Janino. Janino es un compilador super-pequeño, súper-rápido, que compila código Java en tiempo de ejecución.

## Nota

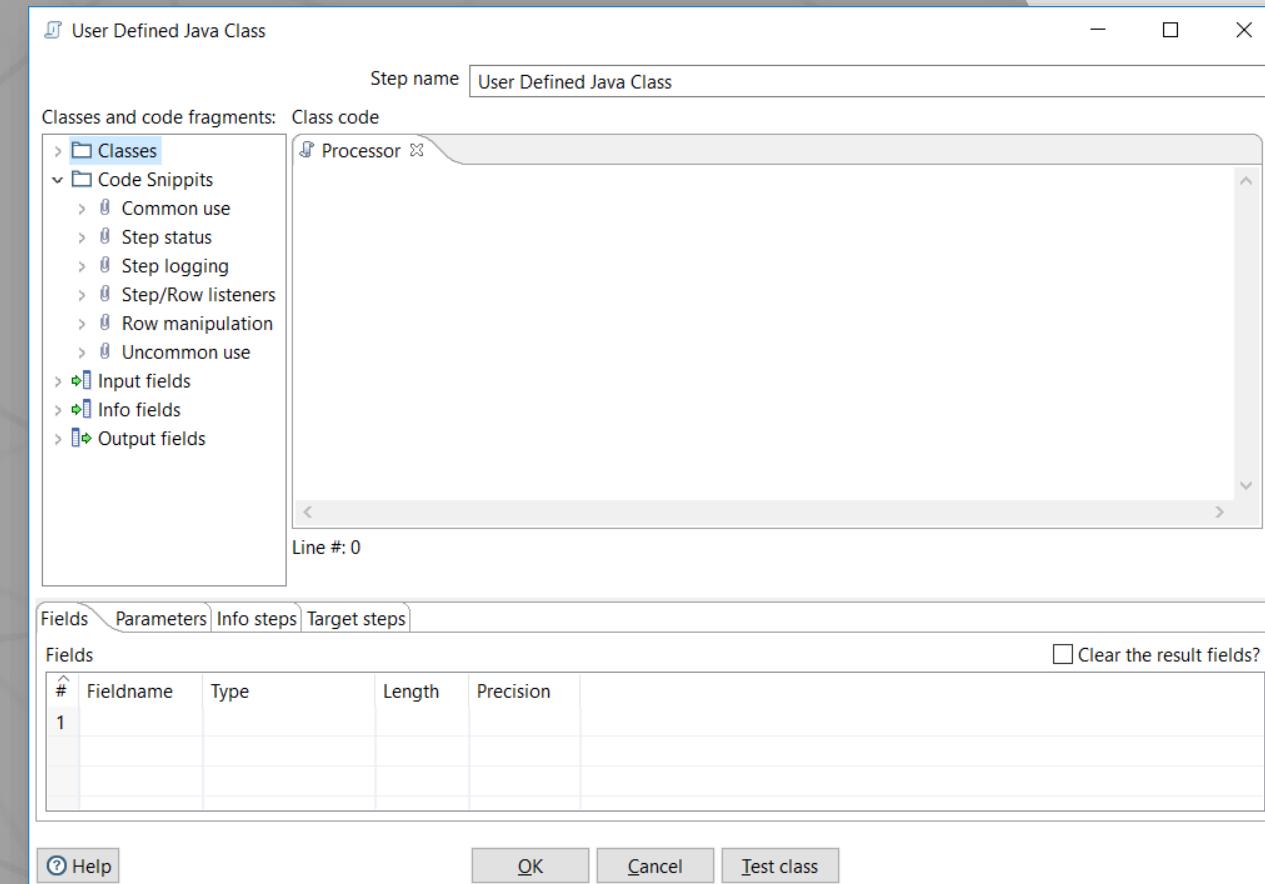
Para ver una lista completa de las características y limitaciones de Janino, puede seguir este enlace:  
<http://janino-compiler.github.io/janino/>



# Inserción de código Java utilizando el paso Java Class

El paso User Defined Java Class permite insertar código Java dentro de su Transformación. El código que escribe aquí se ejecuta una vez por cada fila que llega al paso.

La IU para el paso UDJC es muy similar a la IU para el paso JavaScript, como se muestra a continuación.



## Nota

El código que ves en los Fragmentos de código no es Java puro. Tiene muchas funciones predefinidas de PDI para manipular filas, ver el estado de los pasos y más.

