

HOP

Abraham Trashorras Rivas

Importar trabajo de Kettle

Para este apartado importo los trabajos creados en la Practica Individual 3, una carpeta **etl** que contiene otras dos, la carpeta **trabajo** con los archivos:

- CargarDimensiones.ktr
- CargarFinanzas.ktr
- CargarPeliculas.ktr
- CargarSatisfaccion.ktr
- EtlPeliculas.kjb
- ProduccionCine.xml

Y la carpeta **entrada** con los archivos *.xml* y *.csv* que conforman los datos.

Lo primero que puedo observar es que se me crean 3 carpetas, una de **trabajo** donde los archivos *.ktr* y *.kjb* se han transformado a *.hpl* y *.hwl* respectivamente, otra de **entrada** que se mantiene igual y una de **metadata/rdbms/** que contiene *cine.json*, con datos sobre la conexión a la base de datos POSTGRES:

```
{
  "rdbms": {
    "POSTGRESQL": {
      "databaseName": "cine",
      "pluginId": "POSTGRESQL",
      "accessType": 0,
      "hostname": "localhost",
      "password": "Encrypted
456e6372797074656420326265393861666338366163739353934dffd9ef80bc4c5d0f31aad22d896f7d8",
      "pluginName": "PostgreSQL",
      "port": "5432",
      "attributes": {
        "PORT_NUMBER": "5432"
      },
      "username": "alumnogreibd"
    }
  },
  "name": "cine"
}
```

A mayores se han creado dos archivos, *connections.csv* que incluye los trabajos de Kettle y la BBDD a la que se conectan:

```
file:///home/alumnogreibd/IN/etl/trabajo/CargarDimensiones.ktr,cine
file:///home/alumnogreibd/IN/etl/trabajo/CargarFinanzas.ktr,cine
file:///home/alumnogreibd/IN/etl/trabajo/CargarPeliculas.ktr,cine
file:///home/alumnogreibd/IN/etl/trabajo/CargarSatisfaccion.ktr,cine
```

y *project-config.json* que contiene los datos del propio proyecto de HOP:

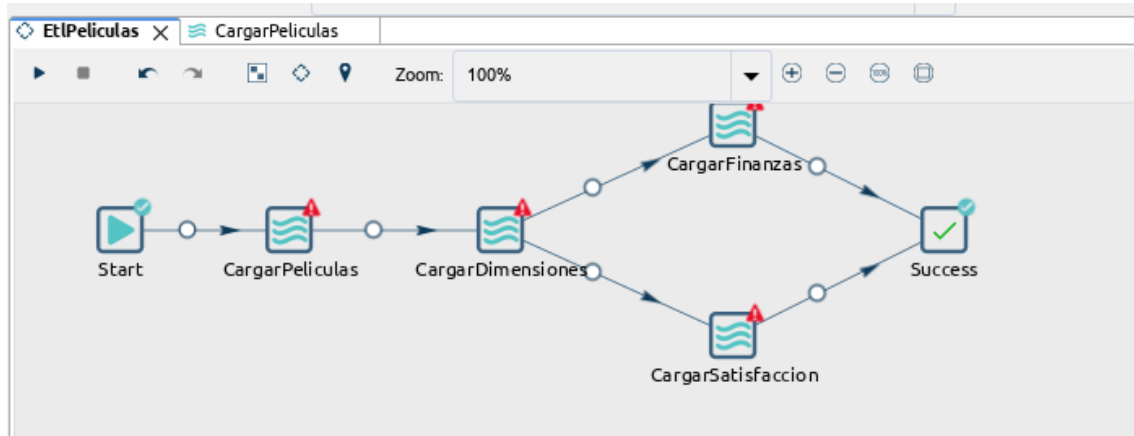
```
{
  "metadataBaseFolder" : "${PROJECT_HOME}/metadata",
  "unitTestsBasePath" : "${PROJECT_HOME}",
}
```

```

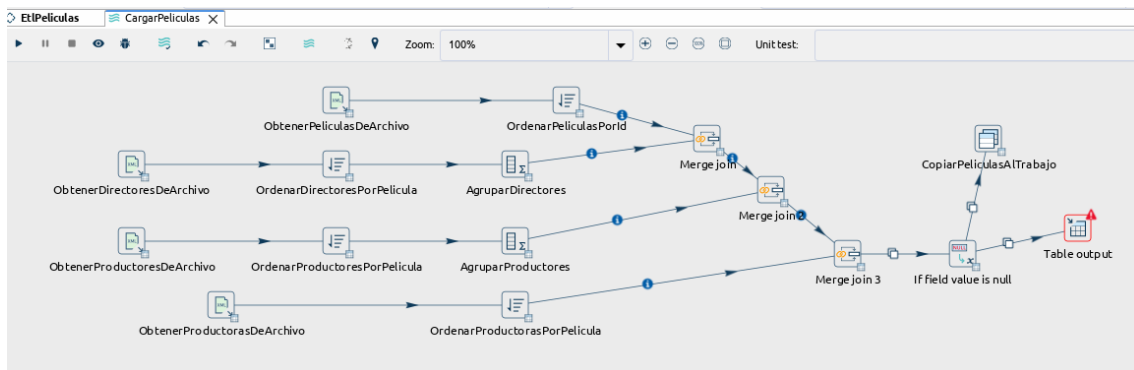
"dataSetsCsvFolder" : "${PROJECT_HOME}/datasets",
"enforcingExecutionInHome" : true,
"config" : {
  "variables" : []
}
}

