



Curso:

Desenvolvendo Processos Avançados de ETL com Pentaho Data  
Integration  
(cód. ETL2000)

# Quem somos

- Empresa nacional com 10 anos de mercado
- Pioneira na América Latina no uso do Pentaho há mais de 10 anos
- Estamos localizados estrategicamente na região de Jundiaí/SP
- Especialista em dados
  - Integração
  - Qualidade
  - Enriquecimento
  - ETL
  - Ingestão
  - Data Prep
  - Big Data
  - Data Science



# Quem somos

## ■ Desenvolvimento

- In-house
- Fábrica

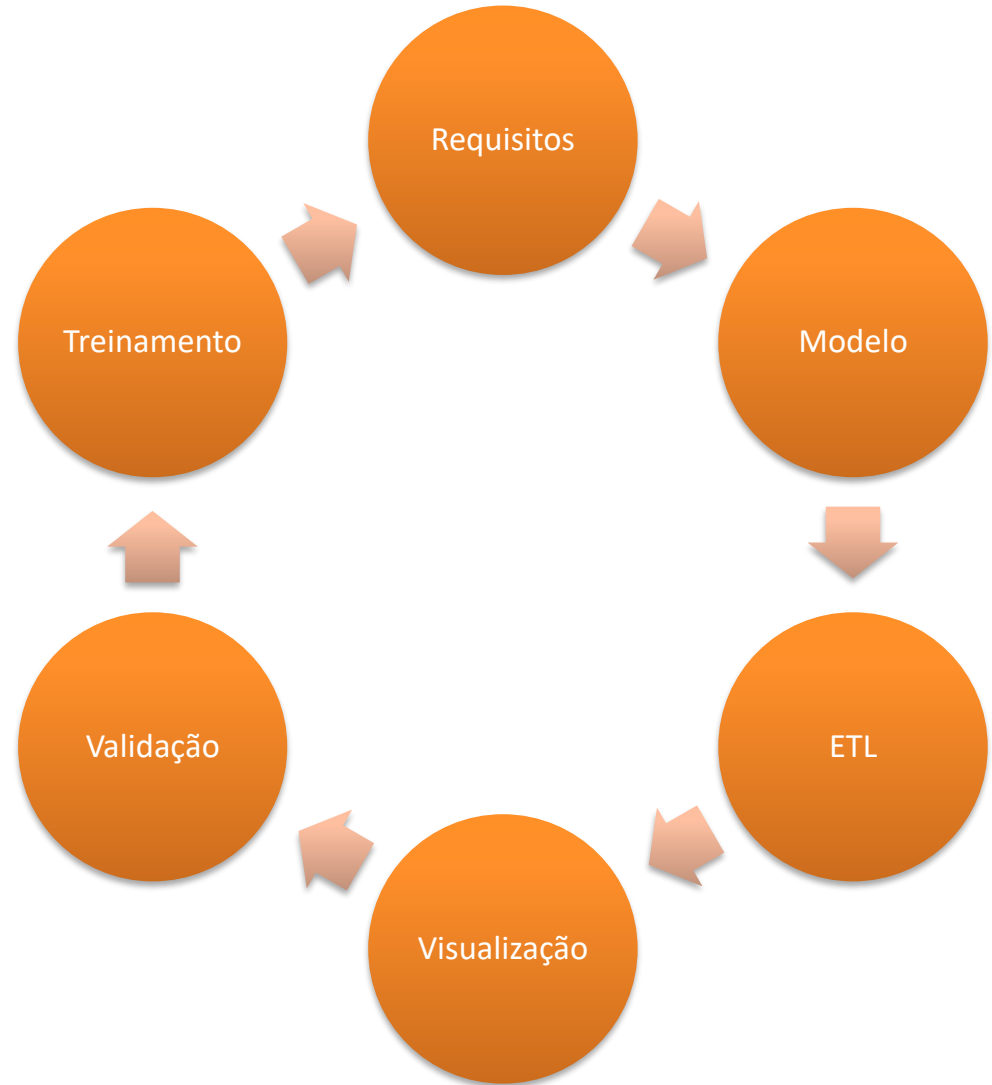
## ■ Treinamento

- EaD
- Online Ao Vivo
- Presencial
- Turmas Abertas/In-company

## ■ Suporte Especializado

- Ambiente Dev/QA/Prod
- Time de desenvolvimento

## ■ BlaaS (BI Como Serviço)



# Alguns de nossos Clientes



Governo

Software

Varejo

Indústria

Saúde

Outros



Os softwares utilizados

# Os softwares utilizados

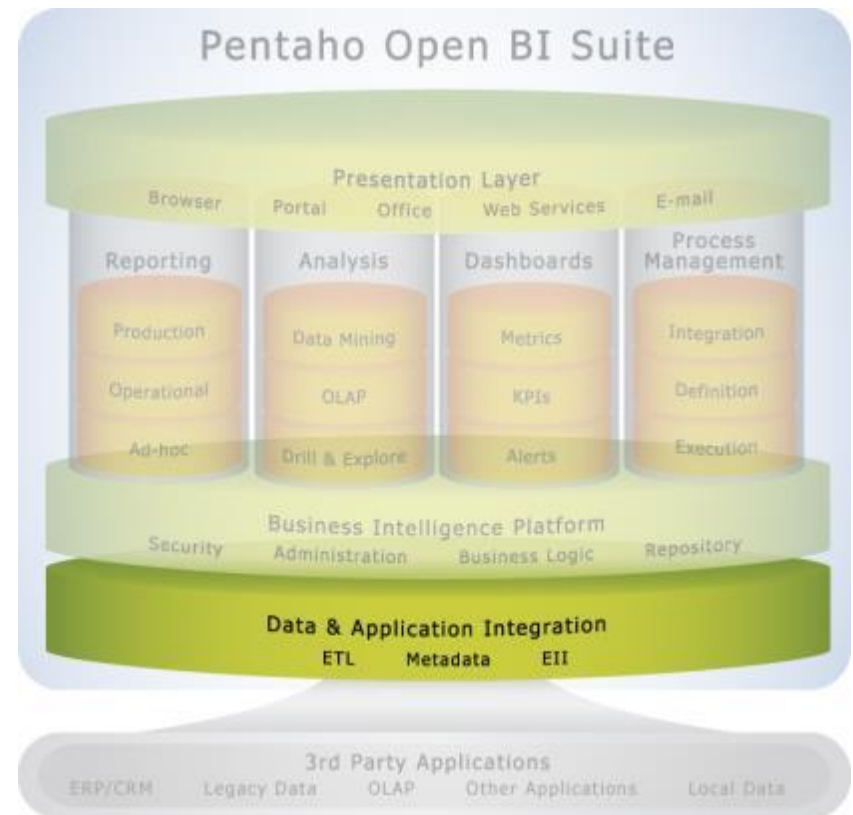
- Softwares Pentaho Community Edition
  - Pentaho Data Integration [package: pdi-ce-8.0.0.0-28.zip]



Apresentando o Pentaho Data Integration

# Apresentando o Pentaho Data Integration

- O mais popular projeto open source de integração de dados (Kettle)
  - Conduzido e patrocinado pela Pentaho
  - Possui uma larga contribuição da comunidade
  - Tecnologia em franca evolução

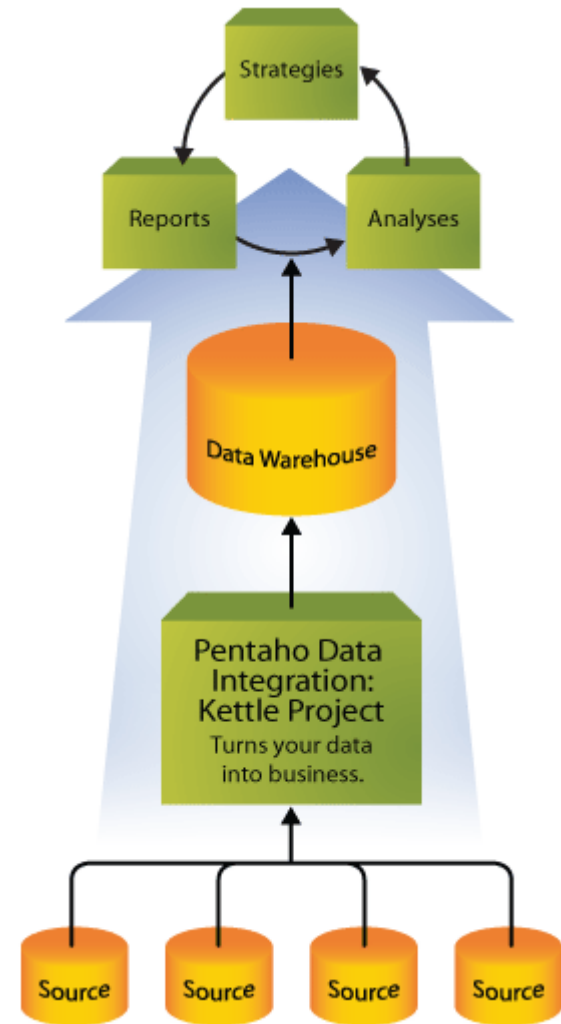




# Apresentando o Pentaho Data Integration

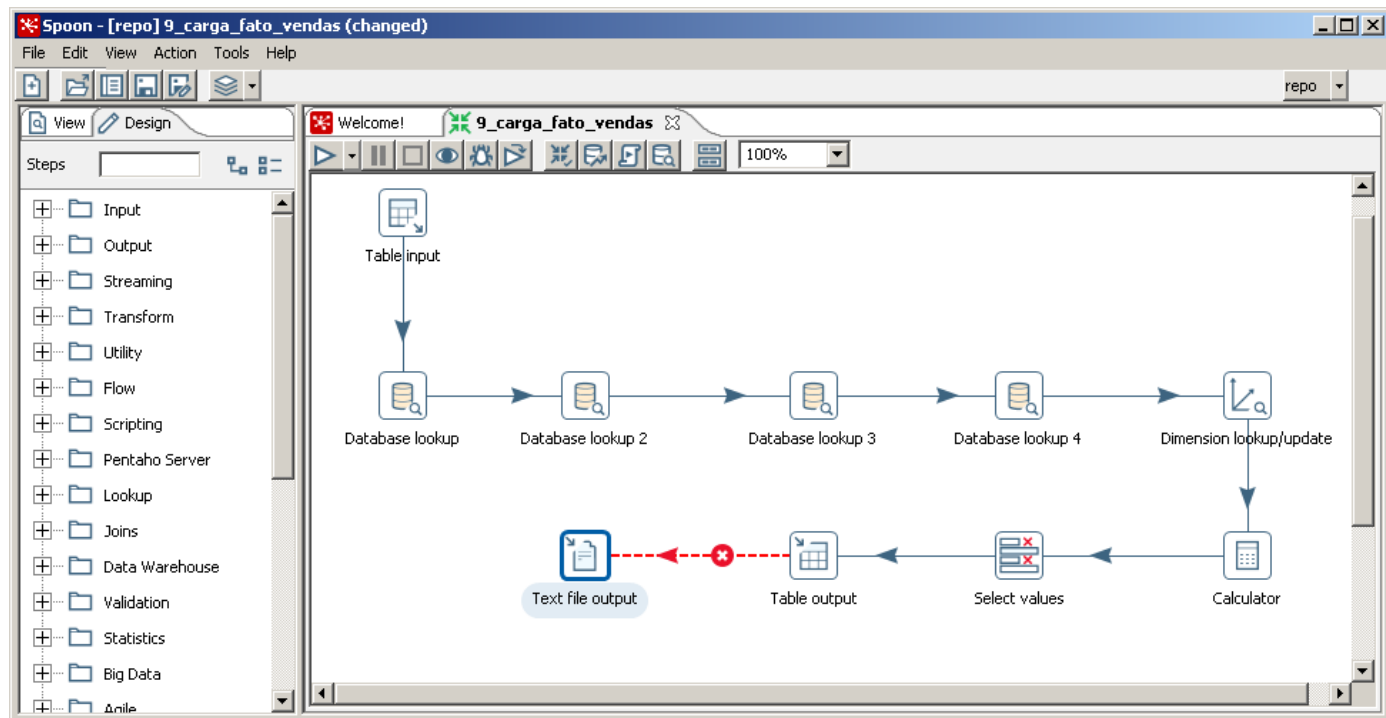
## Oportunidades de uso:

- Popular Data warehouse
- Exportar dados para vários formatos
- Importar dados oriundos de diversas fontes
- Migração de dados entre aplicações
- Integração de dados entre aplicações
- Apoio a metodologias ágeis
- Suporte ao Big Data



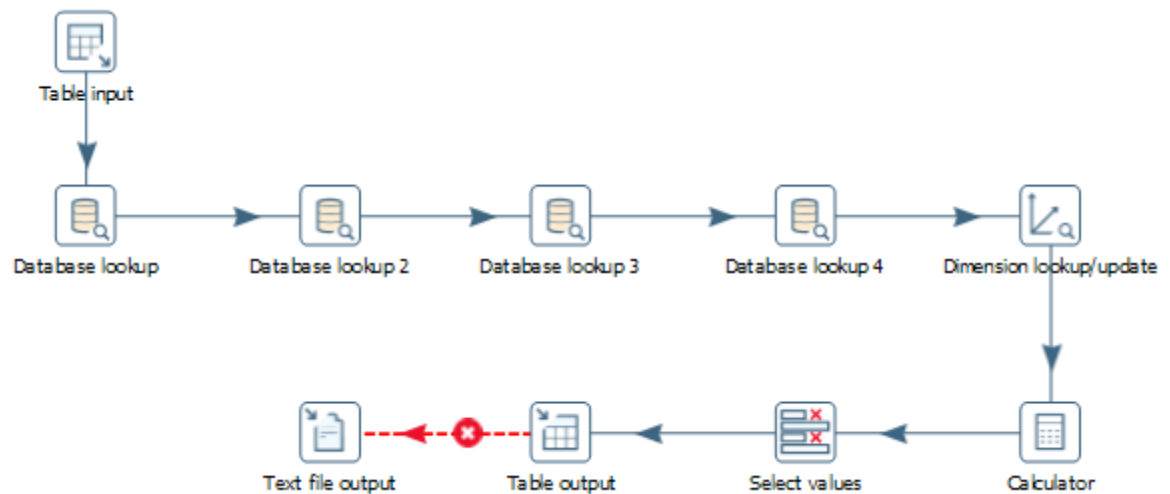
# Componentes do PDI - Spoon

- Spoon
- Pan
- Kitchen
- Carte



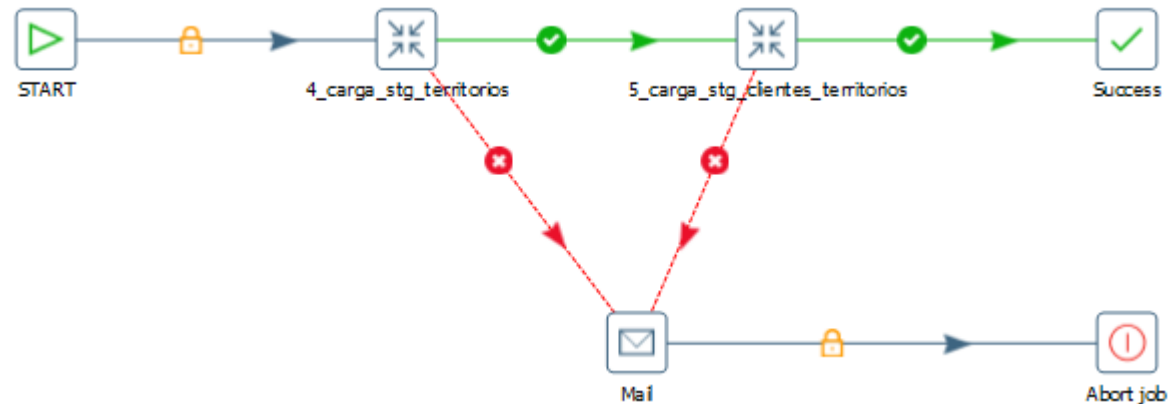
# Componentes do PDI – Pan

- Spoon
- **Pan**
- Kitchen
- Carte



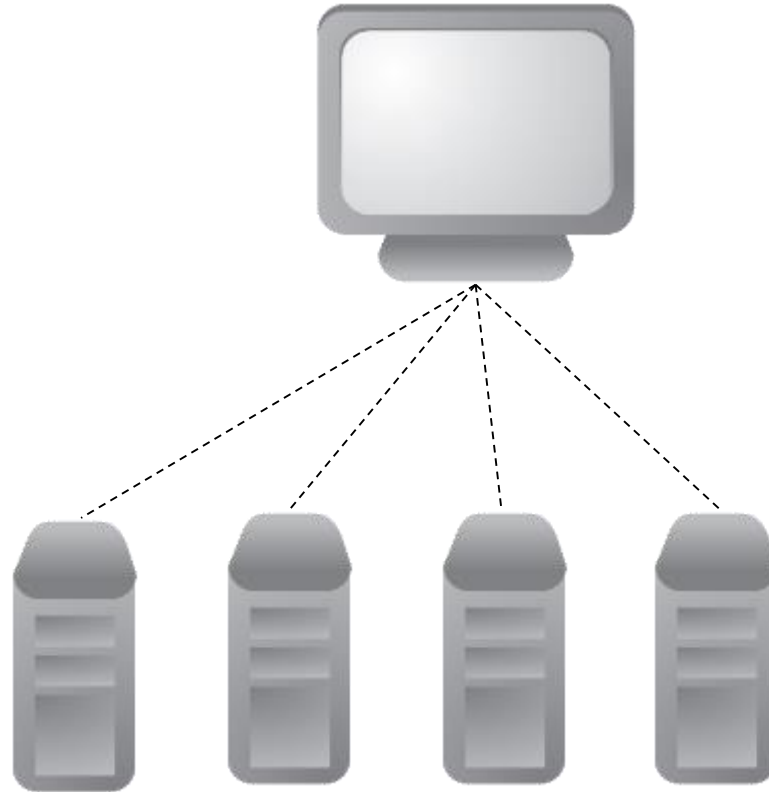
# Componentes do PDI – Kitchen

- Spoon
- Pan
- **Kitchen**
- Carte



# Componentes do PDI - Carte

- Spoon
- Pan
- Kitchen
- **Carte**



# Pré-requisitos

Para instalar o Pentaho Data Integration você deve possuir familiaridade em administração de sistemas e execução de comando via linha de comando.

## Software

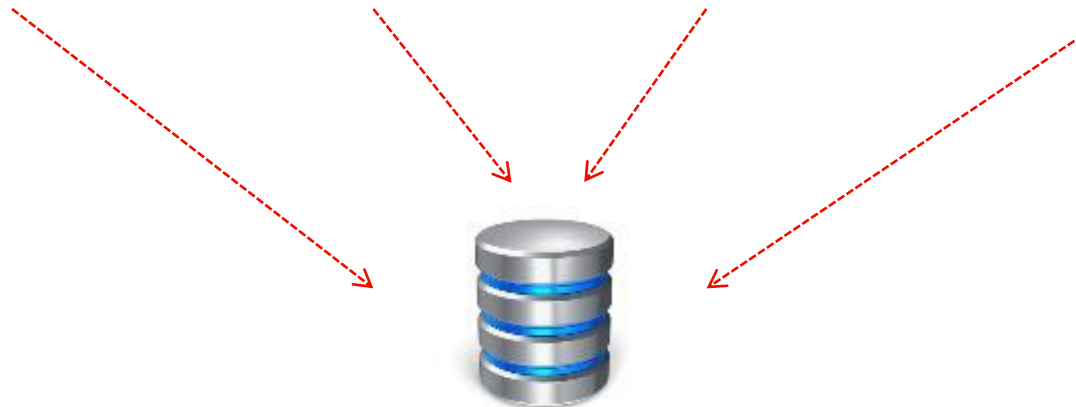
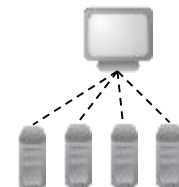
- Windows, Mac, Unix e Linux (32/64 Bits)
- Java Runtime Environment 1.8 (JRK8 - 32/64 Bits)

## Hardware

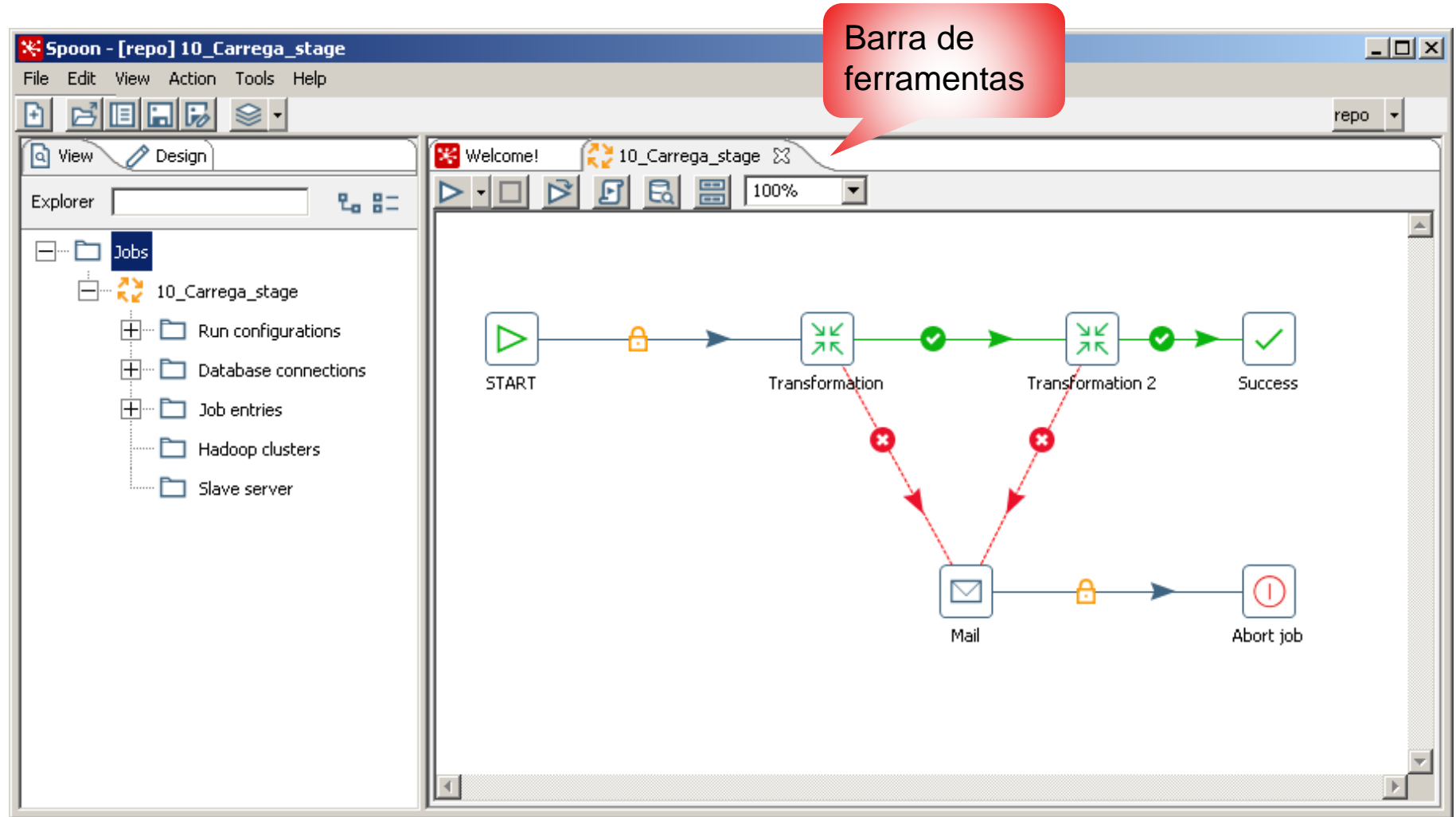
- 4 GB RAM (recomendado)
- 5 GB Espaço em disco (recomendado)
- Processador 32/64 Bits Dual-Core ou Core-2-Dual
- CPU 1.8GHz ou superior

# Usando e Iniciando o Spoon

- Em nosso treinamento faremos uso de repositório baseado em arquivo, todos o metadados estarão armazenados em arquivos de sistema com extensão KTR (Transformations) ou KJB (Jobs), em formato XML
- Iniciar **Spoon.bat** (Windows) ou **Spoon.sh** (Linux, MacOS) na pasta do Pentaho data Integration

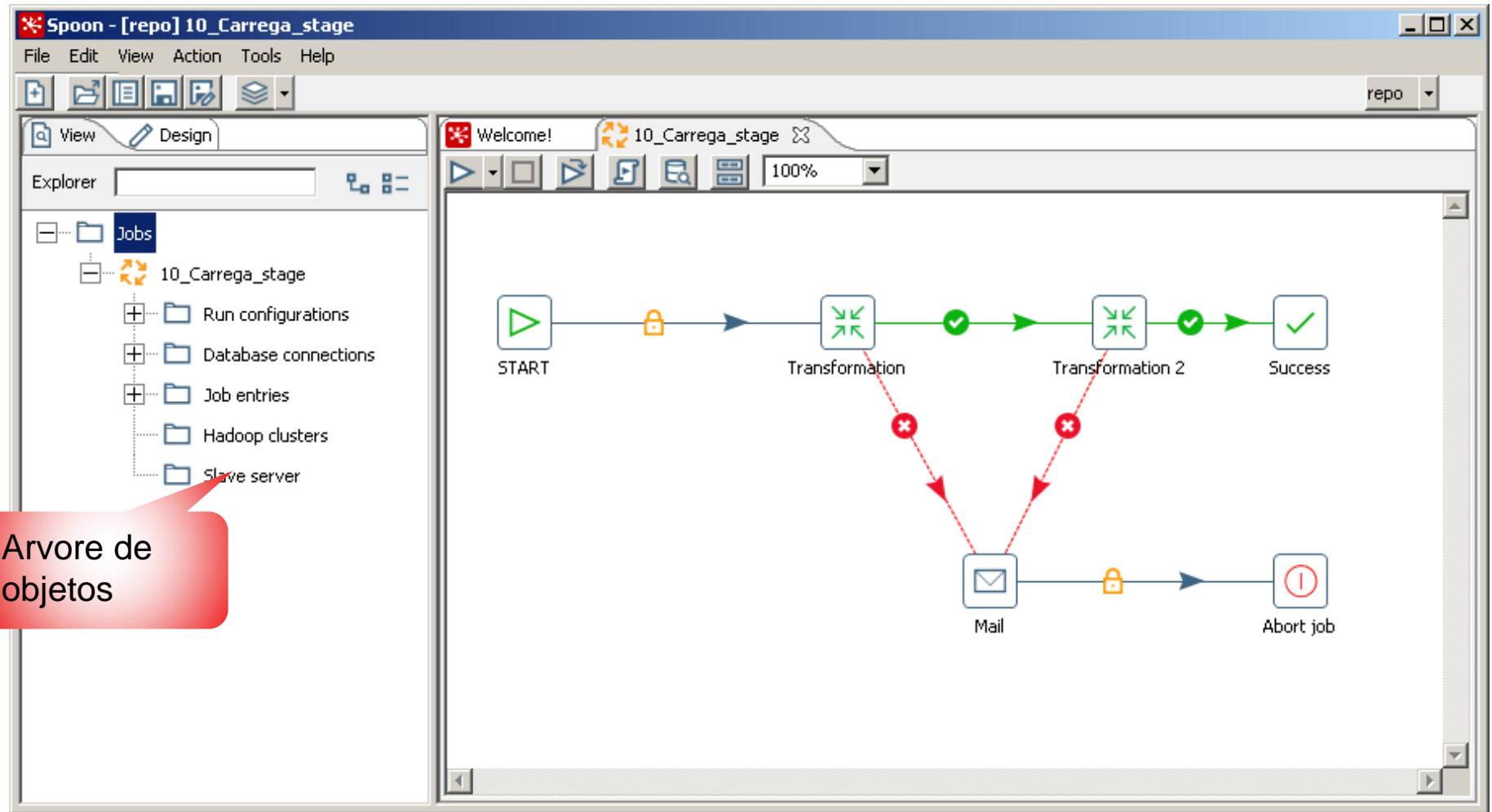


# A interface Spoon

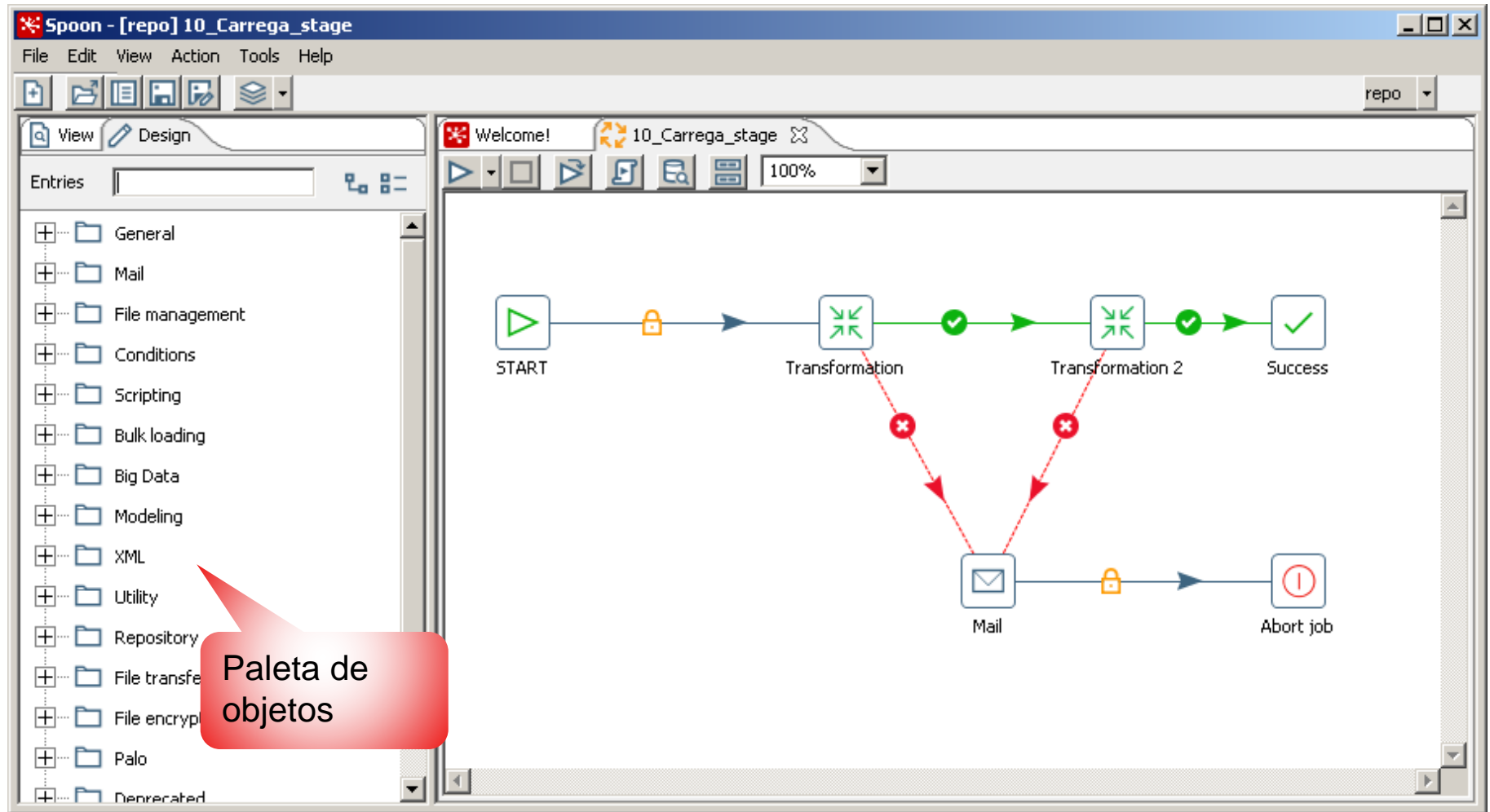




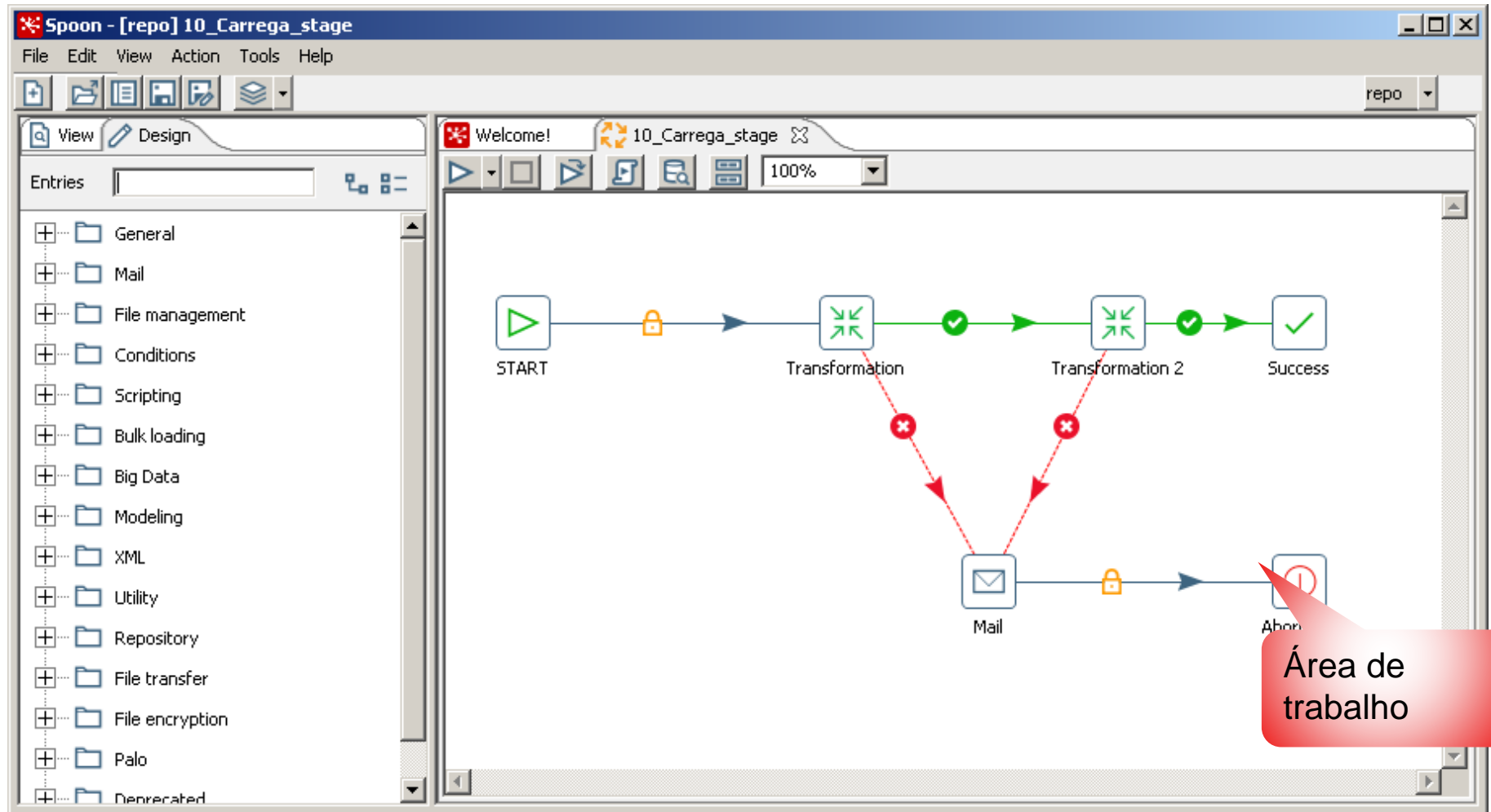
# A interface Spoon



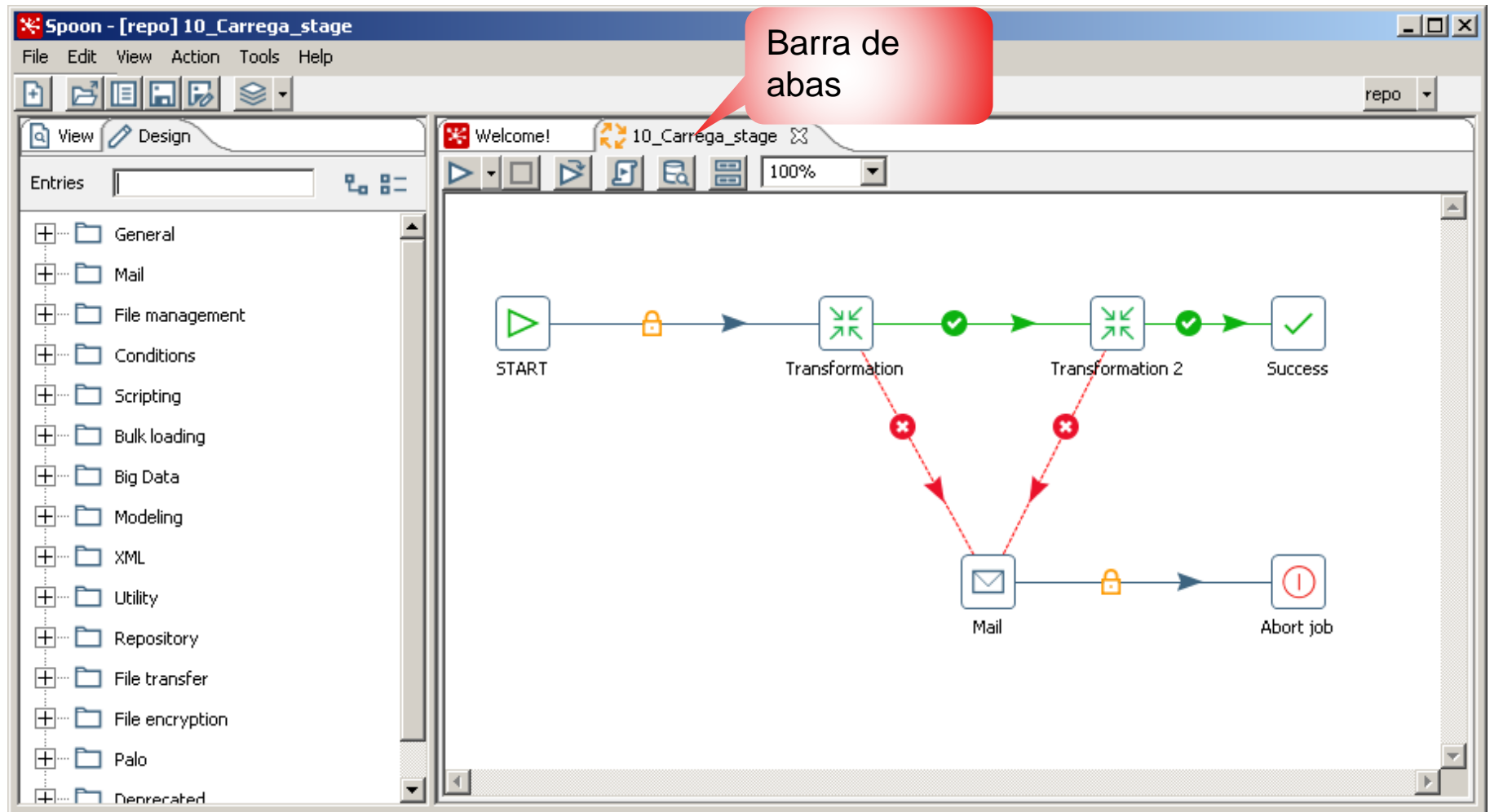
# A interface Spoon



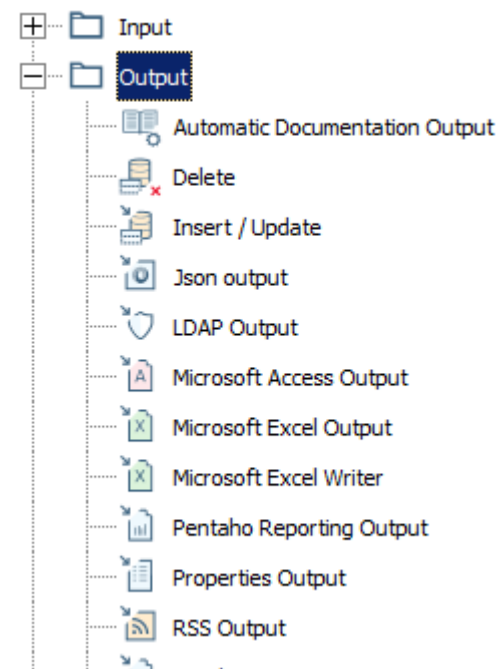
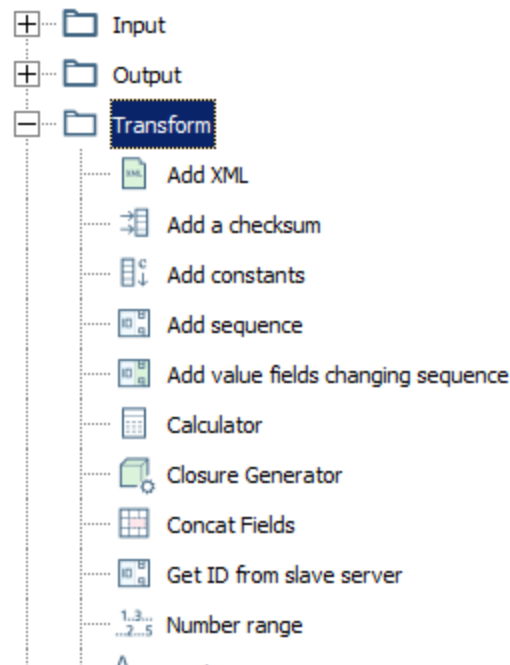
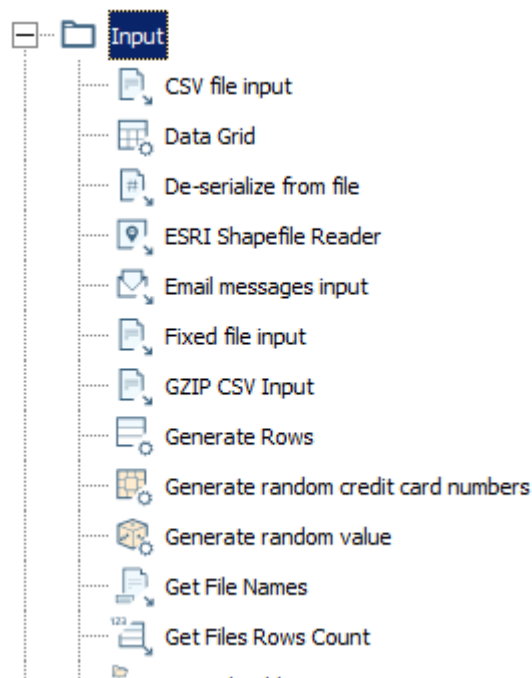
# A interface Spoon



# A interface Spoon



# Steps



# Perguntas e Respostas





Preparação complementar do ambiente

# Preparação complementar do ambiente

- Criar o repositório etl2000
- Restaurar o banco etl2000
- Criar a conexão etl2000





## Exercícios do treinamento

# Exercícios do treinamento

- Otimizando o tempo para:
  - atualização de grandes massas de dados
  - comparação de grandes massas de dados entre origem e destino
  - Aplicando multithreads
  - carga de grandes massas de dados através do paralelismo
- Extraíndo dados de fontes como:
  - Excel, CSV, **XML**, **HTTP** e Web Service
- Efetuando laços (loop) com transformações
- Usando o serviço CARTE para processamento remoto
- Usando sub-transformações para reutilização de processos prontos
- Monitorando o desempenho dos processos de ETL através das tabelas de log



# + Exercício

## ■ Inserindo e Atualizando dados de forma convencional

- O objetivo deste exercício é apresentar ao aluno um processo convencional de inserção e atualização de dados usando o step Insert/Update

Observem que na origem somente o código 7891151028400 possui duplicado com nomes de produtos diferentes

Anote o tempo final de execução da transformação

Tempo médio para a construção do exercício: **15 minutos**

Complexidade para a construção do exercício: **baixa**



# - Exercício

## Iniciar uma transformação nova

Microsoft Excel input

Step name:

Files | Sheets | Content | Error Handling | Fields | Additional output fields

Spread sheet type (engine):

File or directory:  Add Browse...

Regular Expression:

Exclude Regular Expression:

Selected files:

#	File/Directory	Wildcard (RegExp)
1	C:\treinamento\materiais\etl2000\Cadastro_Produtos_Supermercados_NCM.xlsx	

Delete Edit

Accept filenames from previous steps

Accept filenames from previous step ☐

Step to read filenames from:

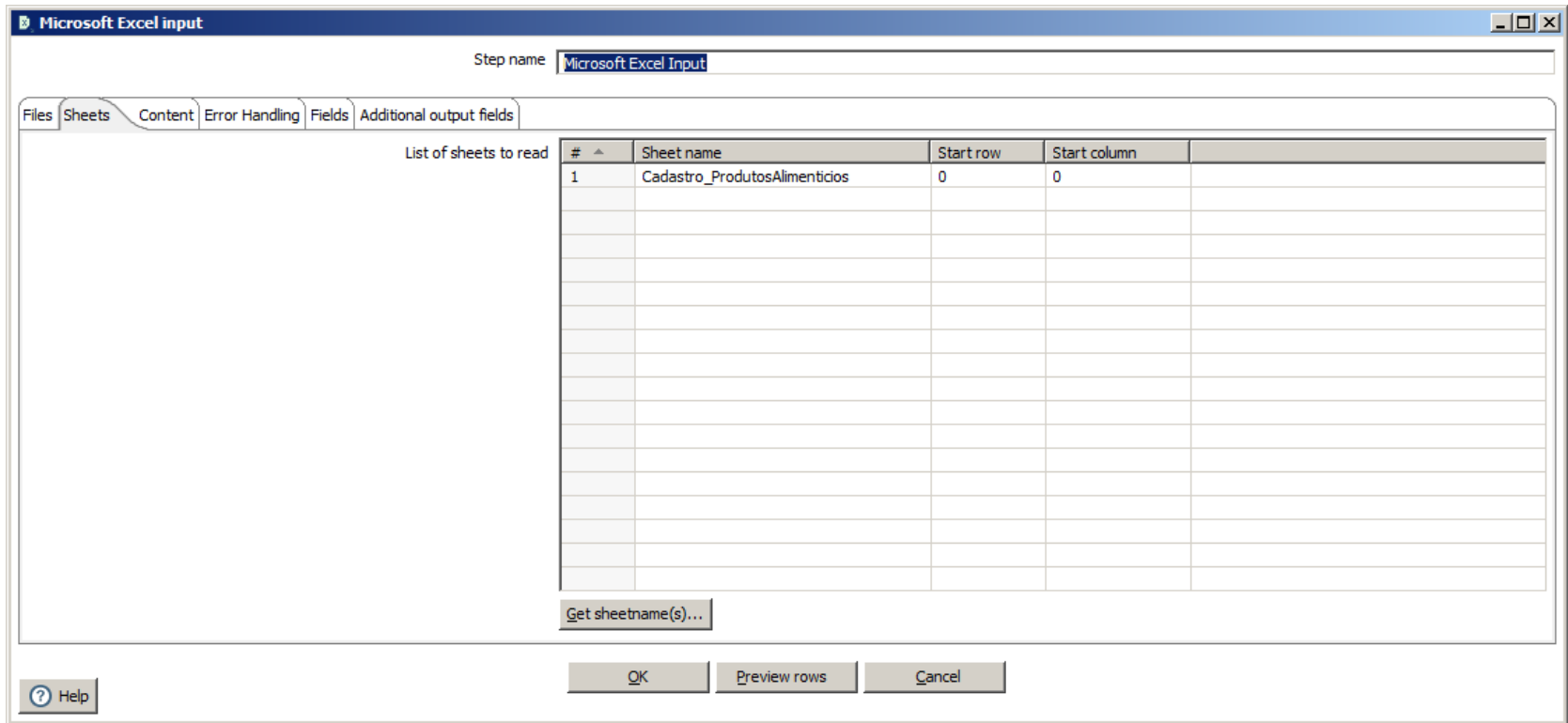
Field in the input to use as filename:

Show filename(s)...

Help OK Preview rows Cancel



## - Exercício



## - Exercício

[illegible]

# - Exercício

Salvar como exercicio1a

**Insert / Update**

Step name:

Connection:

Target schema:

Target table:

Commit size:

Don't perform any updates: ☐

The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1	Stream field2
1	codigo	=	Código	

Update fields:

#	Table field	Stream field	Update
1	ncm	NCM	Y
2	codigo	Código	N
3	descricao	Descrição	Y





# + Exercício

## ■ Trabalhando com Multi-threads em transformações

- O objetivo deste exercício é apresentar ao aluno como escalar o processamento usando a opção de multi-thread de steps em uma transformação

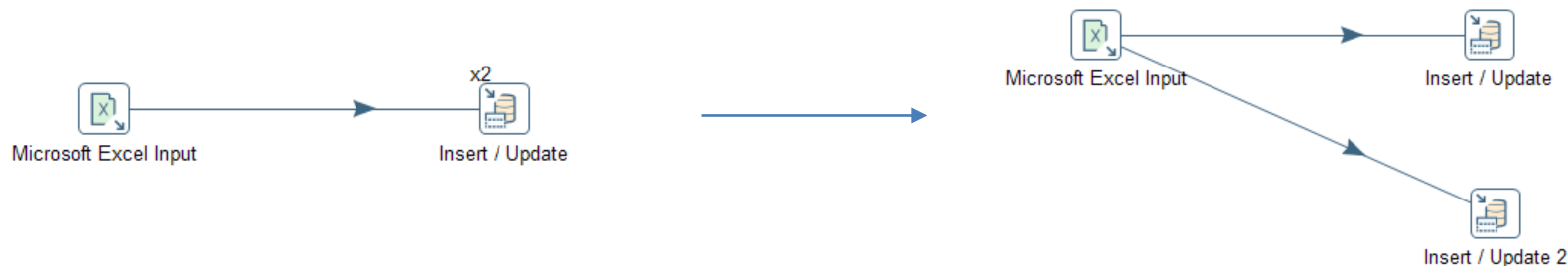
Não usar a opção de multi-threads em steps como Sort rows, Unique rows e Row denormalizer são bons exemplos, pois você precisa aplicar a regra em todas as linhas

Continue com o exercício1a, salve como exercício1b e anote o tempo de execução

Zere a tabela para ter o mesmo cenário da execução anterior

Tempo médio para a construção do exercício: **10 minutos**

Complexidade para a construção do exercício: **baixa**



# + Exercício

## ■ Inserindo e atualizando dados com maior performance

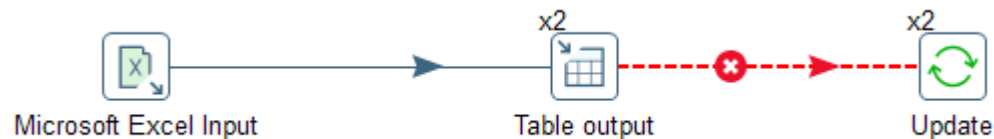
- O objetivo deste exercício é apresentar ao aluno como fazer para obter uma performance melhor utilizando o recurso do tratamento de exceção de erro

Anote o tempo final de execução da transformação

Zere a tabela para ter o mesmo cenário da execução anterior

Tempo médio para a construção do exercício: **20 minutos**

Complexidade para a construção do exercício: **baixa**



# - Exercício

## Iniciar uma transformação nova

Microsoft Excel input

Step name: Microsoft Excel Input

Files | Sheets | Content | Error Handling | Fields | Additional output fields

Spread sheet type (engine): Excel 2007 XLSX (Apache POI)

File or directory:  Add Browse...

Regular Expression:

Exclude Regular Expression:

Selected files:

#	File/Directory	Wildcard (RegExp)
1	C:\treinamento\materiais\eti2000\Cadastro_Produtos_Supermercados_NCM.xlsx	

Delete Edit

Accept filenames from previous steps

Accept filenames from previous step ☐

Step to read filenames from:

Field in the input to use as filename:

Show filename(s)...

OK Preview rows Cancel

Help



## - Exercício

**Microsoft Excel input**

Step name: Microsoft Excel Input

Files | Sheets | Content | Error Handling | Fields | Additional output fields

List of sheets to read

#	Sheet name	Start row	Start column
1	Cadastro_ProdutosAlimenticios	0	0

Get sheetname(s)...

OK Preview rows Cancel

? Help



## - Exercício

[illegible]

## - Exercício

Table output

Step name

Table output

Connection

etl2000

Edit...

New...

Wizard...

Target schema

Browse...

Target table

exercicio1

Browse...

Commit size

100

Truncate table

☐

Ignore insert errors

☐

Specify database fields

☒

Main options

Database fields

Partition data over tables

☐

Partitioning field

Partition data per month

☐

Partition data per day

☐

Use batch update for inserts

☒

Is the name of the table defined in a field?

☐

Field that contains name of table:

Store the tablename field

☒

Return auto-generated key

☐

Name of auto-generated key field

?

Help

OK

Cancel

SQL

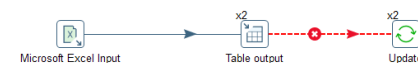
Table output

Main options

Database fields

Fields to insert:

#	Field name	Table
1	desc	Table
2	ncm	Table
3	codig	Table

[illegible]

# - Exercício

Salvar como exercicio1c

**Update**

Step name: Update

Connection: etl2000 [Edit...] [New...] [Wizard...]

Target schema: [Browse...]

Target table: exercicio1 [Browse...]

Commit size: 100

Use batch updates? ☒

Skip lookup? ☒

Ignore lookup failure? ☐ Flag field (key found) [ ]

The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1	Stream field2
1	codigo	=	Código	

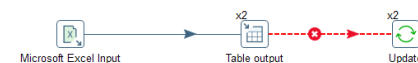
[Get fields]

Update fields:

#	Table field	Stream field
1	codigo	Código
2	descricao	Descrição
3	ncm	NCM

[Get update fields]

[?] Help [OK] [Cancel] [SQL]



# + Exercício

## ■ Deduplicando uma massa de dados

- O objetivo deste exercício é apresentar ao aluno um caso de uso para deduplicação dos dados

Tempo médio para a construção do exercício: **15 minutos**

Complexidade para a construção do exercício: **baixa-média**





# - Exercício

## Iniciar uma transformação nova

Microsoft Excel input

Step name:

Files | Sheets | Content | Error Handling | Fields | Additional output fields

Spread sheet type (engine):

File or directory:  Add Browse...

Regular Expression:

Exclude Regular Expression:

Selected files:

#	File/Directory	Wildcard (RegExp)
1	C:\treinamento\materiais\eti2000\Cadastro_Produtos_Supermercados_NCM.xlsx	

Delete Edit

Accept filenames from previous steps

Accept filenames from previous step ☐

Step to read filenames from:

Field in the input to use as filename:

Show filename(s)...

Help OK Preview rows Cancel



## - Exercício

[illegible]

## - Exercício

[illegible]

# - Exercício

**Sort rows**

Step name:

Sort directory:

TMP-file prefix:

Sort size (rows in memory):

Free memory threshold (in %):

Compress TMP Files? ☐

Only pass unique rows? (verifies keys only) ☐

Fields :

#	Fieldname	Ascending	Case sensitive compare?	Presorted?
1	Código	Y	N	N



# - Exercício

**Add fields changing sequence**

Step name:

Result field:

Start at value:

Increment by:

Init sequence if value of following fields change

#	Field
1	Código



# - Exercício

**Filter rows**

Step name:

Send 'true' data to step:

Send 'false' data to step:

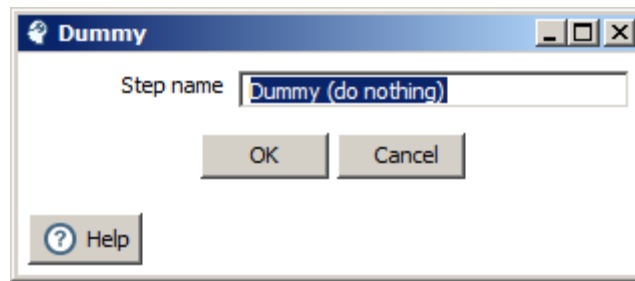
The condition:

(Number)



# - Exercício

Salvar como exercicio4



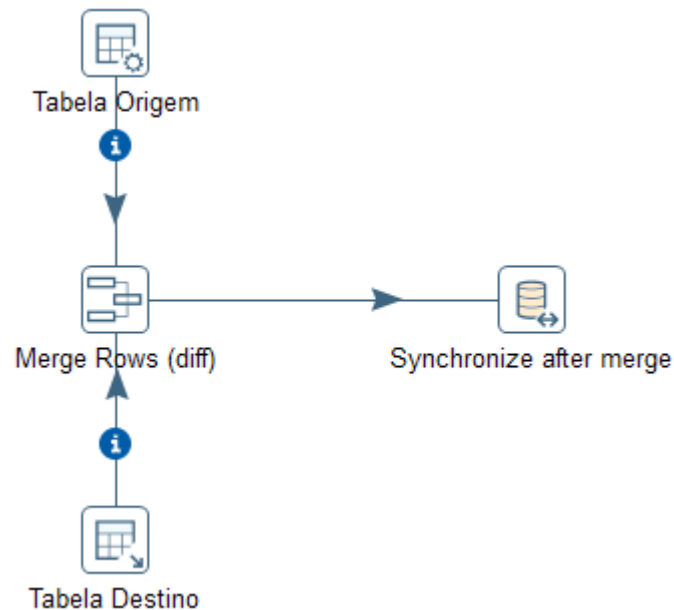
# + Exercício

## ■ Comparando dados entre Origem e Destino

- O objetivo deste exercício é apresentar ao aluno como trabalhar num ambiente onde há necessidade de comparação de grande massa de dados e nenhum dos recursos de CDC e Timestamp estão disponíveis

Tempo médio para a construção do exercício: **20 minutos**

Complexidade para a construção do exercício: **alta**





# - Exercício

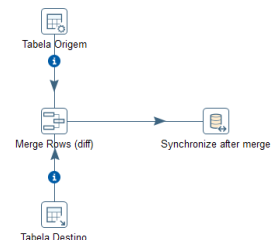
Iniciar uma transformação nova

**Add constant rows**

Step name:

Meta Data

	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Set empty string?
1	id	Integer							
2	nome	String		30					
3	descricao	String		30					



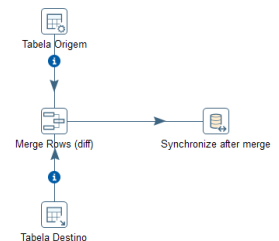
# - Exercício

**Add constant rows**

Step name:

Meta Data

#	id	nome	descricao
1	4	Produto D	Descricao D. 1
2	5	Produto E	Descricao E
3	6	Produto F	Descricao F



# - Exercício

**Table input**

Step name:

Connection:

SQL

```
SELECT
  id
, nome
, descricao
FROM exercicio5
order by id
```

Line 1 Column 0

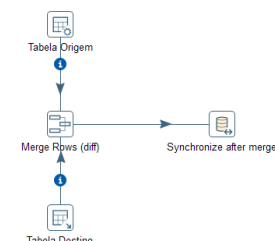
Enable lazy conversion ☐

Replace variables in script? ☐

Insert data from step

Execute for each row? ☐

Limit size



# - Exercício

**Merge rows (diff)**

Step name:

Reference rows origin:

Compare rows origin:

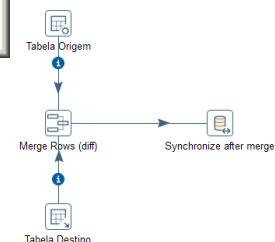
Flag fieldname:

Keys to match :

#	Key field
1	id

Values to compare :

#	Value field
1	nome
2	descricao



# - Exercício

**Synchronize after merge**

Step name: Synchronize after merge

General Advanced

Connection: etl2000 Edit... New... Wizard...

Target schema: Browse...

Target table: exercicio5 Browse...

Commit size: 100

Use batch update ☒

Tablename is defined in a field ☐

Tablename field:

The key(s) to look up the value(s):

#	Table field	Comparator	Stream field1	Stream field2
1	id	=	id	

Get fields

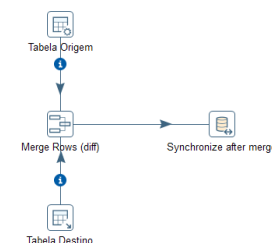
Update fields:

#	Table field	Stream field	Update
1	id	id	N
2	nome	nome	Y
3	descricao	descricao	Y

Get update fields

Edit mapping

Help OK SQL Cancel



# - Exercício

Salvar como exercicio5

Synchronize after merge

Step name Synchronize after merge

General Advanced

Operation

Operation fieldname flagfield

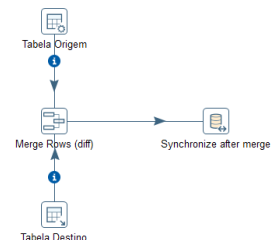
Insert when value equal new

Update when value equal changed

Delete when value equal deleted

Perform lookup ☐

Help OK SQL Cancel



# + Exercício

- Criando o processo de captura de coordenadas geográficas com Google Maps
  - O objetivo deste exercício é apresentar ao aluno como utilizar uma steps como javascript, http client e xml para recuperar coordenadas geográficas usando o Google Maps

Tempo médio para a construção do exercício: **20 minutos**

Complexidade para a construção do exercício: **média**



# - Exercício

Iniciar uma transformação nova

**Generate Rows**

Step name:

Limit:

Never stop generating rows: ☐

Interval in ms (delay):

Current row time field name:

Previous row time field name:

Fields :

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value	Set empty string?
1	bairro	String							Vila Imape	N
2	logradouro	String							Rua Nove de Julho	N
3	numero	String							319	N
4	cep	String							13231130	N
5	uf	String							SP	N
6	localidade	String							Campo Limpo Paulista	N





# - Exercício

Script Values / Mod

Step name: Cria URL Maps

Java script functions :

- Transform Scripts
- Transform Constants
- Transform Functions
- Input fields
  - bairro
  - logradouro
  - numero
  - cep
  - uf
  - localidade
- Output fields

Please use the 'Replace value' Field

Java script :

```
Script 1
//Script here
var endereco = logradouro + ', ' + numero + ', ' + localidade;
endereco = endereco.replace(/ /gi, '%20');
var url_final = 'http://maps.google.com/maps/api/geocode/xml?sensor=false&address=' + endereco;
```

Linerr: 0

Compatibility mode? ☐ Optimization level 9

Fields

#	Fieldname	Rename to	Type	Length	Precision	Replace value 'Fieldname' or 'Rename to'
1	endereco		String			N
2	url_final		String			N

Help OK Cancel Get variables Test script



# - Exercício

HTTP Client

Step name: HTTP Client

General Fields

Settings

URL:

Accept URL from field? ☒

URL field name: url\_final

Encoding (empty means standard): UTF-8

Connection timeout: 10000

Socket timeout: 10000

Connection close wait time: -1

Output fields

Result field name: result

HTTP status code field name:

Response time (milliseconds) field:

Response header field name:

HTTP authentication

Http Login:

HTTP Password:

Proxy to use

Proxy Host:

Proxy Port:

Help OK Cancel



# - Exercício

Colocar o nome do proximo slide

Get data from XML

Step name: Get data from XML

File Content Fields Additional output fields

XML source from field

XML source is defined in a field? ☒

XML source is a filename? ☐

Read source as Url ☐

get XML source from a field: result

File or directory: Add Browse

Regular Expression:

Exclude Regular Expression:

Selected files:

	File/Directory	Wildcard (RegExp)	Exclude wildcard	Required	Include subfolders	
1						

Delete Edit

Show filename(s)...

OK Preview rows Cancel

Help



# - Exercício

Get XML Data

Step name: Trata XML (Coordenadas)

File Content Fields Additional output fields

Settings

Loop XPath: /GeocodeResponse/result/geometry/location Get XPath nodes

Encoding: UTF-8

Namespace aware? ☐

Ignore comments? ☐

Validate XML? ☐

Use token ☐

Ignore empty file ☐

Do not raise an error if no files ☒

Limit: 0

Prune path to handle large files:

Additional fields:

Help OK Preview rows Cancel



# - Exercício

Get data from XML

Step name:

File Content Fields Additional output fields

	Name	XPath	Element	Result type	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type	Repeat
1	lat	lat	Node	Value of	String								
2	lng	lng	Node	Value of	String								

Get fields

OK Preview rows Cancel

Help



# - Exercício

**Select / Rename values**

Step name:

Select & Alter Remove Meta-data

Fields :

	Fieldname	Rename to	Length	Precision
1	bairro			
2	logradouro			
3	numero			
4	cep			
5	uf			
6	localidade			
7	lat			
8	lng			

Get fields to select  
Edit Mapping

Include unspecified fields, ordered by name ☐

Help OK Cancel



# - Exercício

**Text file output**

Step name: Text file output

File Content Fields

Filename: C:\treinamento\exercicios\etl2000\coordenada Browse...

Run this as command instead? ☐

Pass output to servlet ☐

Create Parent folder ☒

Do not create file at start ☐

Accept file name from field? ☐

File name field:

Extension: txt

Include stepnr in filename? ☐

Include partition nr in filename? ☐

Include date in filename? ☐

Include time in filename? ☐

Specify Date time format ☐

Date time format:

Show filename(s)...

Add filenames to result ☒

? Help OK Cancel







# + Exercício

## ■ Trabalhando com Web Services

- O objetivo deste exercício é apresentar ao aluno como conectar um web services

Tempo médio para a construção do exercício: **10 minutos**

Complexidade para a construção do exercício: **baixa-média**



# - Exercício

Iniciar uma transformação nova

**Generate Rows**

Step name:

Limit:

Never stop generating rows: ☐

Interval in ms (delay):

Current row time field name:

Previous row time field name:

Fields :

#	▲	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value	Set empty string?
1		nFahrenheit	Integer	#	8					68	N



# - Exercício

1o Clique em Load

The screenshot shows the 'Web Services Lookup' dialog box. The 'Step name' is 'Web Services Lookup (Conversao Celsius)'. The 'Web Service' tab is selected, showing 'FahrenheitToCelsiusResult'. The 'URL' is 'http://webservices.daehosting.com/services/TemperatureConversions.ws?WSDL'. The 'Operation' is 'FahrenheitToCelsius'. The 'Operation request name (optional)' is empty. 'The number of rows per call' is '1'. 'Pass input data to output' is checked. 'v2.x/3.0 Compatibility mode' is unchecked. 'Repeating element name' is empty. 'Return the complete reply from the service as a String' is unchecked. The 'HTTP authentication' section is collapsed. The 'Proxy to use' section is collapsed. At the bottom are buttons for 'OK', 'Add Input', 'Add Output', 'Cancel', and a 'Help' button. Two red arrows point to the 'Load' button and the 'Operation' dropdown menu.

Web Services Lookup

Step name: Web Services Lookup (Conversao Celsius)

Web Service in: FahrenheitToCelsiusResult

URL: http://webservices.daehosting.com/services/TemperatureConversions.ws?WSDL

Operation: FahrenheitToCelsius

Operation request name (optional):

The number of rows per call: 1

Pass input data to output: ☒

v2.x/3.0 Compatibility mode: ☐

Repeating element name:

Return the complete reply from the service as a String: ☐

HTTP authentication:

HTTP Login:

HTTP Password:

Proxy to use:

Proxy Host:

Proxy Port:

OK Add Input Add Output Cancel

Help

2o Clique em Operation

## - Exercício

[illegible]



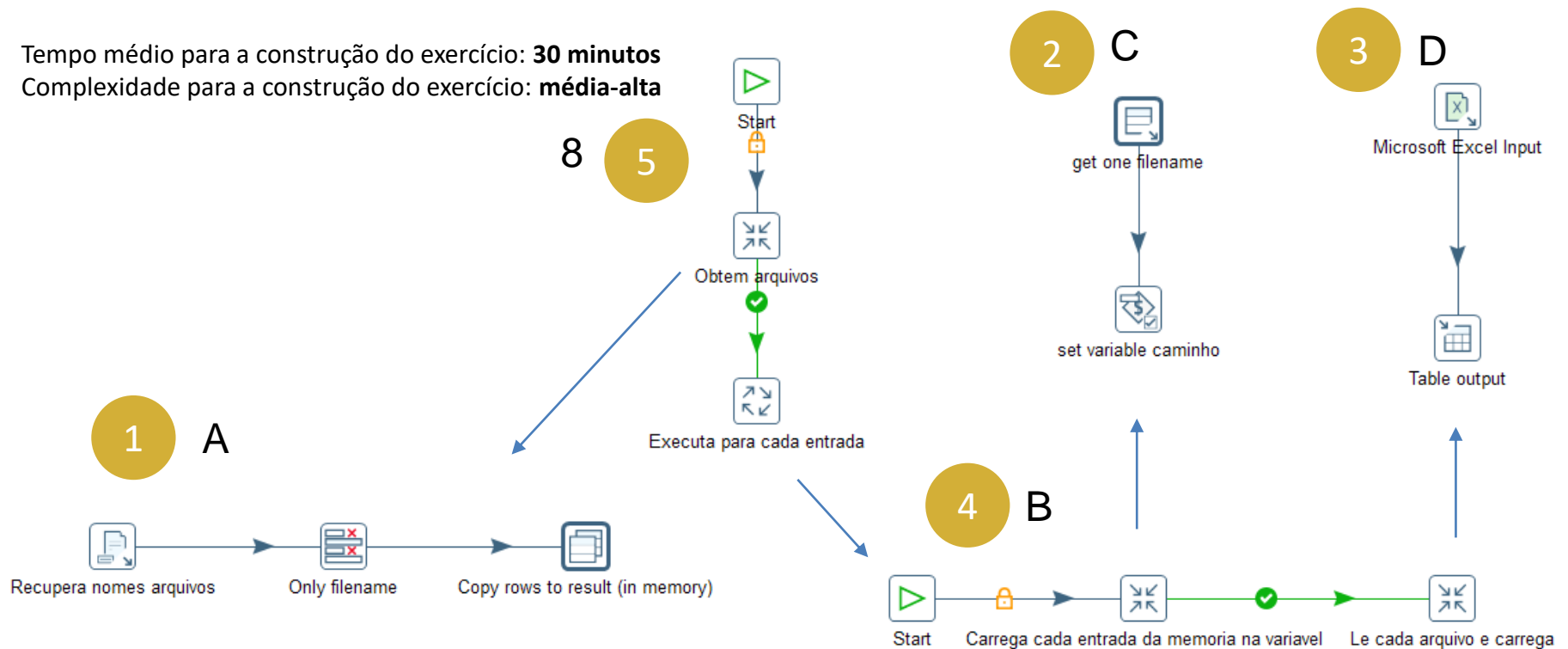
# + Exercício

## ■ Trabalhando com variáveis entre Jobs e transformações

- O objetivo deste exercício é apresentar ao aluno a construção de um processo usando fluxo de dados em memória entre Jobs e Transformações, usando variáveis

Tempo médio para a construção do exercício: **30 minutos**

Complexidade para a construção do exercício: **média-alta**



# - Exercício

## Iniciar uma transformação nova

Get file names

Step name: **Get File Names**

File Filters

Filename is defined in a field? ☐

Get filename from field:

Get wildcard from field (RegExp):

Exclude wildcard field:

Include subfolders: ☐

File or directory:  Add Browse...

Regular Expression:

Exclude Regular Expression:

Selected files:

	File/Directory	Wildcard (RegExp)	Exclude wildcard	Required	Include subfolders
1	C:\treinamento\materiais\etl2000\exercicio	arquivo*.xls		N	N

Show filename(s)... Delete Edit

Help OK Preview rows Cancel



# - Exercício

**Select / Rename values**

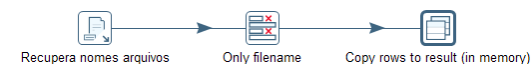
Step name:

Select & Alter Remove Meta-data

Fields :

#	Fieldname	Rename to	Length	Precision
1	filename	arquivo		

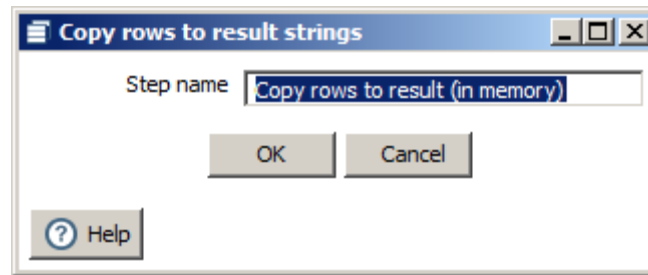
☐ Include unspecified fields, ordered by name





# - Exercício

Salvar como exercicio8\_a



# - Exercício

Iniciar uma transformação nova

Get rows from previous result

Step name:

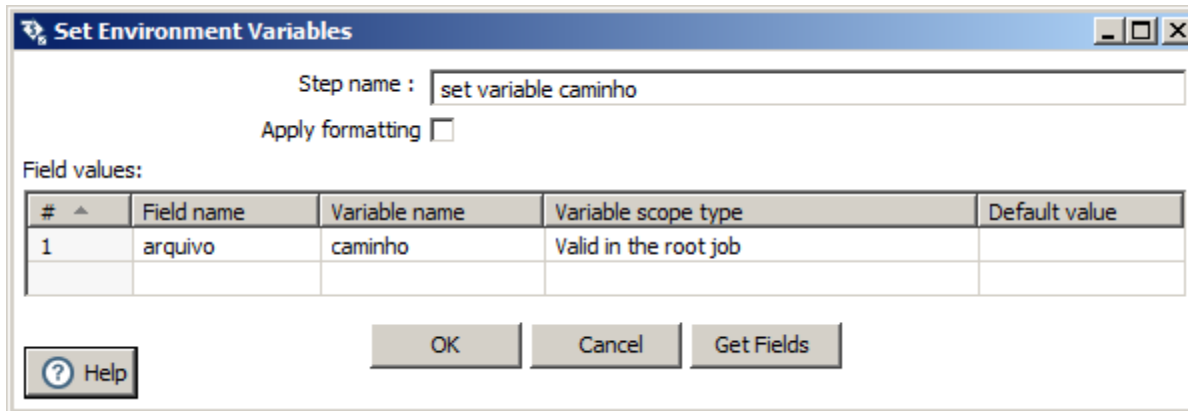
Fields :

#	Fieldname	Type	Length	Precision
1	arquivo	String	256	



# - Exercício

Salvar como exercicio8\_c



Step name :

Apply formatting ☐

Field values:

#	Field name	Variable name	Variable scope type	Default value
1	arquivo	caminho	Valid in the root job	



# - Exercício

## Iniciar uma transformação nova

Microsoft Excel input

Step name: Microsoft Excel Input

Files | Sheets | Content | Error Handling | Fields | Additional output fields

Spread sheet type (engine): Excel 97-2003 XLS (JXL)

File or directory:  Add Browse...

Regular Expression:

Exclude Regular Expression:

Selected files:

#	File/Directory	Wildcard (RegExp)	Exclude wildcard	Required	Include subfolders
1	\${caminho}				

Delete

Accept filenames from previous steps

Accept filenames from previous step ☐

Step to read filenames from:

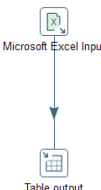
Field in the input to use as filename:

Edit

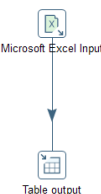
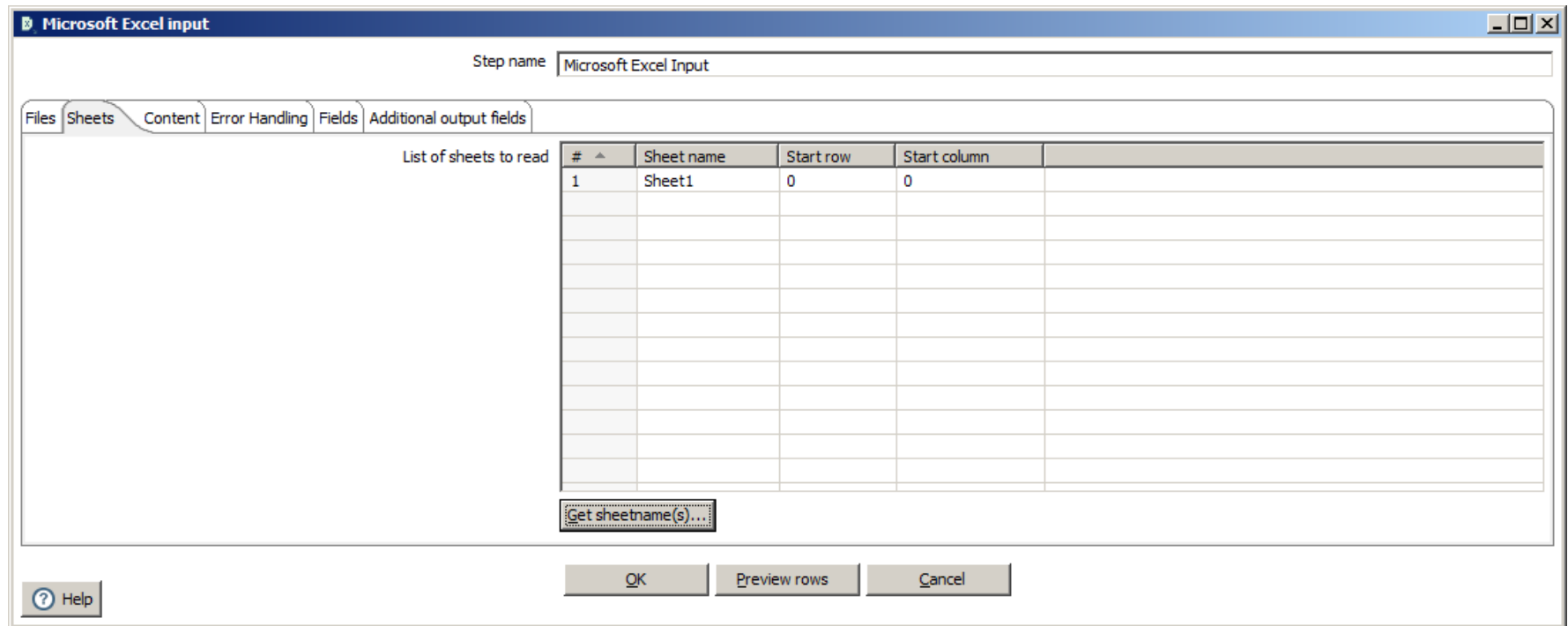
Show filename(s)...

