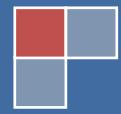


TREINAMENTO

Talend Data Integration

Apostila com os comandos das vídeoaulas e exercícios de fixação



Informações

CURSO: Talend Data Integration	DIREITOS: Uso da apostila deve ser autorizado, caso o estudante venha a utilizá-la fora do ambiente virtual do curso. Enviar e-mail para solicitação.
DATA CRIAÇÃO: 17/04/2017	REVISÃO: 4.0
DATA ÚLTIMA MODIFICAÇÃO: 18/08/2017	VERSÃO: 7.0
AUTOR Grimaldo Lopes de Oliveira	EMAIL: grimaldo_lopes@hotmail.com
SOBRE O PROFESSOR Grimaldo é baiano e soteropolitano. Fez graduação em Estatística pela Universidade Federal da Bahia e logo em seguida uma especialização na Área de Mineração de Dados/BI na Faculdade Visconde de Cairu. Após esta formação, "mergulhou" na área. Trabalha com Ferramentas OLAP e Estatísticas: OBIEE, DISCOVERY, SAS, R, QLIK SENSE, ORACLE DATA VISUALIZATION, TABLEAU, POWER BI, WEKA, OWB, TALEND quando começou a extrair dados dos bancos de dados e passou a gerar análises estatísticas nestas ferramentas. Atualmente trabalha com Business Intelligence na PRODEB. Também é mestre em Gestão e Tecnologia Aplicadas à Educação (Gestec), pela Universidade do Estado da Bahia, editor do blog BI com Vatapá, Autor do Livro BI COMO DEVE SER e do Guia prático – Resumo BI - Concurso e Profissionais.	
<p>Conheça Grimaldo:</p> <p>Livro: BI COMO DEVE SER - www.bicomodeveser.com.br Blog: www.bicomvatapa.blogspot.com Site: www.aprendavirtual.com - Seu Site de Cursos EAD Facebook: www.facebook.com/groups/bicomvatapa/ Perfil: br.linkedin.com/in/grimaldo</p>	

Sumário

INTRODUÇÃO	4
APRESENTAÇÃO	6
SOBRE ESTE DOCUMENTO.....	6
RECURSOS NECESSÁRIOS	6
OBJETIVO.....	8
ENTENDENDO O TALEND DATA INTEGRATION	9
INSTALAÇÃO O TALEND DATA INTEGRATION.....	10
ESTRUTURA DE FUNCIONAMENTO DO TALEND DATA INTEGRATION.....	12
PRIMEIROS PASSOS COM O TALEND DATA INTEGRATOR	14
Construção do DW	23
TRABALHO PRÁTICO	24
CARGA DA STAGING.....	26
TRANSFORMAÇÃO DOS DADOS.....	29
CARGA TABELA STAGING FATO.....	31
CARGA DA DIMENSÃO	33
CARGA DA DIMENSÃO TEMPO	42
CARGA DA FATO	44
AUTOMATIZAÇÃO DAS CARGAS	52
PRINCIPAIS COMPONENTES DO TALEND DATA INTEGRATION	56
TFILEARCHIVE	57
TFILEUNARCHIVE	59
TFILECOMPARE E TFILEOUTPUTDELIMETED	60
TFILECOPY E TFILELIST	61
TFILEDELETE E TJAVA.....	63
TFILEEXIST ETMSGBOX	65
TFILEROWCOUNT	67
TAGGREGATEROW, TSORTROW E TLOGROW	69
TREPLACE E TFILTERCOLUMN	72
TFILTERROW	76
TSPLITROW.....	78
TFILEINPUTXML.....	80
TLOGCATCHER , TROWGENERATION,TDIE E TLOGROW	86

TLOGCATCHER , TROWGENERATION,TWARN E TLOGROWERRO!	INDICADOR NÃO DEFINIDO.
TSENDMAIL	90
VARIÁVEL DE CONTEXTO	91
EXPORTAR JOB.....	95
EXPORTAR PROJETO	97
IMPORTAR PROJETO	100
EXERCÍCIOS FINAIS PARA EMISSÃO DO CERTIFICADO DE PARTICIPAÇÃO.....	102
Primeira Bateria de Exercícios.....	103
Segunda Bateria de Exercícios.....	104
CONSIDERAÇÕES IMPORTANTES	105
CONSIDERAÇÕES IMPORTANTES	105
Futuras atualizações	105
APÊNDICE.....	106
GLOSSÁRIO DE SIGLAS E TERMOS	106

Conheça o Professor Grimaldo Oliveira



Sou professor das pós-graduações das universidades **UNIFACS**, **CATÓLICA DO SALVADOR** e **ISL Wyden**. Mestre pela **Universidade do Estado da Bahia** (UNEBA) no Curso de Mestrado Profissional Gestão e Tecnologias Aplicadas à Educação (GESTEC). Possuo Especialização em Análise de Sistemas pela Faculdade Visconde de Cairu e Bacharelado em **Estatística** pela Universidade Federal da Bahia. Atuo profissionalmente como consultor há mais de **15 anos** nas **áreas de Data Warehouse, Mineração de Dados, Ferramentas de Tomada de Decisão e Estatística**.



