

Ejercicio 2a: Creación de la Transformation (CSV a XML)

1. **Lectura del CSV con Codificación:**
 - Arrastra el Step **CSV file input**.
 - Al editarlo, además de seleccionar la ruta del archivo, es crucial **especificar el "File encoding"** (codificación) en el que está guardado el fichero para asegurar que los datos se recuperen correctamente. Usa el botón "Get Fields" para validar la carga. **UTF-8**
 - **Pulsar en traer campos.**
2. **Lógica en JavaScript: SCRIPTING**
 - Añade **Valor JavaScript Modificado** y conéctalo mediante un **Hop** (flecha).
 - Escribe el script para generar la variable msg y asegúrate de añadirla en el listado de variables de salida inferior.
 - `var msg = '¡Hola ' + nombre + '!';`
 - Obtener variables.
3. **Salida XML:**
 - Añade **XML Output**.
 - Nombre del fichero `.csv`.
 - En Campo: obtener y borrar nombre y Apellidos.

Ejercicio 2b: Ejecución por Línea de Comandos

- Para ejecutar el archivo `.ktr` diseñado sin usar la interfaz gráfica, se utiliza la utilidad **Pan**.
- Desde la terminal:

`cd` (dirección de pentaho)

`.\pan.bat -file` (dirección del archivo)

Ejercicio 2c: Flexibilidad y Automatización (Jobs)

1. **Uso de Variables:** Crea una transformación que use

Información de sistema:

inputFile - > Argumento de comando de línea 1.

Filtrar filas:

SI inputFile IS NOT NULL

Si Verdadero: Asignar valor variable. Si Falso: Abortar.

Asignar valor a variables:

Taer campos.

2. **Configuración del Job.**

Cambiar en el programa 2a los ficheros CSV input y XLM output por variables: \${INPUTFILE}.

Crear un job:

- **Start.**
- **Transformación 1. Cargue el archivo transformación ejercicio 2c.**
- **Check if fields exists.**
 - o **Si: Transformación 2. Cargue el archivo transformación ejercicio 2a.**
 - o **No: Abort.**

Para ejecutar, meter en parámetros de ejecución la ruta completa del .csv.

Ejercicio 2d: Ejecución por Línea de Comandos

- Para ejecutar el flujo completo de trabajo o Job (.kjb), se utiliza la utilidad **Kitchen**.

.\Kitchen.bat -file (dirección del archivo) (dirección del csv)

Valores de Script

Nombre de paso Valor JavaScript modificado

JavaScript functions:

- Transform Scripts
- Transform Constants
- Transform Functions
- Input fields
 - nombre
 - apellido
- Output fields
 - Please use the 'Replace value 'Fieldname' or 'Rename to'

JavaScript:

```
Script 1
//Script here
var msg = '¡Hola ' + nombre + '!';
```

Compatibilidad mode? Optimization level 9

Campos

#	Nombre de campo	Renombar a	Tipo	Longitud	Precisión	Replace value 'Fieldname' or 'Rename to'
1	msg		String			N

Buttons: Help, Vale, Cancelar, Obtener Variables, Probar script

Salida de fichero XML

Nombre paso XML output

Fichero | Contenido | Campos

Nombre fichero: C:\Users\OscarC\Downloads\ejercicio2a.xml Examinar...

Do not create file at start

Pass output to servlet

Extensión: xml

Incluir stepnr en nombre fichero?
 Incluir fecha en nombre fichero?
 Incluir hora en nombre fichero?
 Specify Date time format
 Date time format:

Mostrar nombre fichero(s)...

