

APACHE HOP

Grimaldo Oliveira

O que é o APACHE HOP

1. Abreviação de Hop Orchestration Platform, é uma plataforma de orquestração de dados e engenharia de dados que visa facilitar todos os aspectos da orquestração de dados e metadados.
2. A instalação padrão do Hop vem com cerca de 400 plugins ou componentes.
3. São criados fluxos de trabalho e pipelines em um ambiente de desenvolvimento visual chamado Hop Gui.
4. Com o APACHE HOP é possível combinar, enriquecer, limpar e de muitas outras maneiras manipular dados.

Sites Importantes

APACHE HOP

<https://hop.apache.org/>

Documentação

<https://hop.apache.org/manual/latest/getting-started/>

APACHE HOP Notícias

<https://hop.apache.org/blog/>

Arquitetura

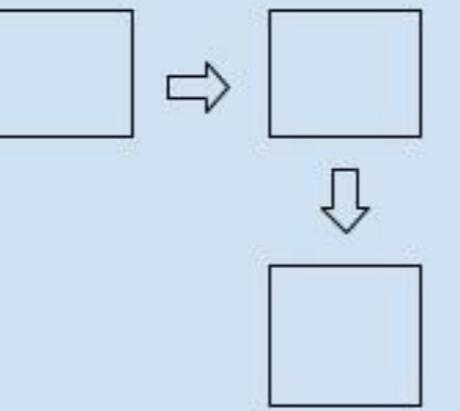
Como funciona a estrutura do APACHE HOP.

JAVA

APACHE HOP

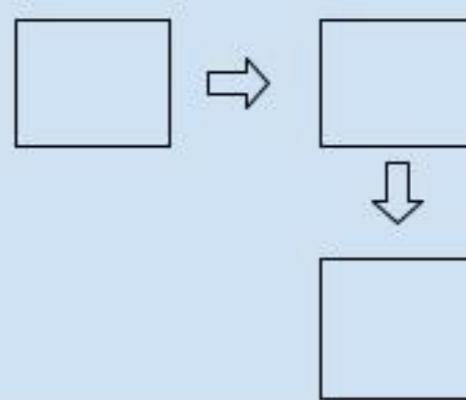
PIPELINE

TRANSFORMAÇÕES



WORKFLOW

FLUXO DE TRABALHO



APACHE HOP

Uma plataforma aberta para execução de pipelines e fluxos de trabalho.

APACHE HOP

Uma plataforma aberta destinada a criação de estruturas de ingestão de dados, tratamento de dados e fluxos de dados em diversas plataformas como bancos de dados relacionais, arquivos, bancos de dados NoSQL e dados na nuvem.

Tem por característica:

BASE DE DADOS: carga de grandes bases de dados, independente de plataforma seja on-premise ou cloud.

DATA WAREHOUSE: possui componentes ou plugins que facilitam a construção de projetos de Data Warehouse como Dimensões, controle de versionamento, criação de chaves artificiais, etc.

INTEGRAÇÃO: permite a integração e construção de projetos para popular bases heterogêneas, persistência em diversos banco de dados.

TRATAMENTO DE DADOS: perfilização e tratamento em dados de forma personalizada.

PROJETO

Faz todo o gerenciamento e
armazenamento dos metadados
criados no APACHE HOP.

PROJETO

Hop Projects é um agrupamento conceitual de configurações, variáveis, objetos metadados , fluxos de trabalho e pipelines. Pode haver mais de um ambiente dentro de um mesmo projeto no APACHE HOP.

DESENVOLVIMENTO



PRODUÇÃO



TESTE

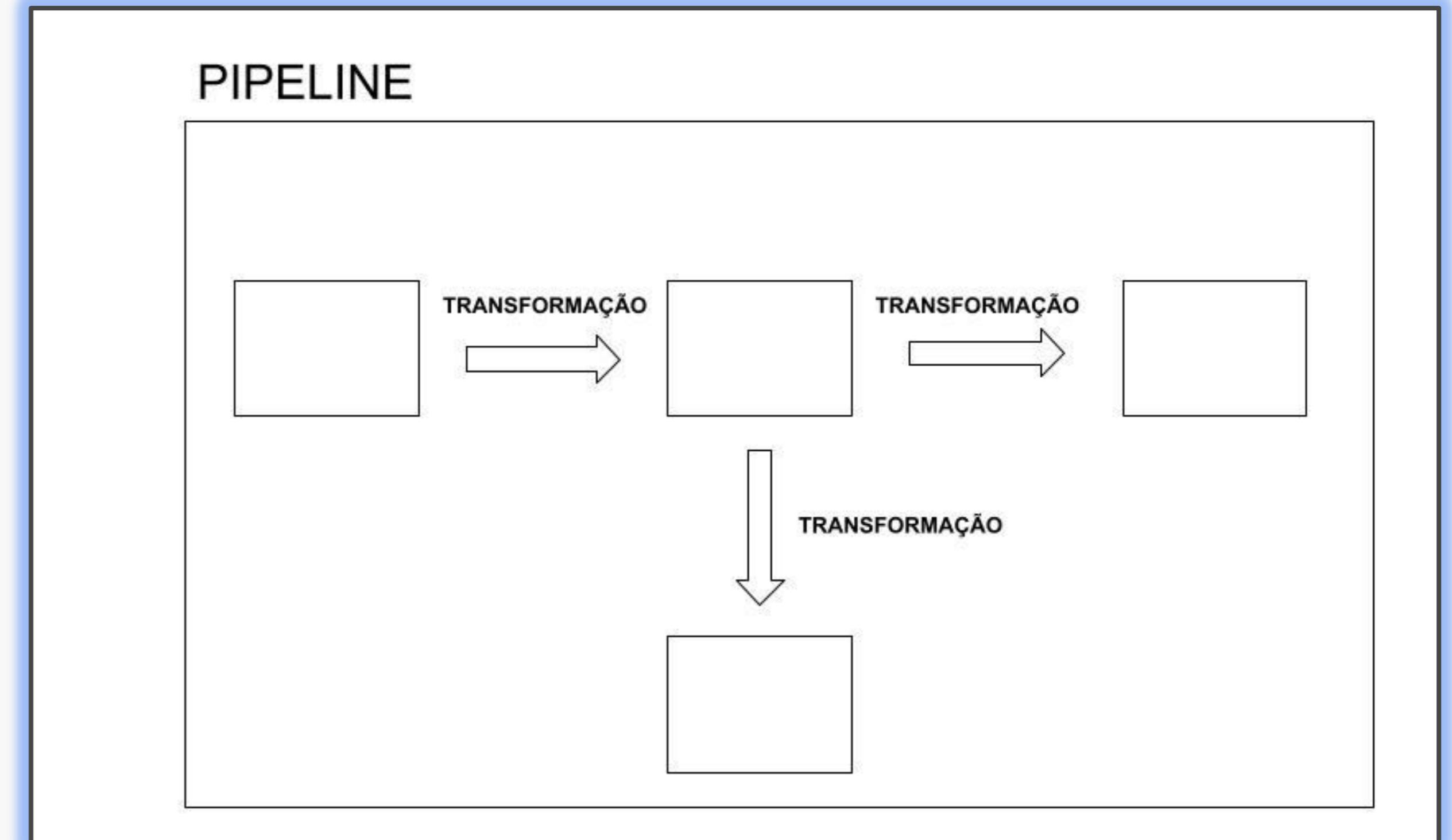


PIPELINE

São os componentes orquestrados de forma a resolver uma determinada operação.

PIPELINE

É uma composição de links e ações que são realizadas por quem desenvolve o pipeline. É aqui que os componentes são interligados e permitem a execução de uma determinada operação como limpeza, ajuste de campos, retirada de registros, gravação em banco de dados, etc.



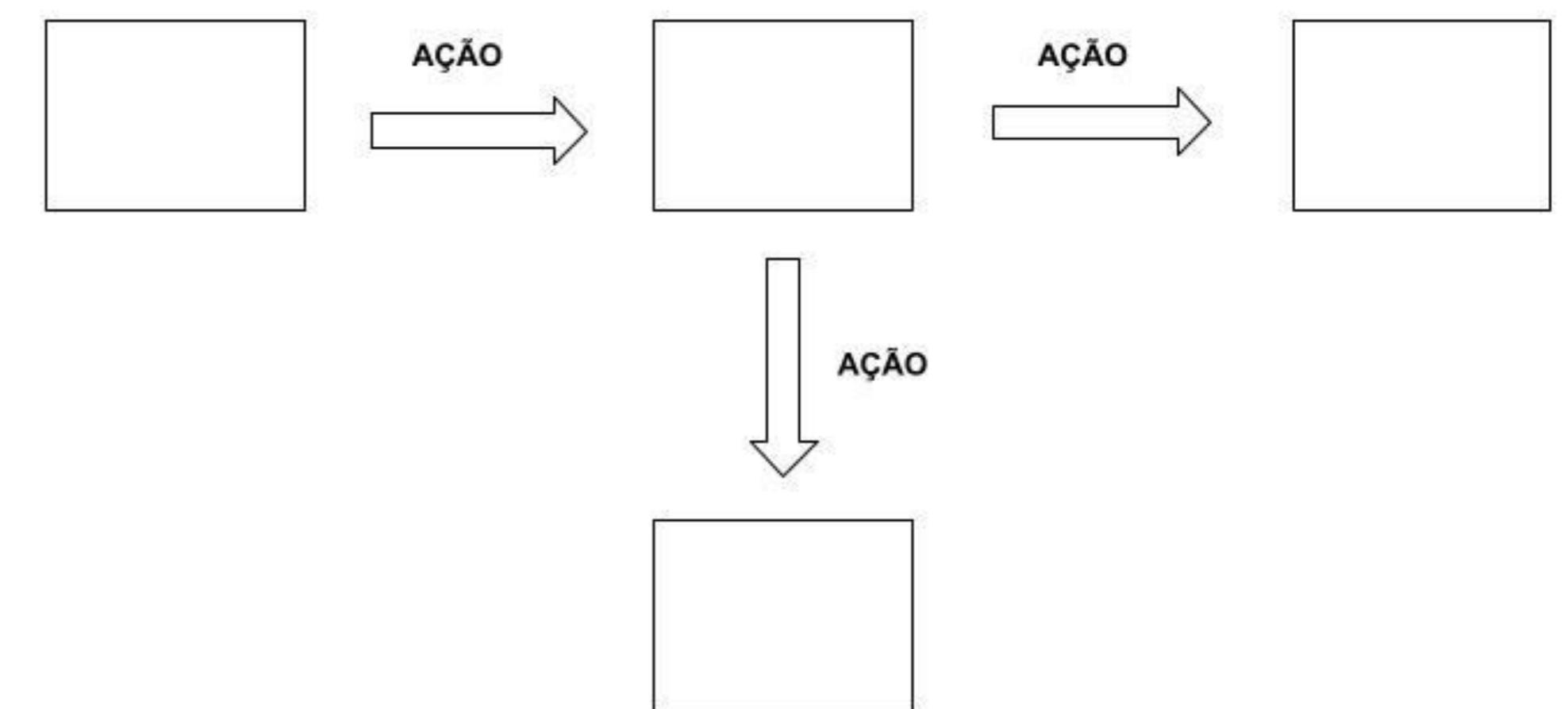
WORKFLOW

Determina uma sequência de operações .

WORKFLOW

É um orquestrador da sequência das operações criadas, também conhecido como fluxo de trabalho. Uma vez criados os pipelines, você poderá encadeá-los de forma a executá-los em ordem sequencial. Uma orquestração típica é verificar se o pipeline foi bem sucedido ou não.

WORKFLOW



HOP GUI

É local de trabalho dentro do APACHE HOP.

INTERFACE

Um espaço de trabalho, um ambiente para acessar todos os seus projetos. O espaço de trabalho organiza objetos, e fornece acesso a dados e recursos computacionais, pipelines e workflow.



HOP GUI - MENUS

Desatcando o local de trabalho dentro do APACHE HOP.

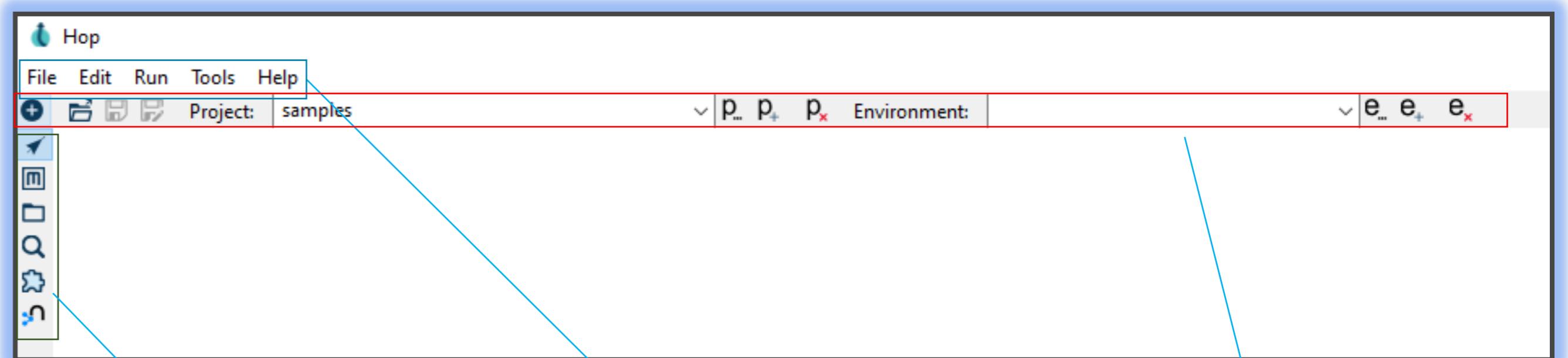
MENUS

Para o nosso trabalho temos três(03) menus principais .

Menu Barras: opções para o gerenciamento de pipelines e fluxos de trabalho.

Menu Principal: gerenciamento dos projetos, ambientes , etc.

Menu Perspectivas: Alterna entre opções nos outros menus com visual ampliado.



MENU PERSPECTIVAS

MENU BARRAS

MENU PRINCIPAL

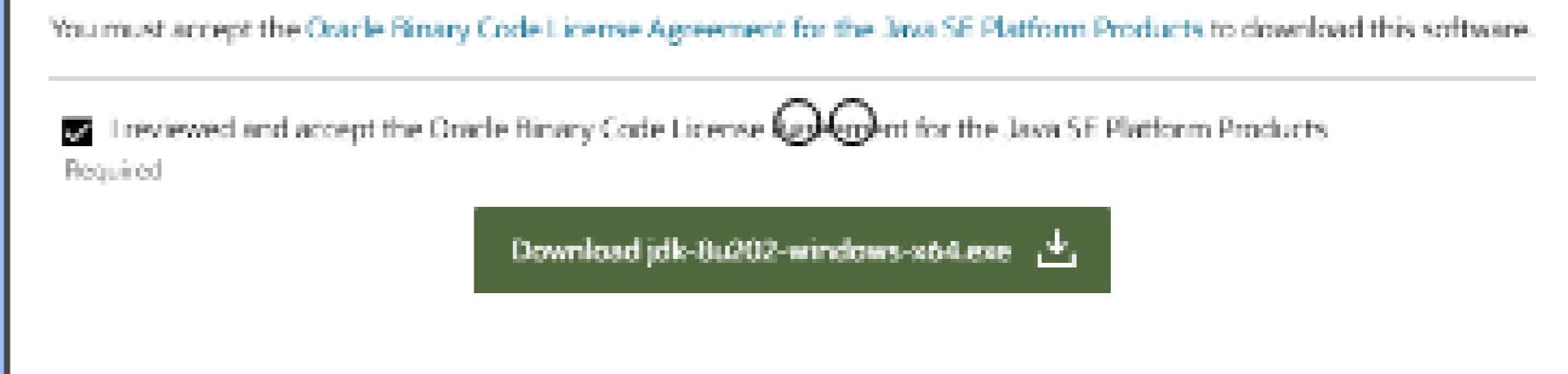
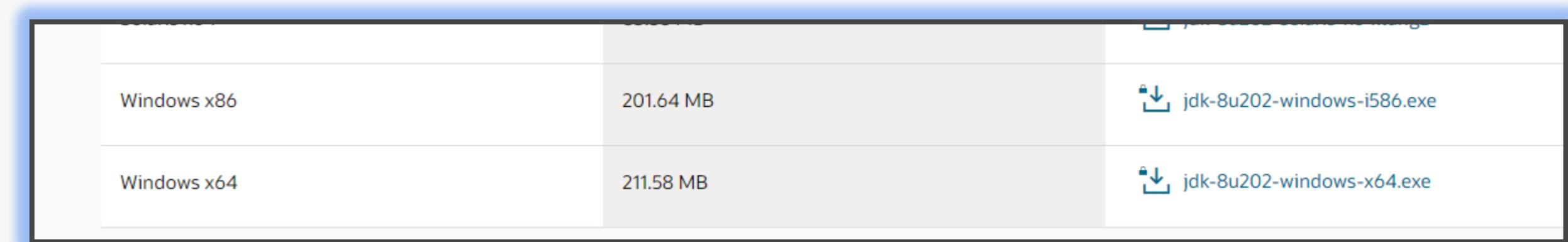
Instalação

Vamos nos preparar para a instalação do java jdk.

Instalação APACHE HOP

Vamos baixar o Java JDK e a aplicação APACHE HOP.

<https://www.oracle.com/java/technologies/javase/javase8-archive-downloads.html>



Instalação

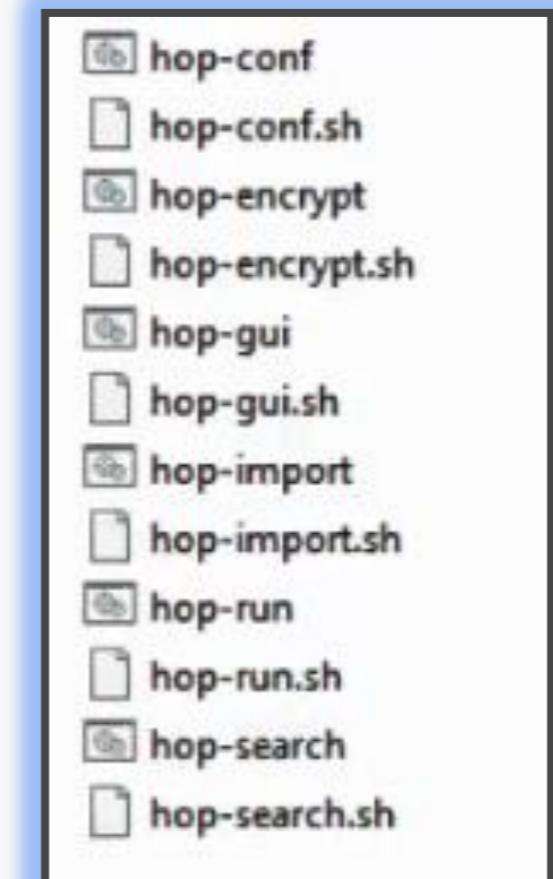
Vamos nos preparar para a instalação
do APACHE HOP.

Instalação APACHE HOP

Vamos baixar o Java JDK e a aplicação APACHE HOP.

<https://hop.apache.org/download/>

Version	Date	Description	Download Link	PGP Signature file of download	SHA512 Checksum file of download
1.0.0	October, 5th, 2021	Binaries	apache-hop-client-1.0.0-incubating.zip	asc	sha512
		Sources	apache-hop-1.0.0-incubating-src.tar.gz	asc	sha512
0.99	August, 5th, 2021	Binaries	apache-hop-client-0.99-incubating.zip	asc	sha512
		Sources	apache-hop-0.99-incubating-src.tar.gz	asc	sha512



Atenção Importante!

Vamos preparar e entender ajustes
para executar o APACHE HOP.

Preparação antes de executar o APACHE HOP

Preste atenção na configuração depois de instalados JAVA e APACHE HOP.

```
:NORMALSTART
REM set java primary is HOP_JAVA_HOME Fallback to JAVA_HOME or default java
if not "%HOP_JAVA_HOME%"==""
    set _HOP_JAVA="%HOP_JAVA_HOME%\bin\java"
) else if not "%JAVA_HOME%"==""
    set _HOP_JAVA="%JAVA_HOME%\bin\java"
) else (
    set _HOP_JAVA=java
)
```

DriverData	C:\Windows\System32\Drivers\DriverData
HOP_JAVA_HOME	C:\Java\jdk1.8
JAVA_HOME	C:\Java\jdk1.8\bin

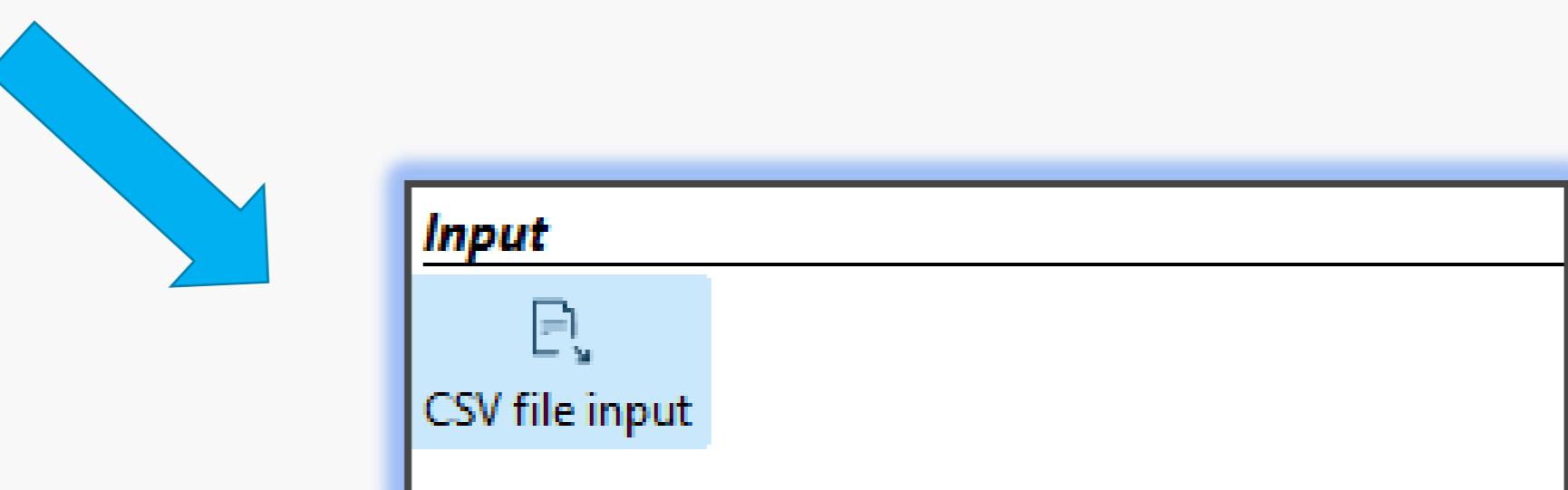
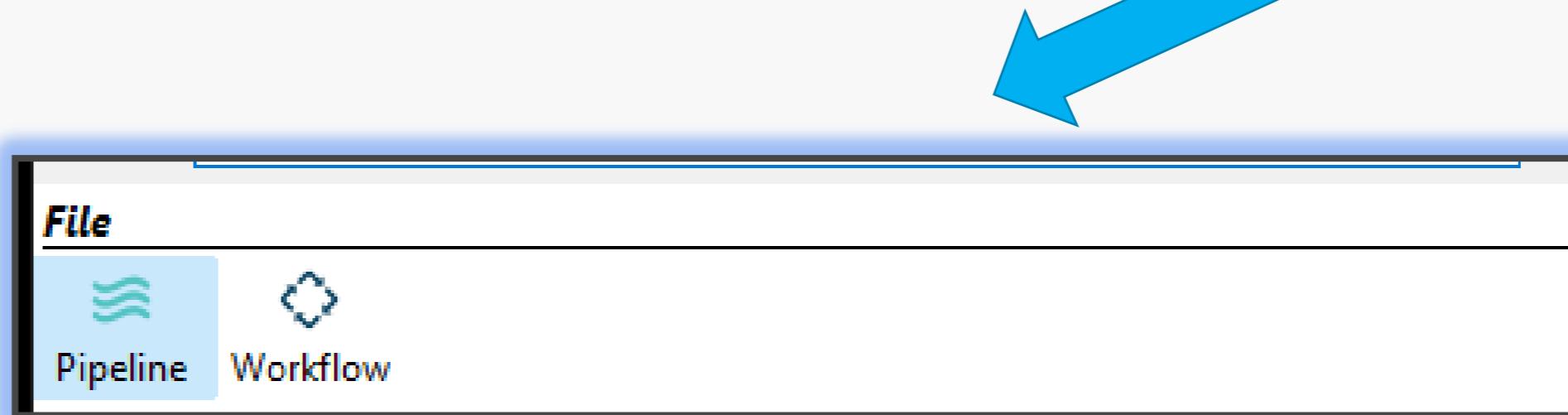
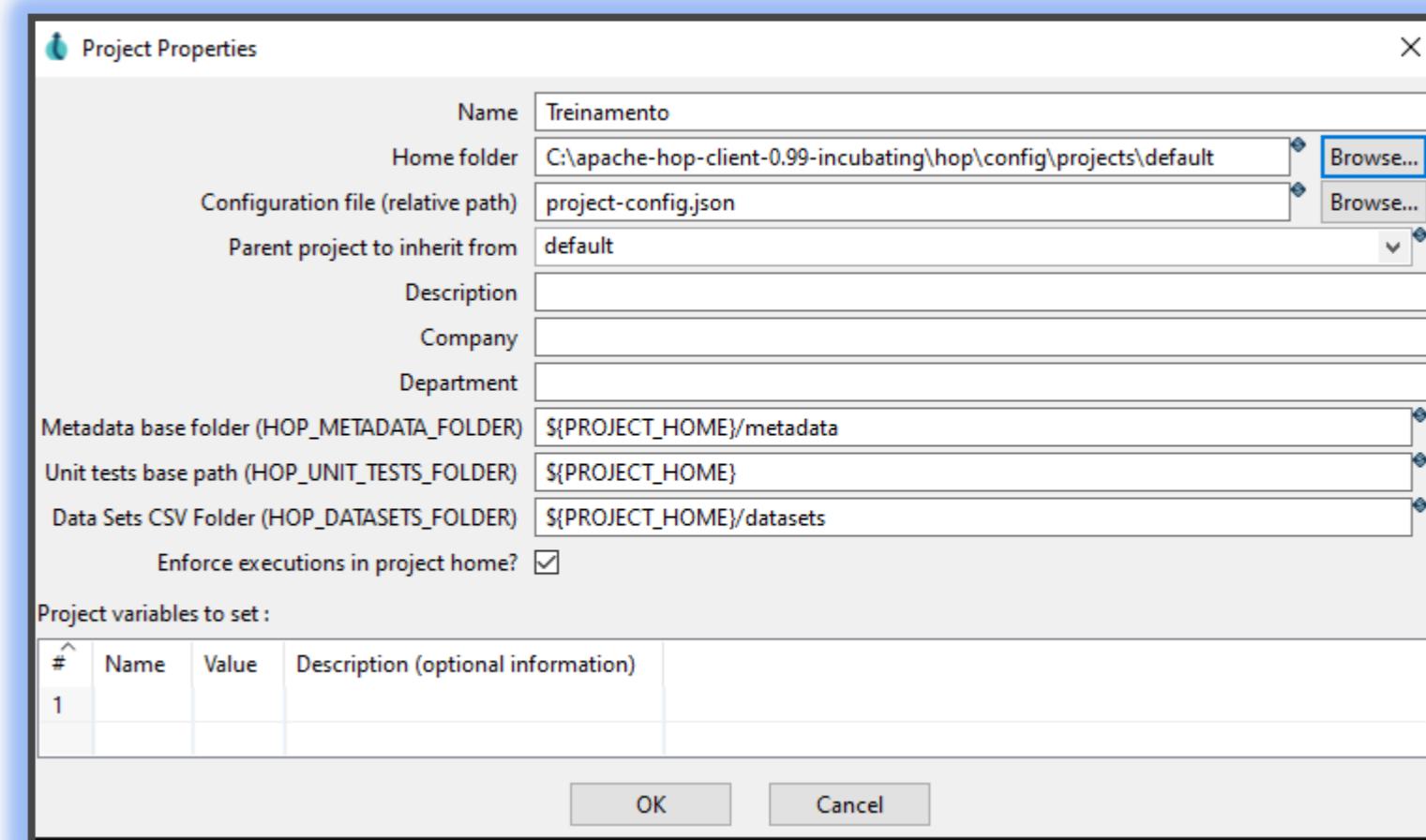
```
REM # Settings for all OSses
if "%HOP_OPTIONS%"=="" set HOP_OPTIONS=-Xmx2048m
```

Começando a trabalhar

Vamos criar um projeto.

Criando uma área de projetos

Precisamos informar, aonde estarão os metadados do projeto. É armazenado em um arquivo JSON.

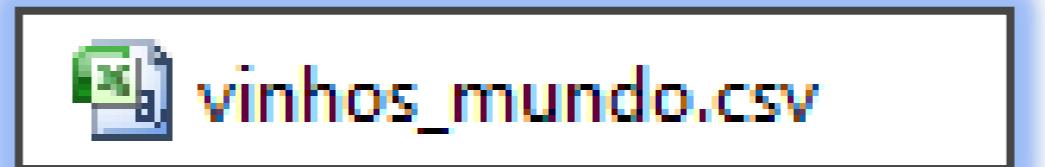


Começando a trabalhar

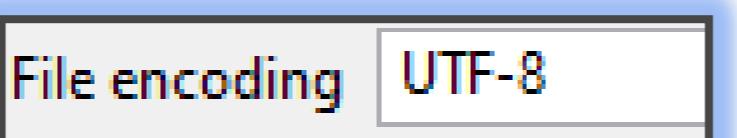
Vamos carregar os primeiros dados.

Carregando dados

Vamos carregar 129 mil registros em csv do arquivo **vinhos_mundo.csv**



The screenshot shows the configuration of a 'CSV file input' component. The 'Transform name' is 'CSV file input'. The 'Filename' is set to 'C:\Users\grima\Downloads\Hop\Dados\vinhos_mundo.csv'. The 'Delimiter' is a comma (,), 'Enclosure' is an empty string, and 'NIO buffer size' is 50000. 'Lazy conversion?' and 'Header row present?' are checked. The 'File encoding' is set to 'UTF-8'. Below the configuration, a preview table shows 14 columns: #, Name, Type, Format, Length, Precision, Currency, Decimal, Group, register (Integer), country (String), description (String), designation (String), points (Integer), price (Number), province (String), region_1 (String), region_2 (String), taster_name (String), taster_twitter_handle (String), title (String), variety (String), and winery (String). The preview table has 14 rows of sample data.



The screenshot shows a preview of the data with 1000 rows. The columns are 'register', 'country', and 'description'. The first few rows of data are:

#	register	country	description
1	0	Italy	Aromas include tropical fruit, broom, brimstone and dried herb. The palate isn't overly expressive
2	1	Portugal	This is ripe and fruity, a wine that is smooth while still structured. Firm tannins are filled out with j
3	2	US	Tart and snappy, the flavors of lime flesh and rind dominate. Some green pineapple pokes through
4	3	US	Pineapple rind, lemon pith and orange blossom start off the aromas. The palate is a bit more opu
5	4	US	Much like the regular bottling from 2012, this comes across as rather rough and tannic, with rusti
6	5	Spain	Blackberry and raspberry aromas show a typical Navarran whiff of green herbs and, in this case, h
7	6	Italy	Here's a bright, informal red that opens with aromas of candied berry, white pepper and savory h
8	7	France	This dry and restrained wine offers spice in profusion. Balanced with acidity and a firm texture, it'
9	8	Germany	Savory dried thyme notes accent sunnier flavors of preserved peach in this brisk, off-dry wine. It's
10	9	France	This has great depth of flavor with its fresh apple and pear fruits and touch of spice. It's off dry w
11	10	US	Soft, supple plum envelopes an oaky structure in this Cabernet, supported by 15% Merlot. Coffe
12	11	France	This is a dry wine, very spicy, with a tight, taut texture and strongly mineral character layered with
13	12	US	Slightly reduced, this wine offers a chalky, tannic backbone to an otherwise juicy explosion of ric
14	13	Italy	This is dominated by oak and oak-driven aromas that include roasted coffee bean, espresso, coco
15	14	US	Building on 150 years and six generations of winemaking tradition, the winery trends toward a lea
16	15	Germany	Zesty orange peels and apple notes abound in this sprightly, mineral-toned Riesling. Off dry on th

Começando a trabalhar

Vamos carregar os primeiros dados.

Carregando dados

Entendendo o log, saídas diversas de análise.

```
7 - Hop - Pipeline opened.  
7 - Hop - Launching pipeline [estudo]...  
7 - Hop - Started the pipeline execution.  
7 - estudo - Executing this pipeline using the Local Pipeline Engine with run configuration 'local'  
7 - estudo - Expedindo início para Pipeline [estudo]  
7 - CSV file input.0 - Header row skipped in file 'C:\Users\grima\Downloads\Hop\Dados\vinhos_mundo.csv'  
3 - CSV file input.0 - Finished processing (I=129972, O=0, R=0, W=129971, U=0, E=0)  
3 - Value mapper.0 - Finished processing (I=0, O=0, R=129971, W=129971, U=0, E=0)  
3 - estudo - Pipeline duration : 0.805 seconds [ 0.805" ]  
3 - estudo - Execution finished on a local pipeline engine with run configuration 'local'
```

#	Nome do transform	Copia nr	Input	Read	Written	Output	Updated	Rejected	Errors	Buffers Input	Buffers Output	Duration	Speed	Status
1	CSV file input	0	129.972	0	129.971	0	0	0	0	0	0	0.738"	173.527	Finished
2	Value mapper	0	0	129.971	129.971	0	0	0	0	0	0	0.755"	169.897	Finished

Preview

View output Preview output Debug output Sniff output Add data probe

Basic

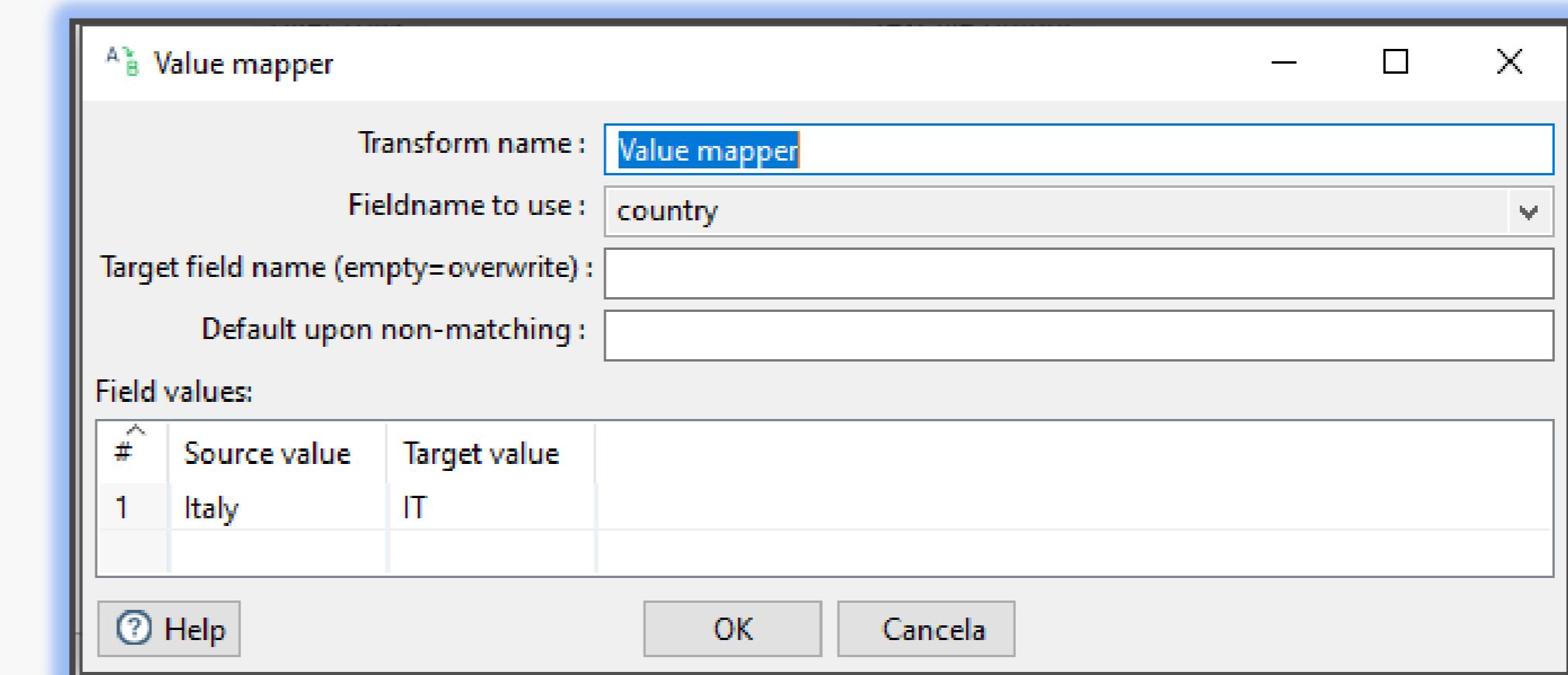
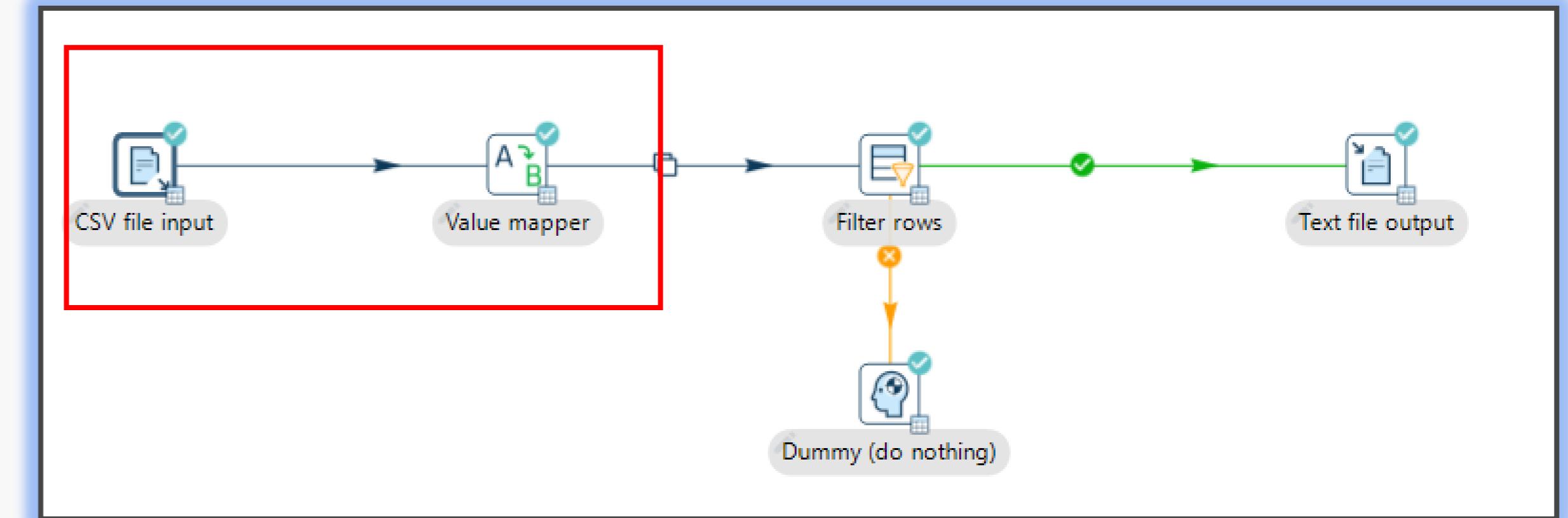
Edit Copy to clipboard Create hop Detach transform Show input fields Show output fields
Edit description Delete

Começando a trabalhar

Vamos carregar os primeiros dados.

Criando o Pipeline

Vamos criar um pipeline para tratar os dados via APACHE HOP

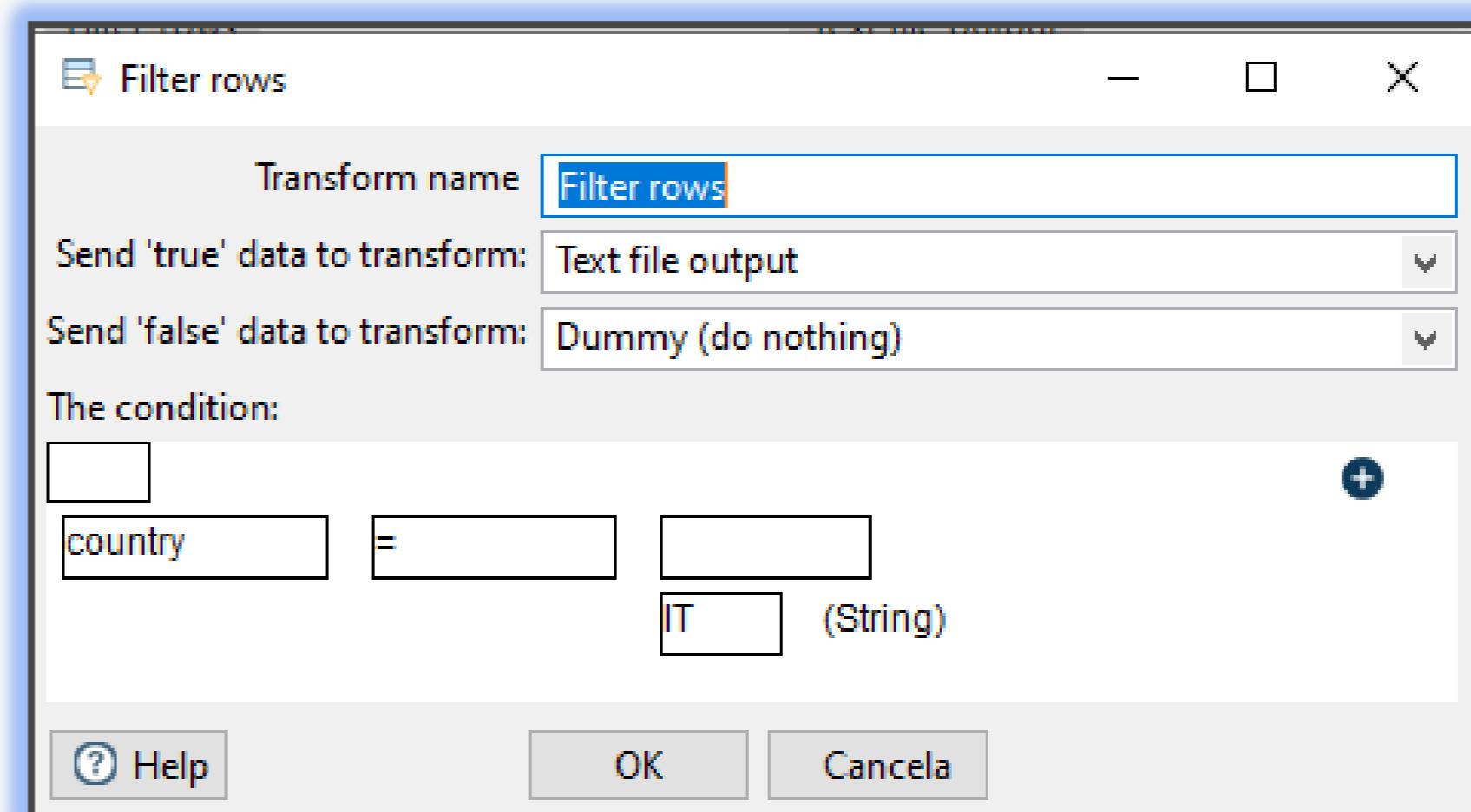
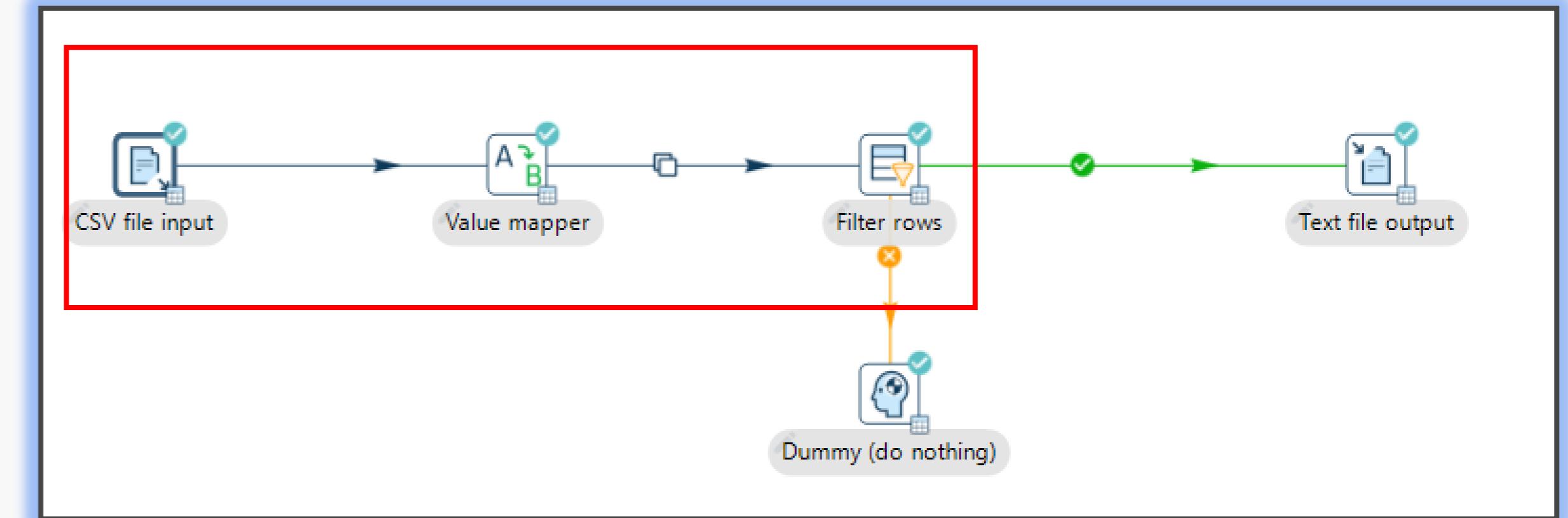


Começando a trabalhar

Vamos carregar os primeiros dados.

Criando o Pipeline

Vamos criar um pipeline para tratar os dados via APACHE HOP

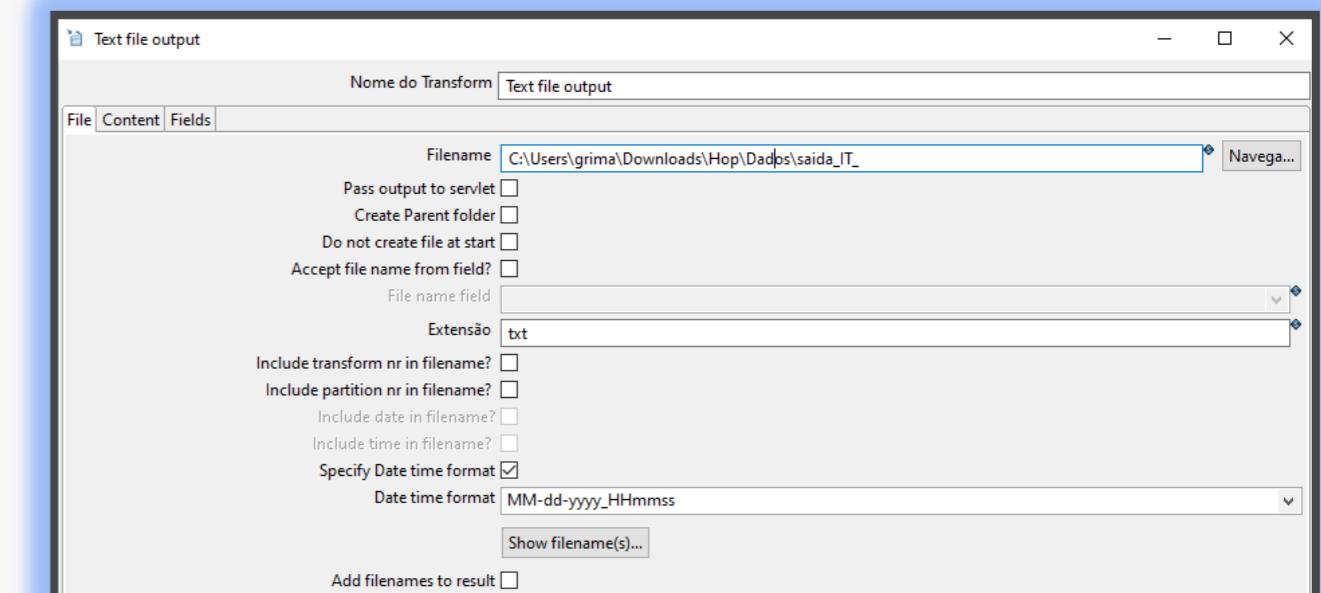
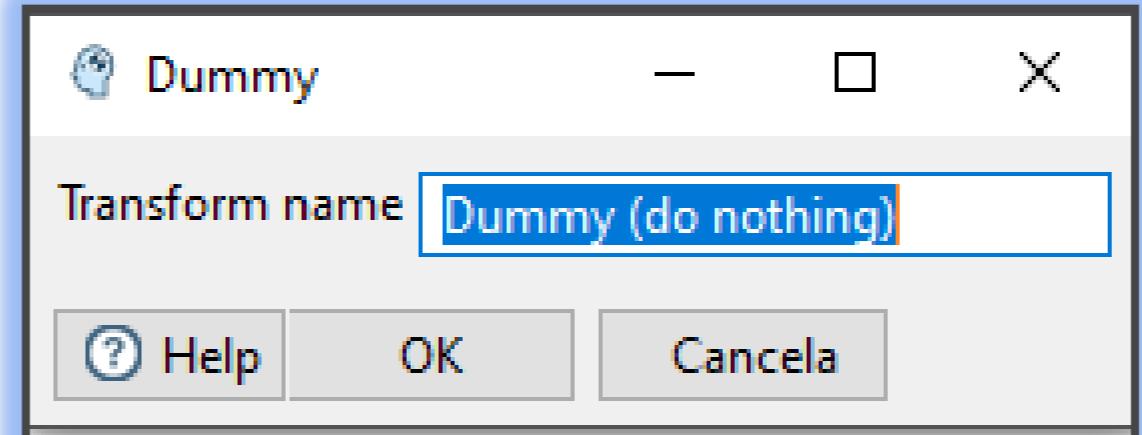
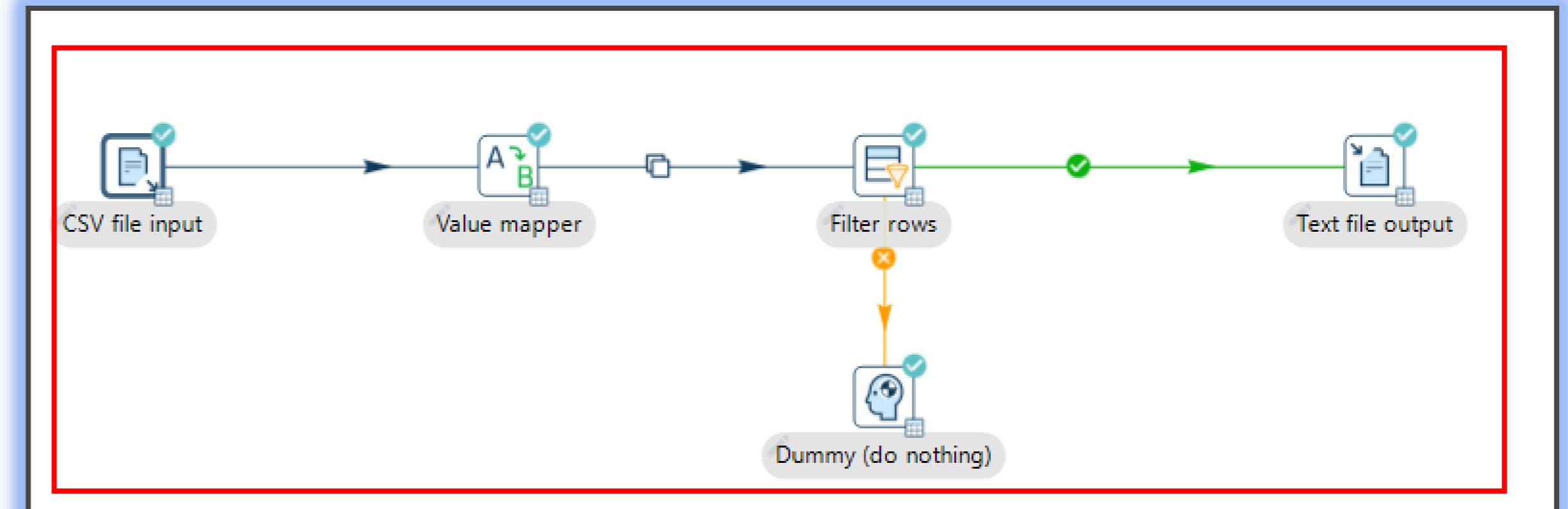


Começando a trabalhar

Vamos carregar os primeiros dados.

Criando o Pipeline

Vamos criar um pipeline para tratar os dados via APACHE HOP

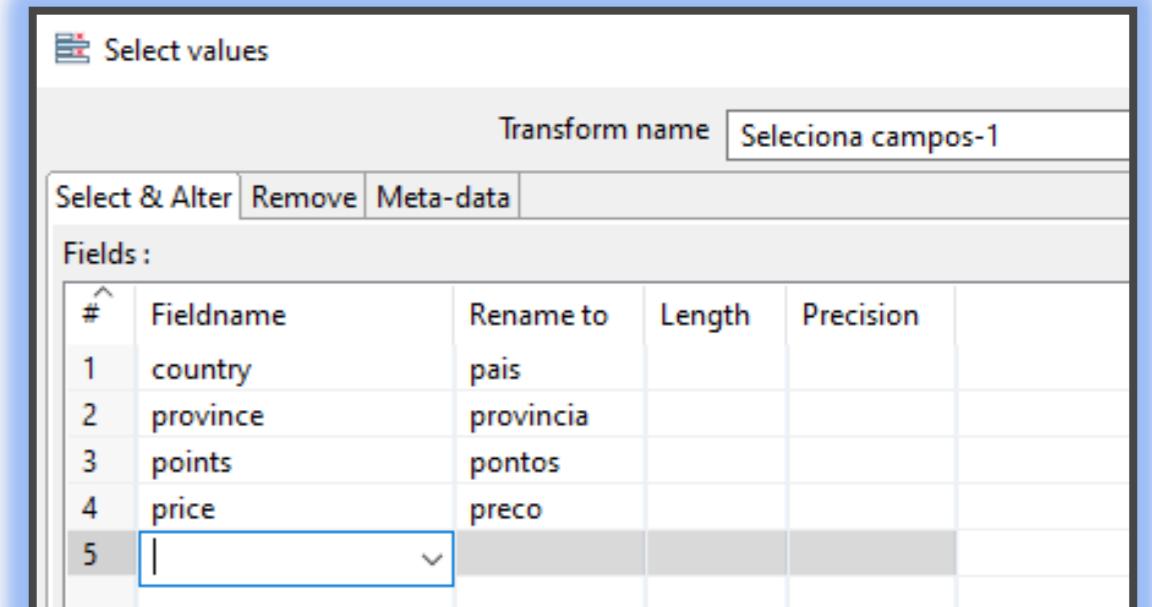
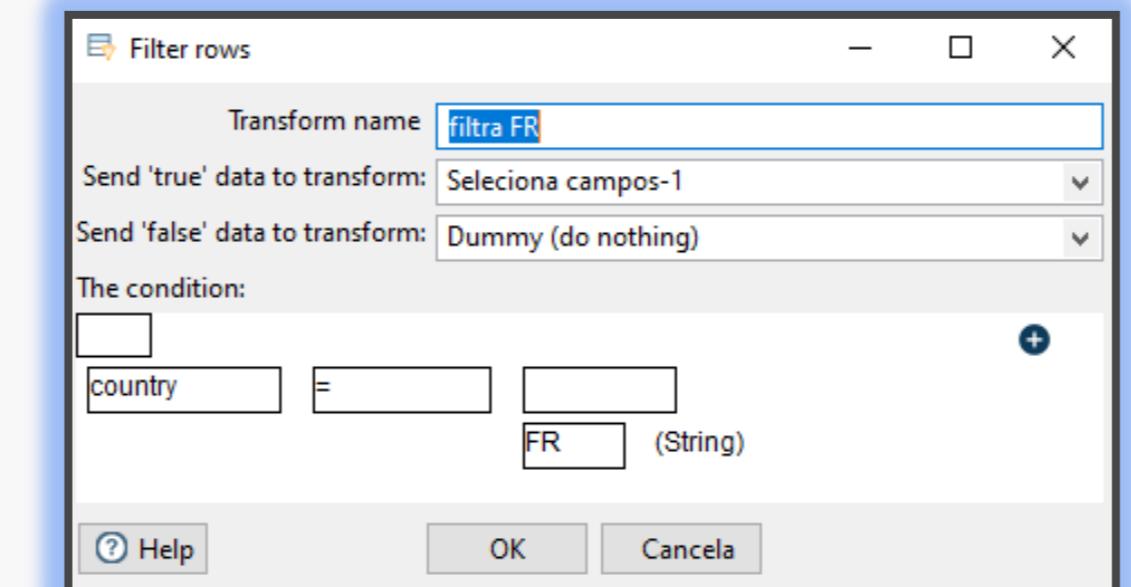
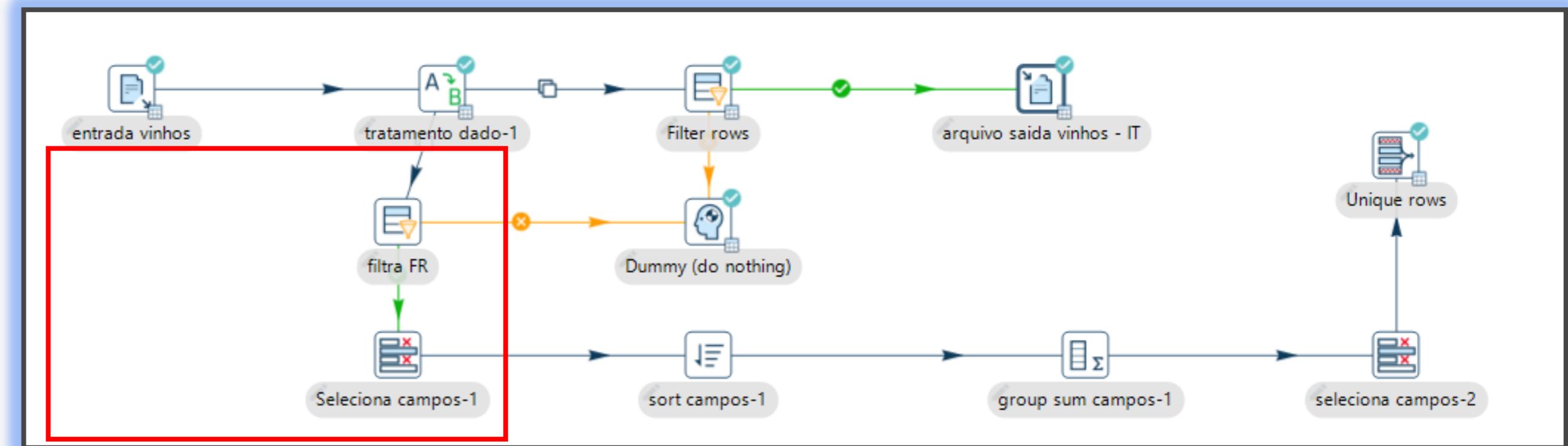


Totalizar os dados

Vamos fazer um tratamento de consolidação dos dados.

Incrementando o Pipeline

Vamos acrescentar mais funcionalidades ao pipeline para tratar os dados via APACHE HOP.

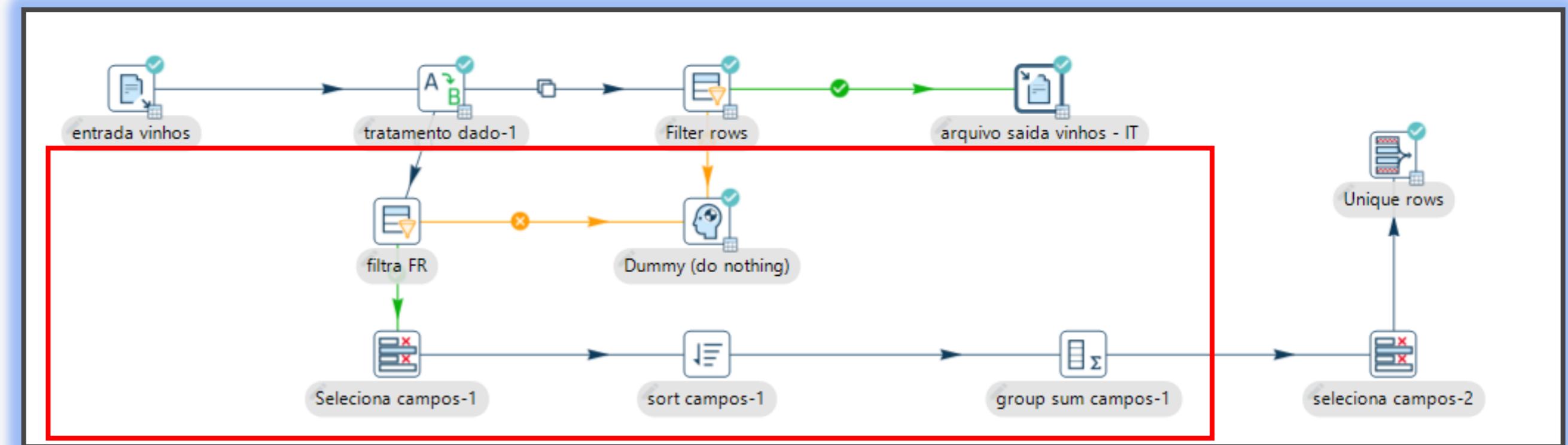


Totalizar os dados

Vamos fazer um tratamento de consolidação dos dados.

Incrementando o Pipeline

Vamos acrescentar mais funcionalidades ao pipeline para tratar os dados via APACHE HOP.



Two configuration windows are shown, connected by blue arrows pointing to the highlighted section of the pipeline.

Sort rows configuration:

- Transform name: sort campos-1
- Sort directory: \${java.io.tmpdir}
- TMP-file prefix: out
- Sort size (rows in memory): 1000000
- Free memory threshold (in %):
- Compress TMP Files?:
- Only pass unique rows? (verifies keys only):
- Fields:

#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength	Presorted?
1	país	S	N	N	0	N
2	provincia	S	N	N	0	N
3	pontos	S	N	N	0	N
4	preco	S	N	N	0	N

Group by configuration:

- Transform name: group sum campos-1
- Include all rows?:
- Temporary files directory: \${java.io.tmpdir}
- TMP-file prefix: grp
- Add line number, restart in each group:
- Line number field name:
- Always give back a result row:
- The fields that make up the group:

#	Group field
1	país
2	provincia

- Aggregates:

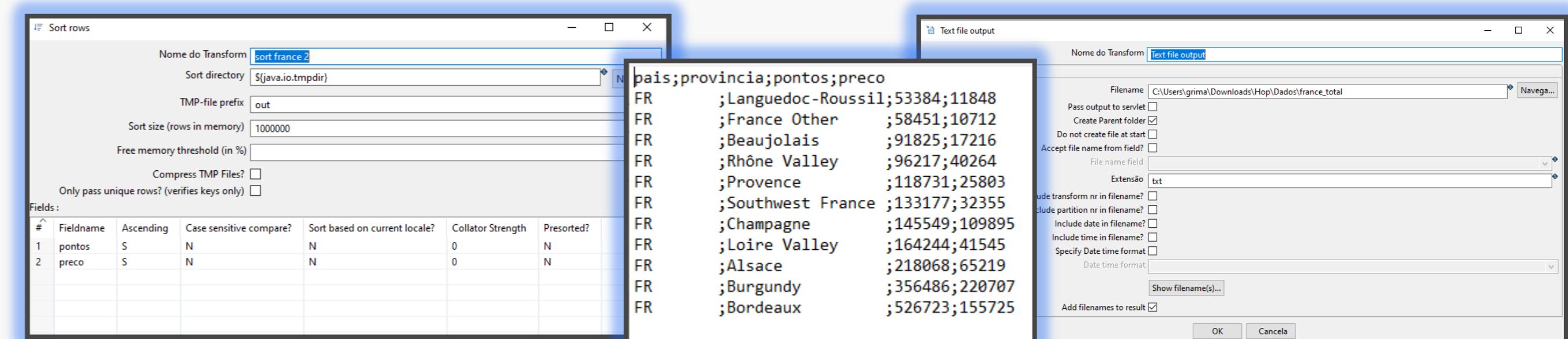
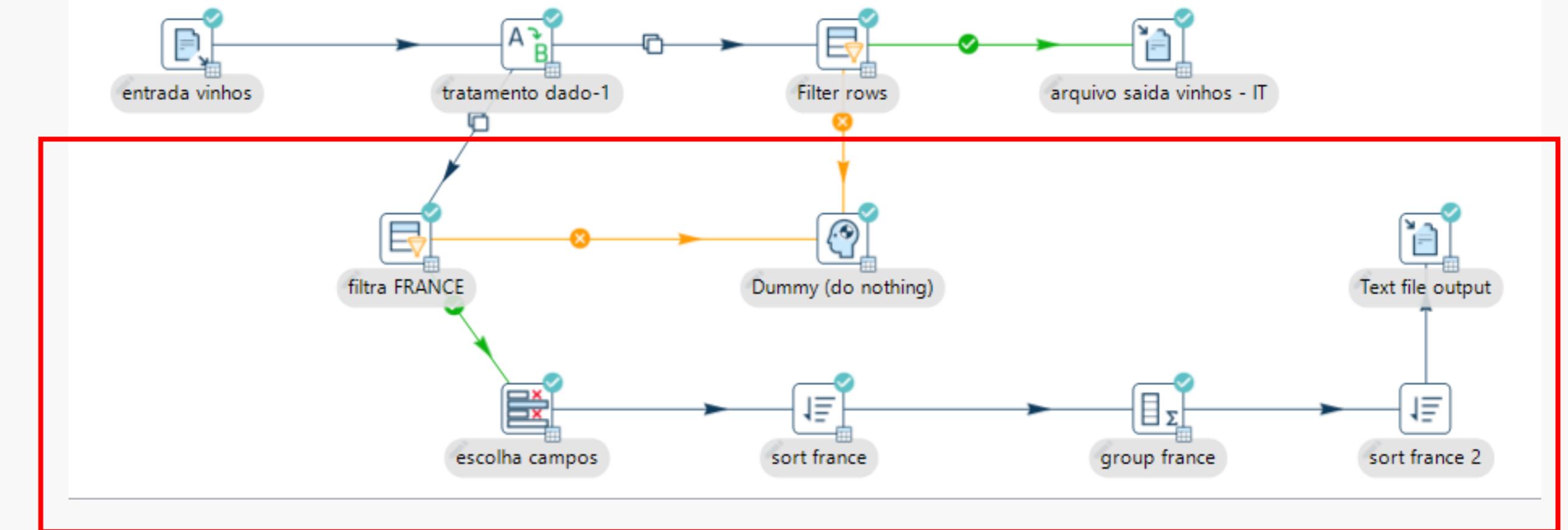
#	Name	Subject	Type
1	pontos	pontos	Sum
2	preco	preco	Sum

Totalizar os dados

Vamos fazer um tratamento de consolidação dos dados.

