

## Ejemplo 2 - Filtrando datos

En este caso de uso, vamos a leer un archivo CSV y filtrar los datos para quedarnos con un subconjunto de los mismos. Además, aprenderemos a gestionar los errores y ejecutar la transformación desde el terminal.

### Lectura CSV

Tras crear la nueva transformación (CTRL + N), desde *Input* arrastraremos el paso de *CSV input file* para seleccionar el archivo

`samples\transformations\files\Zipssortedbycitystate.csv` dentro de nuestra instalación de Pentaho.

Tras seleccionar el archivo, mediante el botón *Get Fields* cargaremos y comprobaremos que los campos que vamos a leer son correctos (nombre y tipo de los datos).

CSV file input

Step name: CSV file input

Filename: C:\data-integration\samples\transformations\files\Zipssortedbycitystate.csv

Delimiter: ,

Enclosure:

NIO buffer size: 50000

Lazy conversion? ☒

Header row present? ☒

Add filename to result ☐

The row number field name (optional):

Running in parallel? ☐

New line possible in fields? ☐

Format: mixed

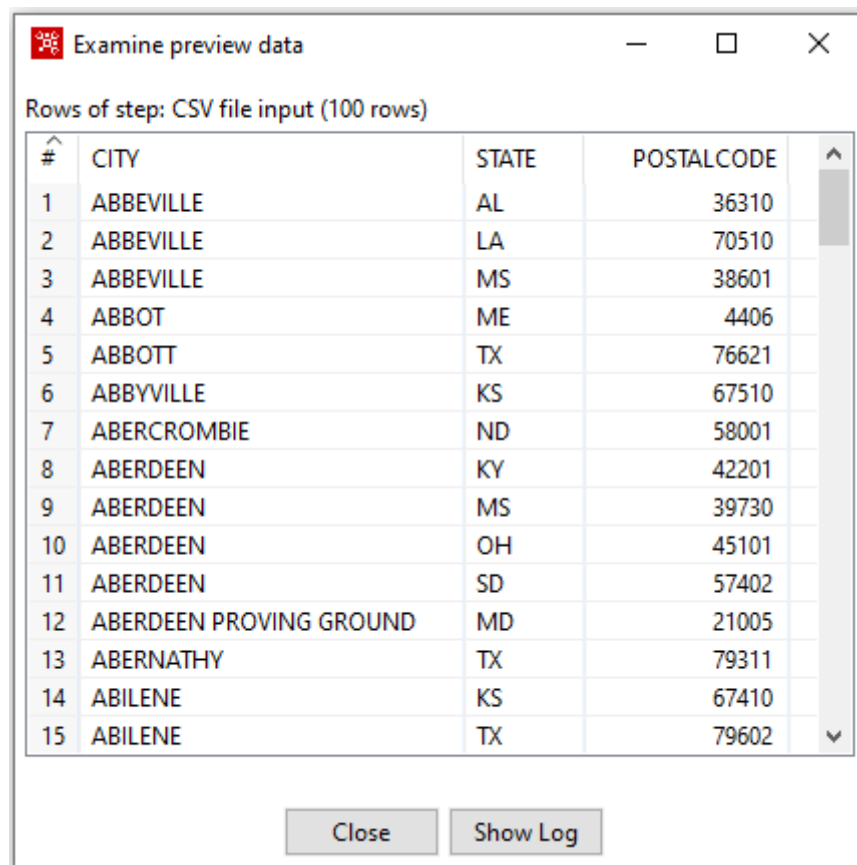
File encoding:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	CITY	String		23		\$	,	.	none
2	STATE	String		2		\$	,	.	none
3	POSTALCODE	Integer	#	15	0	\$	,	.	none

Buttons: Help, OK, Get Fields, Preview, Cancel

Caso de Uso 1 - Tras pulsar sobre Get Fields

Tras ello, mediante el botón *Preview* comprobaremos que los datos se leen correctamente.



The screenshot shows a window titled "Examine preview data" with a close button. Below the title bar, it says "Rows of step: CSV file input (100 rows)". The main content is a table with 4 columns: "#", "CITY", "STATE", and "POSTALCODE". The table contains 15 rows of data. At the bottom of the window, there are two buttons: "Close" and "Show Log".

#	CITY	STATE	POSTALCODE
1	ABBEVILLE	AL	36310
2	ABBEVILLE	LA	70510
3	ABBEVILLE	MS	38601
4	ABBOT	ME	4406
5	ABBOTT	TX	76621
6	ABBYVILLE	KS	67510
7	ABERCROMBIE	ND	58001
8	ABERDEEN	KY	42201
9	ABERDEEN	MS	39730
10	ABERDEEN	OH	45101
11	ABERDEEN	SD	57402
12	ABERDEEN PROVING GROUND	MD	21005
13	ABERNATHY	TX	79311
14	ABILENE	KS	67410
15	ABILENE	TX	79602

*Caso de Uso 1 - Resultado de la opción Preview sobre Ciudades*

## Filtrado de datos

Una vez leído, el siguiente paso es filtrar las filas. Para ello, desde la categoría de *Flow*, arrastramos el paso *Filter* (Filtrar filas), y las conectamos tal como hemos realizado en el caso anterior. Al soltar la flecha, nos mostrará dos opciones:

- *Main output of step*: define los pasos con un flujo principal, donde todo funciona bien
- *Error handling of step*: define los pasos a seguir en caso de encontrar un error

De momento elegimos la primera y configuramos el filtro para solo seleccionar aquellos datos cuyo estado sea NY (`STATE = NY`)

Filter rows

Step name: Filtro NY

Send 'true' data to step: [dropdown]

Send 'false' data to step: [dropdown]

The condition:

[input] STATE = [input] NY (String)

[Help] [OK] [Cancel]

### Caso de Uso 1 - Configuración del filtro

Para configurar el resultado, seleccionamos el paso del filtro, y bien pulsamos sobre el icono del ojo de la barra de herramientas, o sobre el paso, tras pulsar con el botón derecho, seleccionamos la opción *Preview*.

Examine preview data

Rows of step: Filtro NY (1000 rows)

#	CITY	STATE	POSTALCODE
1	ACRA	NY	12405
2	ADAMS	NY	13605
3	ADAMS CENTER	NY	13606
4	ADIRONDACK	NY	12808
5	AKRON	NY	14001
6	ALABAMA	NY	14003
7	ALBANY	NY	12201
8	ALBANY	NY	12203
9	ALBANY	NY	12205
10	ALBANY	NY	12207
11	ALBANY	NY	12209
12	ALBANY	NY	12211
13	ALBANY	NY	12214
14	ALBANY	NY	12222
15	ALBANY	NY	12224

Close Stop Get more rows

Process flow: Ciudades → Filtro NY

Metrics for Filtro NY:

Copynr	0
Read	18708
Written	1001
Input	0
Output	0
Updated	0
Rejected	0
Errors	0
Active	Paused
Time	15.1s
Speed (r/s)	1,236
input/output	2671/0

### Caso de Uso 1 - Resultado de hacer Preview sobre Filtro NY


Por defecto se precargan 1000 filas. Tras comprobar el resultado, pulsamos sobre *Stop* para detener el proceso de previsualización. Las métricas que aparecen nos informan del proceso y su rendimiento.

## Ordenación

El siguiente paso que vamos a realizar es ordenar los datos por su código *Postal Code*. Para ello, desde la categoría de *Transform*, arrastramos el paso de *Sort rows* (Ordenar filas), y conectamos la salida del filtrado con la ordenación eligiendo la salida principal (*main output of step*).

### Forzando un error

Vamos a forzar un error para comprobar cómo lo indica *Spoon*. Si al elegir el nombre del campo, en vez de *POSTAL CODE* escribimos *CP*, cuando previsualizamos el resultado, podremos ver como aparece la marca de prohibido en la esquina superior derecha del paso, y si visualizamos el log y las métricas de los pasos, veremos el error:



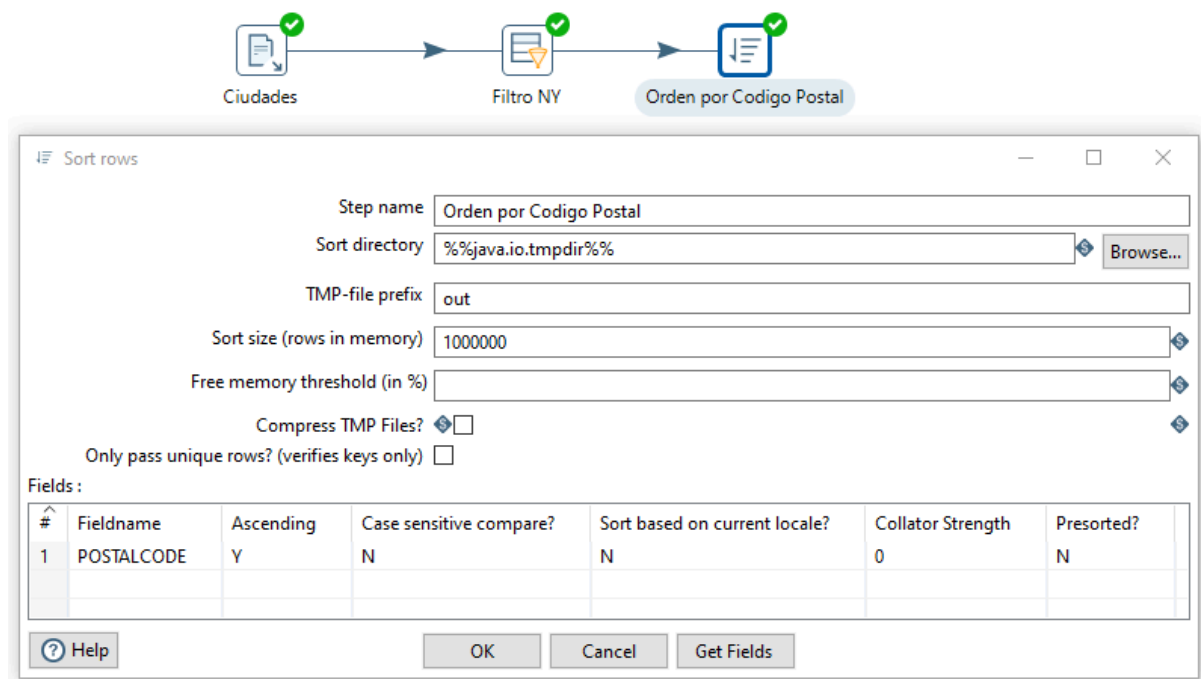
Execution Results

Logging Execution History Step Metrics Performance Graph Metrics Preview data

```
2021/10/24 17:00:14 - caso1-filterNY - Dispatching started for transformation [caso1-filterNY]
2021/10/24 17:00:14 - Ciudades.0 - Header row skipped in file 'C:\data-integration\samples\transformations\files\Zipssortedbycitystate.csv'
2021/10/24 17:00:15 - Orden porCodigo Postal.0 - ERROR (version 9.1.0.0-324, build 9.1.0.0-324 from 2020-09-07 05.09.05 by buildguy) : Unexpected error
2021/10/24 17:00:15 - Orden porCodigo Postal.0 - ERROR (version 9.1.0.0-324, build 9.1.0.0-324 from 2020-09-07 05.09.05 by buildguy) : org.pentaho.di.core.exception.KettleException:
2021/10/24 17:00:15 - Orden porCodigo Postal.0 - The field CP specified in the "Orden porCodigo Postal" step is not in the steps input stream.
2021/10/24 17:00:15 - Orden porCodigo Postal.0 -
2021/10/24 17:00:15 - Orden porCodigo Postal.0 - at org.pentaho.di.trans.steps.sort.SortRows.processRow(SortRows.java:426)
2021/10/24 17:00:15 - Orden porCodigo Postal.0 - at org.pentaho.di.trans.step.RunThread.run(RunThread.java:62)
2021/10/24 17:00:15 - Orden porCodigo Postal.0 - at java.lang.Thread.run(Unknown Source)
2021/10/24 17:00:15 - Orden porCodigo Postal.0 - Finished processing (I=0, O=0, R=1, W=0, U=0, E=1)
2021/10/24 17:00:15 - Ciudades.0 - Finished processing (I=10948, O=0, R=0, W=10947, U=0, E=0)
2021/10/24 17:00:15 - Filtro NY.0 - Finished processing (I=0, O=0, R=1046, W=61, U=0, E=0)
2021/10/24 17:00:15 - caso1-filterNY - Transformation detected one or more steps with errors.
2021/10/24 17:00:15 - caso1-filterNY - Transformation is killing the other steps!
2021/10/24 17:00:15 - caso1-filterNY - ERROR (version 9.1.0.0-324, build 9.1.0.0-324 from 2020-09-07 05.09.05 by buildguy) : Errors detected!
2021/10/24 17:00:15 - Spoon - The transformation has finished!!
2021/10/24 17:00:15 - caso1-filterNY - ERROR (version 9.1.0.0-324, build 9.1.0.0-324 from 2020-09-07 05.09.05 by buildguy) : Errors detected!
2021/10/24 17:00:15 - caso1-filterNY - ERROR (version 9.1.0.0-324, build 9.1.0.0-324 from 2020-09-07 05.09.05 by buildguy) : Errors detected!
```

### Caso de Uso 1 - Forzando un error

Volvemos a editar el paso, corregimos el nombre del campo (escribimos *POSTAL CODE*) y comprobamos que ahora sí que funciona correctamente

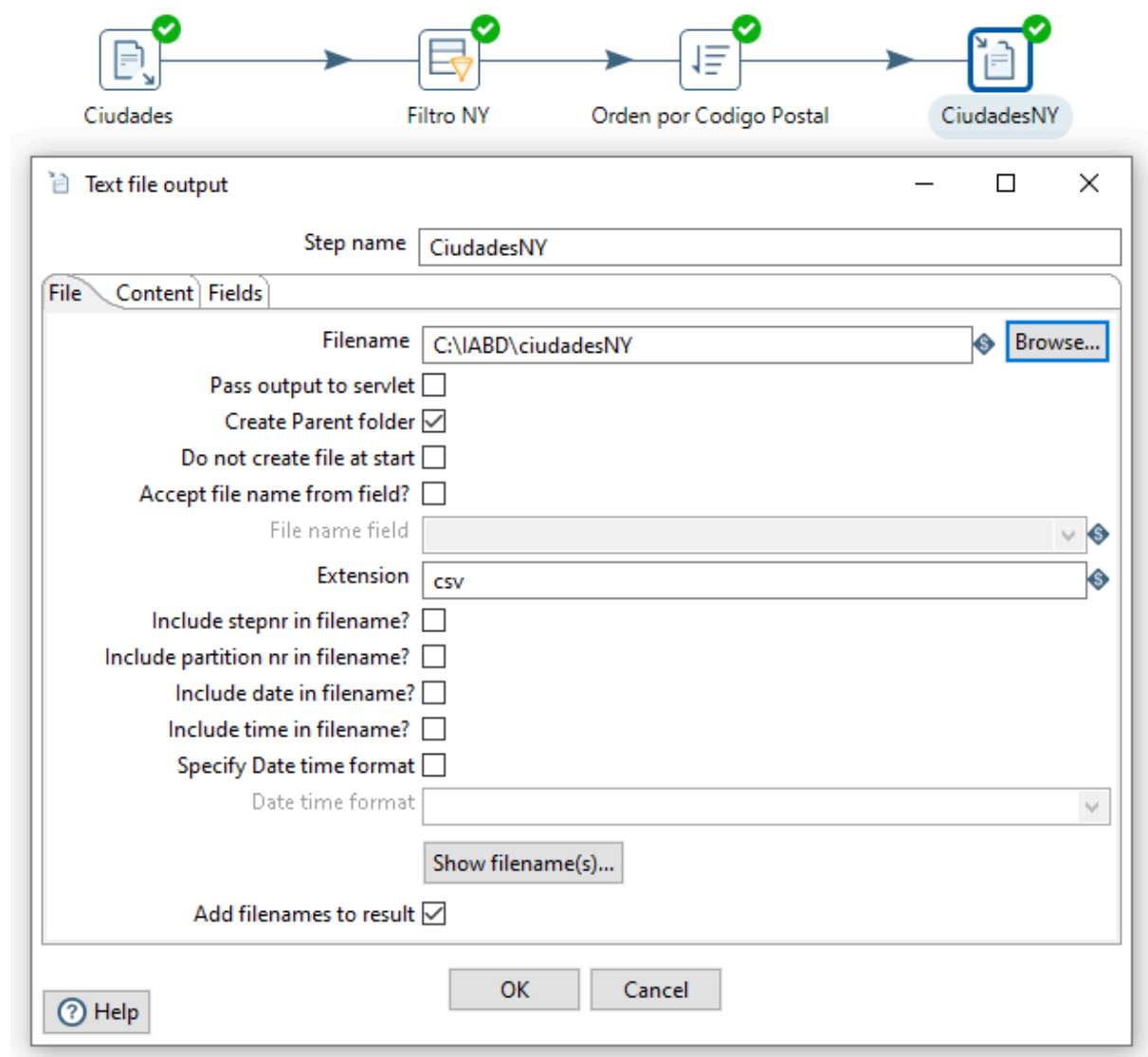


### Caso de Uso 1 - Ordenando

## Escritura del resultado

Una vez realizados todos los pasos, sólo nos queda enviar el resultado a un fichero para persistir la transformación.

Para ello, desde la categoría de *Output* arrastramos el paso *Text file output*, y lo conectamos desde la salida del paso de ordenación. Tras ello, editar este paso para indicar el archivo donde almacenar el resultado.

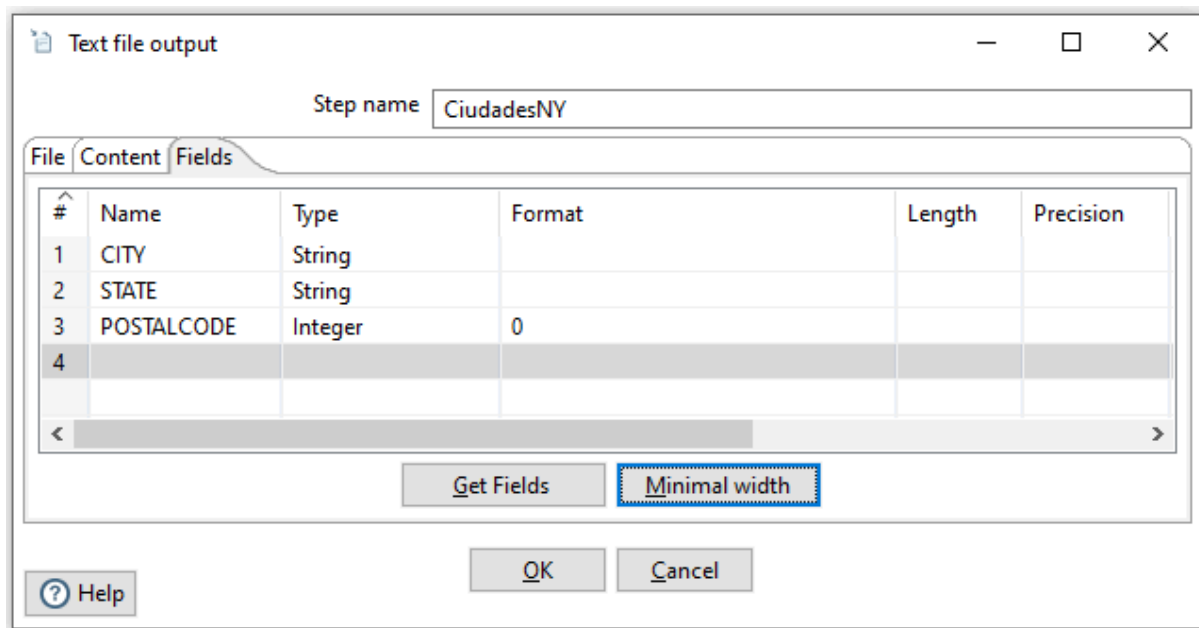


### Caso de Uso 1 - Guardando el resultado

Tras ello, podremos ejecutar la transformación (icono del triángulo, menú *Action* -> *Run* o F9) y comprobar el resultado en el fichero:

```
CITY;STATE;POSTALCODE
HOLTSVILLE      ;NY;501
FISHERS ISLAND   ;NY;6390
NEW YORK          ;NY;10001
NEW YORK          ;NY;10003
NEW YORK          ;NY;10005
NEW YORK          ;NY;10007
NEW YORK          ;NY;10009
```

Al comprobar el fichero, vemos que se han quedado espacios en blanco a la derecha del nombre de las ciudades, ya que la columna tenía un tamaño configurado. Si volvemos a editar el último paso, en la pestaña de *Fields* (Campos) podemos indicar mediante el botón de *Minimal width* que reduzca su anchura al mínimo:



### Caso de Uso 1 - Anchura mínima de los campos

Y tras volver a ejecutar la transformación, veremos que ahora sí que obtenemos los datos que esperábamos:

```
CITY;STATE;POSTALCODE
HOLTSVILLE;NY;501
FISHERS ISLAND;NY;6390
NEW YORK;NY;10001
NEW YORK;NY;10003
NEW YORK;NY;10005
NEW YORK;NY;10007
NEW YORK;NY;10009
```

## Uso de Pan

Mediante la utilidad *Pan*, podemos ejecutar las transformaciones sin necesidad de arrancar *Spoon*. Para indicarle el archivo que contiene la transformación, al comando `pan.bat` (o `pan.sh` en el caso de Ubuntu) le pasamos el parámetro `/file=rutaArchivo.ktr`.

Para comprobar su funcionamiento, vamos a eliminar el fichero generado. A continuación, ejecutamos `pan`:

```
pan.bat /file=c:/IABD/caso1filtradoNY.ktr
```

Tras algunos segundos y varias líneas de debug del arranque de `pan`, tendremos un mensaje similar al siguiente:

2021/10/24 18:01:42 - Start of run.  
2021/10/24 18:01:42 - caso1filtradoNY - Dispatching started for transformation [caso1filtradoNY]  
2021/10/24 18:01:42 - Ciudades.0 - Header row skipped in file  
'C:\data-integration\samples\transformations\files\Zipssortedbycitystate.csv'  
2021/10/24 18:01:42 - Ciudades.0 - Finished processing (I=21380, O=0, R=0, W=21379, U=0, E=0)  
2021/10/24 18:01:42 - Filtro NY.0 - Finished processing (I=0, O=0, R=21379, W=1146, U=0, E=0)  
2021/10/24 18:01:42 - Orden porCodigo Postal.0 - Finished processing (I=0, O=0, R=1146, W=1146, U=0, E=0)  
2021/10/24 18:01:42 - CiudadesNY.0 - Finished processing (I=0, O=1147, R=1146, W=1146, U=0, E=0)  
2021/10/24 18:01:43 - Carte - Installing timer to purge stale objects after 1440 minutes.  
2021/10/24 18:01:43 - Finished!  
2021/10/24 18:01:43 - Start=2021/10/24 18:01:42.424, Stop=2021/10/24 18:01:43.041  
2021/10/24 18:01:43 - Processing ended after 0 seconds.  
2021/10/24 18:01:43 - caso1filtradoNY -  
2021/10/24 18:01:43 - caso1filtradoNY - Step Ciudades.0 ended successfully, processed 21379 lines.  
( - lines/s)  
2021/10/24 18:01:43 - caso1filtradoNY - Step Filtro NY.0 ended successfully, processed 21379 lines. ( - lines/s)  
2021/10/24 18:01:43 - caso1filtradoNY - Step Orden porCodigo Postal.0 ended successfully, processed 1146 lines. ( - lines/s)  
2021/10/24 18:01:43 - caso1filtradoNY - Step CiudadesNY.0 ended successfully, processed 1146 lines. ( - lines/s)

Y si comprobamos el fichero, veremos que ha vuelto a aparecer.