Add filenames to result

Buttons: Help, Vale, Cancelar

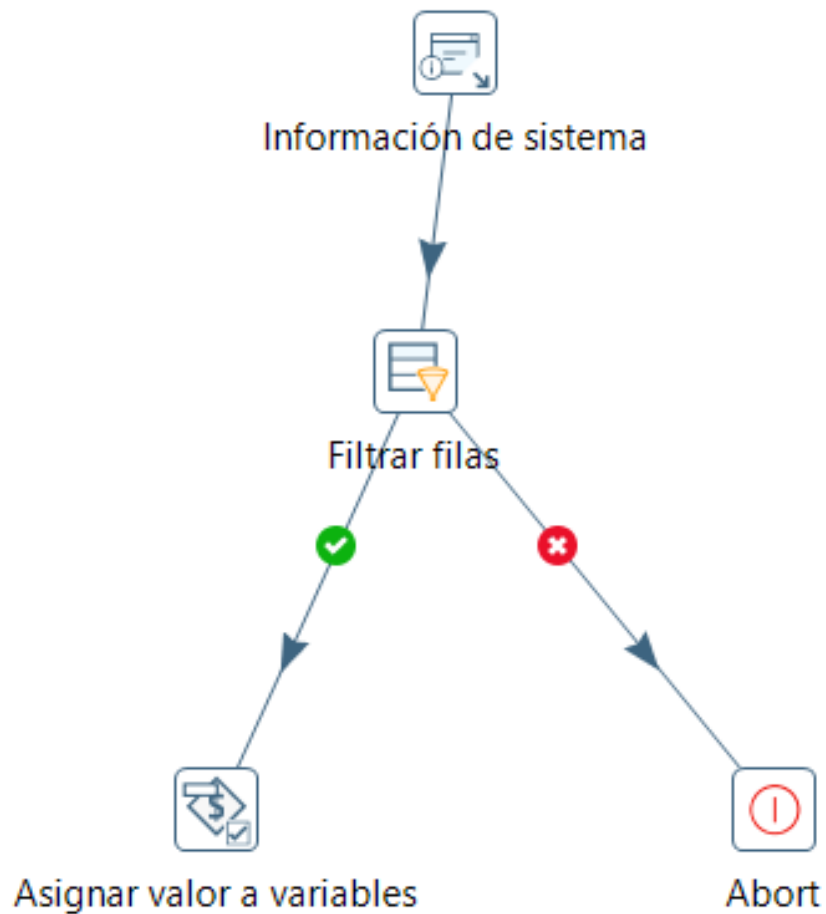
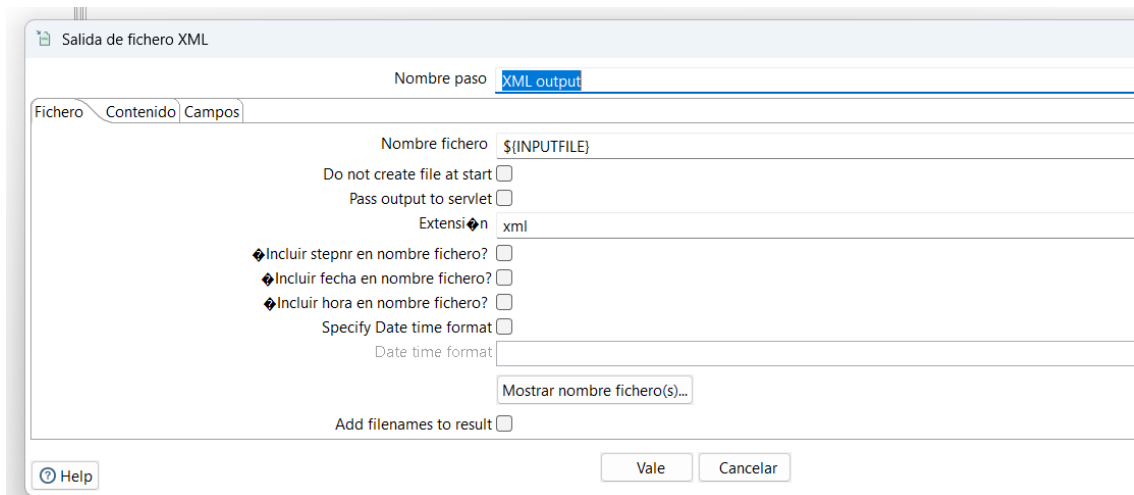
Salida de fichero XML

Nombre paso XML output

Fichero | Contenido | Campos

#	Nombre campo	Nombre elemento	Content type	Tipo	Formato	Longitud	Precisión	Moneda	Decimal	Grupo	Nulo
1	msg		Element	String							

Buttons: Obtener Campos, Anchura Mínima, Help, Vale, Cancelar



Información de sistema

Nombre paso: Información de sistema

Campos:

#	Nombre	Tipo
1	inputFile	argumento de línea de comando 1

Asignar valor a variables Abort

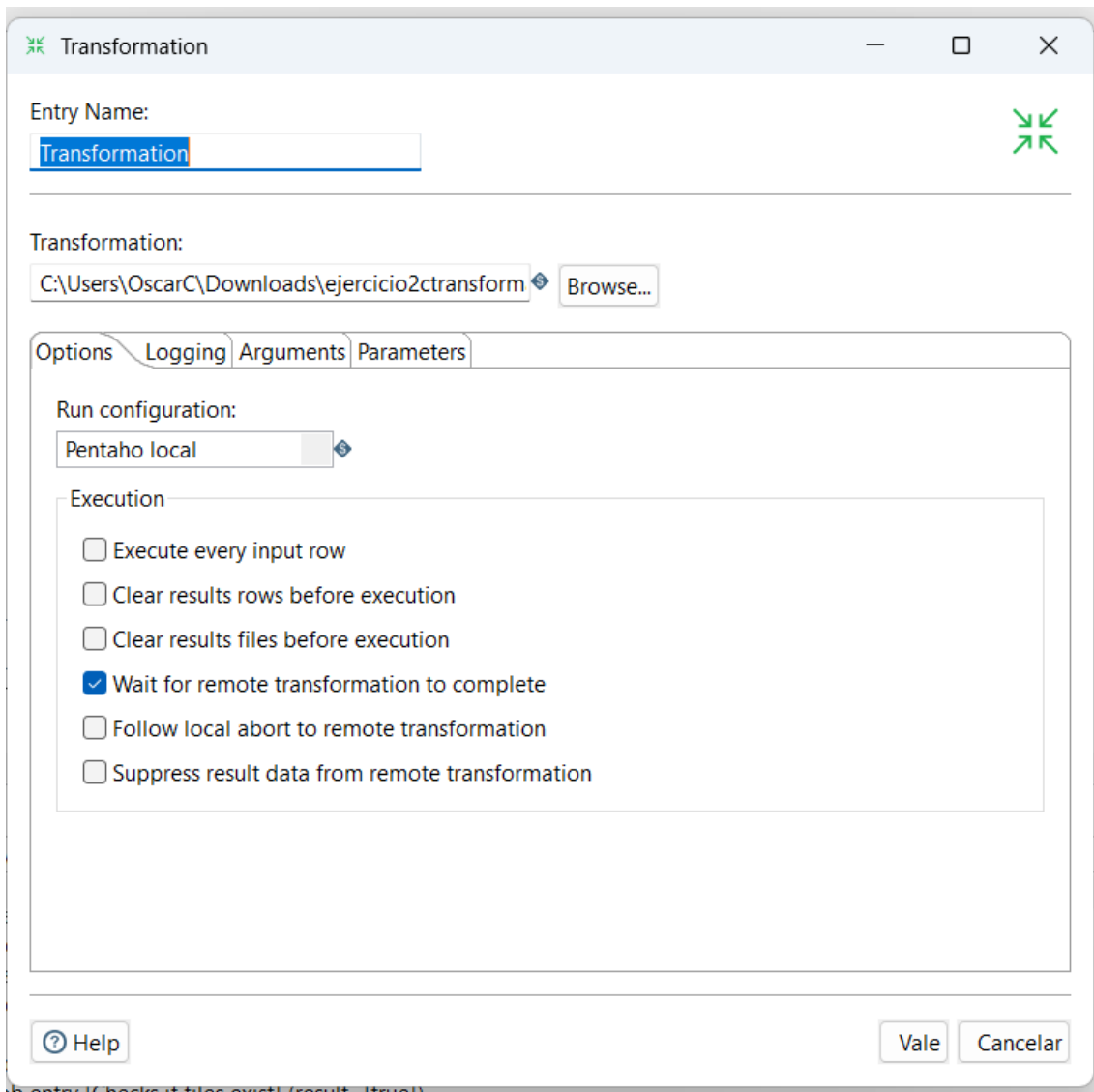
Filtrar filas

Nombre de paso: Filtrar filas

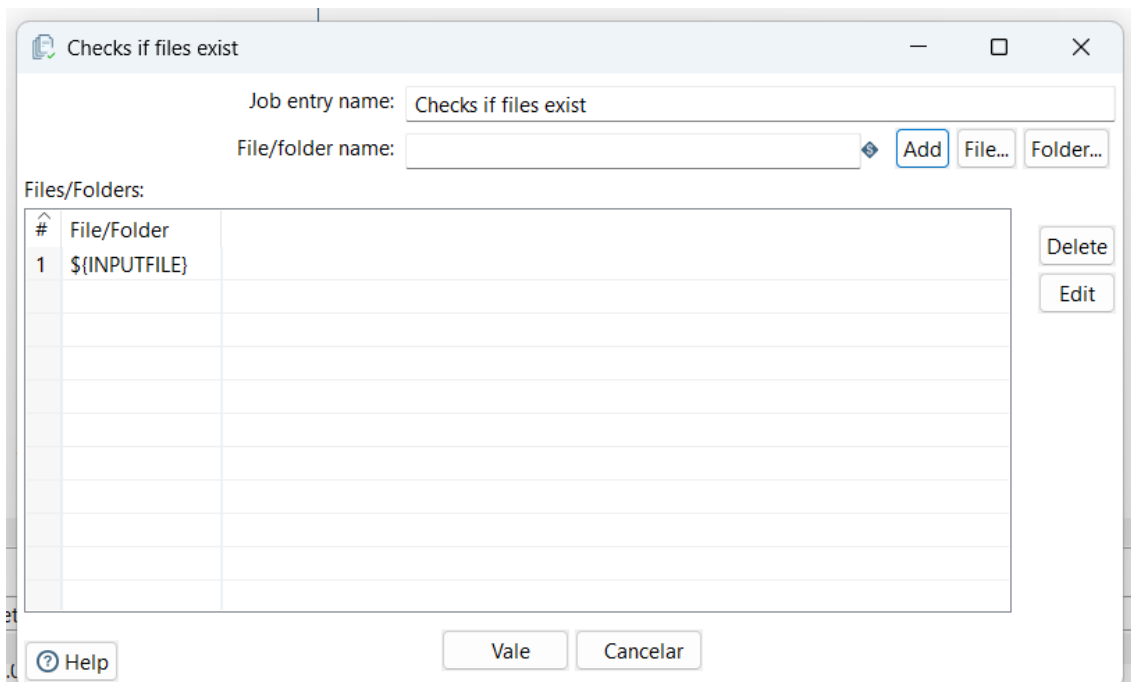
Enviar 'verdadero' a: Asignar valor a variables

Enviar 'falso' a paso: Abort

La condición:



Job entry (Checks if files exist) (result=true)



Job entry (Checks if files exist) (result=true)