Incrementando o Pipeline

Vamos acrescentar mais funcionalidades ao pipeline para tratar os dados via APACHE HOP.



Preparando os dados

Vamos conhecer os dados que serão utilizados neste novo pipeline.

Vendas.xlsx

id_venda	cod_cliente	cod_produto	cod_marca	Ano_venda	Mês_venda	Dia_venda	dtc_venda	qtd_vend	val_venda	num_nota
1	3	4	4	2015	8	22	22/08/2015	1	1300	1033
2	4	10	8	2014	8	15	15/08/2014	1	2,15	812
3	5	11	8	2014	4	26	26/04/2014	2	0,5	447
4	5	19	13	2014	1	9	09/01/2014	1	1500	509
5	6	16	11	2015	5	24	24/05/2015	1	135	1337
6	4	11	8	2014	7	25	25/07/2014	1	0,5	1337
7	3	9	8	2015	3	12	12/03/2015	1	12,25	724
8	2	19	13	2014	8	17	17/08/2014	2	1500	375
9	2	20	12	2015	2	7	07/02/2015	2	350	416

Merge nos dados

Vamos fazer um tratamento para realizar união entre dados.

Cliente.xlsx

Cod_Cliente	Des_Cliente
1	João do Pulo
2	Benito de Paula
3	Renato Russo
4	Tom Jobim
5	Edson Arantes
6	Reginaldo Rossi

Marca.xlsx

cod_marca	des_marca
1	APPLE
2	MICROSOFT
3	DELL
4	IBM
5	SAMSUNG
6	MOTOROLA
7	TILIBRA
8	FABER CASTELL
9	AVON
10	BOTICARIO
11	PHILIPS
12	ELETROLUX
13	BRASTEMP
14	DA CASA

Produto.xlsx

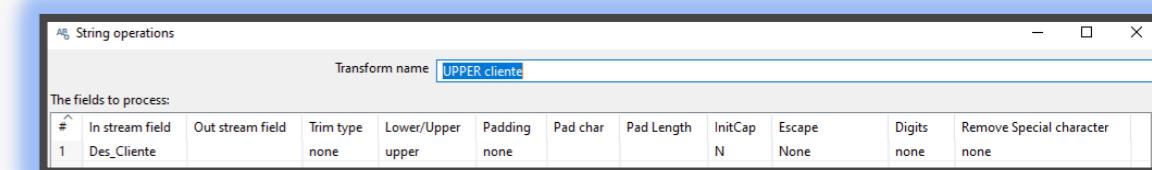
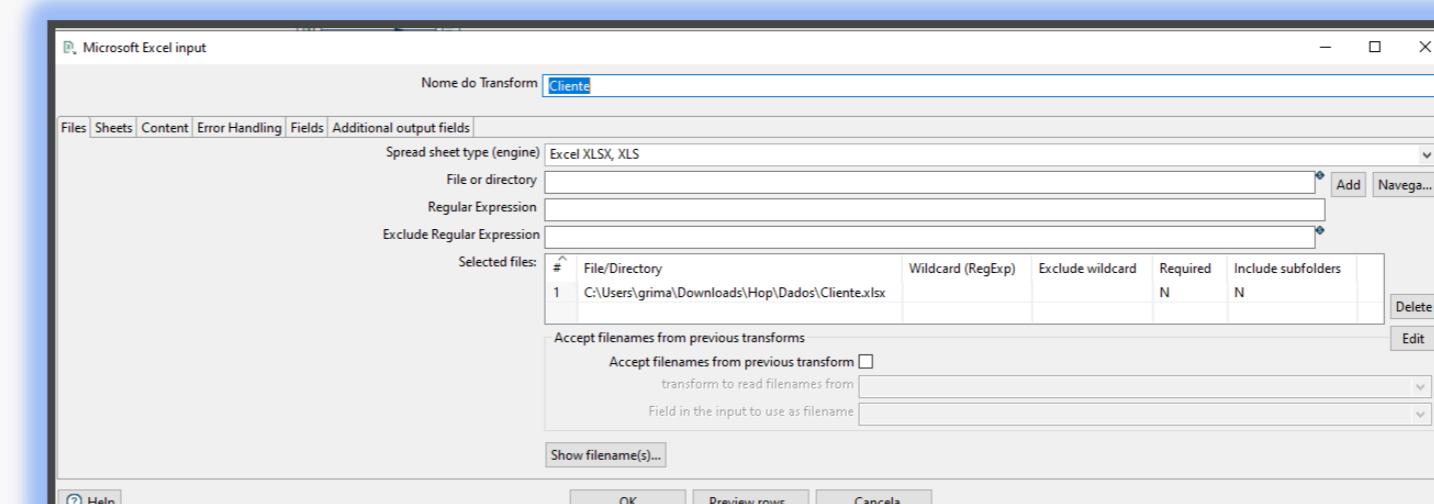
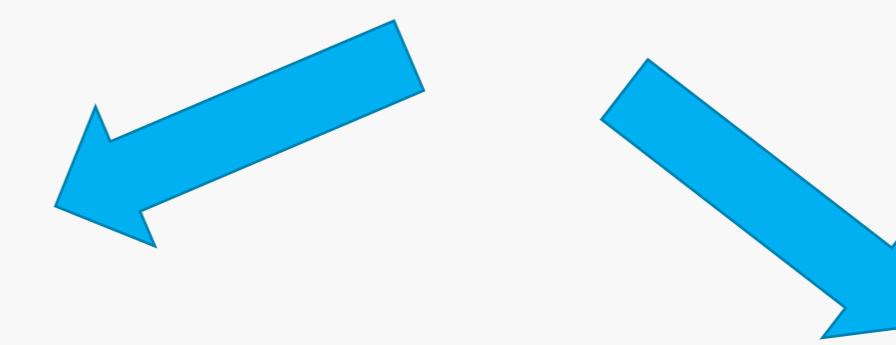
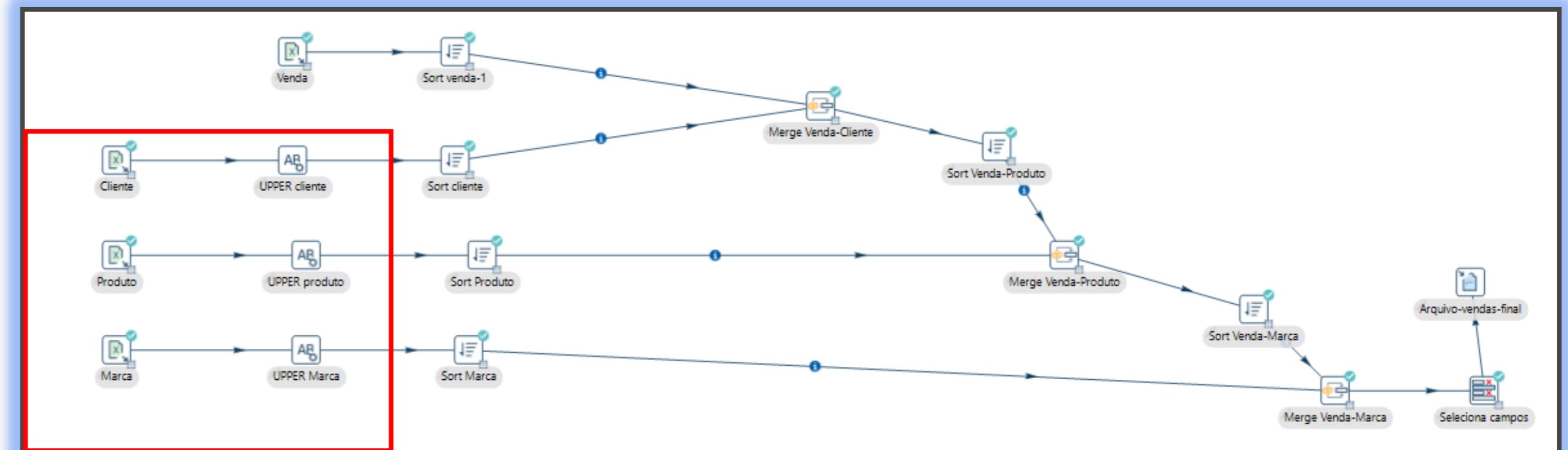
cod_produto	nom_produto
1	TV
2	Mouse
3	Teclado
4	Monitor
5	Computador
6	Notebook
7	Celular 4Gb
8	Telefone Sem Fio
9	Celular 8GB
10	Papel A4
11	Caneta
12	Borracha
13	Estojo
14	Classificador
15	Baton
16	Perfume
17	Aparalho de Barbear Elétrico
18	Xbox
19	Fogão
20	Geladeira
21	microondas
22	Mesa
23	Sofá

Merge nos dados

Vamos fazer um tratamento para realizar união entre dados.

Preparando os dados

Vamos dar os primeiros passos realizando a leitura dos dados e colocando em maiúsculo as informações string.

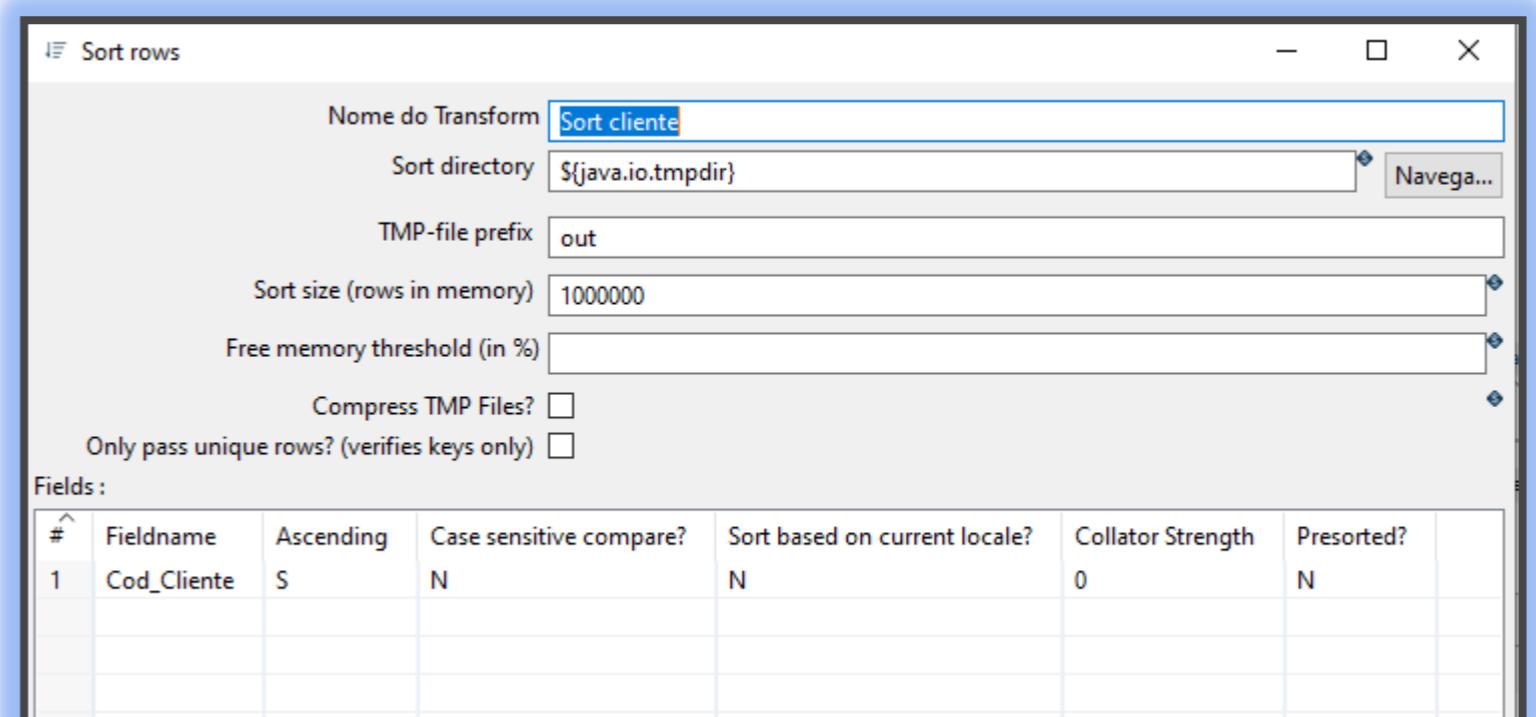
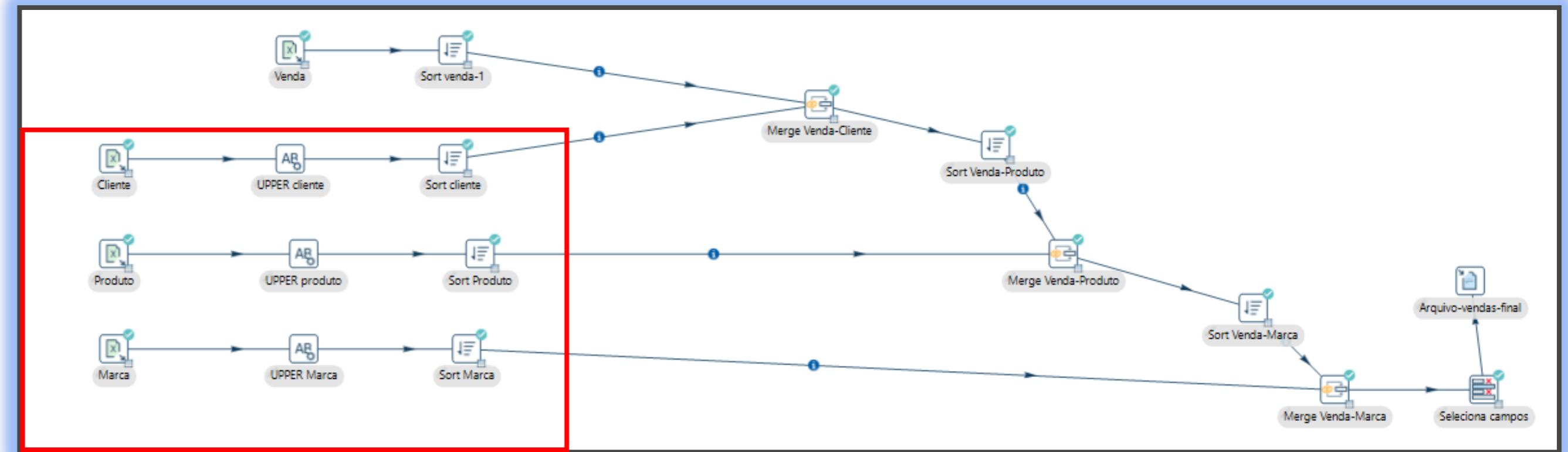


Merge nos dados

Vamos fazer um tratamento para realizar união entre dados.

Sort nos dados

Para fazer a união dos dados é necessário que os dados estejam ordenados pela chave. Cada arquivo tem um campo chave.

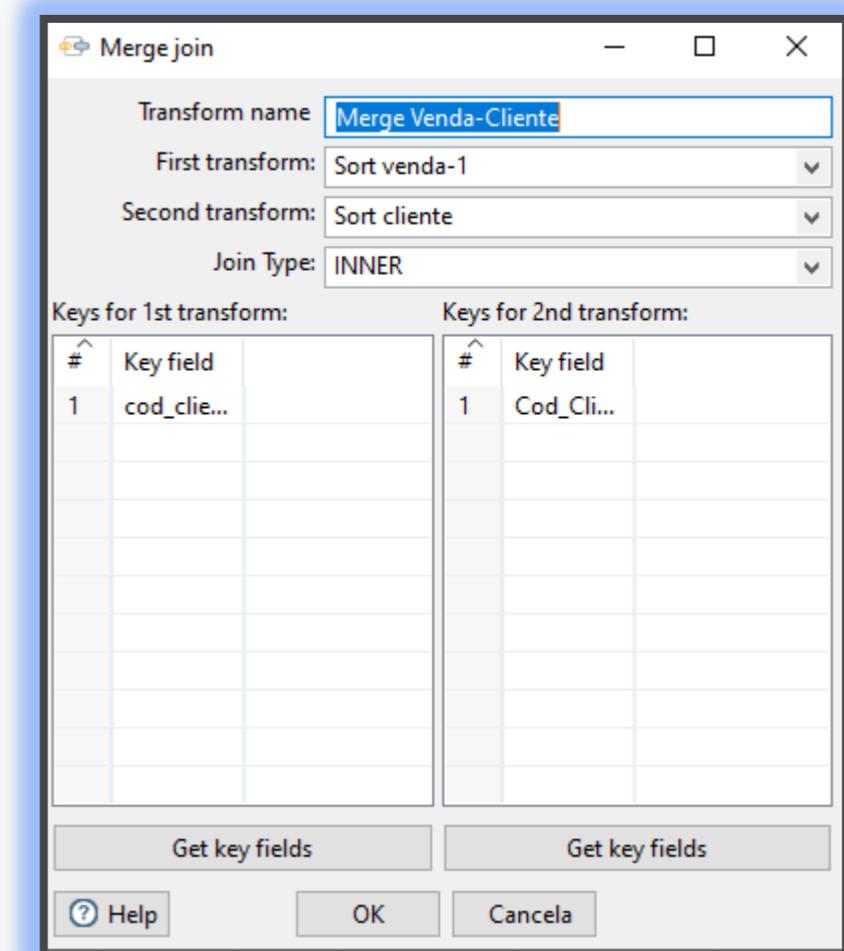
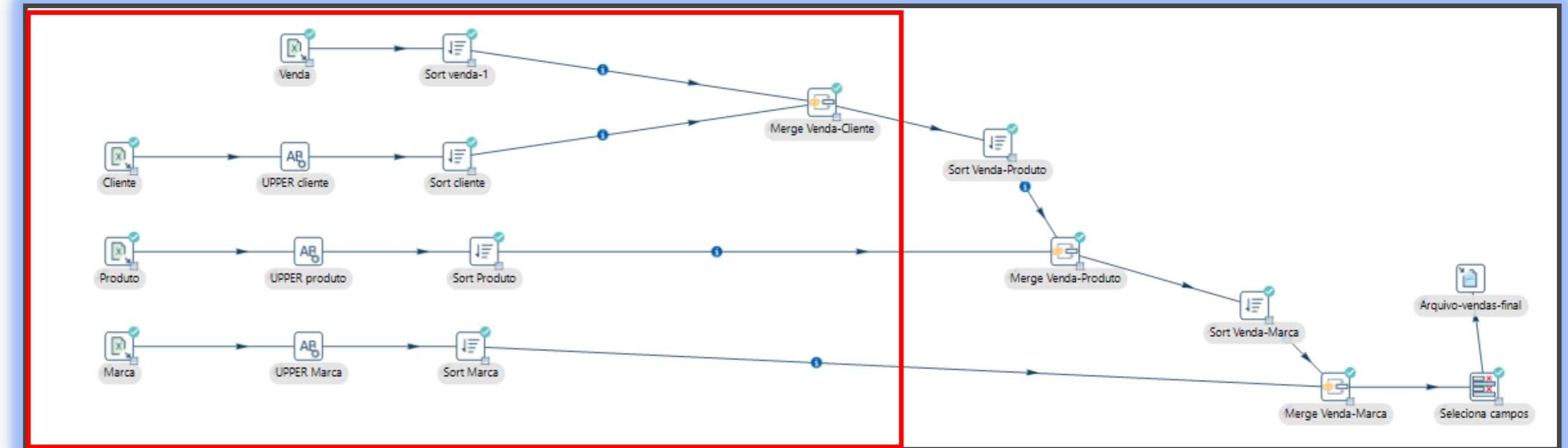


Merge nos dados

Vamos fazer um tratamento para realizar união entre dados.

Merge com venda

Utilizaremos 3 vezes o componente Merge para unir os arquivos: Venda, Produto, Marca e Cliente.

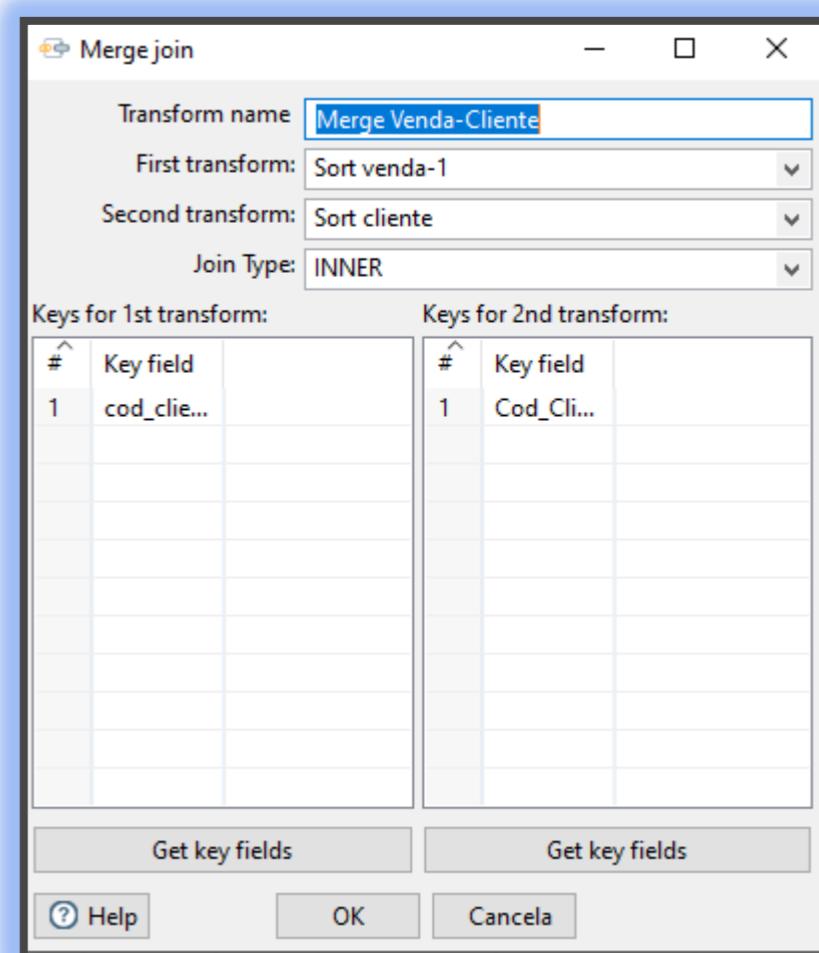
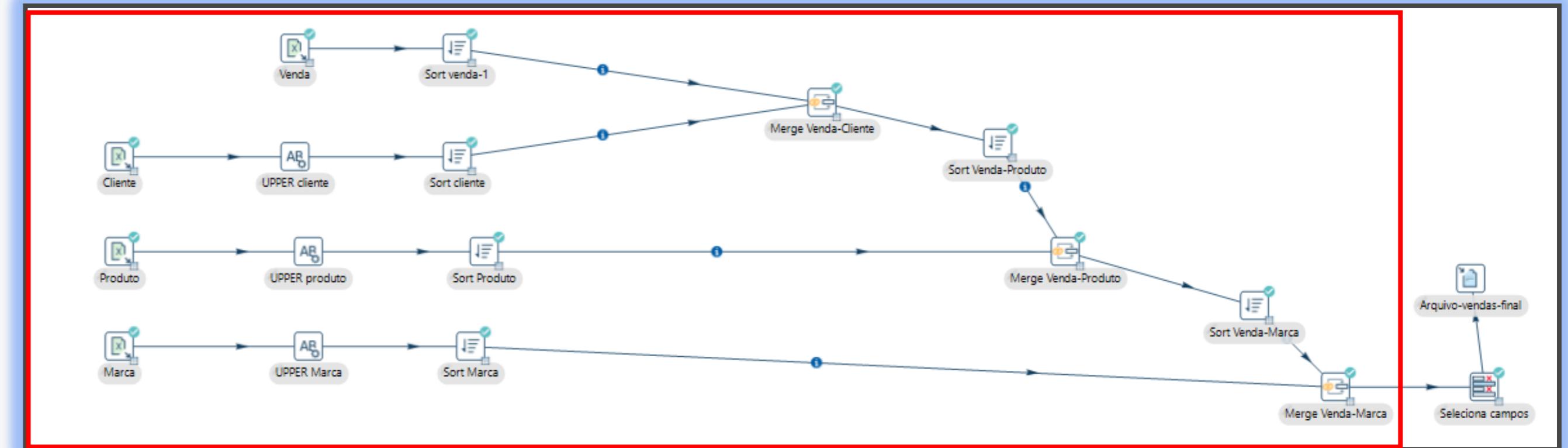


Merge nos dados

Vamos fazer um tratamento para realizar união entre dados.

Merge com venda

Utilizaremos 3 vezes o componente Merge para unir os arquivos: Venda, Produto, Marca e Cliente.

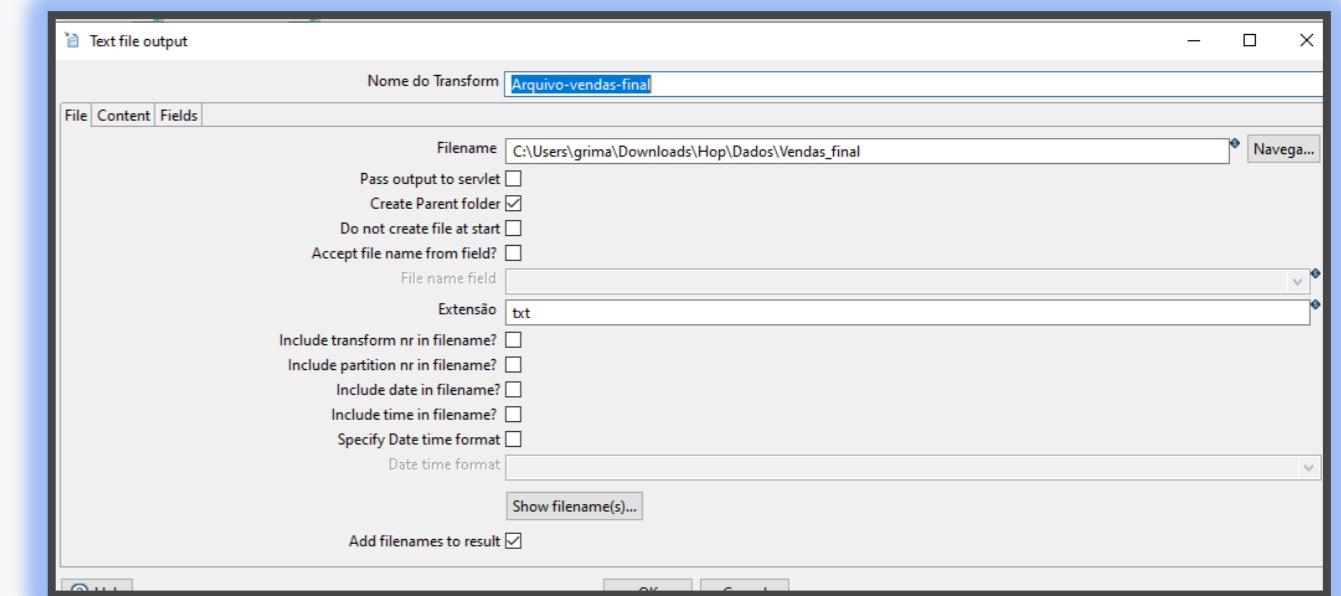
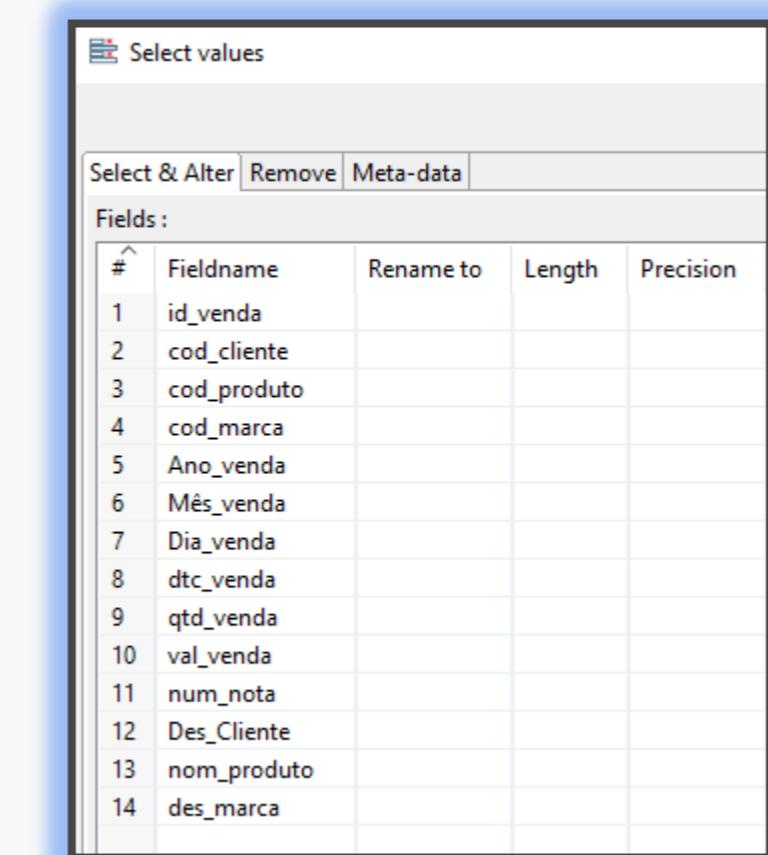
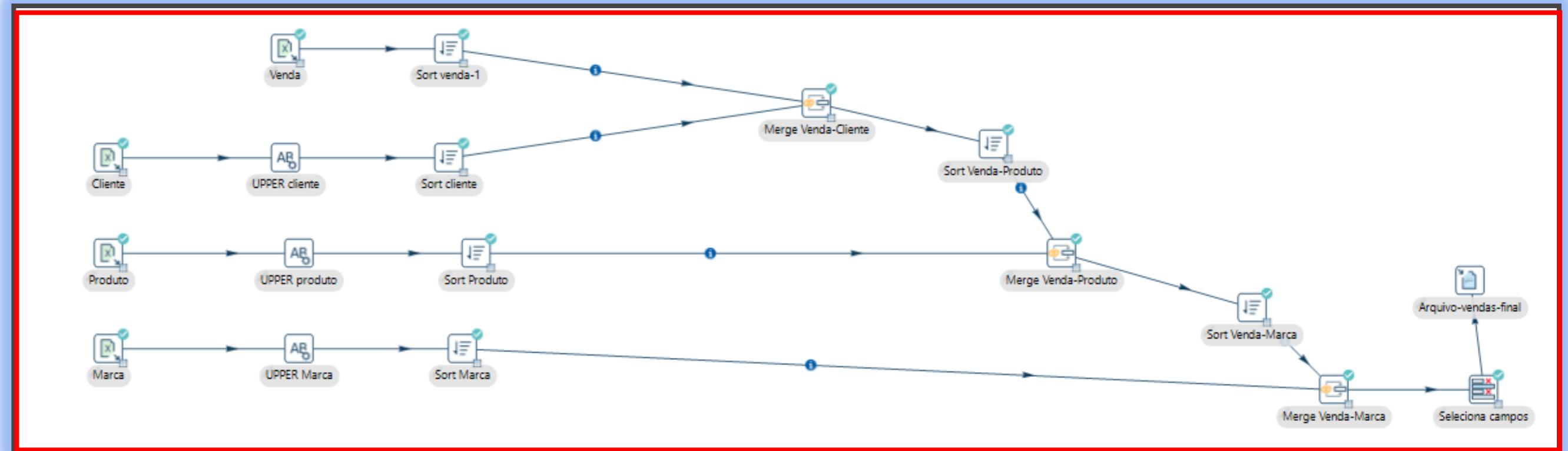


Merge nos dados

Vamos fazer um tratamento para realizar união entre dados.

Faremos agora a seleção dos campos e gravação final

Selecionaremos os campos finais e gravaremos em um arquivo de saída.

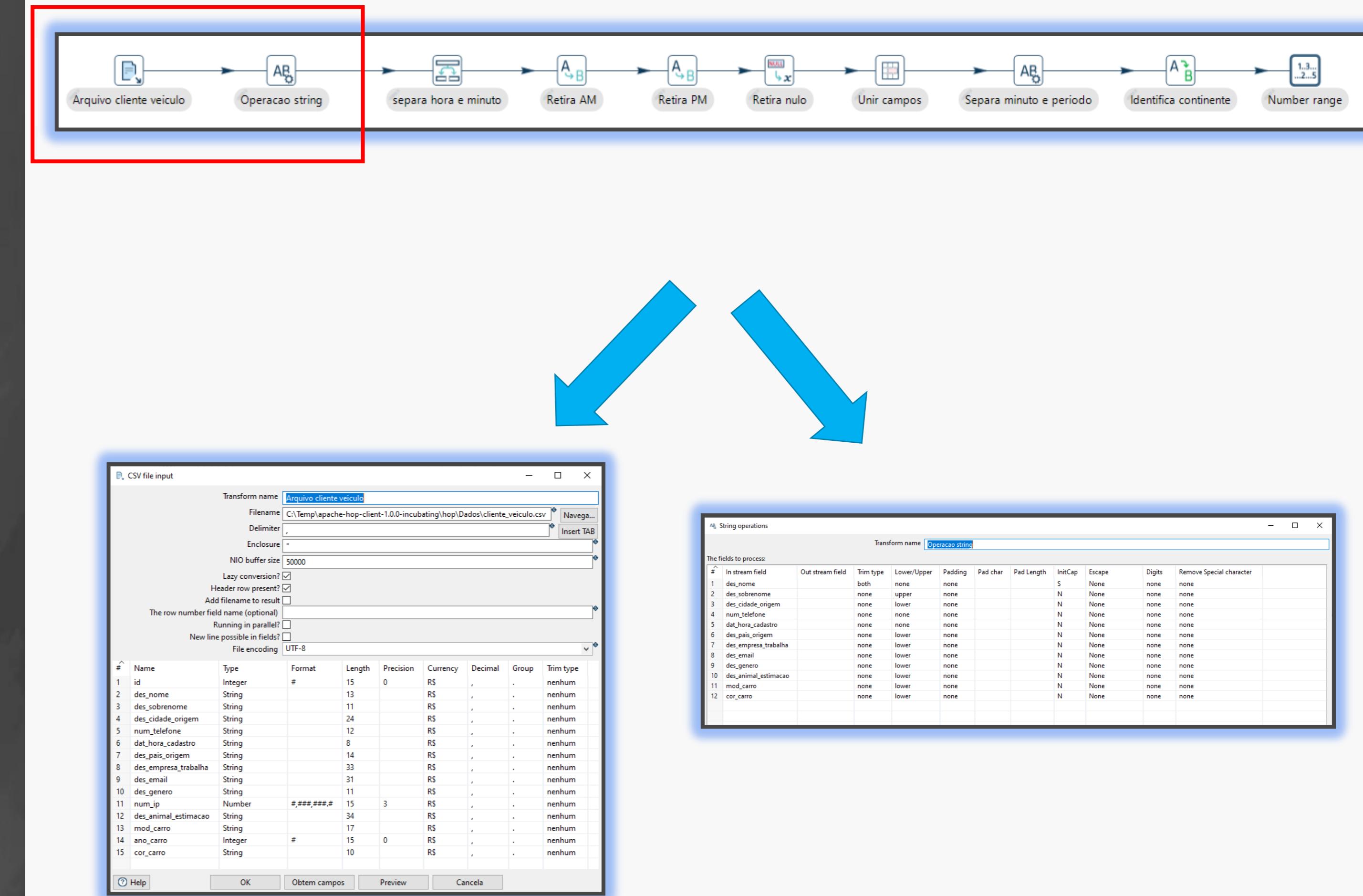


Pipeline de tratamento com diversos componentes

A ideia é conhecermos as diversas funcionalidades.

Tratamento nos dados

Vamos fazer uma série de tratamentos nos dados.

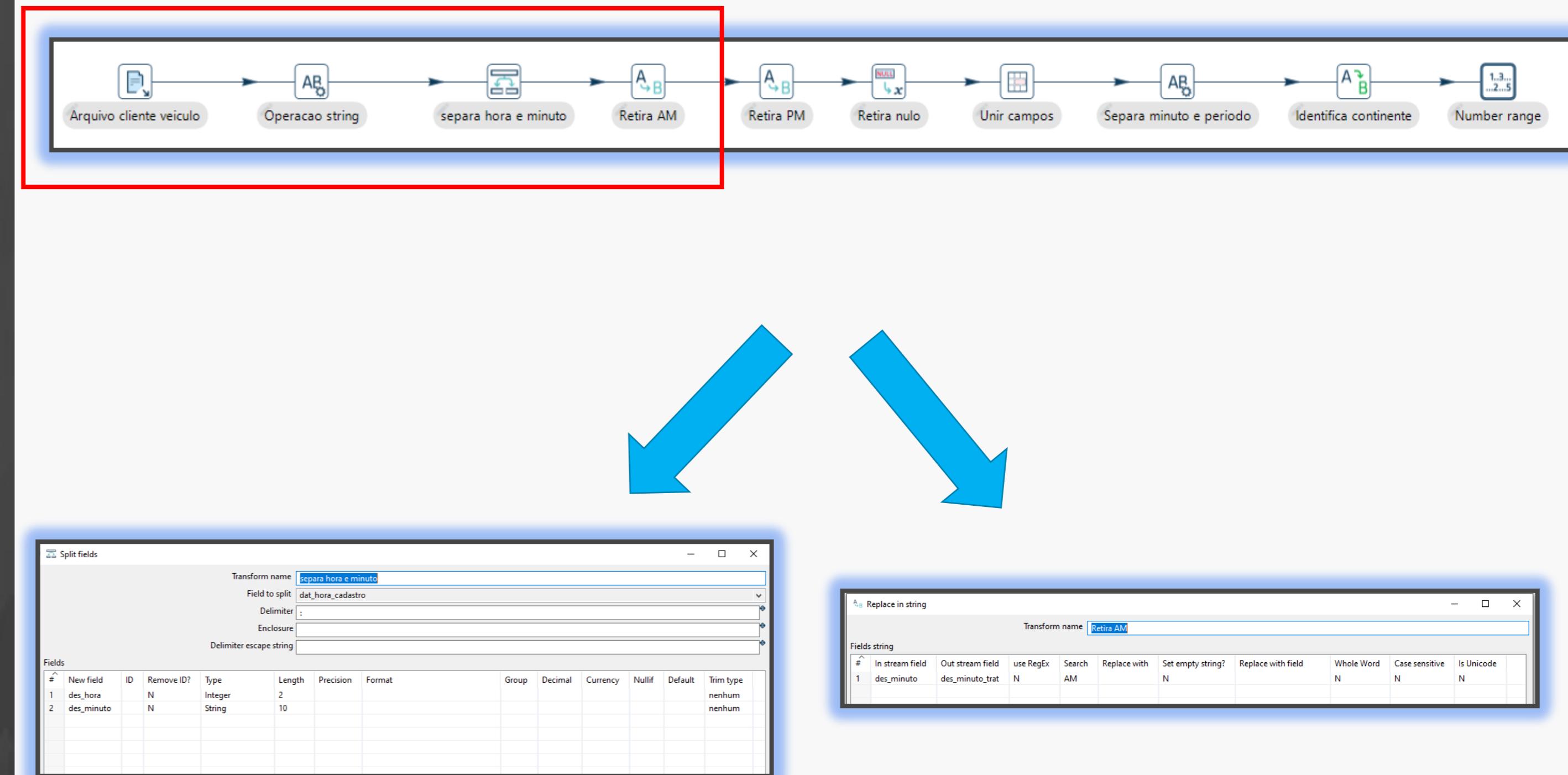


Pipeline de tratamento com diversos componentes

A ideia é conhecermos as diversas funcionalidades.

Tratamento nos dados

Vamos fazer uma série de tratamentos nos dados.

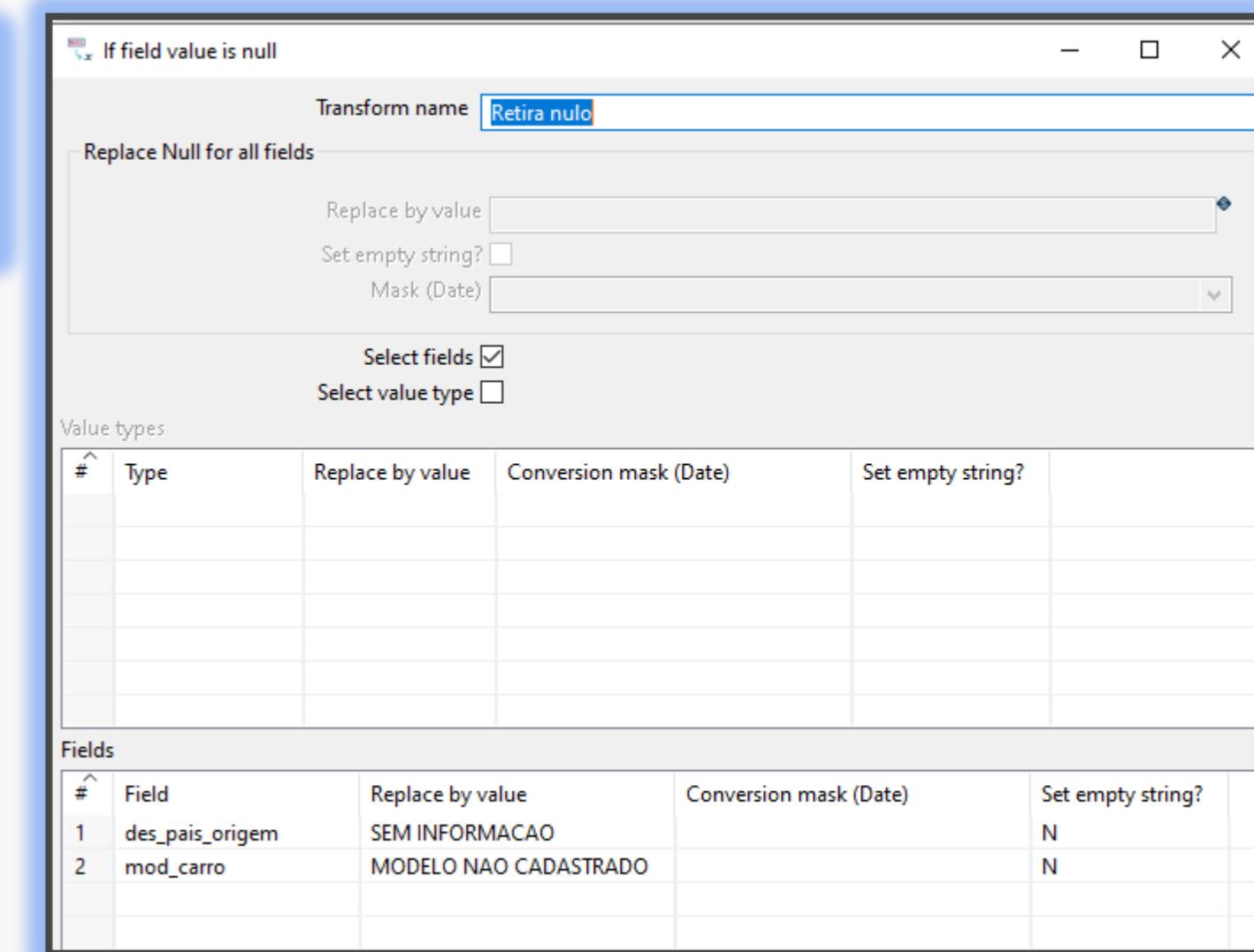
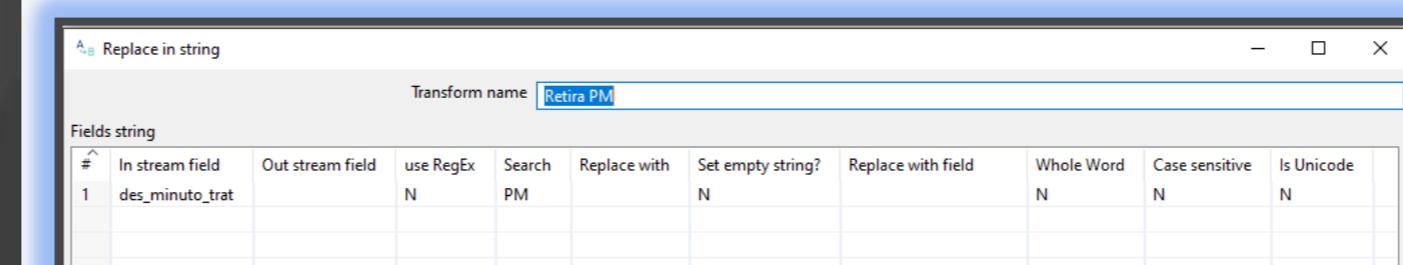
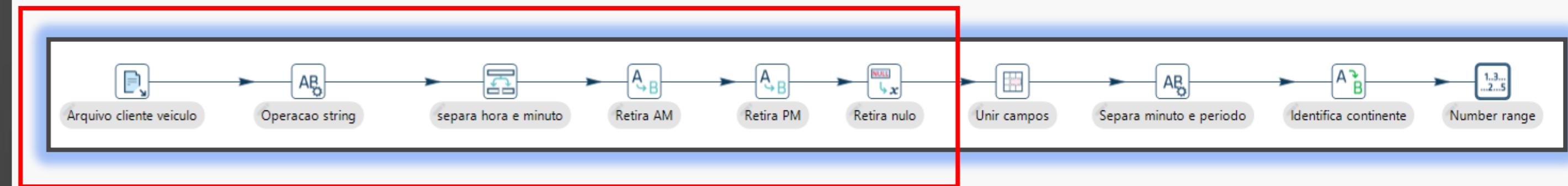


Pipeline de tratamento com diversos componentes

A ideia é conhecermos as diversas funcionalidades.

Tratamento nos dados

Vamos fazer uma série de tratamentos nos dados.

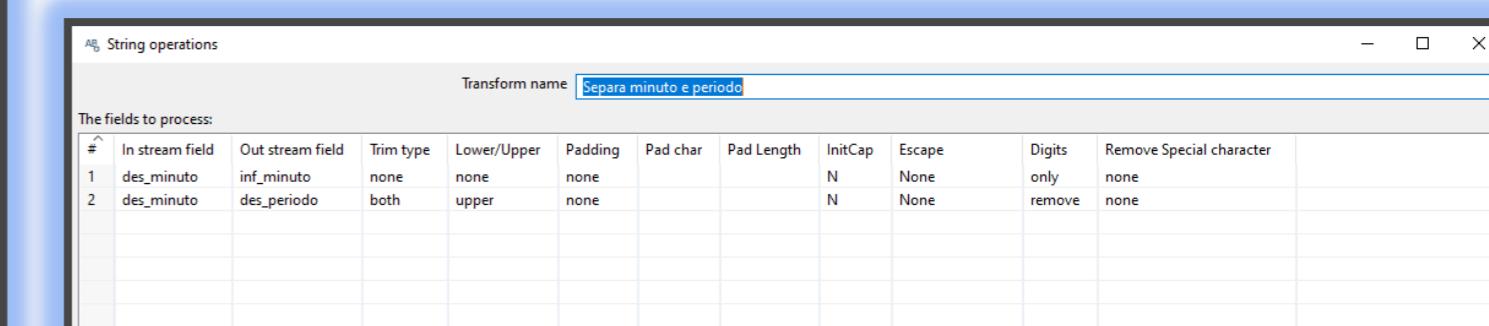
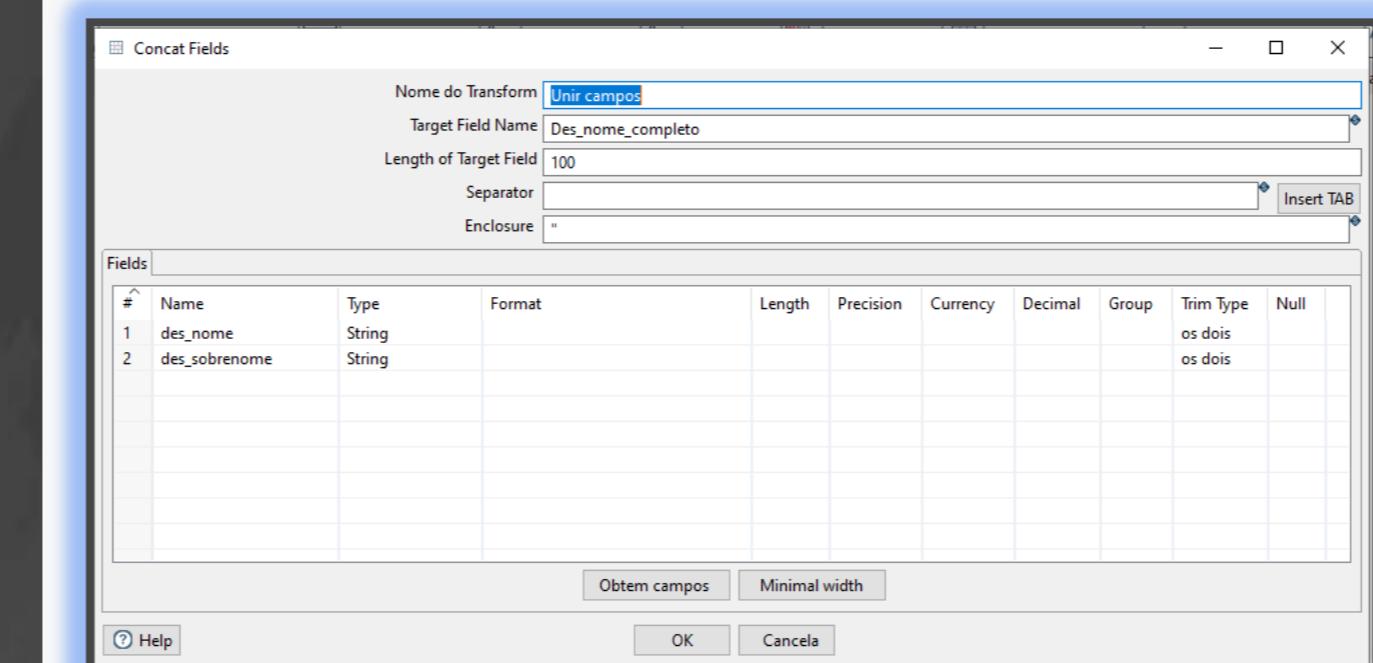
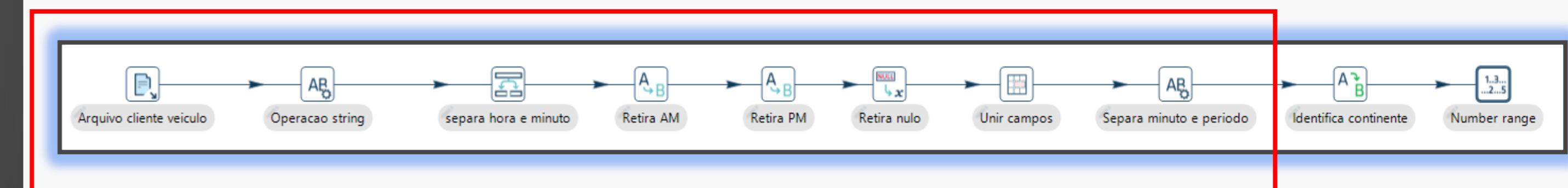


Pipeline de tratamento com diversos componentes

A ideia é conhecermos as diversas funcionalidades.

Tratamento nos dados

Vamos fazer uma série de tratamentos nos dados.

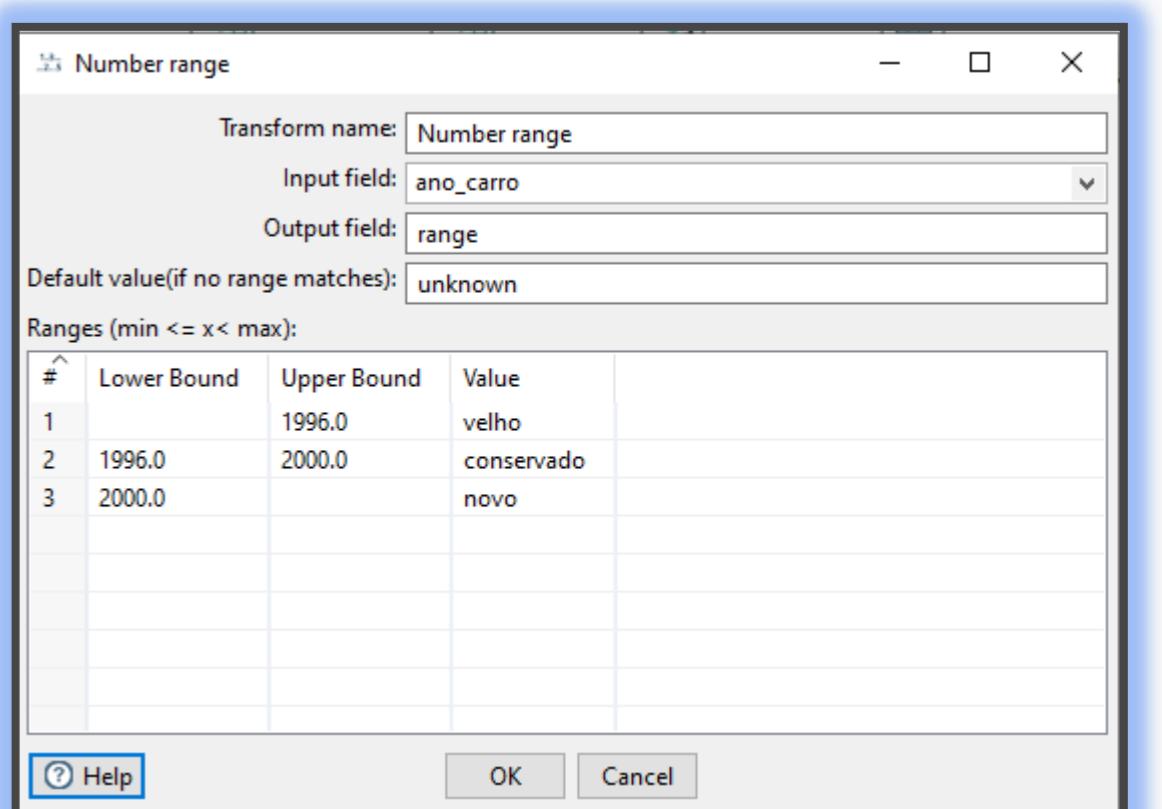
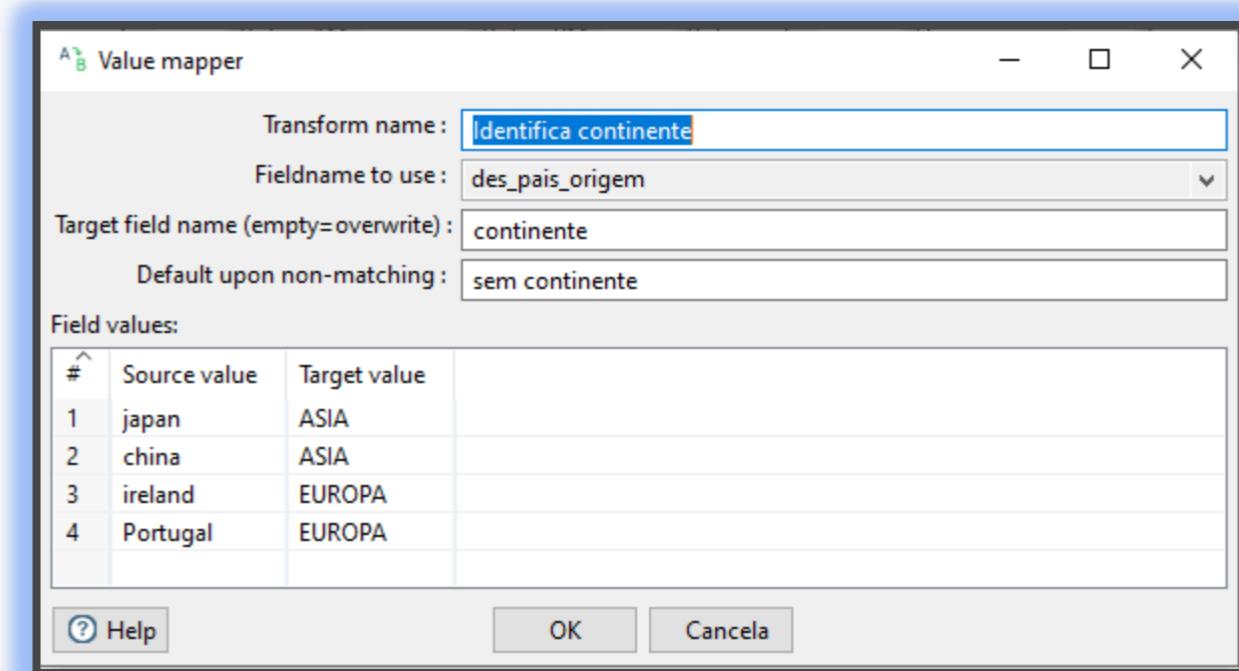
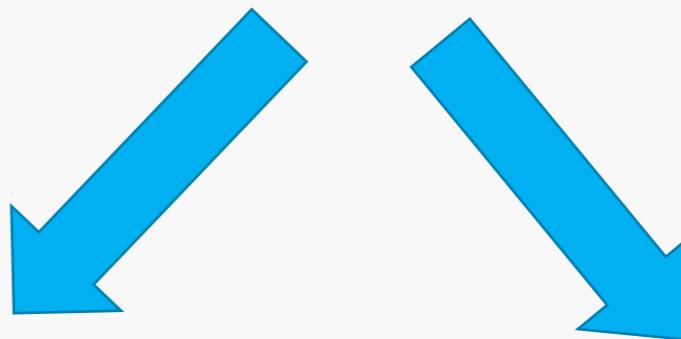
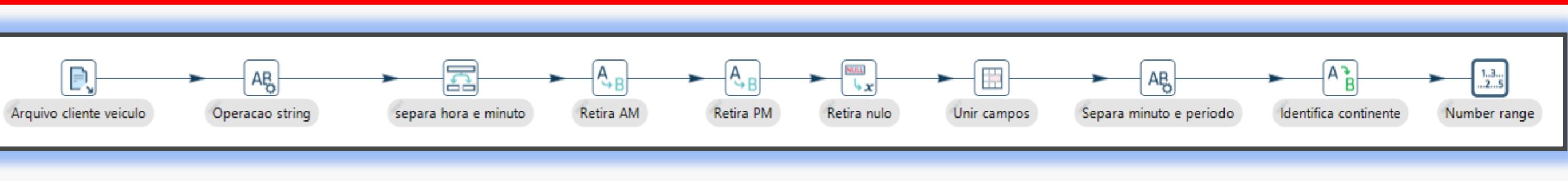


Pipeline de tratamento com diversos componentes

A ideia é conhecermos as diversas funcionalidades.

Tratamento nos dados

Vamos fazer uma série de tratamentos nos dados.

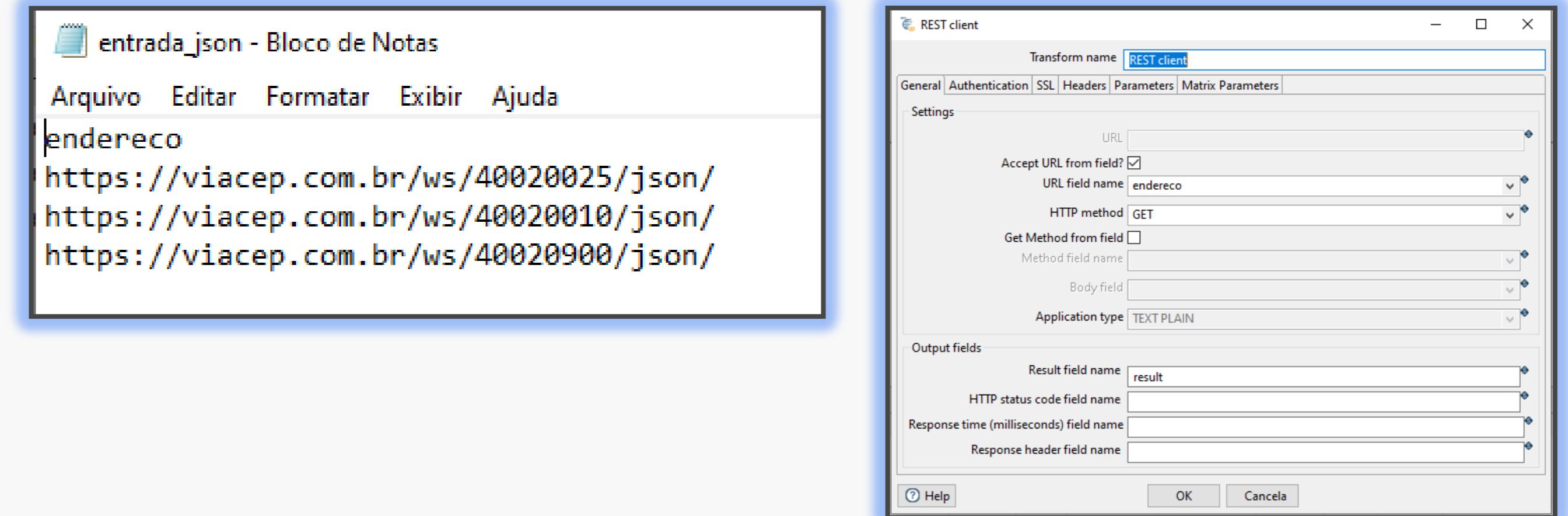
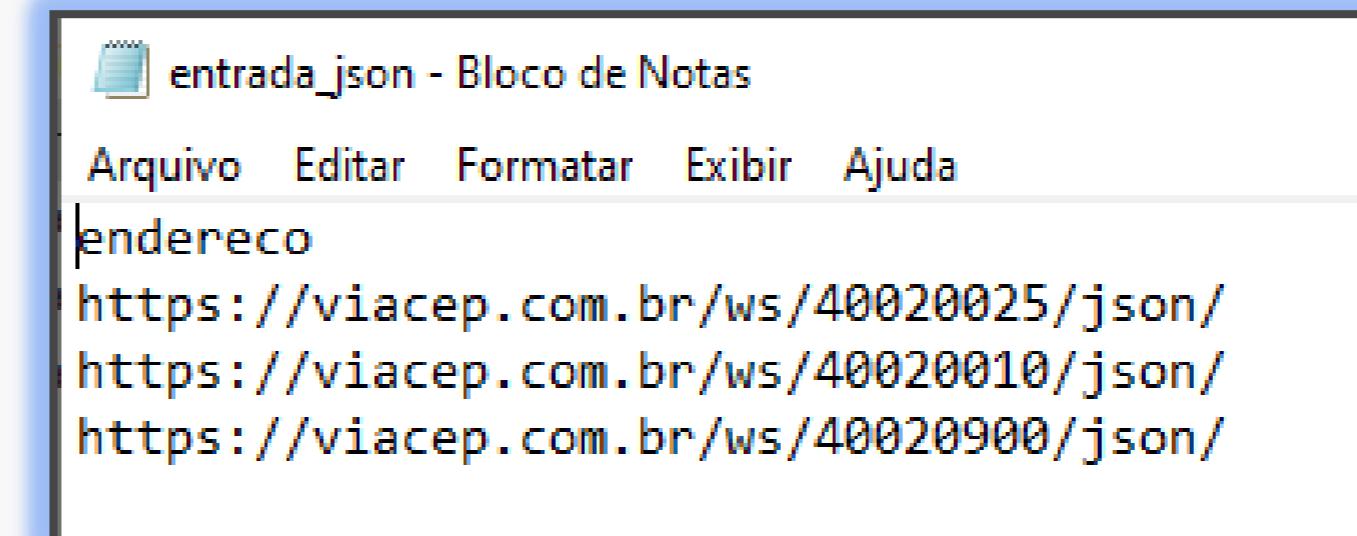
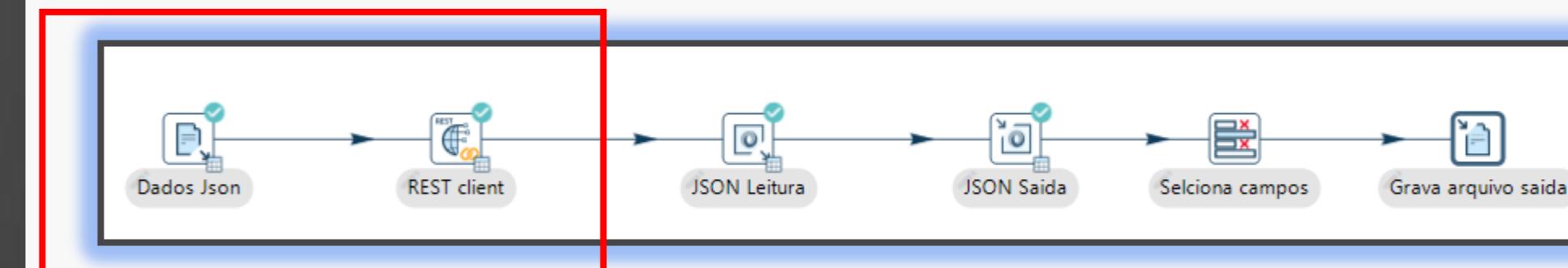


Carregando dados API

Vamos fazer uma pipeline para carga de dados em uma API

Trazendo dados de uma API

Faremos uma consulta Rest Método GET para a leitura do dados.

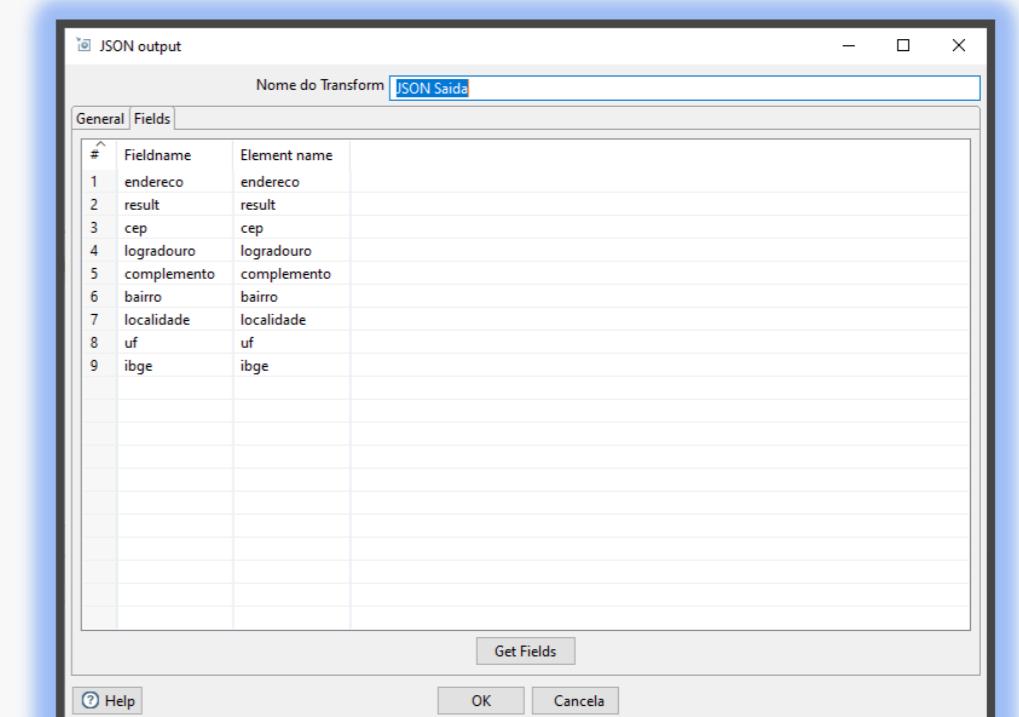
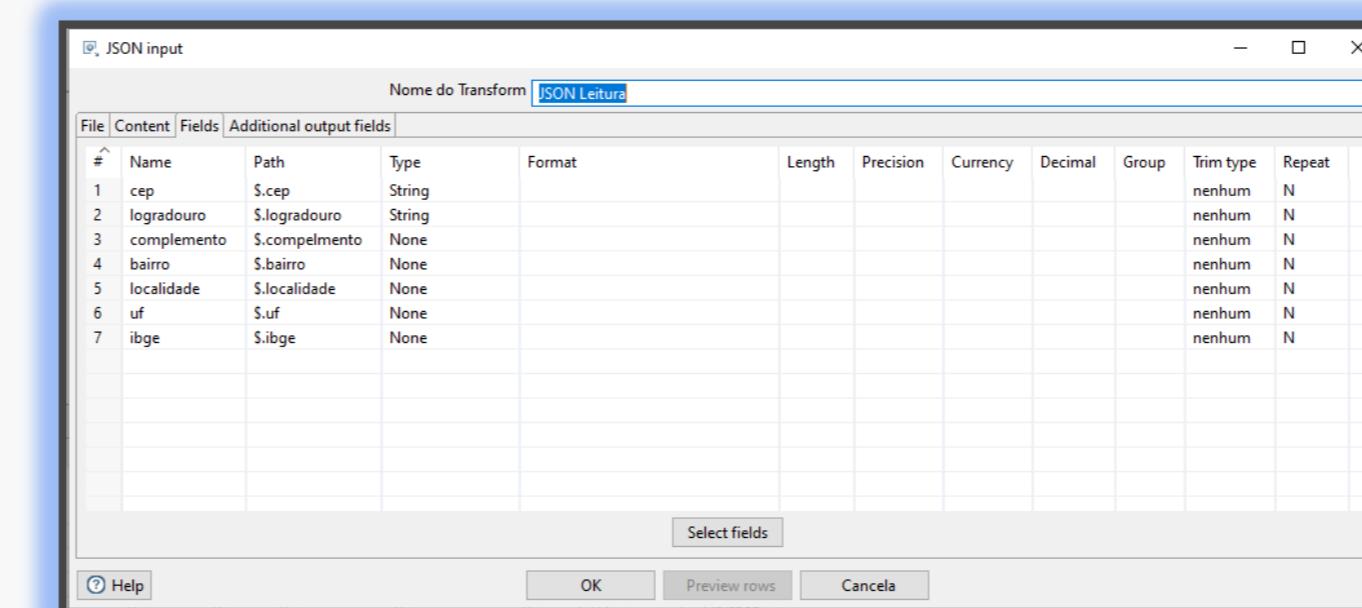
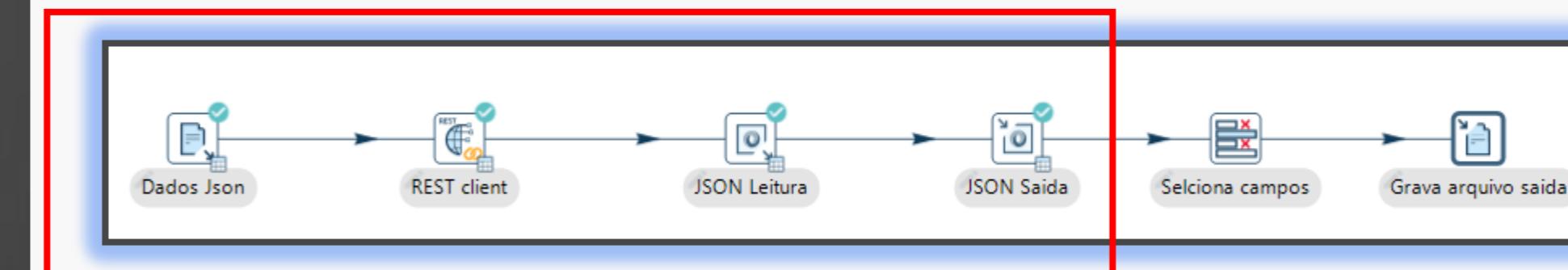


Carregando dados API

Vamos fazer uma pipeline para carga de dados em uma API

Trazendo dados de uma API

Faremos uma consulta Rest Método GET para a leitura do dados.

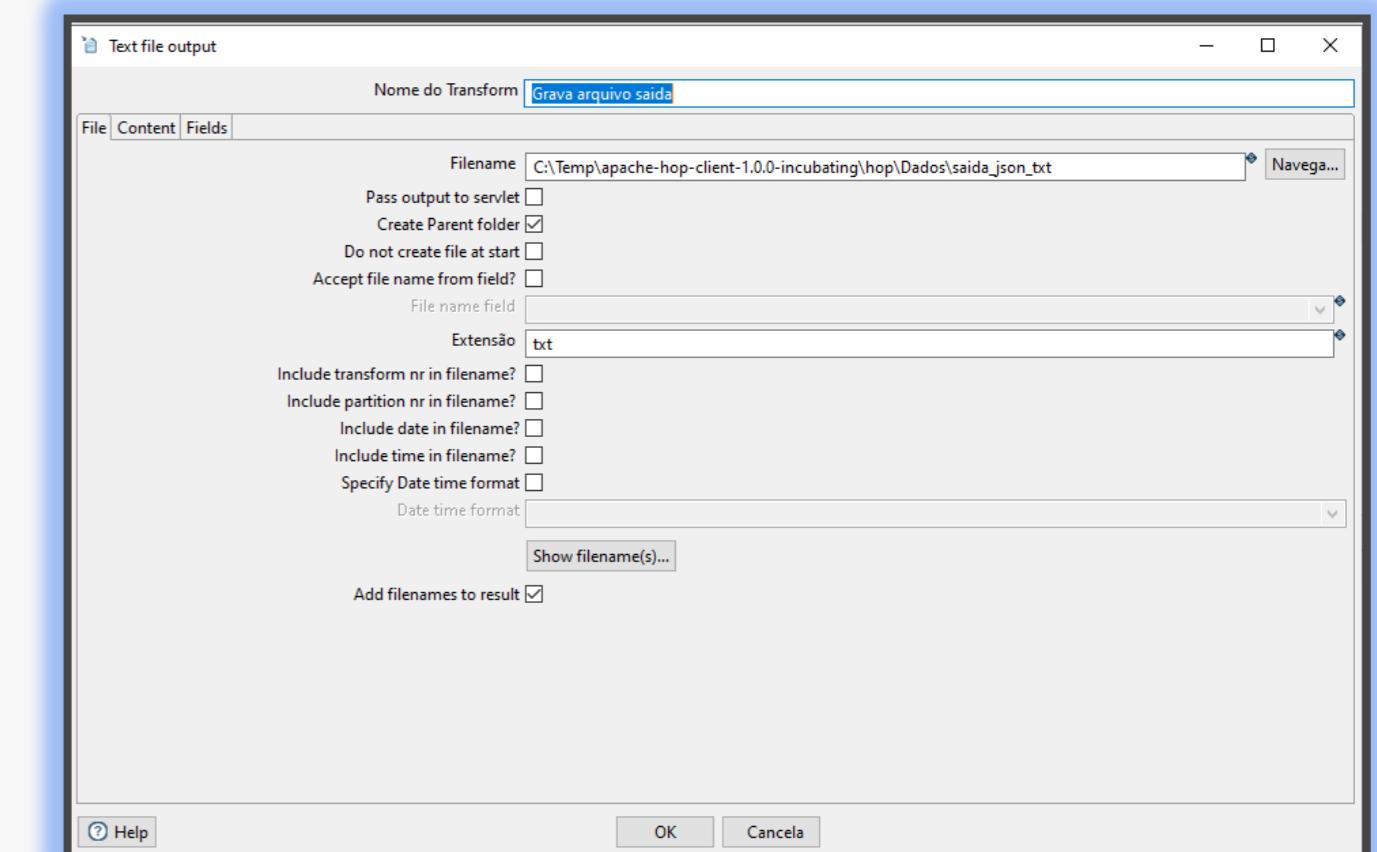
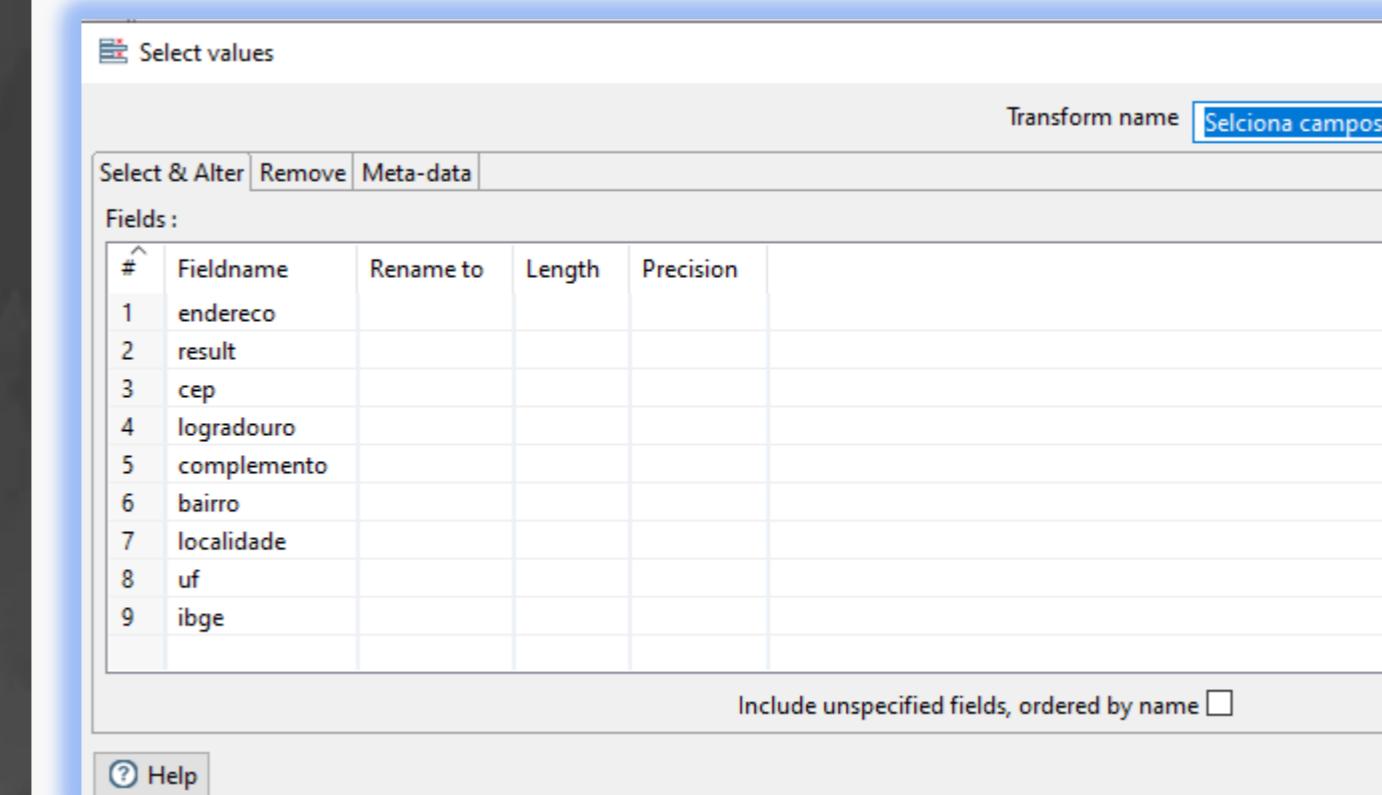
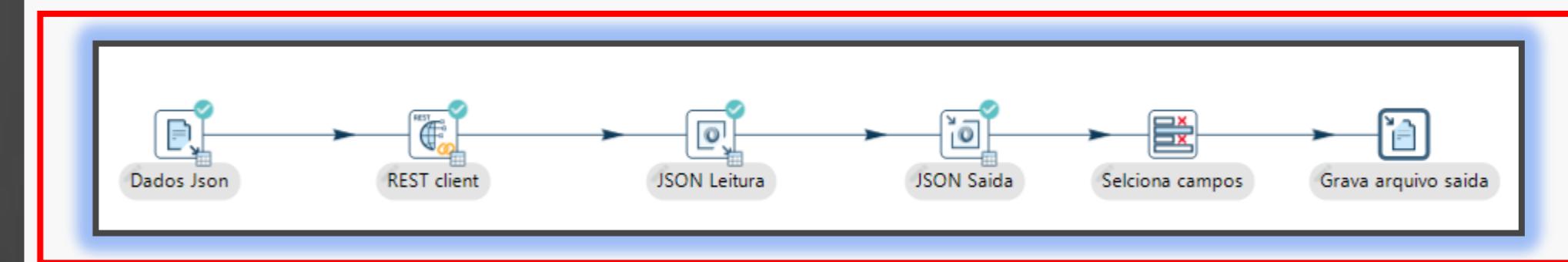


Carregando dados API

Vamos fazer uma pipeline para carga de dados em uma API

Trazendo dados de uma API

Faremos uma consulta Rest Método GET para a leitura do dados.

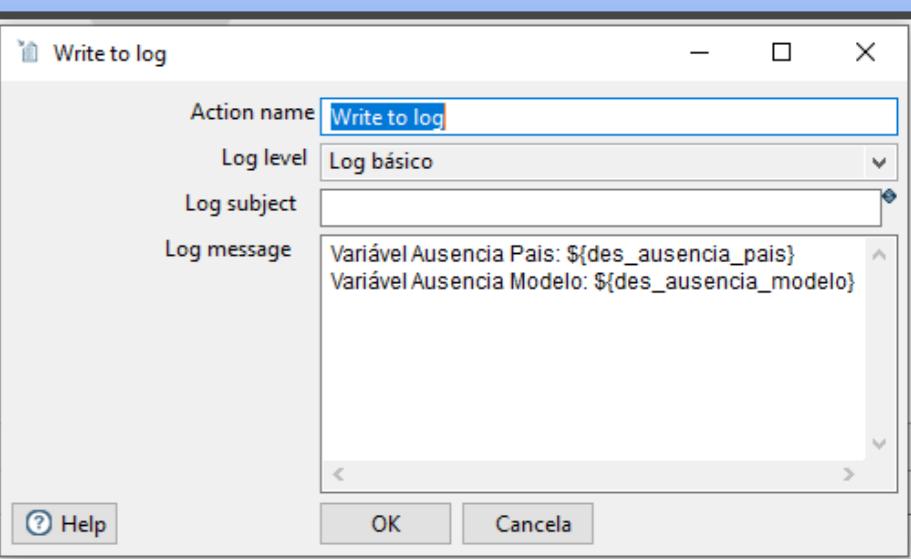
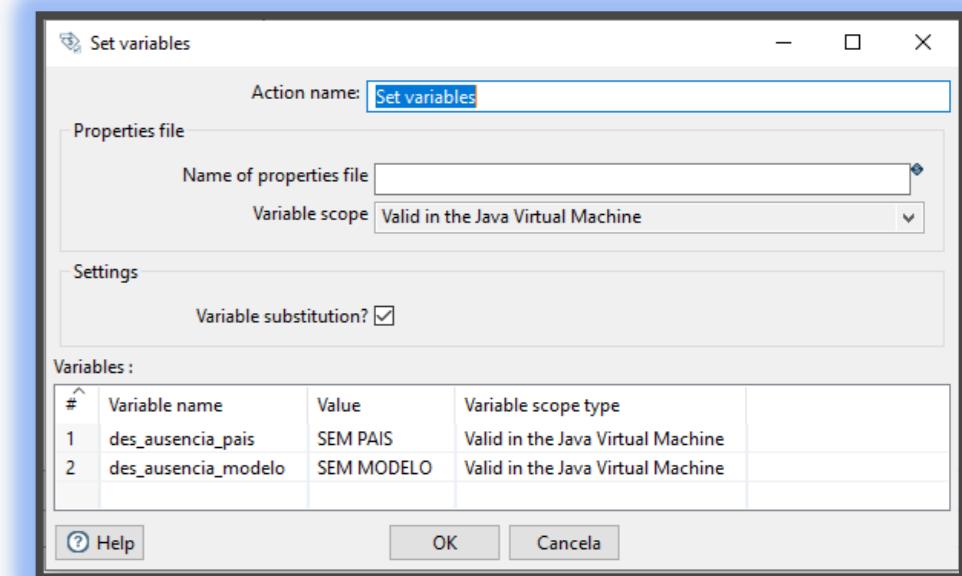
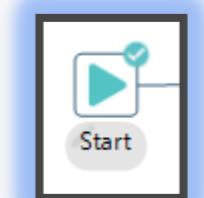
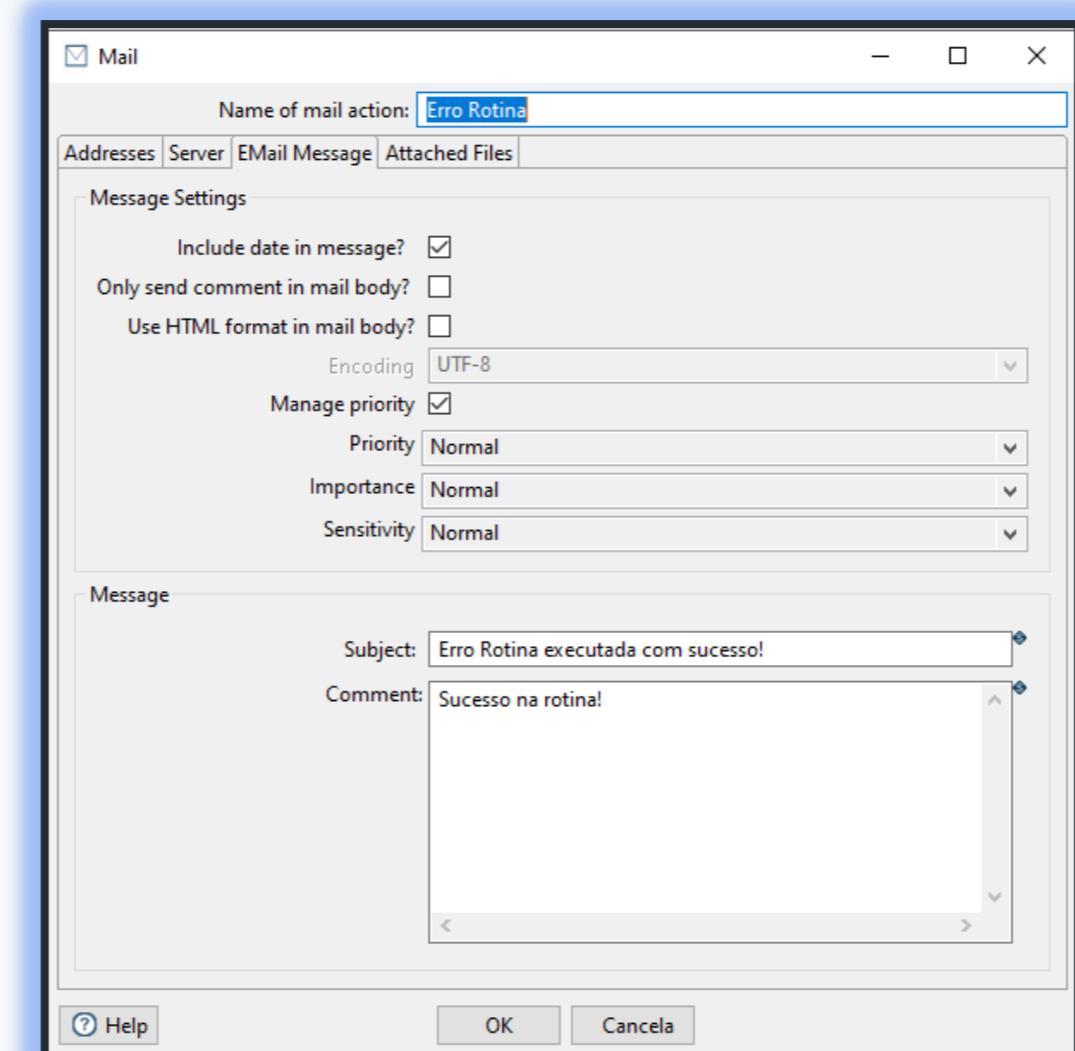
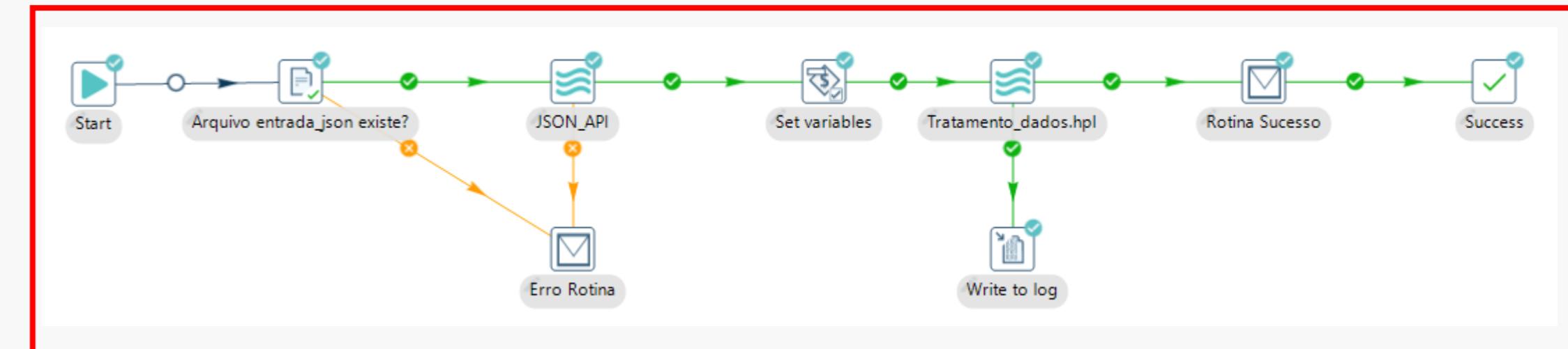


Trabalhando com Workflow

Vamos preparar um workflow de execução dos nossos pipelines.