Idealizador do treinamento online **BI PRO** com + de 10 módulos contendo todas as disciplinas para formação completa na área de dados. Quem participa do **BI PRO** tem acesso gratuito: todos os meus cursos de dados da Udemy, + ebook BI COMO DEVE SER - O Guia Definitivo, espaço de mentoria para retirada de dúvidas, respostas das atividades. Acesse www.bipro.com.br



Autor do eBook **BI COMO DEVE SER - O Guia Definitivo**, com ele você poderá entender os conceitos e técnicas utilizados para o desenvolvimento de uma solução BI, tudo isso de forma objetiva e prática, com linguagem acessível tanto para técnicos quanto gestores e analista de negócio. Acesse www.bicomodeveser.com.br



Site de **cupons** do prof. Grimaldo, com desconto de todos os seus cursos de dados da Udemy, atualizado diariamente com diversas promoções, incluindo cursos gratuitos.

Acesse <https://is.gd/CUPOMCURSOSPROFGRIMALDO>

Curso EAD Talend Data Integration

Prof. Grimaldo Oliveira

grimaldo_lopes@hotmail.com



Idealizador do Blog **BI COM VATAPÁ** reúne informações diversas sobre a área de dados com detalhes sobre o mundo de Business Intelligence, Big Data, Ciência de dados, Mineração de dados e muitos outros. Acesse <http://bicomvatapa.blogspot.com/>

Introdução

"O pessimista vê dificuldade em cada oportunidade; o otimista vê oportunidade em cada dificuldade."

Winston Churchill (1874-1965)

Apresentação

O curso de Talend Data Integration, foi idealizado para permitir um rápido aprendizado prático ao aluno, através de uma interação com o professor a partir de vídeoaulas, que facilitam o entendimento dos diversos comandos do Talend Data Integration. Durante o curso, o aluno terá que assistir as vídeoaulas e praticar diretamente no seu computador. Será necessário que o aluno tenha em sua máquina uma versão da ferramenta instalada. O aluno terá um prazo máximo de curso de 3 meses para sua conclusão, mas poderá solicitar uma única prorrogação pelo tempo que desejar, entretanto o aluno será avisado por e-mail sobre o término do curso. Durante todo o curso, o aluno poderá retirar dúvidas com o professor e colegas, através do fórum de dúvidas ou por e-mail.

Sobre este documento

O objetivo deste documento é fornecer ao aluno, os passos necessários para que este aprenda a trabalhar com a ferramenta Talend Data Integration na sua plenitude, onde detalhes mais específicos sobre cada comando serão explicados, através de uma linguagem direta, facilitando a construção das suas análises.

Recursos necessários

Para acesso ao curso é necessário que o aluno tenha uma internet de rápido acesso para assistir aos vídeos, além de um leitor pdf da apostila do curso*, devem ser utilizados os navegadores de internet Firefox(Mozilla) ou Google Chrome com às versões mais atuais para acesso ao site de aulas EAD, além do software Talend Data Integration* instalado na máquina do aluno.

****Este software não é fornecido pelo treinamento, o aluno é responsável pela aquisição, caso o mesmo seja pago.***

Atualizações deste documento

Futuras modificações poderão ocorrer no conteúdo deste documento em decorrência de possíveis ajustes na documentação do curso, sejam elas oriundas do professor ou devido a atualizações pertinentes que possam ser demandadas pelos alunos, dentro de critérios lógicos que não afetarão os objetivos para o qual este documento foi criado.

Aulas

“Todo o conhecimento humano começou com intuições, passou daí aos conceitos e terminou com ideias.”

Immanuel Kant

As informações abaixo estão nas vídeoaulas do curso, acompanhe os vídeos e re-execute os procedimentos para fixação

Objetivo

O objetivo principal do curso é permitir que aspectos básicos com ênfase no entendimento da ferramenta Talend Data Integration e sua forma de operação sejam compreendidos. O curso em questão não pretende ensinar análise de dados ao aluno, será necessário que o aluno tenha conhecimento em indicadores e saiba correlacionar informações para entender as saídas de dados dos comandos que serão apresentados. Os comandos são usados ao longo do texto simplesmente para ilustrar aspectos do uso da operacionalização da ferramenta.

Embora, na maior parte do texto assume-se familiaridade com conceitos e métodos básicos de análise de dados, alguns tópicos especializados são usados em algumas sessões e, não sendo de interesse de leitor, podem ser ignorados, sem prejuízo ao acompanhamento das demais partes do texto. De todo modo, não será assumido nenhum conhecimento prévio em ferramentas de ETL. O curso foi preparado e estruturado para que desde o iniciante até o aluno mais avançado, este utilize técnicas de análise de dados e aprenda a trabalhar com a ferramenta Talend Data Integration.

Portanto, mãos a obra.

Entendendo o Talend Data Integration

O Talend Data Integration é um conjunto de ferramentas da família ETL (**Extração, Transformação e Carga**) para integração de dados. Fornece meios para integrar e processar todos os seus dados com um designer visual fácil de usar.

O Talend Data Integration para Integração de Dados oferece soluções para os problemas que as empresas enfrentam devido à crescente complexidade dos sistemas, abordando a ETL para análise e ETL para necessidades de integração operacional.

Existe hoje no mercado três versões para o Talend Data Integration, ambas com a seguinte nomenclatura:

- TOS_*-win-x86_64.exe, para Windows.
- TOS_*-linux-gtk-x86_64, para Linux.
- TOS_*-macosx-cocoa.app, para Mac.

Ao trabalhar com o Talend Data Integration para Integração de Dados, você será apresentado a estrutura de trabalho do ambiente do programa, muitas vezes encontrará palavras como **repositório, projeto, workspace, Job, componente**.

Vamos detalhá-los para que você entenda com facilidade.

REPOSITÓRIO: é o local de armazenamento que o Talend utiliza para Integração de dados, de forma a reunir os dados relacionados a construção de uma carga de dados, podendo ser ela: cargas ETL, cargas de Data Quality ou simplesmente uma cópia de bases.

PROJETO: os projetos são conjuntos estruturados, elementos e seus metadados associados. Tudo que você construir de cargas será organizado em um projeto. Você pode criar tantos projetos quanto você precisa em um repositório.

WORKSPACE: é o diretório onde você armazena todas as pastas do projeto. Você precisa ter um diretório por conexão (conexão de repositório).

JOB: é um desenho gráfico, de um ou mais componentes conectados em conjunto, que permite executar processos de gerenciamento de fluxo de dados. Traduz as necessidades do negócio em código, rotinas e programas. É a carga de dados em si, é todo o código gerado em uma carga.

COMPONENTE: um componente é um conector pré-configurado usado para executar uma integração de dados específica, independentemente das fontes de dados que você está integrando: bancos de dados, aplicativos, arquivos simples, serviços da web, etc.

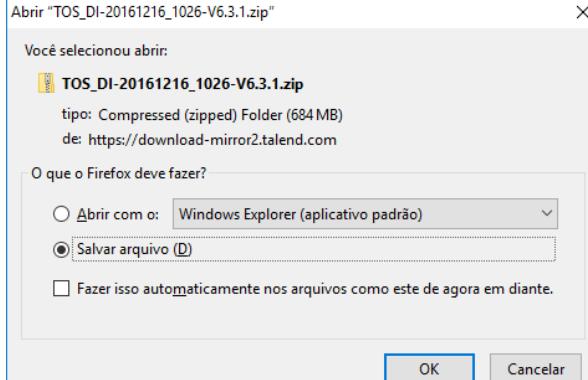
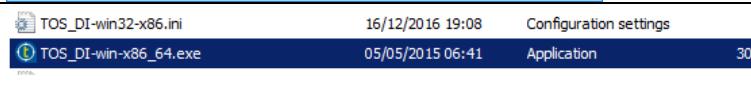
Entender o conceito por trás de cada uma dessas palavras é crucial para entender a funcionalidade do Talend para integração de dados.

Enfim, o Talend Data Integration permite construir carga de dados dos mais variados assuntos, personalizados para os indivíduos em sua empresa. Com ele você poderá realizar descoberta nos dados sem necessitar de programação alguma, basta apenas arrastar os mais diversos objetos que são oferecidos na sua paleta de construção.

Instalação o Talend Data Integration

A instalação será realizada através do download da ferramenta no site da empresa. Ao baixar à ferramenta, sua instalação é como qualquer outro produto para Windows.

Passos:

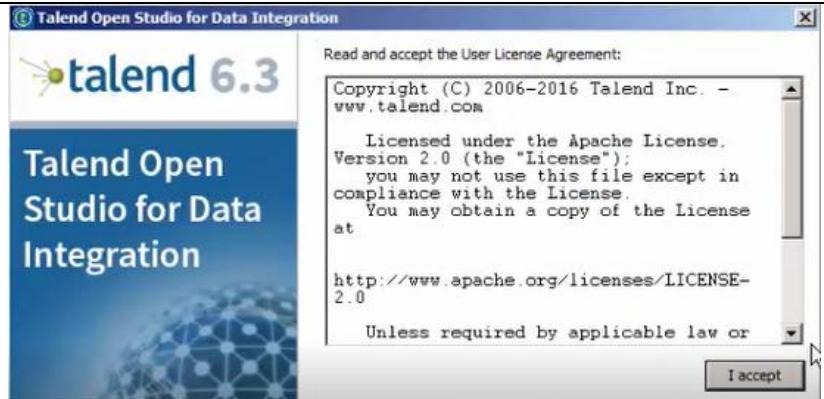
1- Download do Talend Data Integration, versão Open Studio	http://www.talend.com/download/talend-open-studio/#t4
2- Faça o download da ferramenta e execute em sua máquina.	
3- Descompacte o conteúdo e execute o programa.	