```

Al intentar ejecutar EtlPeliculas.hwf me da error en todos los trabajos.



No ha importado bien la conexión a la base de datos y necesito arreglarla a mano. Una vez arreglada la conexión, pruebo a lanzar CargarPeliculas.hpl y me da error por clave duplicada ya que estamos intentando meter los mismos datos que ya están guardados.



2023/12/05 15:23:33 - Table output.0 - Error updating batch

2023/12/05 15:23:33 - Table output.0 - Batch entry 0 INSERT INTO etl.pelicula_productora (pelicula, fecha_emision, id_productora, text_id_director, text_id_productor) VALUES (12, '2003-05-30 00:00:00+02'::date, 3, '7', '9') was aborted: ERROR: llave duplicada viola restricción de unicidad «pelicula_productora_pkey»

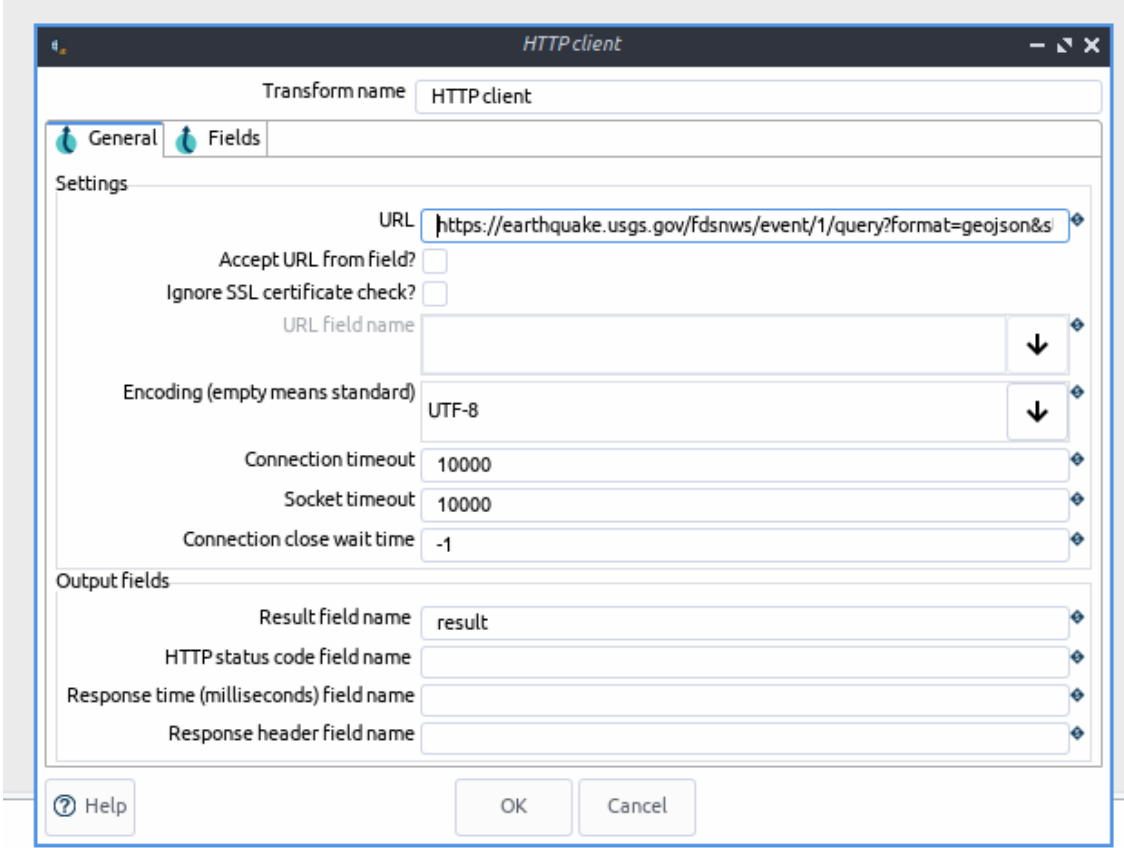
2023/12/05 15:23:33 - Table output.0 - Detail: Ya existe la llave (pelicula, id_productora)=(12, 3). Call getNextException to see other errors in the batch.

2023/12/05 15:23:33 - Table output.0 -

Webs

Para llamar a la api de terremotos uso la url

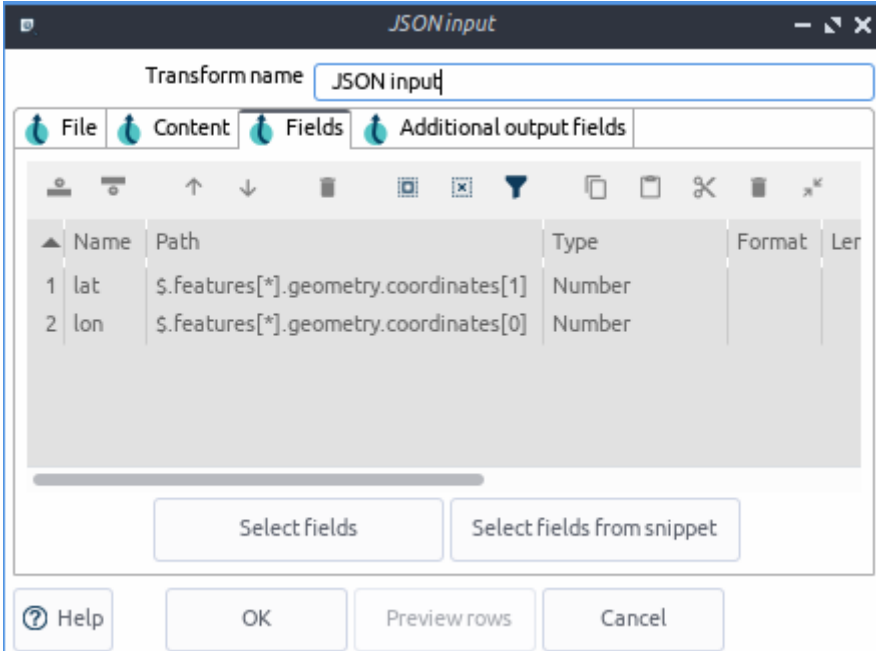
<https://earthquake.usgs.gov/fdsnws/event/1/query?format=geojson&starttime=2023-12-04&endtime=2023-12-05&minmagnitude=5> en un elemento HTTP client:



The screenshot shows the 'HTTP client' configuration window. The 'Transform name' is 'HTTP client'. The 'General' tab is selected. The 'Settings' section includes a 'URL' field with the value 'https://earthquake.usgs.gov/fdsnws/event/1/query?format=geojson&starttime=2023-12-04&endtime=2023-12-05&minmagnitude=5'. Below the URL are checkboxes for 'Accept URL from field?' and 'Ignore SSL certificate check?'. The 'URL field name' is empty. The 'Encoding' is set to 'UTF-8'. The 'Connection timeout' is '10000', the 'Socket timeout' is '10000', and the 'Connection close wait time' is '-1'. The 'Output fields' section has 'Result field name' set to 'result', and the other fields are empty. At the bottom are 'Help', 'OK', and 'Cancel' buttons.

Field	Value
URL	https://earthquake.usgs.gov/fdsnws/event/1/query?format=geojson&starttime=2023-12-04&endtime=2023-12-05&minmagnitude=5
Accept URL from field?	<input type="checkbox"/>
Ignore SSL certificate check?	<input type="checkbox"/>
URL field name	
Encoding	UTF-8
Connection timeout	10000
Socket timeout	10000
Connection close wait time	-1
Result field name	result
HTTP status code field name	
Response time (milliseconds) field name	
Response header field name	

Con el cual alimento a un JSON input donde proceso los datos y extraigo las coordenadas



The screenshot shows the 'JSON input' configuration window. The 'Transform name' is 'JSON input'. The 'Fields' tab is selected. The 'Name' and 'Path' columns are visible. The 'Name' column has two rows: '1 lat' and '2 lon'. The 'Path' column has two rows: '\$.features[*].geometry.coordinates[1]' and '\$.features[*].geometry.coordinates[0]'. The 'Type' column has two rows: 'Number' and 'Number'. The 'Format' and 'Ler' columns are empty. At the bottom are 'Help', 'OK', 'Preview rows', and 'Cancel' buttons.

Name	Path	Type	Format	Ler
1 lat	\$.features[*].geometry.coordinates[1]	Number		
2 lon	\$.features[*].geometry.coordinates[0]	Number		

Ahora tenemos que repetir lo mismo otra vez, un HTTP Client para la API de temperaturas:

The screenshot shows the 'HTTP client' configuration window with the 'General' tab selected. The 'Transform name' is 'HTTP client 2'. The 'Settings' section includes a 'URL' field with the value 'https://api.open-meteo.com/v1/forecast?daily=temperature_2m_min', and checkboxes for 'Accept URL from field?' and 'Ignore SSL certificate check?'. Below these are fields for 'URL field name', 'Encoding' (set to 'UTF-8'), 'Connection timeout' (10000), 'Socket timeout' (10000), and 'Connection close wait time' (-1). The 'Output fields' section has fields for 'Result field name' (set to 'temperatures'), 'HTTP status code field name', 'Response time (milliseconds) field name', and 'Response header field name'. At the bottom are 'Help', 'OK', and 'Cancel' buttons.

Transform name: HTTP client 2

General Fields

Settings

URL: https://api.open-meteo.com/v1/forecast?daily=temperature_2m_min

Accept URL from field? ☐

Ignore SSL certificate check? ☐

URL field name: [empty]

Encoding (empty means standard): UTF-8

Connection timeout: 10000

Socket timeout: 10000

Connection close wait time: -1

Output fields

Result field name: temperatures

HTTP status code field name: [empty]

Response time (milliseconds) field name: [empty]

Response header field name: [empty]

Help OK Cancel

The screenshot shows the 'HTTP client' configuration window with the 'Fields' tab selected. The 'Transform name' is 'HTTP client 2'. The 'Parameters' section has a table with two rows: '1 lat latitude' and '2 lon longitude'. The 'Custom HTTP Headers' section has a table with one row: '1 [empty] [empty]'. Both sections have a 'Get Fields' button. At the bottom are 'Help', 'OK', and 'Cancel' buttons.