Workflow de execução dos pipelines

Criaremos o encadeamento das rotinas para execução.

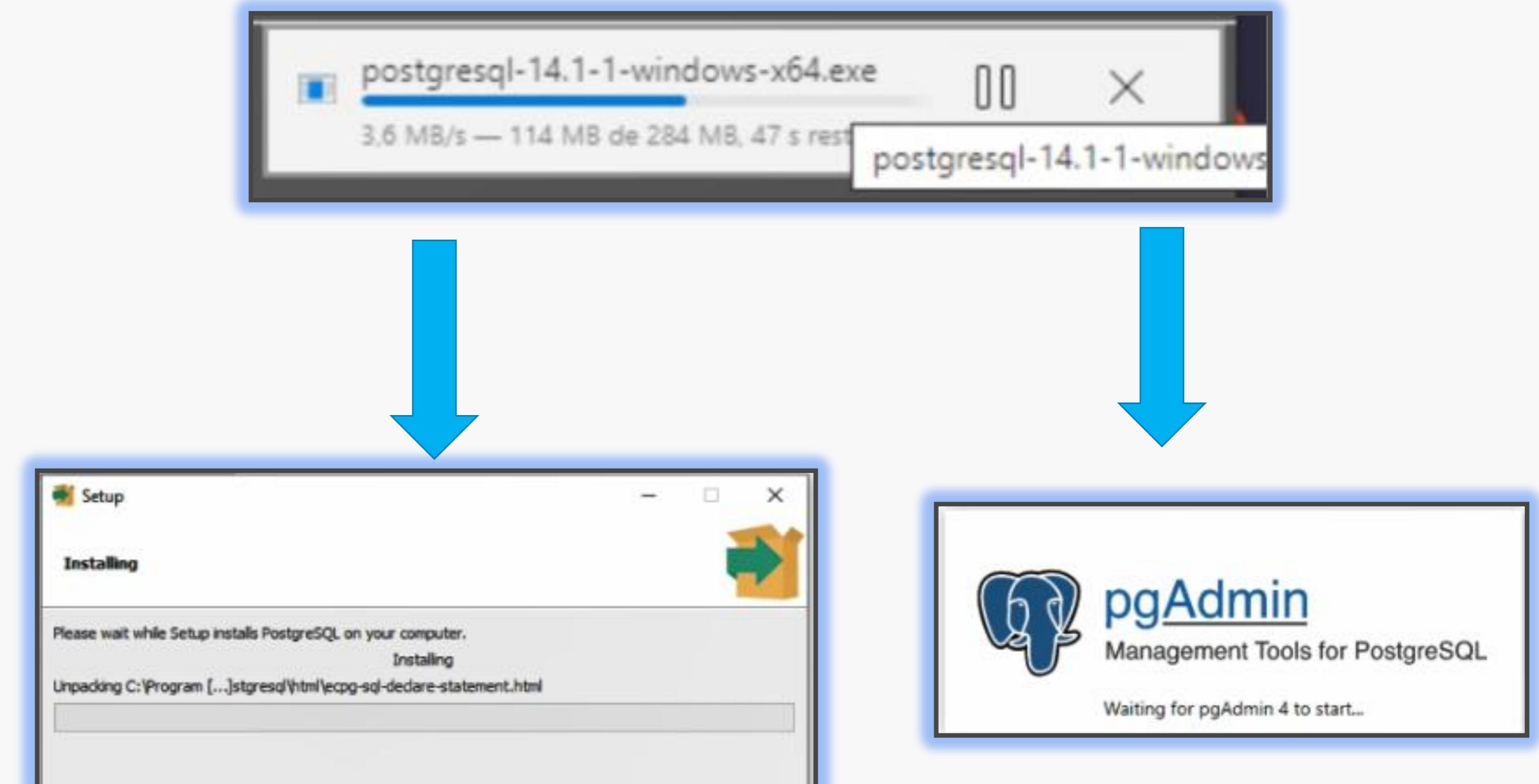


Instalando Postgresql

Vamos instalar o banco de dados.

Instalação do Postgresql

Utilizaremos este banco de dados para leitura e gravação.



Trabalhando com Pipeline tabelas de banco de dados

Vamos preparar um pipeline
mostrando as operações em banco de
dados.

Pipeline com as tabelas de banco de dados

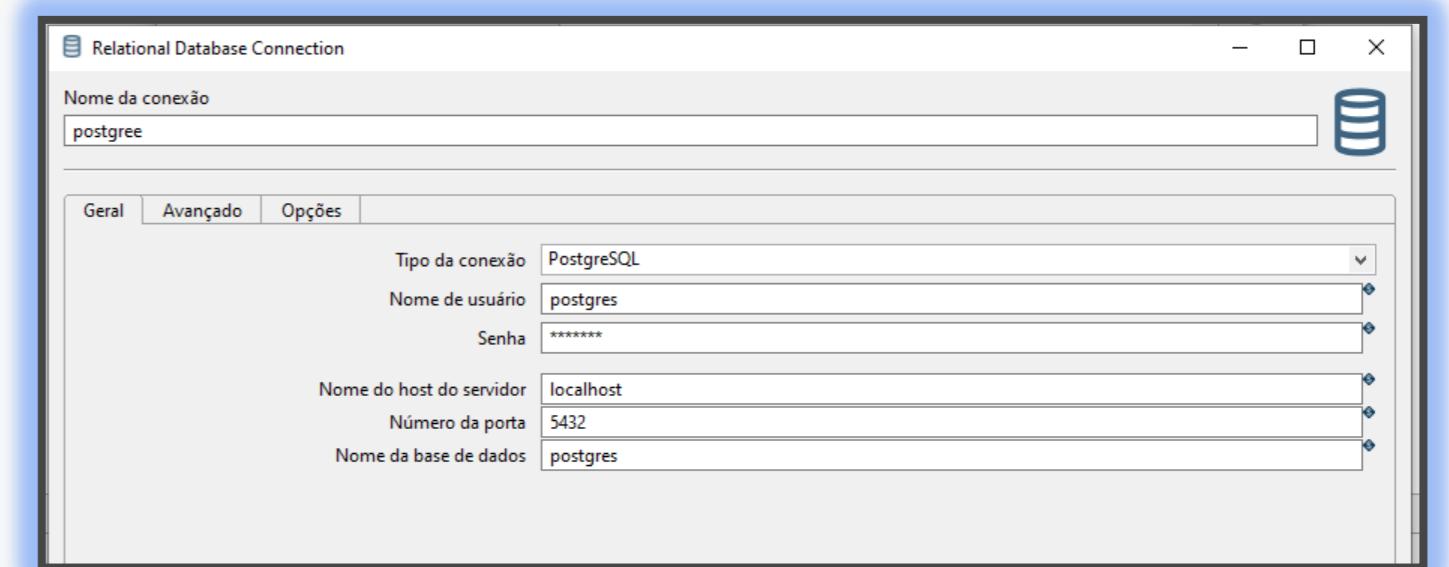
Configuração da conexão e execução dos script : tabelas e dados.

Query Editor Query History

```
1 SELECT "ID_AUTOR", "NOME", "SEXO", "DATA_NASCIMENTO"
2   FROM public."TB_AUTOR";
```

Data Output Explain Messages Notifications

ID_AUTOR	NOME	SEXO	DATA_NASCIMENTO
1	João	M	2020-07-10
2	Ana	F	2020-05-12
3	Sandra	F	2020-12-21
4	Carlos	M	2020-04-13
5	Fernanda	F	2020-06-05
6	Teresa	F	2020-01-02
7	Manoel	M	2020-03-22



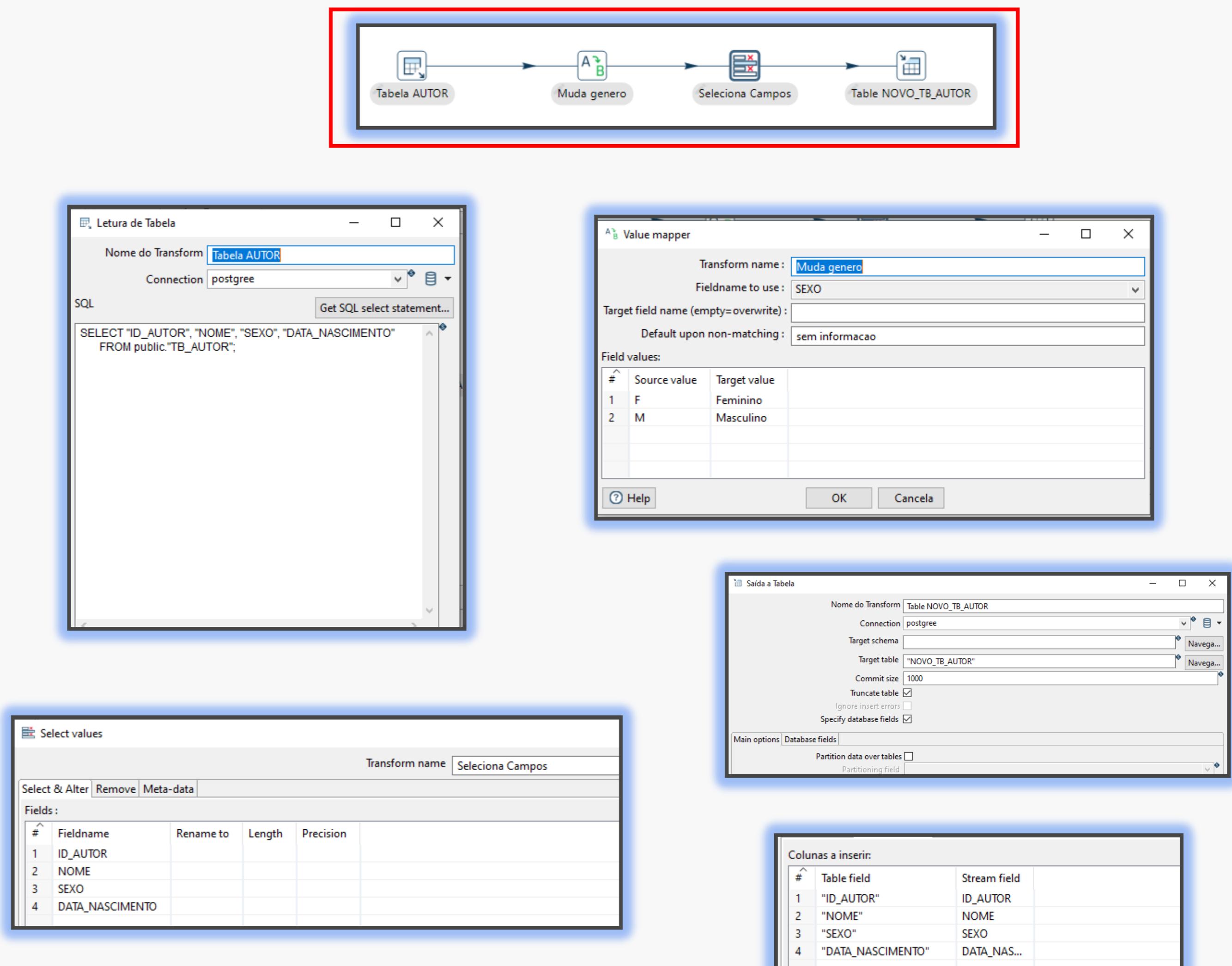
```
-- INSERTING into TB_AUTOR
INSERT INTO public."TB_AUTOR"("ID_AUTOR", "NOME", "SEXO", "DATA_NASCIMENTO") values (1,'João','M','10/07/20');
INSERT INTO public."TB_AUTOR"("ID_AUTOR", "NOME", "SEXO", "DATA_NASCIMENTO") values (2,'Ana','F','12/05/20');
INSERT INTO public."TB_AUTOR"("ID_AUTOR", "NOME", "SEXO", "DATA_NASCIMENTO") values (3,'Sandra','F','21/12/20');
INSERT INTO public."TB_AUTOR"("ID_AUTOR", "NOME", "SEXO", "DATA_NASCIMENTO") values (4,'Carlos','M','13/04/20');
INSERT INTO public."TB_AUTOR"("ID_AUTOR", "NOME", "SEXO", "DATA_NASCIMENTO") values (5,'Fernanda','F','05/06/20');
INSERT INTO public."TB_AUTOR"("ID_AUTOR", "NOME", "SEXO", "DATA_NASCIMENTO") values (6,'Teresa','F','02/01/20');
INSERT INTO public."TB_AUTOR"("ID_AUTOR", "NOME", "SEXO", "DATA_NASCIMENTO") values (7,'Manoel','M','22/03/20');
```

Trabalhando com Pipeline tabelas de banco de dados

Vamos preparar um pipeline mostrando as operações em banco de dados.

Pipeline com as tabelas de banco de dados

Configuração da conexão e execução dos script : tabelas e dados.



Automatizando Pipeline e Workflow

Automatizar pipeline e workflow

Automatizando Pipeline e Workflow

Criando uma automatização para as cargas.

```
===[Environment Settings - hop-run.bat]=======
Java identified as "C:\Program Files\Java\jdk-11.0.9\bin\java"
HOP_OPTIONS="-Xmx2048m" -DHOP_PLATFORM_OS=Windows -DHOP_PLATFORM_RUNTIME=Run -DHOP_AUTO_CREATE_CONFIG=Y

Consolidated parameters to pass to HopRun are
-j treinamento -r local -f C:\Users\grima\Downloads\Hop\Pipeline\trata_banco.hpl

Command to start HopRun will be:
"C:\Program Files\Java\jdk-11.0.9\bin\java" -classpath lib\*;libswt\win64\* -Djava.library.path=lib "-Xmx2048m" -DHOP_PLATFORM_OS=Windows -DHOP_PLATFORM_RUNTIME=Run -DHOP_AUTO_CREATE_CONFIG=Y org.apache.hop.run.HopRun -j treinamento -r local -f C:\Users\grima\Downloads\Hop\Pipeline\trata_banco.hpl

===[Starting HopRun]=======
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hop.core.plugins.HopURLClassLoader (file:/C:/Users/grima/Downloads/Hop/lib/hop-core-0.99.jar) to field java.net.URLClassLoader.ucp
WARNING: Please consider reporting this to the maintainers of org.apache.hop.core.plugins.HopURLClassLoader
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
2021/12/02 16:19:43 - HopRun - Enabling project 'Treinamento'
2021/12/02 16:19:54 - HopRun - Starting pipeline: C:\Users\grima\Downloads\Hop\Pipeline\trata_banco.hpl
2021/12/02 16:19:54 - trata_banco - Executing this pipeline using the Local Pipeline Engine with run configuration 'local'
2021/12/02 16:19:54 - trata_banco - Expedindo início para Pipeline [trata_banco]
2021/12/02 16:19:55 - Tabela NOVO_AUTOR.0 - Connected to database [treinamento] (commit=1000)
2021/12/02 16:19:55 - Tabela Autor.0 - Finished reading query, closing connection.
2021/12/02 16:19:55 - Tabela Autor.0 - Finished processing (I=7, O=0, R=0, W=7, U=0, E=0)
2021/12/02 16:19:55 - Muda genero.0 - Finished processing (I=0, O=0, R=7, W=7, U=0, E=0)
2021/12/02 16:19:55 - Seleciona campos.0 - Finished processing (I=0, O=0, R=7, W=7, U=0, E=0)
2021/12/02 16:19:55 - Tabela NOVO_AUTOR.0 - Finished processing (I=0, O=7, R=7, W=7, U=0, E=0)
2021/12/02 16:19:55 - trata_banco - Pipeline duration : 1.256 seconds [ 1.256" ]
2021/12/02 16:19:55 - trata_banco - Execution finished on a local pipeline engine with run configuration 'local'
```

HOP RUN

Link: <https://hop.apache.org/manual/latest/hop-run/index.html>

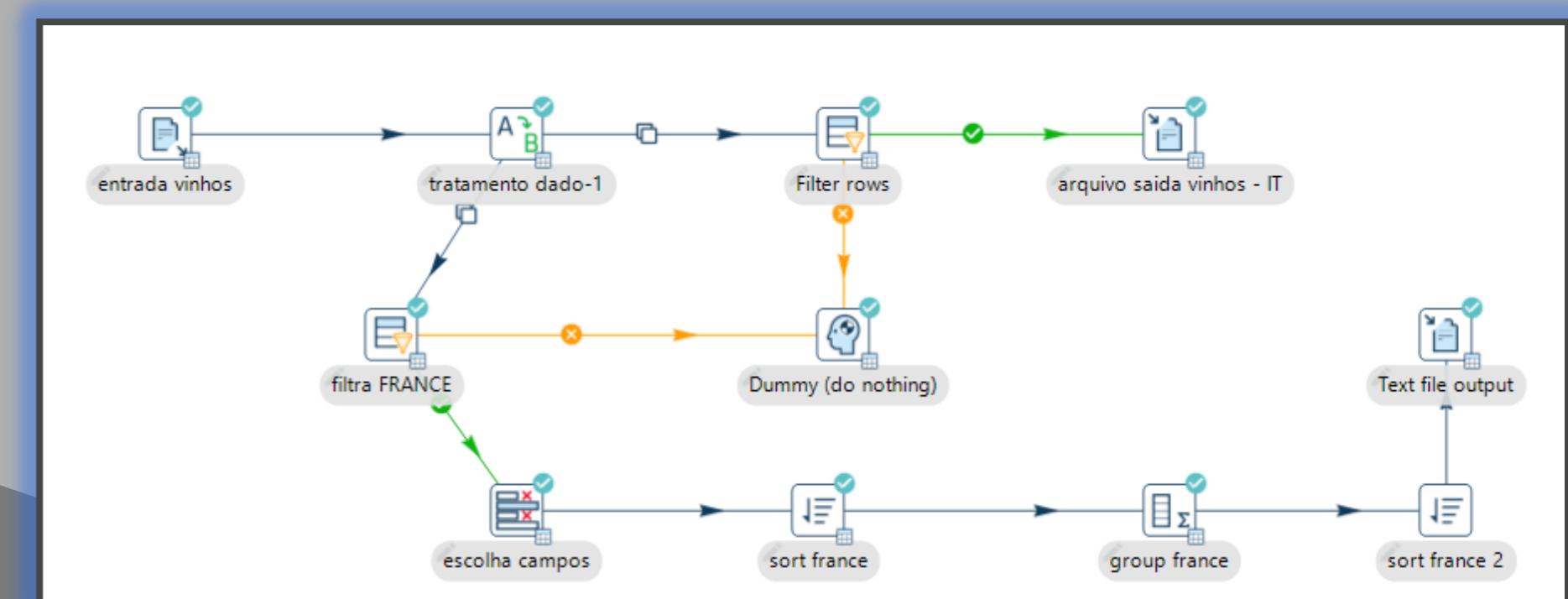
PRÁTICA

ENVIE AO PROFESSOR

PREPARE UM ESTUDO

Carregando qualquer arquivo do site **kaggle.com** e crie um Workflow com seus pipelines, grave um video e envie para correção.

Muito boa sorte e conte comigo!



```
7 - Hop - Pipeline opened.  
7 - Hop - Launching pipeline [estudo]...  
7 - Hop - Started the pipeline execution.  
7 - estudo - Executing this pipeline using the Local Pipeline Engine with run configuration 'local'  
7 - estudo - Expedindo inicio para Pipeline [estudo]  
7 - CSV file input.0 - Header row skipped in file 'C:\Users\grima\Downloads\Hop\Dados\vinhos_mundo.csv'  
8 - CSV file input.0 - Finished processing (I=129972, O=0, R=0, W=129971, U=0, E=0)  
8 - Value mapper.0 - Finished processing (I=0, O=0, R=129971, W=129971, U=0, E=0)  
8 - estudo - Pipeline duration : 0.805 seconds [ 0.805" ]  
8 - estudo - Execution finished on a local pipeline engine with run configuration 'local'
```