OK Preview rows Cancel

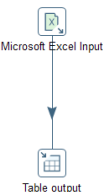
Help



## - Exercício



## - Exercício

[illegible]

# - Exercício

Salvar como exercicio8\_d

Table output

Step name: Table output

Connection: etl2000 [Edit...] [New...] [Wizard...]

Target schema: [Browse...]

Target table: exercicio8 [Browse...]

Commit size: 1000

Truncate table: ☐

Ignore insert errors: ☐

Specify database fields: ☐

Main options Database fields

Partition data over tables: ☐

Partitioning field: [Browse...]

Partition data per month: ☐

Partition data per day: ☐

Use batch update for inserts: ☒

Is the name of the table defined in a field?: ☐

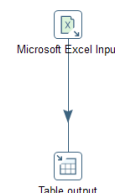
Field that contains name of table: [Browse...]

Store the tablename field: ☒

Return auto-generated key: ☐

Name of auto-generated key field: [Browse...]

[?] Help [OK] [Cancel] [SQL]



# - Exercício

## Iniciar um job novo

The screenshot shows the 'Transformation' dialog box in Pentaho. The 'Entry Name' field contains the text 'Carrega cada entrada da memória na variável'. The 'Transformation' field contains the path '/exercicio8\_c'. Below these fields are tabs for 'Options', 'Logging', 'Arguments', and 'Parameters'. The 'Options' tab is active, showing a 'Run configuration' dropdown set to 'Pentaho local'. Under the 'Execution' section, there are five checkboxes: 'Execute every input row' (unchecked), 'Clear results rows before execution' (unchecked), 'Clear results files before execution' (unchecked), 'Wait for remote transformation to complete' (checked), and 'Follow local abort to remote transformation' (unchecked). At the bottom of the dialog are 'Help', 'OK', and 'Cancel' buttons.

Transformation

Entry Name:  
Carrega cada entrada da memória na variável

Transformation:  
/exercicio8\_c Browse...

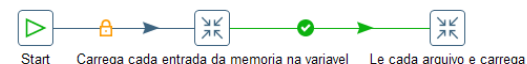
Options Logging Arguments Parameters

Run configuration:  
Pentaho local

Execution

- ☐ Execute every input row
- ☐ Clear results rows before execution
- ☐ Clear results files before execution
- ☒ Wait for remote transformation to complete
- ☐ Follow local abort to remote transformation

Help OK Cancel





# - Exercício

Salvar como exercicio8\_b

Transformation

Entry Name:  
Le cada arquivo e carrega

Transformation:  
/exercicio8\_d Browse...

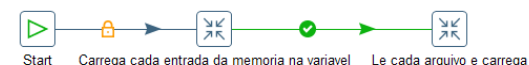
Options Logging Arguments Parameters

Run configuration:  
Pentaho local

Execution

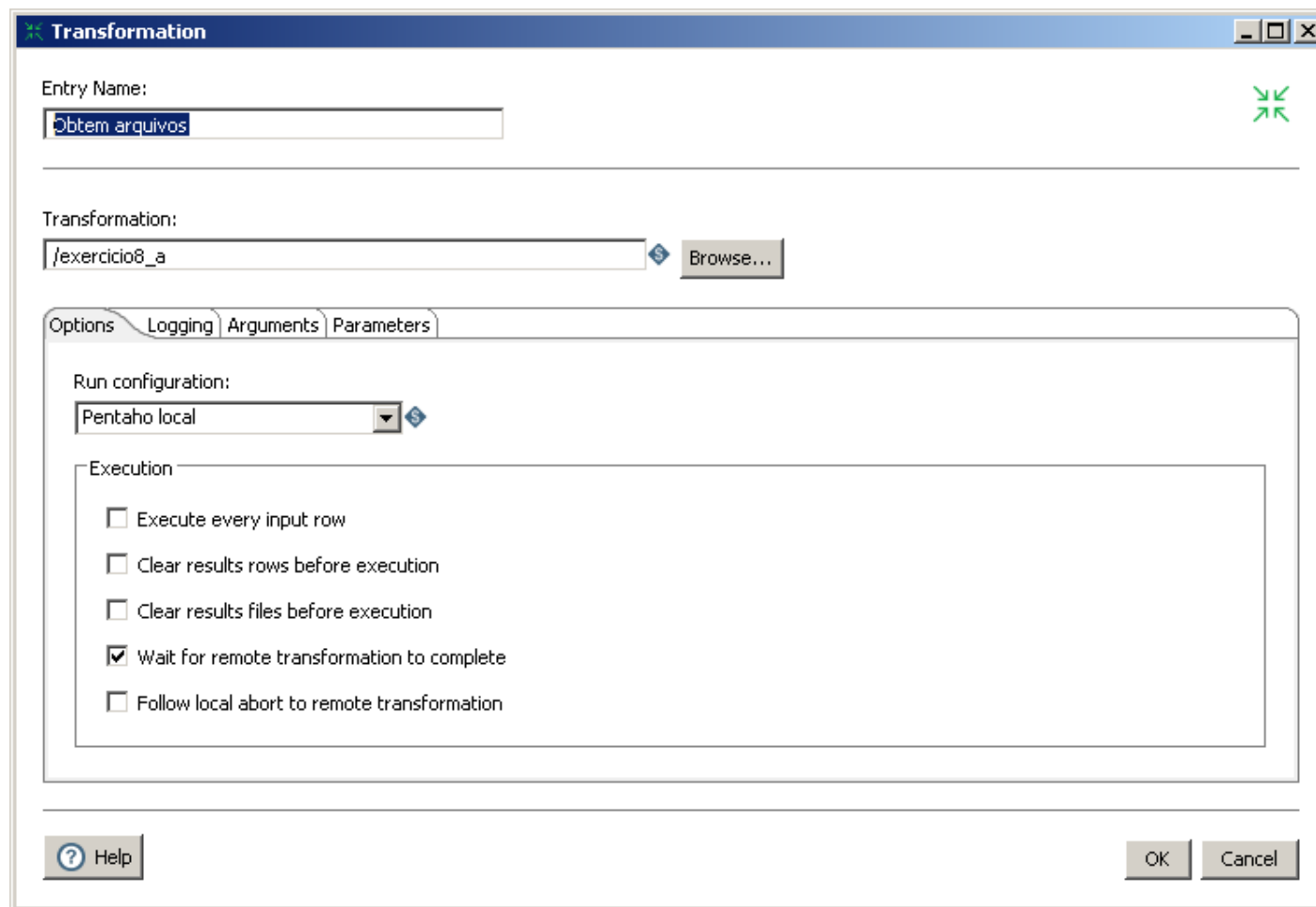
- ☐ Execute every input row
- ☐ Clear results rows before execution
- ☐ Clear results files before execution
- ☒ Wait for remote transformation to complete
- ☐ Follow local abort to remote transformation

Help OK Cancel



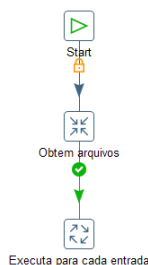
# - Exercício

## Iniciar um job novo



The image shows a 'Transformation' dialog box with the following fields and options:

- Entry Name:** A text field containing 'Obtem arquivos'.
- Transformation:** A text field containing '/exercicio8\_a' with a 'Browse...' button next to it.
- Options tab:** Contains a 'Run configuration:' dropdown menu set to 'Pentaho local'.
- Execution section:** A group box containing five checkboxes:
  - ☐ Execute every input row
  - ☐ Clear results rows before execution
  - ☐ Clear results files before execution
  - ☒ Wait for remote transformation to complete
  - ☐ Follow local abort to remote transformation
- Buttons:** 'Help', 'OK', and 'Cancel' at the bottom.



# - Exercício

## Salvar como exercicio8

The screenshot shows the 'Job' configuration window in Pentaho. The 'Entry Name' field is set to 'Executa para cada entrada'. The 'Job' field is set to '/exercicio8\_b' with a 'Browse...' button next to it. The 'Options' tab is selected, showing the 'Run configuration' dropdown set to 'Pentaho local'. Under the 'Execution' section, the following options are checked: 'Execute every input row', 'Wait for remote job to complete'. The other options, 'Pass the sub jobs and transformations to the server', 'Enable monitoring for sub jobs and transformations', and 'Follow local abort to remote job', are unchecked. At the bottom, there are 'Help', 'OK', and 'Cancel' buttons.

Job

Entry Name:  
Executa para cada entrada

Job:  
/exercicio8\_b Browse...

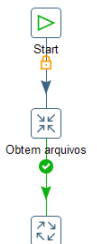
Options Logging Arguments Parameters

Run configuration:  
Pentaho local

Execution

- ☒ Execute every input row
- ☐ Pass the sub jobs and transformations to the server
- ☐ Enable monitoring for sub jobs and transformations
- ☒ Wait for remote job to complete
- ☐ Follow local abort to remote job

Help OK Cancel



Executa para cada entrada

# Argumentos e Parâmetros

## ■ O que é e suas diferenças

*Argumentos* são usados no contexto de linha de comando tanto para o Pan quanto para o Kitchen como passagem de variáveis não explicitas, porém devem ser sequenciadas como se espera nos steps que irão consumir estes argumentos.

*Parâmetros* são usados no contexto de linha de comando também, mas são declarados, evitando trocas de valores

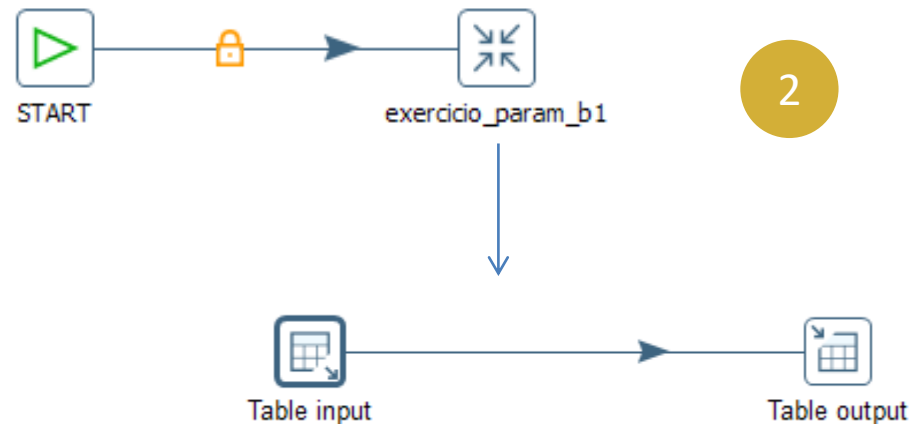
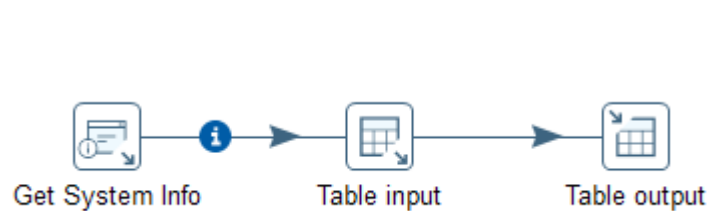
Vantagens no uso de parâmetros:

- Um valor padrão pode ser definido
- Uma descrição pode ser adicionada
- Não há necessidade de uma transformação adicional para recuperar o valor

# + Exercício

## ■ Trabalhando com Argumentos e Parâmetros

- O objetivo deste exercício é apresentar ao aluno as diferenças entre usar argumentos e parâmetros



# - Exercício

Iniciar uma transformação nova

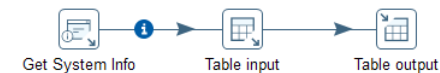
Get System Data

Step name

Fields:

#	Name	Type	
1	data_inicial	command line argument 1	
2	data_final	command line argument 2	

Help



# - Exercício

**Table input**

Step name:

Connection:

SQL

```
SELECT
  data
  , tipo_produto
  , quantidade
FROM
  vendas
WHERE
  data >=? AND
  data <=?
```

Line 1 Column 0

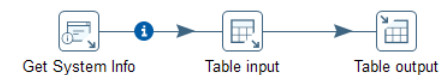
Enable lazy conversion ☐

Replace variables in script? ☒

Insert data from step:

Execute for each row? ☐

Limit size:



# - Exercício

**Table output**

Step name: Table output

Connection: etl2000 [Edit...] [New...] [Wizard...]

Target schema: [Browse...]

Target table: vendas\_temporaria [Browse...]

Commit size: 1000

Truncate table: ☐

Ignore insert errors: ☐

Specify database fields: ☒

**Main options** | **Database fields**

Partition data over tables: ☐

Partitioning field: [Browse...]

Partition data per month: ☐

Partition data per day: ☐

Use batch update for inserts: ☒

Is the name of the table defined in a field?: ☐

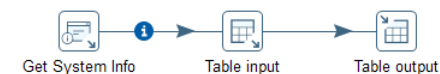
Field that contains name of table: [Browse...]

Store the tablename field: ☒

Return auto-generated key: ☐

Name of auto-generated key field: [Browse...]

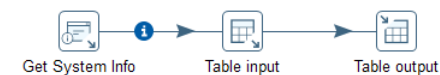
[?] Help [OK] [Cancel] [SQL]





## - Exercício

Salvar como exercicio9\_arg\_a

[illegible]

# - Exercício

Iniciar uma transformação nova

Table input

Step name: Table input

Connection: etl2000

SQL:

```
SELECT
  data
  , tipo_produto
  , quantidade
FROM
  vendas
WHERE
  data >="{VAR_DATA_INICIAL}" AND
  data <="{VAR_DATA_FINAL}"
```

Line 1 Column 0

Enable lazy conversion: ☐

Replace variables in script?: ☒

Insert data from step:

Execute for each row?: ☐

Limit size: 0

Buttons: Help, OK, Preview, Cancel



# - Exercício

**Table output**

Step name: Table output

Connection: etl2000 [Edit... New... Wizard...]

Target schema: [Browse...]

Target table: vendas\_temporaria [Browse...]

Commit size: 1000

Truncate table: ☒

Ignore insert errors: ☐

Specify database fields: ☒

**Main options** | **Database fields**

Partition data over tables: ☐

Partitioning field: [ ]

Partition data per month: ☐

Partition data per day: ☐

Use batch update for inserts: ☒

Is the name of the table defined in a field?: ☐

Field that contains name of table: [ ]

Store the tablename field: ☒

Return auto-generated key: ☐

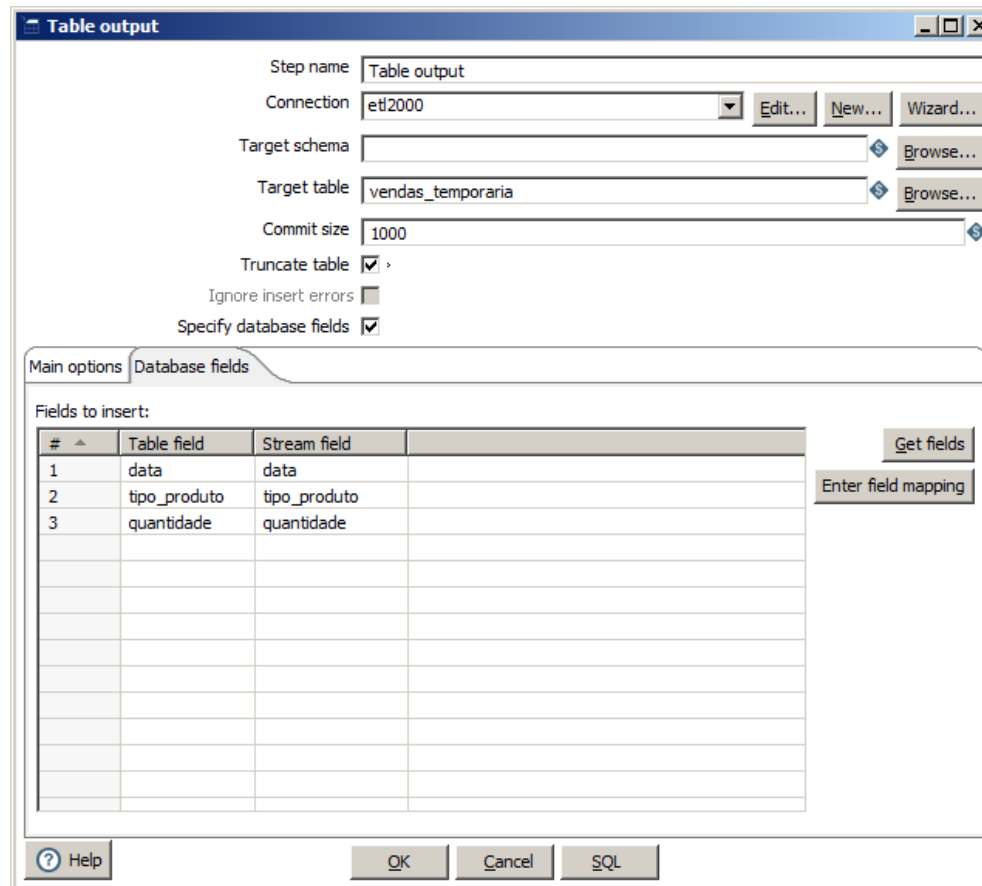
Name of auto-generated key field: [ ]

[?] Help [OK] [Cancel] [SQL]



# - Exercício

Salvar como exercicio9\_param\_b1



The image shows a 'Table output' dialog box with the following fields and options:

- Step name: Table output
- Connection: etl2000 (with Edit..., New..., and Wizard... buttons)
- Target schema: (with a Browse... button)
- Target table: vendas\_temporaria (with a Browse... button)
- Commit size: 1000
- Truncate table: ☒
- Ignore insert errors: ☐
- Specify database fields: ☒

The 'Database fields' tab is active, showing a table with the following data:

#	Table field	Stream field
1	data	data
2	tipo_produto	tipo_produto
3	quantidade	quantidade

Buttons: Get fields, Enter field mapping, Help, OK, Cancel, SQL.



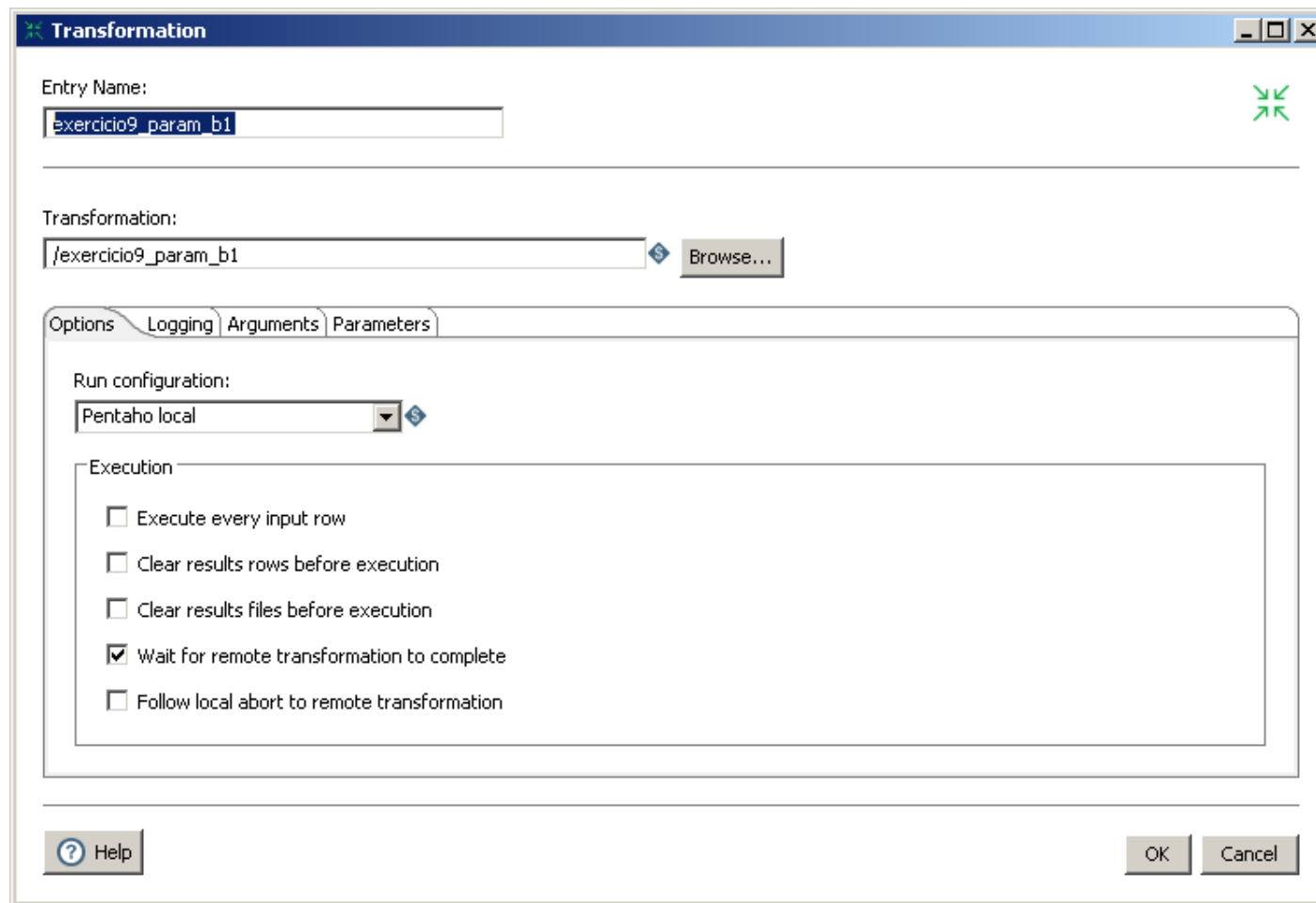
## - Exercício

## Iniciar um novo Job

[illegible]

# - Exercício

Salvar como exercicio9\_param\_b



The image shows a 'Transformation' dialog box with the following fields and options:

- Entry Name:** exercicio9\_param\_b1
- Transformation:** /exercicio9\_param\_b1 (with a 'Browse...' button)
- Options tab:**
  - Run configuration:** Pentaho local
  - Execution:**
    - ☐ Execute every input row
    - ☐ Clear results rows before execution
    - ☐ Clear results files before execution
    - ☒ Wait for remote transformation to complete
    - ☐ Follow local abort to remote transformation

Buttons at the bottom: Help, OK, Cancel.



# + Exercício

## ■ Trabalhando com sub-transformações

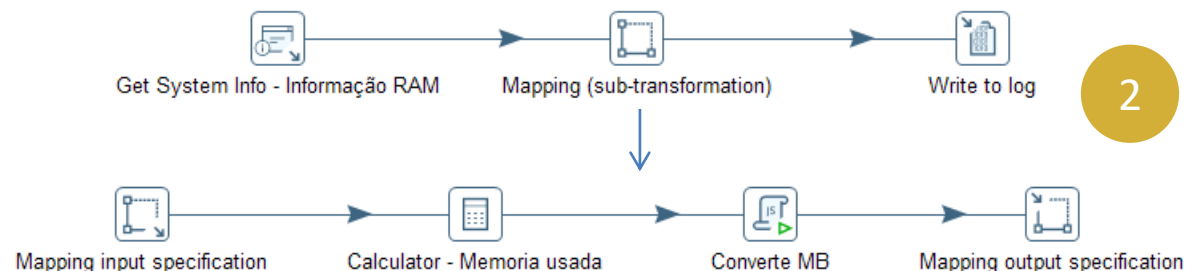
- O objetivo deste exercício é apresentar ao aluno como e o porque de trabalhar com sub-transformações
- Não é como uma chamada de Job e sim parecido como uma função, onde você chama uma transformação de dentro do fluxo de outra transformação.
- Útil para processos que se repetem

Tempo médio para a construção do exercício: **30 minutos**

Complexidade para a construção do exercício: **média-alta**



1



2

# - Exercício

Iniciar uma transformação nova

Get System Data

Step name

Fields:

#	Name	Type
1	tamanho_memoria	Total physical memory size (bytes)
2	memoria_livre	Free physical memory size (bytes)





## - Exercício

**Calculator**

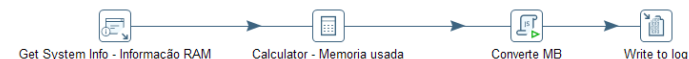
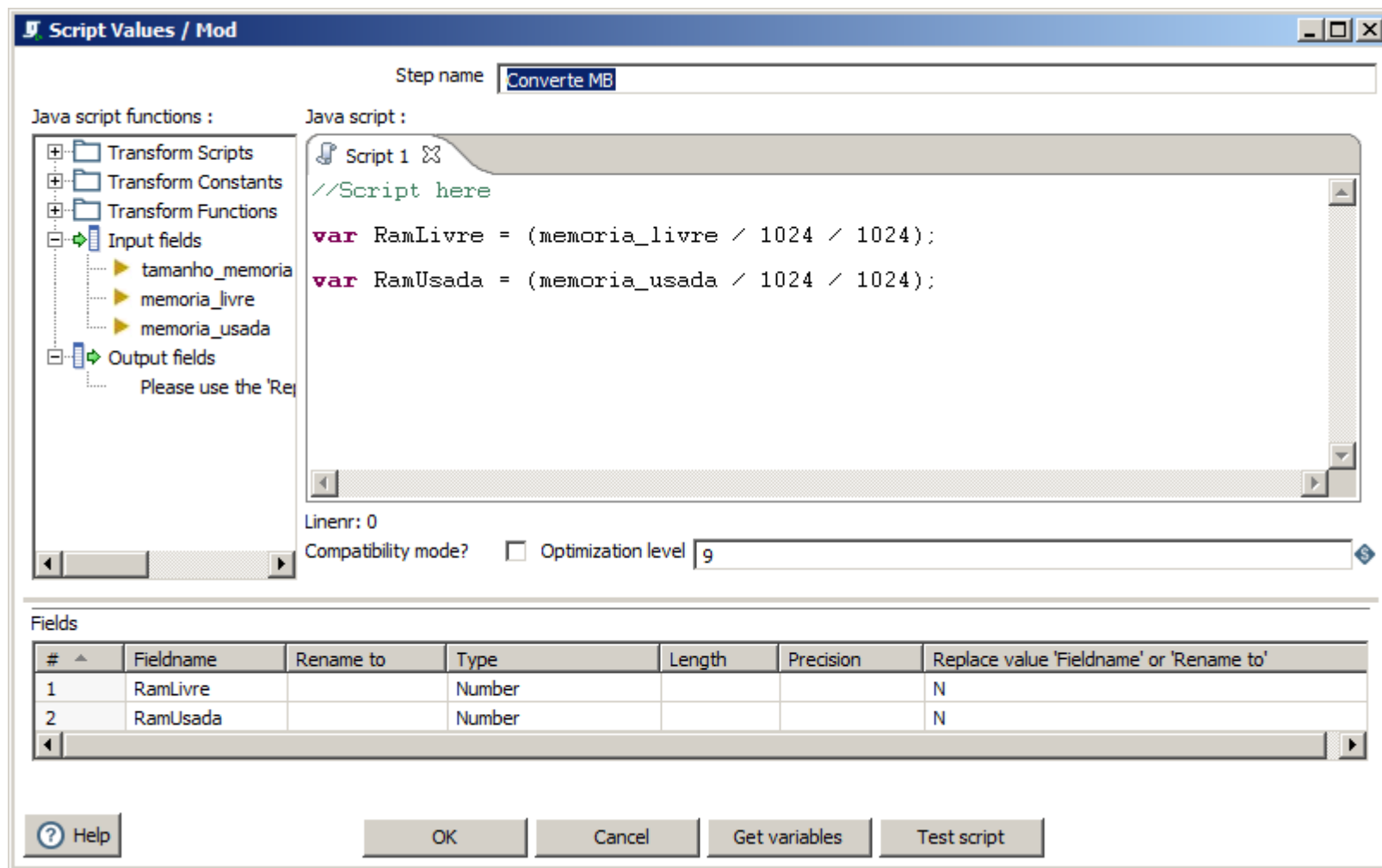
Step name:

Fields:

#	New field	Calculation	Field A	Field B	Field C	Value type	Length	Precision	Remove
1	memoria_usada	A - B	tamanho_memoria	memoria_livre		Number			N



# - Exercício



# - Exercício

Salvar como exercicio10\_trans\_a

Write to log

Step name: Write to log

Log level: Basic

Print header: ☒

Limit rows?: ☐

Nr of rows to print: 0

Write to log

Fields

	Field	
1	RamLivre	
2	RamUsada	

Help OK Get Fields Cancel



# - Exercício

Iniciar uma transformação nova

**Mapping Input Specification**

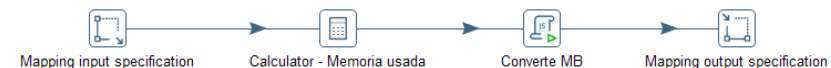
Step name: Mapping input specification

The required input fields for this mapping (sub-transformation) :

#	Name	Type	Length	Precision
1	tamanho_memoria	Number		
2	memoria_livre	Number		

☐ Include unspecified fields, ordered by name

[? Help](#) [OK](#) [Cancel](#)



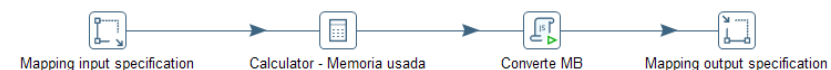
## - Exercício

**Calculator**

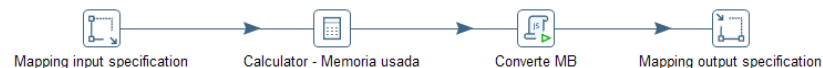
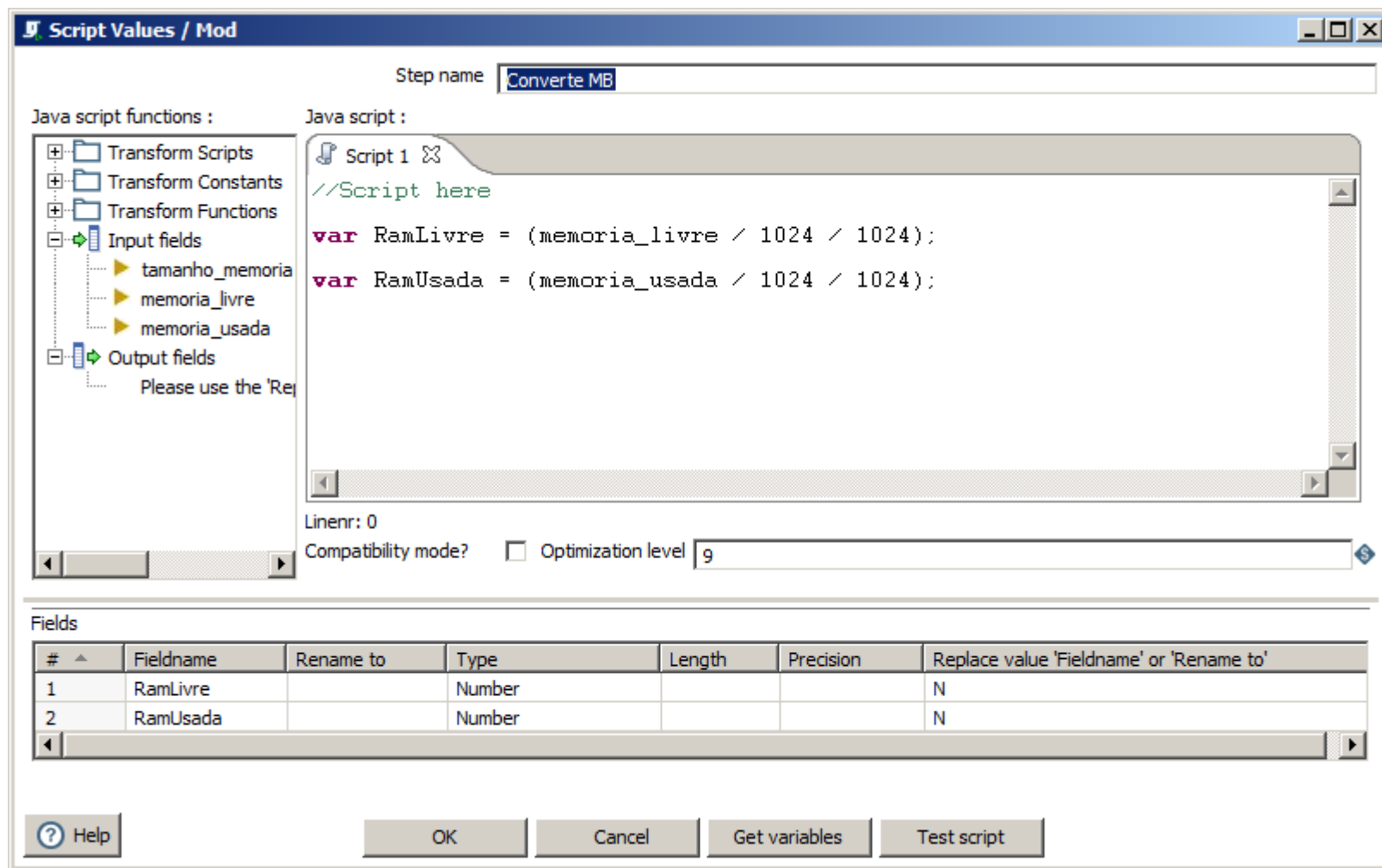
Step name:

Fields:

#	New field	Calculation	Field A	Field B	Field C	Value type	Length	Precision	Remove
1	memoria_usada	A - B	tamanho_memoria	memoria_livre		Number			N

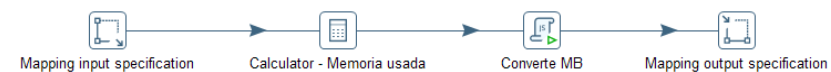
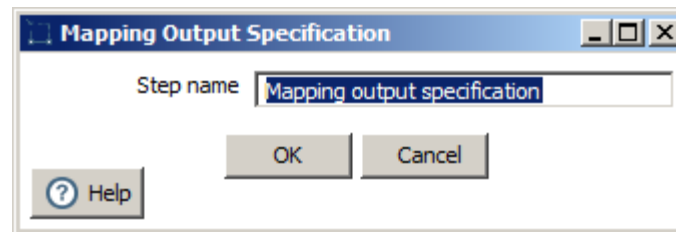


# - Exercício



# - Exercício

Salvar como exercicio10\_map\_c



# - Exercício


Iniciar uma transformação nova

**Get System Data**

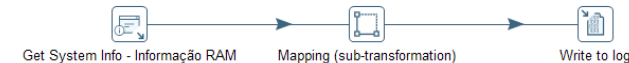
Step name:

Fields:

#	Name	Type
1	tamanho_memoria	Total physical memory size (bytes)
2	memoria_livre	Free physical memory size (bytes)

 Help

OK Preview rows Cancel








# - Exercício

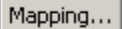
**Simple Mapping (sub-transformation)**



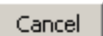
Step Name:  
Simple Mapping (sub-transformation)

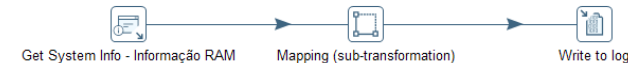
Transformation:  
/exercicio10\_map\_c  Browse...

Parameters  Input  Output

Fieldname from source step	Fieldname to mapping input step
tamanho_memoria	tamanho_memoria
memoria_livre	memoria_livre

☒ Update mapped fieldnames downstream 

 Help  



# - Exercício

Salvar como exercicio10\_map\_b

**Write to log**

Step name: Write to log

Log level: Basic

Print header: ☒

Limit rows?: ☐

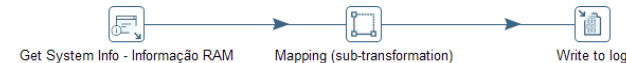
Nr of rows to print: 0

Write to log

Fields

	Field	
1	RamLivre	
2	RamUsada	

? Help OK Get Fields Cancel



# Execução Remota e em Cluster

- Usando o servidor Carte para execução remota e em cluster de processos de ETL
  - Carte é um servidor web utilizado para a execução remota de transformações e Jobs. O Carte também é usado para monitoramento, iniciar e parar transformações e jobs que são executados por ele, além de permitir a criação de um cluster de servidores Carte para processamento distribuído

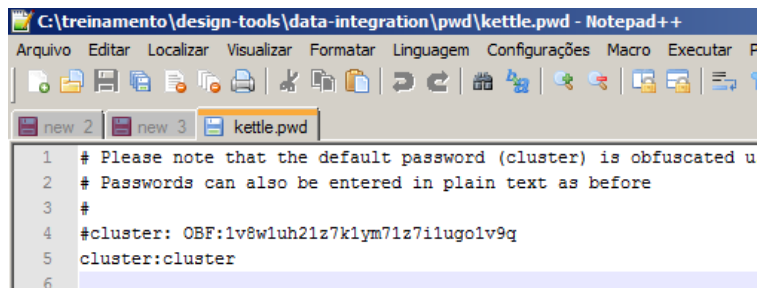
# + Exercício

- Instalar e Configurar o servidor Carte para processamento remoto
  - Neste exercício apresentaremos ao aluno como instalar e configurar o servidor Carte para processamento remoto de transformações e jobs

Tempo médio para a construção do exercício: **30 minutos**

Complexidade para a construção do exercício: **alta**

- Editar o arquivo **kettle.pwd** que se encontra no diretório **pwd** em **data-integration**
- Proceder com a configuração abaixo:

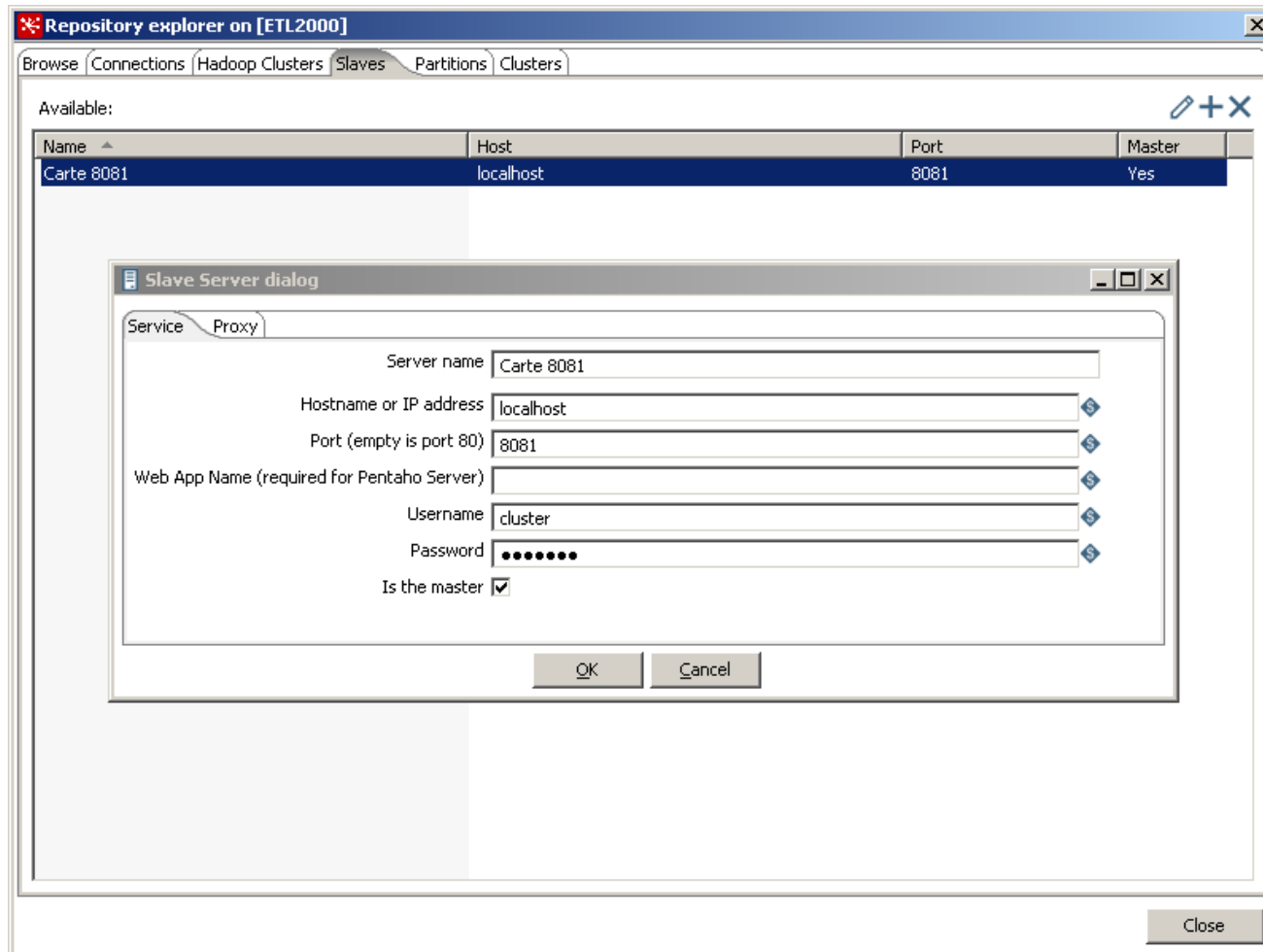


```
1 # Please note that the default password (cluster) is obfuscated u
2 # Passwords can also be entered in plain text as before
3 #
4 #cluster: OBF:1v8w1uh21z7k1ym71z7i1ugo1v9q
5 cluster:cluster
6
```

- Via linha de comando executar: `carte localhost 8081`
- Acessar via browser o endereço `localhost:8081` com usuário/senha `cluster/cluster`
- Importa e salve o arquivo `exercicio11.ktr` do diretório `...treinamento\materiais\etl2000`

# - Exercício

- Definindo o servidor Carte no Spoon
- Vá em Tools > Repository > Explore... e então selecione a aba Slave e configure



# - Exercício

- Criando uma configuração de execução remota
- Vá na aba View e clique com o botão direito do mouse em Run Configuration > New

Run configuration

Name:  
Pentaho Remoto

Description:

Engine:  
Pentaho

Settings

☐ Local

☒ Slave server

Location:  
Carte 8081

☐ Send resources to this server

OK Cancel

# - Exercício

- Executando o exercicio11 com a configuração de execução Pentaho Remoto

Run Options

Run configuration:  
Pentaho Remoto

Options

☒ Clear log before running      Log level: Basic

☐ Enable safe mode

☒ Gather performance metrics

Parameters   Variables

Parameter	Default value	Value	Description

Arguments (legacy)

☒ Always show dialog on run

Help      Run      Cancel

# + Exercício

## ■ Configurando o cluster

- Neste exercício o aluno aprenderá a configurar o Cluster Carte para processamento distribuído

Tempo médio para a construção do exercício: **30 minutos**

Complexidade para a construção do exercício: **alta**

- Colocar no ar mais dois servidores Carte, portas 8082 e 8083
- Adicionar os novos servidores como Slave
- Na aba View em Kettle Cluster Schema, criar um novo

Clustering schema dialog

Schema name: Cluster Local

Port: 40000

Sockets buffer size: 2000

Sockets flush interval (rows): 5000

Sockets data compressed? ☐

Dynamic cluster ☐

Slave servers

#	Name	Service URL	Master?
1	Carte 8081	Carte 8081	Y
2	Carte 8082	Carte 8082	N
3	Carte 8083	Carte 8083	N

Select slave servers

OK Cancel



# - Exercício

- Criando uma configuração de execução em cluster
- Vá na aba View e clique com o botão direito do mouse em Run Configuration > New

The image shows a 'Run configuration' dialog box with the following fields and settings:

- Name:** Pentaho Cluster
- Description:** (Empty text box)
- Engine:** Pentaho
- Settings:**
  - ☐ Local
  - ☒ Slave server
  - Location:** Clustered
  - ☐ Log remote execution locally
  - ☒ Show transformations

Buttons: OK, Cancel

# - Exercício

- No exercício11, no step Memory Group by, selecione-o e clique com o botão direito do mouse e selecione Clusters...
- Na caixa de diálogo aberta, Cluster schema, selecione Cluster local e clique no botão OK.



# - Exercício

- Executando o exercicio11 no Cluster Local

Run Options

Run configuration:  
Pentaho Cluster

Options

☒ Clear log before running      Log level: Basic

☐ Enable safe mode

☒ Gather performance metrics

Parameters   Variables

Parameter	Default value	Value	Description

Arguments (legacy)

☒ Always show dialog on run

Help   Run   Cancel

# Log dos processos de ETL

- Existem duas possibilidades de log, arquivo e banco de dados
  - Log em arquivos podem se tornar muito grandes, arquivar ou excluir
  - Log em banco de dados, mais estruturado e fácil para gestão
  - Motivos de guardar logs
    - Verificar se um processo foi finalizado
    - Rever os erros encontrados
    - Grande parte dos ETL's em produção não rodam em GUI
    - Monitoramento da performance
  - Tanto transformações quanto jobs podem armazenar logs em tabelas configuradas

Vamos acompanhar a demonstração das possibilidades