# Inserción de código Java utilizando el paso Java Class

Los campos de entrada y salida aparecen automáticamente en el árbol cuando el código Java se compila correctamente.

Entonces tienes unas pestañas en la parte inferior. La siguiente tabla resume sus funciones:

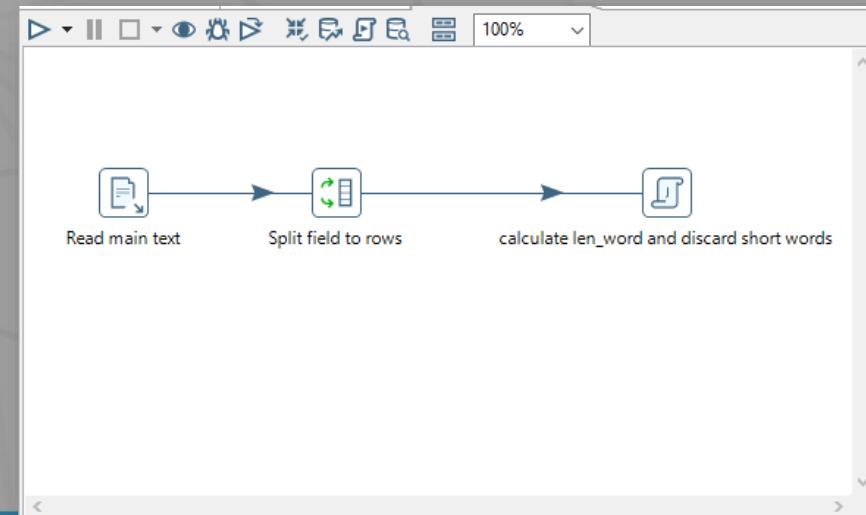
Tab	Función
<b>Fields</b>	Para declarar los nuevos campos añadidos por el paso.
<b>Parameters</b>	Para agregar parámetros a su código junto con sus valores.
<b>Info steps</b>	Para declarar pasos adicionales de PDI en su Transformación que proporcionan información para leer dentro de su código Java
<b>Target steps</b>	Para declarar los pasos PDI a los que se redirigirán las filas, en caso de que desee redirigir las filas a más de un destino

# Aprendiendo a insertar código java en un paso de clase Java

Al igual que hicimos en la sección de JavaScript, para mostrarle cómo usar el paso de Java Class, implementaremos la misma Transformación del Capítulo Controlando el flujo de datos, esta vez usando el código Java:

## Pasos

1. Usamos transformación creada anteriormente y guárdela con un nombre diferente.
2. Eliminar el paso de JavaScript.
3. Desde la categoría de pasos de Scripting, seleccione y arrastre un paso de **User Defined Java Class** al área de trabajo. Crea un salto desde el campo **Split field to rows** hacia este.



# Aprendiendo a insertar código java en un paso de clase Java

## Pasos

1. Haga doble clic en el paso **User Defined Java Class** (UDJC a partir de ahora) y en la pestaña Procesador, escriba lo siguiente:



```
public boolean processRow(StepMetaInterface smi, StepDataInterface sdi) throws KettleException {  
    Object[] r = getRow();  
    if (r == null) {  
        setOutputDone();  
        return false;  
    }  
  
    if (first) {  
        first = false;  
    }  
  
    Object[] outputRow = createOutputRow(r, data.outputRowMeta.size());  
  
    // HERE GOES YOUR CODE  
  
    putRow(data.outputRowMeta, outputRow);  
  
    return true;  
}
```

# Añadiendo campos

Agregar nuevos campos al conjunto de datos es realmente simple. Así es como lo haces:

## Pasos

1. En el código, define el campo como una variable interna y calcula su valor.
2. Entonces tienes que actualizar la fila de salida. Suponiendo que el nombre del nuevo campo es **my\_new\_field** y el nombre de la variable interna es **my\_var**, actualice la fila de salida de la siguiente manera: `get (Fields.Out, "my_new_field").setValue (outputRow, my_var);`
3. Finalmente, debe agregar el campo a la cuadrícula inferior. Simplemente agregue una nueva línea de cada nuevo campo. Debes proporcionar al menos el nombre de campo y el tipo.



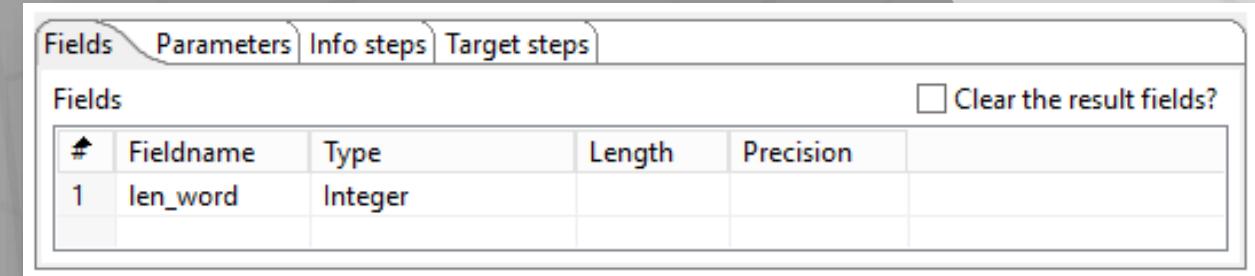
# Añadiendo campos

## Pasos

1. Escriba las siguientes líneas en el código de Java, reemplazando la línea // HERE GOES YOUR CODE:

```
String word = get(Fields.In, "word").getString(r);
long len_word = word.length();
get(Fields.Out, "len_word").setValue(outputRow, len_word);
```

2. Rellene la pestaña Fields en la cuadrícula inferior de la siguiente manera:



The screenshot shows a software interface with a ribbon bar at the top. The "Fields" tab is selected. Below the ribbon, there is a table titled "Fields" with the following data:

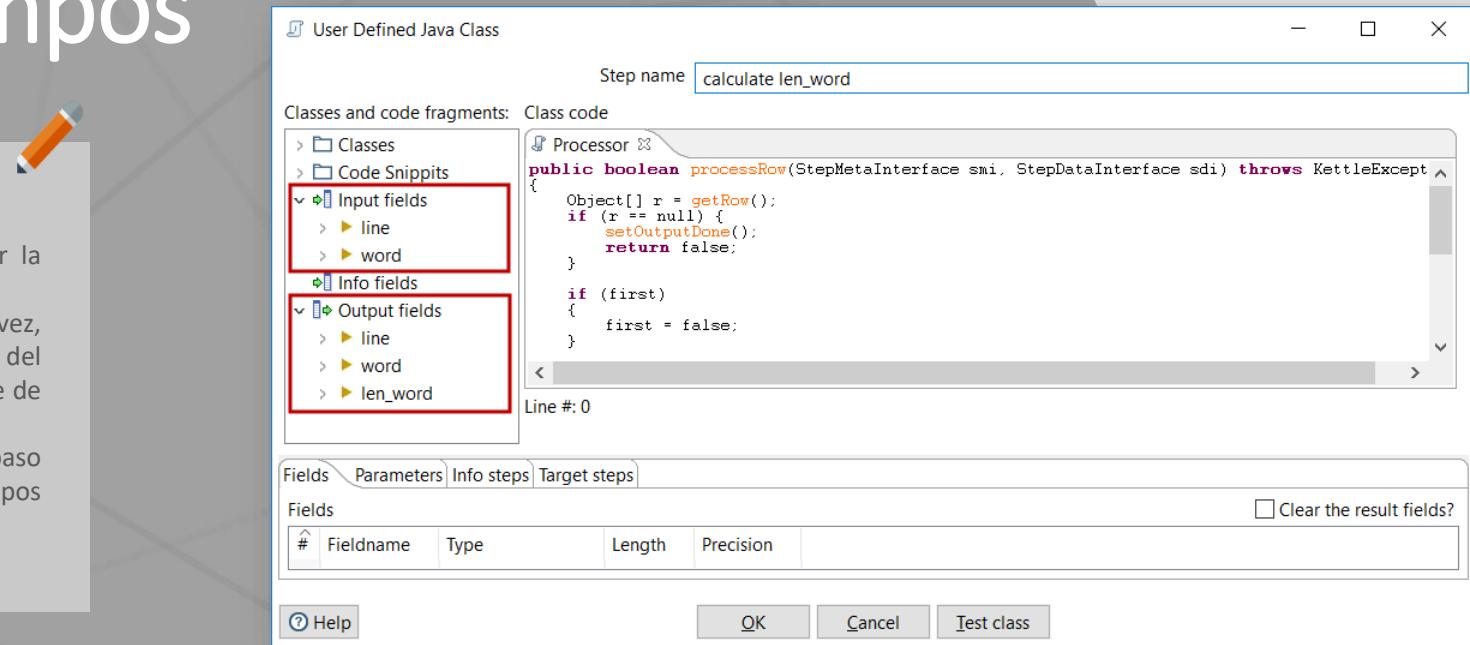
#	Fieldname	Type	Length	Precision
1	len_word	Integer		

Clear the result fields?

# Añadiendo campos

## Pasos

3. Haga clic en **OK** para cerrar la ventana y guardar la transformación.
4. Haga doble clic en el paso Java Class de nuevo. Esta vez, verá que Input fields y las ramas de los Output fields del árbol de la izquierda se han llenado con el nombre de los campos que entran y salen del paso:
5. Una vez más, cierra la ventana. Esta vez, con el paso seleccionado, ejecute una vista previa. Verá los campos antiguos más el nuevo campo **len\_word**:



**Examine preview data**

Rows of step: calculate len\_word and discard short words (1000 rows)

#	line	word	len_word
2..	least attention, up to the present time is that which treats of the geology	time	4
2..	least attention, up to the present time is that which treats of the geology	is	2
2..	least attention, up to the present time is that which treats of the geology	that	4
2..	least attention, up to the present time is that which treats of the geology	which	5
2..	least attention, up to the present time is that which treats of the geology	treats	6
2..	least attention, up to the present time is that which treats of the geology	of	2
2..	least attention, up to the present time is that which treats of the geology	the	3
2..	least attention, up to the present time is that which treats of the geology	geology	7
2..	of South America. The actual writing of this book appears to have occu...	of	2
2..	of South America. The actual writing of this book appears to have occu...	South	5

**Close**   **Stop**   **Get more rows**

# Modificando campos

Modificar un campo en lugar de agregar uno nuevo es aún más fácil. Suponiendo que el nombre de su campo es `my_field` y el valor que desea establecer se almacena en una variable llamada `my_var`; acaba de establecer el campo al nuevo valor utilizando la siguiente sintaxis:

```
| get(Fields.Out, "my_field").setValue(r, my_var);
```

Al hacerlo de esta manera, estás modificando la fila de salida. Cuando envía la fila al siguiente paso utilizando el método `putRow()`, el campo ya tiene su nuevo valor. En nuestra Transformación de muestra, queremos convertir la palabra a mayúsculas. Para esto, debe agregar la siguiente línea a su código que convierte la palabra en mayúsculas:

```
| get(Fields.Out, "word").setValue(outputRow, word.toUpperCase());
```

# Controlando el flujo con la función putRow ()

Con el paso de clase Java, usted controla qué filas pasan al siguiente paso usando el método `putRow ()`. Con este método de forma selectiva, usted decide qué filas enviar y qué filas descartar.

Como ejemplo, si queremos aplicar un filtro y mantener solo las palabras con una longitud mayor que tres, podríamos mover la función `putRow ()` dentro de una cláusula `if`, como se muestra en la siguiente muestra:

```
if (len_word > 3) {  
    putRow(data.outputRowMeta, outputRow);  
}
```

# Controlando el flujo con la función putRow ()

Su código final debe verse como sigue:

```
public boolean processRow(StepMetaInterface smi, StepDataInterface sdi) throws KettleException {
    Object[] r = getRow();
    if (r == null) {
        setOutputDone();
        return false;
    }
    if (first) { first = false; }
    Object[] outputRow = createOutputRow(r, data.outputRowMeta.size());

    String word = get(Fields.In, "word").getString(r);
    get(Fields.Out, "word").setValue(outputRow, word.toUpperCase());
    long len_word = word.length();
    get(Fields.Out, "len_word").setValue(outputRow, len_word);
    if (len_word > 3) {
        putRow(data.outputRowMeta, outputRow);
    }

    return true;
}
```

# Controlando el flujo con la función putRow ()

Si ejecuta una vista previa, verá algo como esto:

Examine preview data

Rows of step: calculate len\_word and discard short words (1000 rows)

#	line	word	len_word
1..	of South America. The actual writing of this book appears to have occupied	WRITING	7
1..	of South America. The actual writing of this book appears to have occupied	THIS	4
1..	of South America. The actual writing of this book appears to have occupied	BOOK	4
2..	of South America. The actual writing of this book appears to have occupied	APPEARS	7
2..	of South America. The actual writing of this book appears to have occupied	HAVE	4
2..	of South America. The actual writing of this book appears to have occupied	OCCUPIED	8
2..	Darwin a shorter period than either of the other volumes of the series; his	DARWIN	6
2..	Darwin a shorter period than either of the other volumes of the series; his	SHORTER	7
2..	Darwin a shorter period than either of the other volumes of the series; his	PERIOD	6
2..	Darwin a shorter period than either of the other volumes of the series; his	THAN	4

[Close](#) [Stop](#) [Get more rows](#)

# Probando la clase Java usando el botón de clase de prueba

El método para probar el código en un paso de clase Java es similar al que viste en la sección de JavaScript

## Pasos

1. Haga clic en el botón **Test class** en la parte inferior de la ventana.
2. Aparece una ventana para crear un conjunto de filas para la prueba. Rellénelo como se muestra en la captura de pantalla:
3. Haga clic en Vista previa y aparecerá una ventana que muestra diez filas idénticas con los valores de muestra proporcionados.
4. Haga clic en Aceptar en la ventana de vista previa para probar el código.
5. Aparece una ventana con el resultado de haber ejecutado el código en los datos de prueba:

2

Generate Rows

Step name: ## TEST DATA ##

Limit: 10

Never stop generating rows:

Interval in ms (delay):

Current row time field name:

Previous row time field name:

Fields :

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value	Set empty
1	line	String			.	,	,	N	test value test value	N
2	word	String			.	,	,	N	geological	N

Examine preview data

Rows of step: calculate len\_word and discard short words (10 rows)

#	line	word	len_word
1	test value test value	GEOLOGICAL	10
2	test value test value	GEOLOGICAL	10
3	test value test value	GEOLOGICAL	10
4	test value test value	GEOLOGICAL	10
5	test value test value	GEOLOGICAL	10
6	test value test value	GEOLOGICAL	10
7	test value test value	GEOLOGICAL	10
8	test value test value	GEOLOGICAL	10
9	test value test value	GEOLOGICAL	10
1..	test value test value	GEOLOGICAL	10

OK Preview Cancel

5

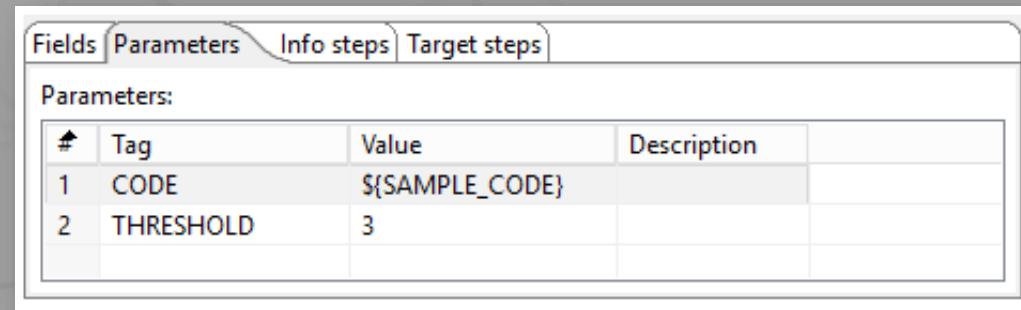
Close Show Log

# Aprovecha al máximo el paso de Java Class

En las secciones anteriores, aprendió a usar el paso de Java Class para realizar tareas básicas. El paso tiene algunas características más que le permiten crear un código enriquecido. Las siguientes subsecciones resumen algunos de ellos.

# Parámetros de recepción

Para escribir un código más flexible, puede agregar parámetros. Puede hacerlo configurando la pestaña **Parameters** en la cuadrícula inferior de la ventana de configuración de Java Class. Para cada nuevo parámetro, debe proporcionar un nombre en la columna Tag y un valor en la columna Value de la siguiente manera:

A screenshot of a software interface showing the 'Parameters' tab selected in a tab bar. The tab bar also includes 'Fields', 'Info steps', and 'Target steps'. Below the tab bar is a section titled 'Parameters:' containing a table. The table has columns labeled '#', 'Tag', 'Value', and 'Description'. There are two rows: Row 1 contains 'CODE' in 'Tag' and '\${SAMPLE\_CODE}' in 'Value'; Row 2 contains 'THRESHOLD' in 'Tag' and '3' in 'Value'.

#	Tag	Value	Description
1	CODE	\${SAMPLE_CODE}	
2	THRESHOLD	3	

## Nota

Tenga en cuenta que el valor de un parámetro de clase Java puede ser un valor fijo, así como una variable PDI.

# Parámetros de recepción

En tu código, lees un parámetro usando la función `getParameter ()`, como sigue:

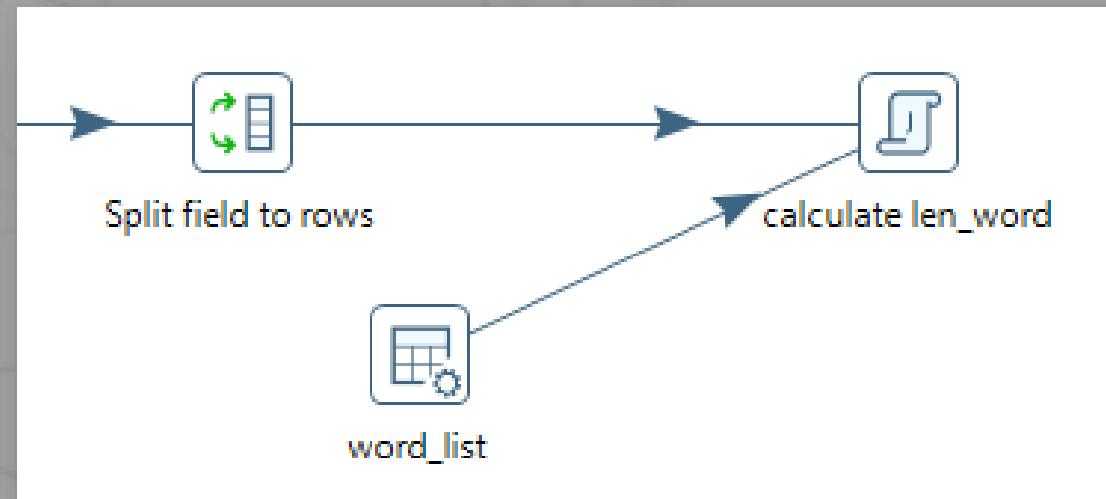
```
String code = getParameter("CODE");
```

Tenga en cuenta que los parámetros no tienen un tipo de datos y se leen como valores de strings. En caso de que necesite usarlos en un formato diferente, debe convertir los valores al tipo de datos adecuado, como se muestra en el siguiente ejemplo:

```
long threshold = Integer.parseInt(getParameter("THRESHOLD"));
```

# Leyendo datos de pasos adicionales.

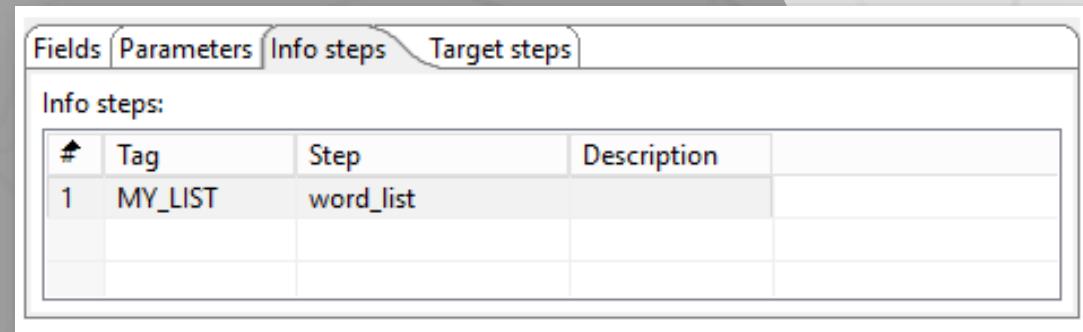
El paso de Java Class le permite leer datos de una o más secuencias secundarias. Para hacer esto, lo primero que debe hacer es conectar los pasos a su ícono de pasos de Java Class de la siguiente manera:



# Leyendo datos de pasos adicionales.

## Transformación de muestra con pasos de información

Luego, debe configurar la pestaña **Info steps** en la cuadrícula inferior de la ventana de configuración de Java Class. Para cada flujo secundario, debe proporcionar un nombre de Tag y seleccionar el nombre del paso entrante, como se muestra en la siguiente captura de pantalla:

A screenshot of a software interface titled "Java Class Configuration". At the top, there are four tabs: "Fields", "Parameters", "Info steps" (which is highlighted in blue), and "Target steps". Below the tabs, the "Info steps" section is displayed with the following data:

#	Tag	Step	Description
1	MY_LIST	word_list	

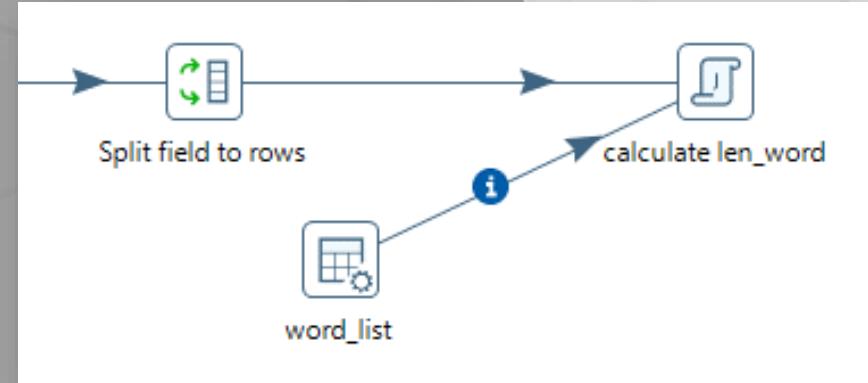
# Leyendo datos de pasos adicionales.

## Configurando la pestaña Info steps

Una vez que haga esto, los saltos de los pasos de información hacia el paso de Java Class cambiarán su apariencia:

## Info steps

En el código Java, puede usar el método `findInfoRowSet()` para hacer referencia al paso de información, y el método `getRowFrom ()` para leer las filas de un ciclo, como se muestra en el siguiente código de ejemplo:



```
RowSet infoStream = findInfoRowSet("my_list");
Object[] infoRow = null;
while((infoRow = getRowFrom(infoStream)) != null){
    < your code here >
}
```

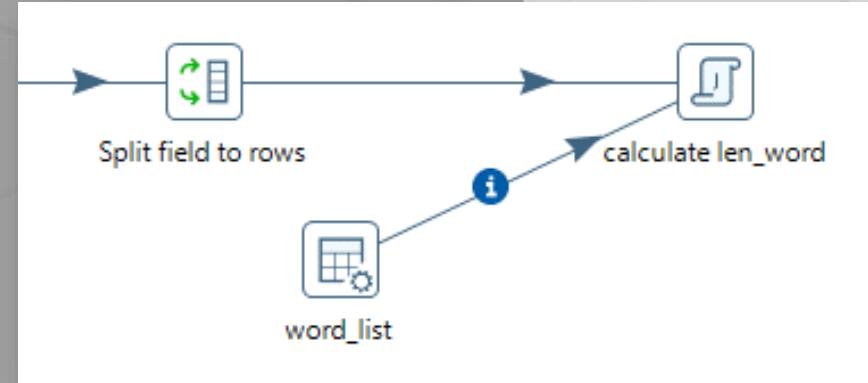
# Leyendo datos de pasos adicionales.

## Configurando la pestaña Info steps

Una vez que haga esto, los saltos de los pasos de información hacia el paso de Java Class cambiarán su apariencia:

## Info steps

En el código Java, puede usar el método `findInfoRowSet()` para hacer referencia al paso de información, y el método `getRowFrom ()` para leer las filas de un ciclo, como se muestra en el siguiente código de ejemplo:



```
RowSet infoStream = findInfoRowSet("my_list");
Object[] infoRow = null;
while((infoRow = getRowFrom(infoStream)) != null){
    < your code here >
}
```

# Redireccionar datos a diferentes pasos de destino

De la misma manera, como puede recibir datos de más de una secuencia entrante, puede redirigir las filas a diferentes pasos de destino. Después de crear los saltos de la Clase Java hacia cada paso de destino, debe configurar la pestaña **Output steps** en la cuadrícula inferior de la ventana de configuración de la Clase Java.

# Redireccionar datos a diferentes pasos de destino

En el código Java, utiliza el método `findTargetRowSet()` para identificar el conjunto de filas de destino y el método `putRowTo()` para especificar a qué paso de destino desea redireccionar cada fila:

```
private RowSet targetStreamA = null;
private RowSet targetStreamB= null;

public boolean processRow( ... ) {

    if (first){
        first = false;
        targetStreamA = findTargetRowSet("target1");
        targetStreamB = findTargetRowSet("target2");
    }

    if (<condition>)
        putRowTo(data.outputRowMeta, r, targetStreamA);
    else
        putRowTo(data.outputRowMeta, r, targetStreamB);

    ...
}
```

# Analizando estructuras JSON

Si necesita analizar estructuras JSON complejas y el paso **JSON Input** no satisface sus necesidades, puede resolver la tarea con un paso de clase Java. Una forma sencilla de hacerlo es con el paquete **org.json.simple**. Entre las clases que se encuentran en este paquete, utilizará principalmente lo siguiente:

org.json.simpleClass	Descripción
JSONParser	Esta clase analiza el texto JSON
JSONValue	Esta clase tiene métodos para analizar cadenas JSON en objetos Java.
JSONValue	Es una colección desordenada de pares name/value
JSONArray	Es una secuencia ordenada de valores.

# Analizando estructuras JSON

Para usar este paquete en un paso de clase Java, solo importa las bibliotecas necesarias al principio del código, de la siguiente manera:

Una vez que importe las bibliotecas, puede usarlas en su código. Por ejemplo, en su conjunto de datos, si tiene un campo llamado **content** que contiene una estructura JSON, puede analizarlo con las siguientes líneas de código

Entonces puede obtener los valores en su objeto simplemente llamando a la función `get()`:

```
import org.json.simple.JSONArray;
import org.json.simple.JSONObject;
import org.json.simple.JSONValue;
```

```
String contentStr = get(Fields.In, "content").getString(r);
JSONObject data = (JSONObject) JSONValue.parse(contentStr);
```

```
String name = (String) data.get("title");
```

# Evitar la codificación utilizando pasos construidos a propósito

Usted vio a través de los ejercicios cuán poderosos son los pasos de JavaScript y Java Class para ayudarlo en sus transformaciones. En versiones anteriores de PDI, la codificación de JavaScript era el único medio que tenía para tareas específicas. En las últimas versiones de PDI, aparecieron pasos reales que eliminan la necesidad de codificar en muchos casos. Aquí tienes algunos ejemplos de estos pasos:

- **Formula step:** antes de la aparición de este paso, había muchas funciones, como la función de texto derecha o izquierda, que solo se podían resolver con JavaScript.
- **Analytic Query:** este paso ofrece una manera de recuperar información de filas antes o después de la fila actual.
- **Split field into rows:** este paso se usa para crear varias filas a partir de un solo valor de cadena.
- **Add value fields changing sequence:** similar al paso **Add sequence**, pero el valor de secuencia se restablece cada vez que cambia un valor en la lista de campos especificados

## Nota



Siempre que haya un paso que haga lo que usted quiere hacer, debería preferir usar ese paso en lugar de codificar.

# ¿Por qué debería preferir utilizar un paso específico en lugar de un código? Hay algunas razones:

- La codificación lleva más tiempo de desarrollo. No tiene que perder tiempo codificando si hay pasos que resuelven su problema.
- El código es difícil de mantener. Si tiene que modificar o corregir una Transformación, será mucho más fácil atacar el cambio si la Transformación se compone de muchos pasos coloridos con nombres significativos, en lugar de que la Transformación se realice con solo un par de iconos de JavaScript o Java Class. .
- Un montón de iconos es auto-documentado. Los pasos de JavaScript o Java Class son como la caja de Pandora. Hasta que no los abra, no sabrá exactamente qué hacen y si contienen solo una línea de código o miles.
- En el caso de JavaScript, debes saber que es inherentemente lento. Las alternativas más rápidas para expresiones simples son la **User Defined Java Expression** (también en la categoría Scripting) y los pasos **Calculator** (en la categoría Transform). Por lo general, son más del doble de rápido.

Por el contrario, hay situaciones en las que puede preferir o tener que codificar. Vamos a enumerar algunos de ellos:

- Para manejar datos de entrada no estructurados
- Manipular objetos XML o JSON complicados.
- Para acceder a las bibliotecas de Java
- Cuando necesita utilizar una función proporcionada por el lenguaje JavaScript o Java y no es proporcionada por ninguno de los pasos regulares de PDI
- En el caso de JavaScript, cuando el código guarda muchos pasos regulares de PDI (así como espacio de pantalla), y cree que no vale la pena mostrar los detalles de lo que hacen esos pasos.
- En el caso de Java, por razones de rendimiento, si tiene que lidiar con millones de filas y operaciones muy complicadas, una clase Java puede ser todo lo que necesita para terminar con una Transformación que funciona muy bien.

Cuando tenga dudas sobre la solución adecuada, codificar o no codificar, tenga en cuenta los siguientes puntos:

- Tiempo de desarrollo, ¿ahorrará tiempo si decide implementar la solución con el código?
- Mantenimiento, ¿será más fácil mantener el código o mantener la solución con pasos específicos de PDI?
- Documentación, ¿tendrá que dedicar más tiempo a documentar la solución?
- Capacidad para manejar datos no estructurados, ¿los pasos de PDI le permiten manejar las estructuras de sus datos?
- Número de pasos necesarios, ¿necesita pocos o muchos pasos PDI para implementar la solución?
- Rendimiento, ¿es la solución elegida performant?