Transform name: HTTP client 2

General Fields

Parameters :

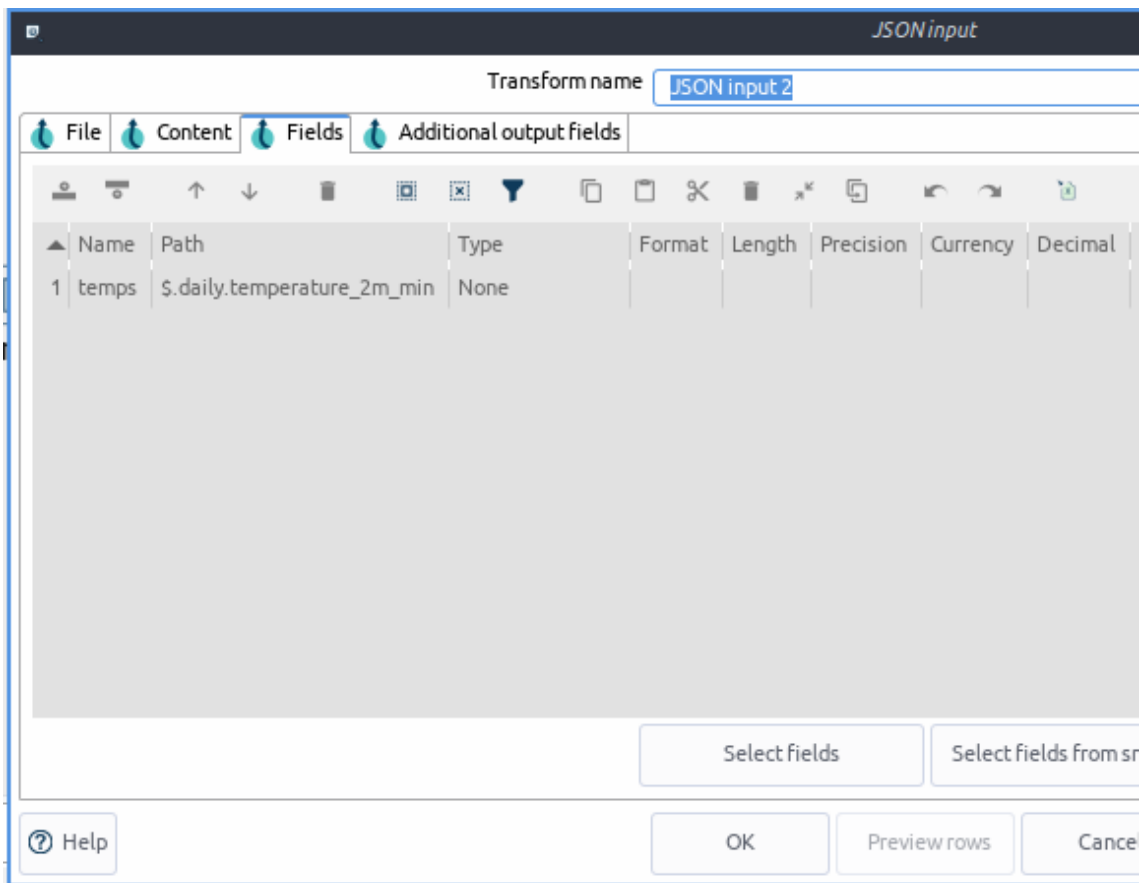
Name	Parameter
1 lat	latitude
2 lon	longitude

Custom HTTP Headers :

Field	Header
1	

Help OK Cancel

Y un JSON input para transformar la salida:



Solo quedaría hacer el mínimo y listo. Me queda así el trabajo:

