Ejercicio 4 - Cuestionarios Airbnb

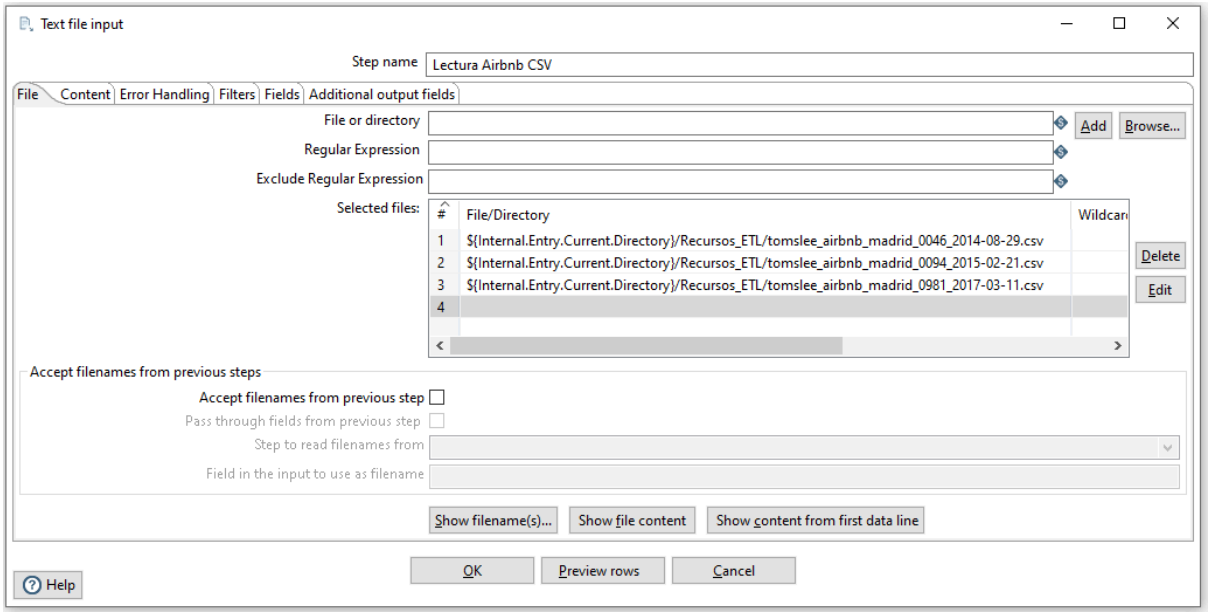
Para el siguiente caso de uso, vamos a utilizar datos de los cuestionarios de AirBnb que se pueden descargar desde

https://insideairbnb.com/get-the-data/

En concreto, nos vamos a centrar en los datos de Málaga.

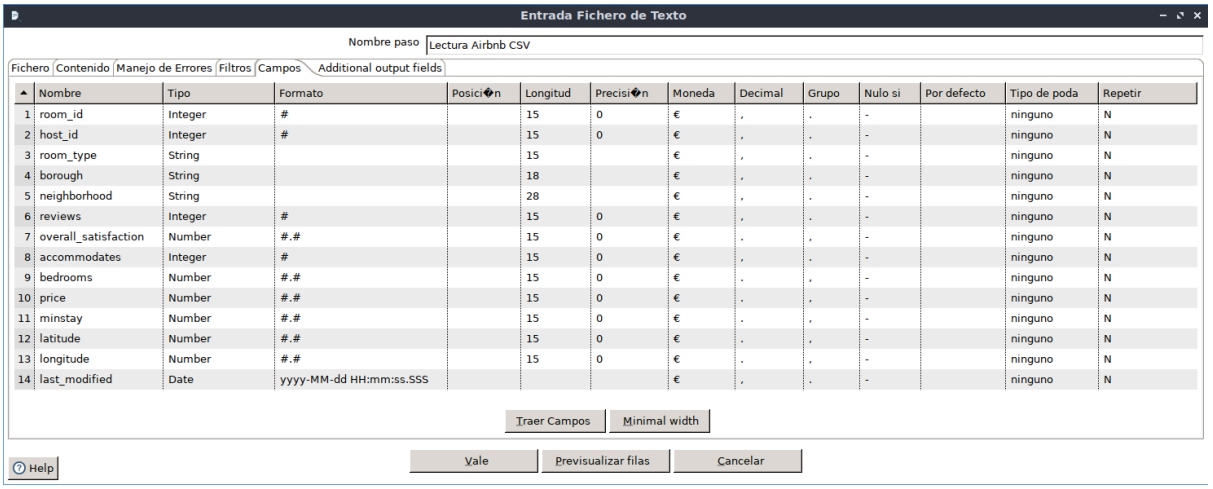
Uniendo datos

Una vez descargados los datos y descomprimidos, vamos a cargar los ficheros en el mismo paso, utilizando dentro de *Input* la opción de *Text File Input/Entrada Fichero de Texto*:



Caso de Uso 3 - Filtrado compuesto

Recordad que antes, en la pestaña *Content/Contenido* elegiremos la , como separador de campos y en *Fields/Campos*, tenemos que obtener los campos a leer:

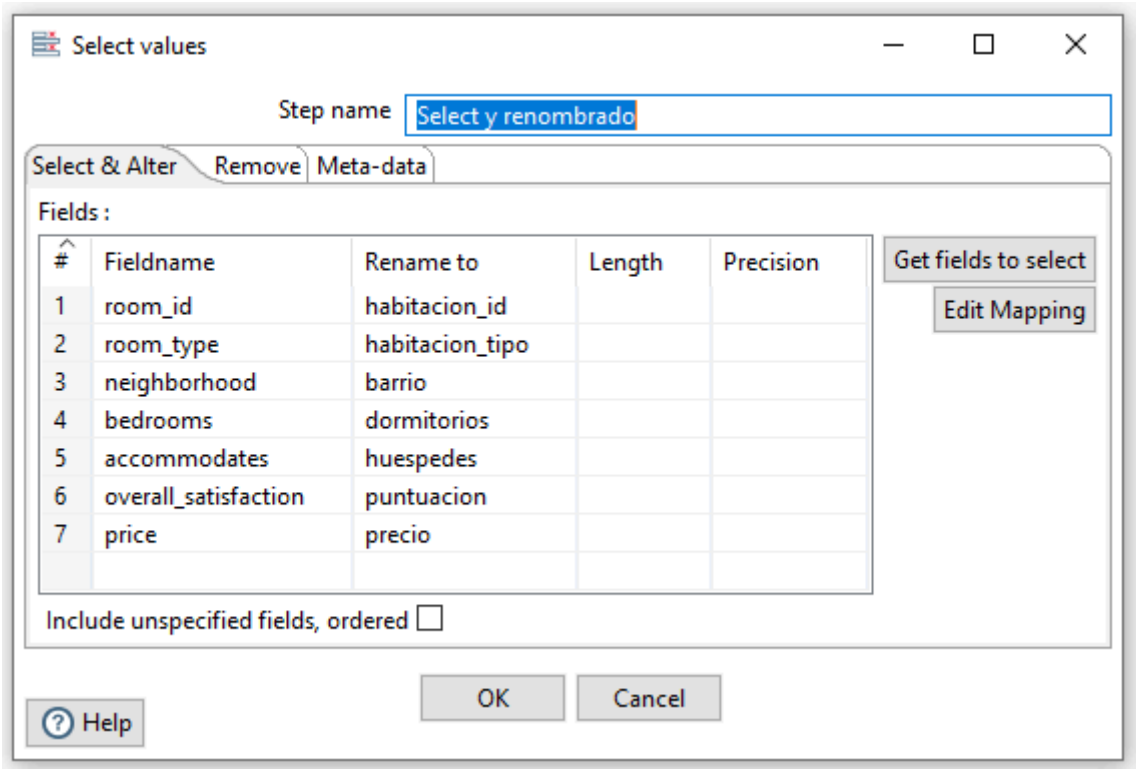


	Nombre	Tipo	Formato	Posici&#65533;n	Longitud	Precisi&#65533;n	Moneda	Decimal	Grupo	Nulo si	Por defecto	Tipo de poda	Repetir
1	room_id	Integer	#		15	0	€	,	.	-		ninguno	N
2	host_id	Integer	#		15	0	€	,	.	-		ninguno	N
3	room_type	String			15		€	,	.	-		ninguno	N
4	borough	String			18		€	,	.	-		ninguno	N
5	neighborhood	String			28		€	,	.	-		ninguno	N
6	reviews	Integer	#		15	0	€	,	.	-		ninguno	N
7	overall_satisfaction	Number	#.#		15	0	€	.	,	-		ninguno	N
8	accommodates	Integer	#		15	0	€	,	.	-		ninguno	N
9	bedrooms	Number	#.#		15	0	€	.	,	-		ninguno	N
10	price	Number	#.#		15	0	€	.	,	-		ninguno	N
11	minstay	Number	#.#		15	0	€	.	,	-		ninguno	N
12	latitude	Number	#.#		15	0	€	.	,	-		ninguno	N
13	longitude	Number	#.#		15	0	€	.	,	-		ninguno	N
14	last_modified	Date	yyyy-MM-dd HH:mm:ss.SSS				€	,	.	-		ninguno	N

Caso de Uso 3 - Campos

Nos vamos a quedar con un subconjunto de las columnas y las vamos a renombrar. Para ello, dentro de la categoría *Transform* elegimos el paso *Select values*/*Selecciona/Renombra valores* y elegimos y renombramos los siguentes campos:

- `room_id`, `room_type`, `neighborhood`, `bedrooms`, `overall_satisfaction`, `accommodates`, y `price` pasarán a ser `habitacion_id`, `habitacion_tipo`, `barrio`, `dormitorios`, `puntuacion`, `huespedes` y `precio`.

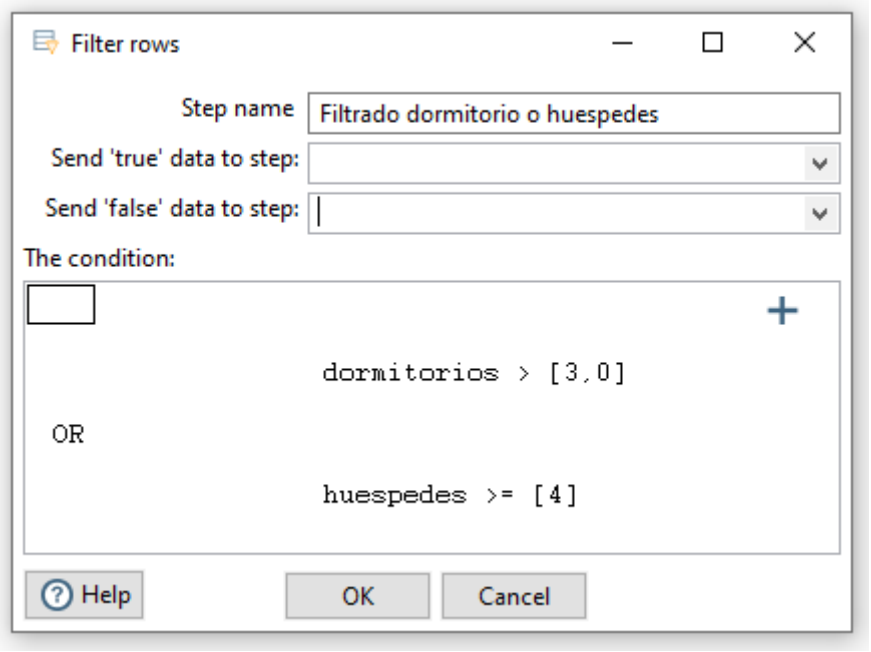


#	Fieldname	Rename to	Length	Precision
1	room_id	habitacion_id		
2	room_type	habitacion_tipo		
3	neighborhood	barrio		
4	bedrooms	dormitorios		
5	accommodates	huespedes		
6	overall_satisfaction	puntuacion		
7	price	precio		

Caso de Uso 3 - Selección y nombrado de campos

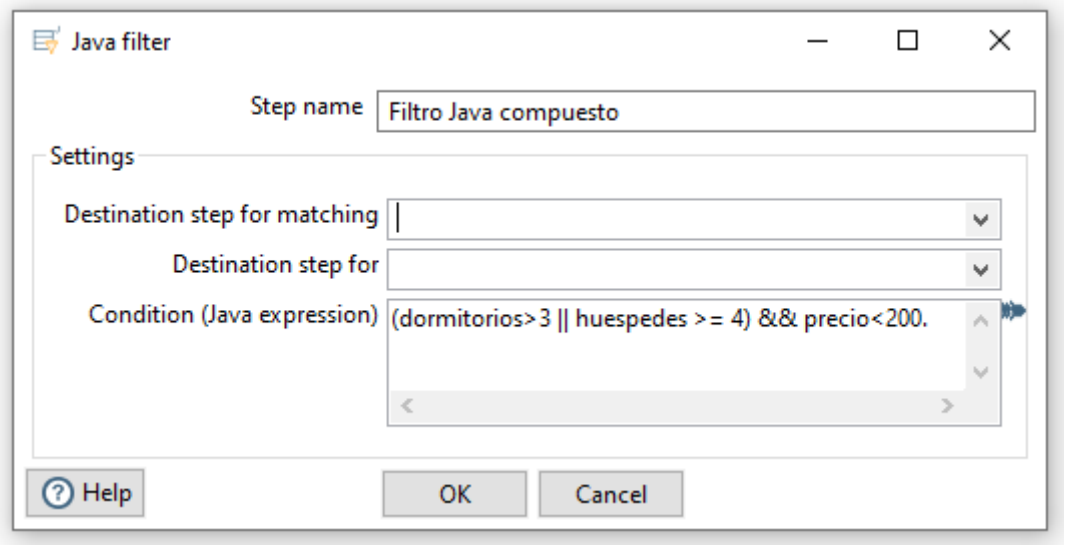
Filtrado compuesto

El siguiente paso que vamos a hacer es quedarnos con aquellos cuestionarios con más de 3 dormitorios o al menos 4 huéspedes. Así pues, con el paso *Filter Rows/Filtrar filas* realizaremos:



Caso de Uso 3 - Filtrado compuesto

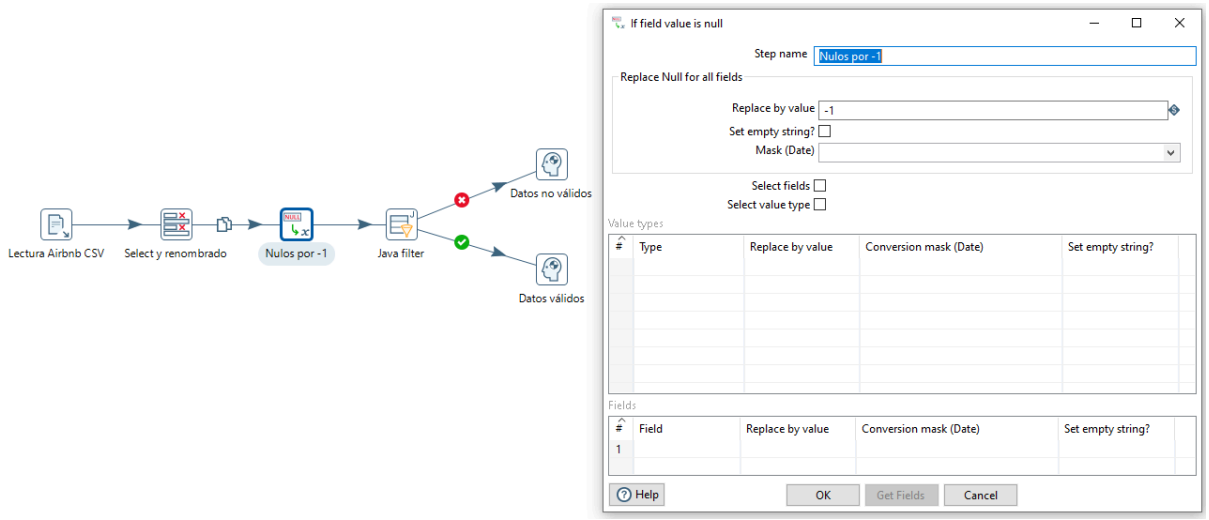
Si el filtrado fuese con condiciones más complejas, en ocasiones es más sencillo utilizar el paso *Java filter* (de la categoría *Flow*), el cual utilizando la notación de Java, podemos indicar la condición a cumplir. Por ejemplo, vamos filtrar los de más de 3 dormitorios o al menos 4 huéspedes, y que su precio sea inferior a 200$ → `(dormitorios>3 || huespedes>=4) && precio <200`:



Caso de Uso 3 - Filtrado Java

Para comprobar su funcionamiento, vamos a añadir un par de pasos *dummy/transformación simulada* (no realizan nada, pero sirven para finalizar tareas). Al ejecutarlo, veremos que nos da un error. Si algún dato es nulo, el filtrado Java provocará un error de transformación.

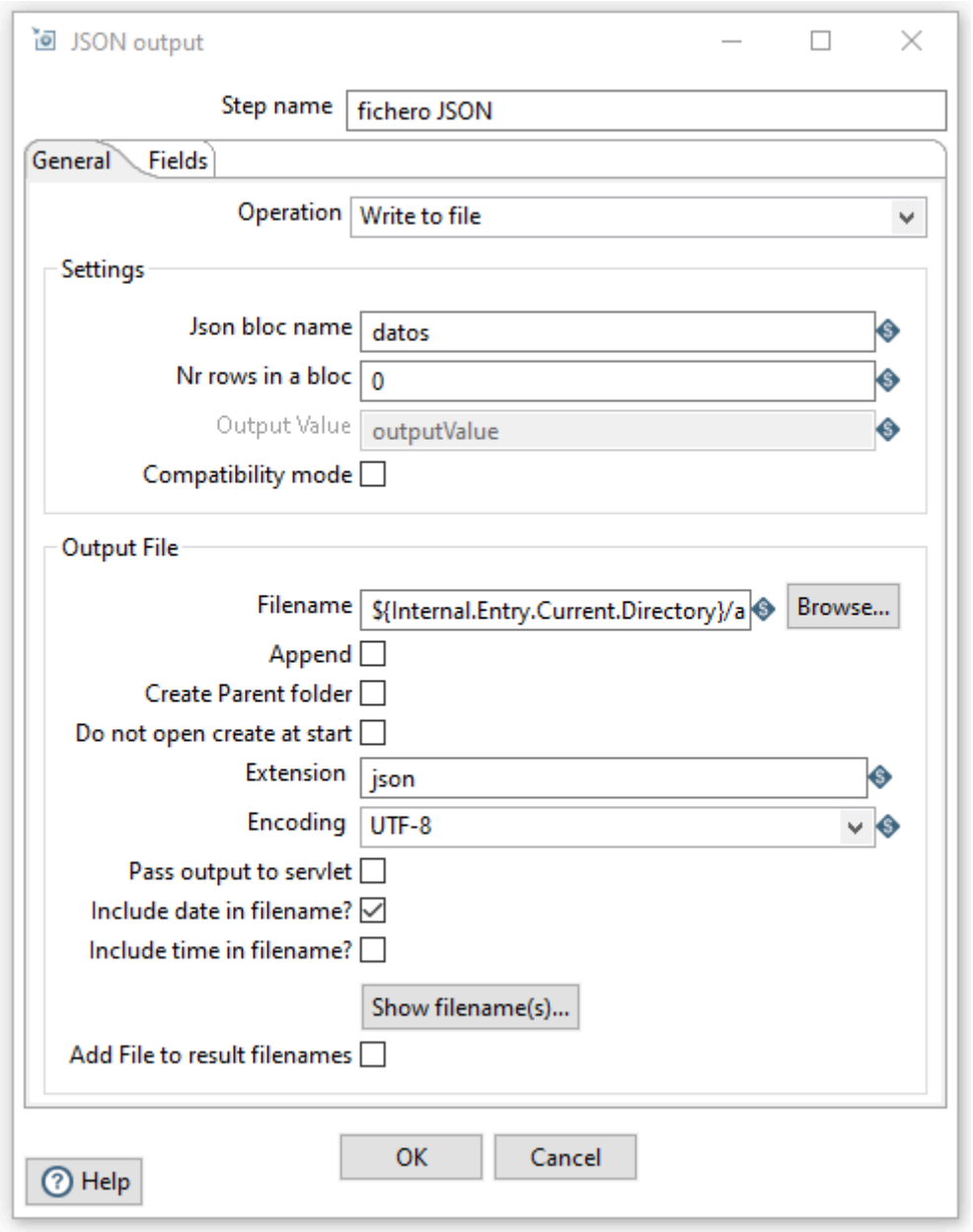
Una posibilidad es que introduzcamos un paso de la categoría *Utility* denominado *If value is null*. Con este paso, podemos indicar el valor a tomar a todos los campos o hacerlo de forma concreta en los campos que queramos. En nuestro caso, vamos a indicar que cambie todos los nulos por `-1`.



Caso de Uso 3 - Cambiando nulos por -1

Debemos tener en cuenta que como ahora podemos tener precios con `-1`, para evitar recogerlos en el filtrado Java, deberíamos modificarlo por `(dormitorios > 3 || huespedes >=4) && ( precio >= 0 && precio < 200)`.

Generando JSON



Caso de Uso 3 - Configuración JSON

Finalmente queremos almacenar los datos que cumplen el filtro en un fichero JSON.

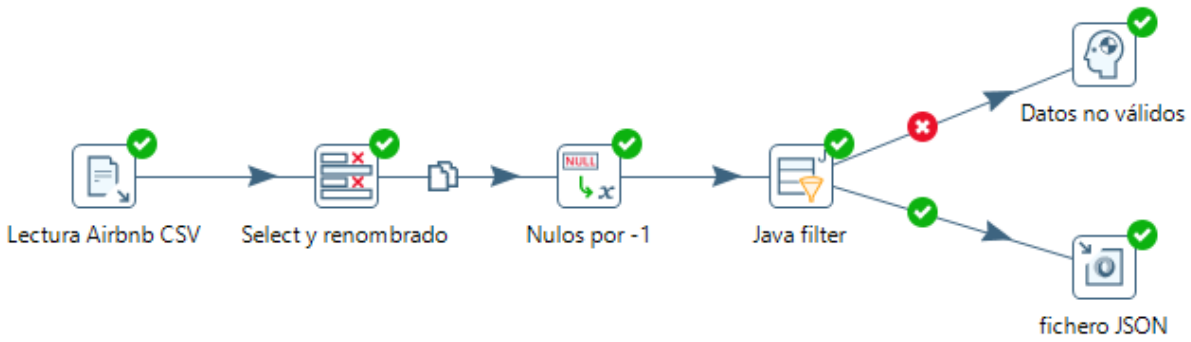
Para ello, sustituimos el *dummy* del camino exitoso por un paso *JSON output*. Tras seleccionar los campos que nos interesan, configuraremos:

- *Filename*: La ruta y el nombre del archivo
- *Json bloc name*: nombre de la propiedad que contendrá un objeto o un array de objetos con los datos.
- *Nr rows in a bloc*: Cantidad de datos del archivo. Si ponemos 0, coloca todos los datos en el mismo fichero. Si ponemos 1, generará un fichero por cada registro.

En la imagen que tenemos a la derecha puedes comprobar los valores introducidos.

Resultado final

En la siguiente imagen puedes visualizar la transformación completa:



Caso de Uso 3 - Transformación final