Curso EAD Talend Data Integration

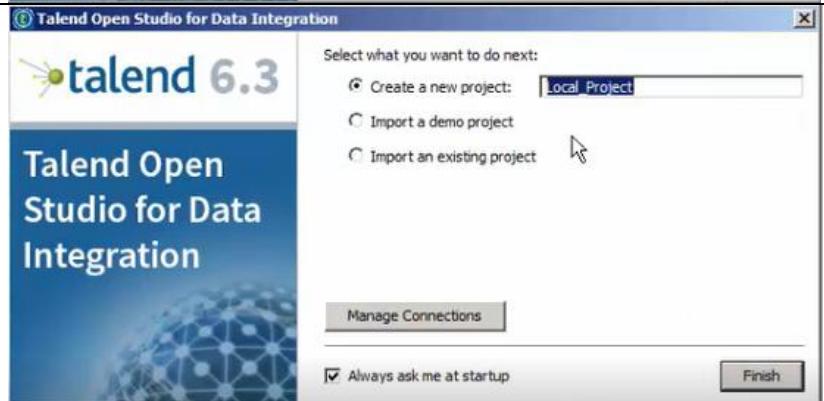
Prof. Grimaldo Oliveira

grimaldo_lopes@hotmail.com

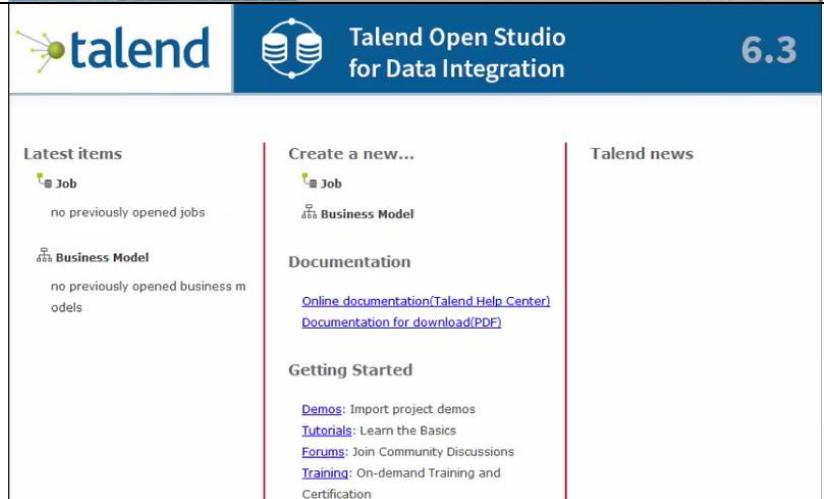
4- Aceite a licença.



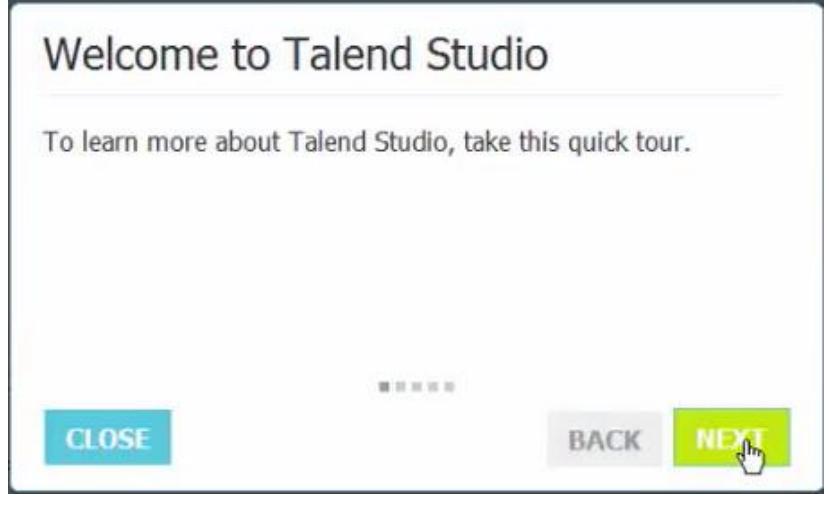
5- Será solicitada a criação de um novo projeto e automaticamente o repositório de dados.



6- Será exibida a tela de entrada.

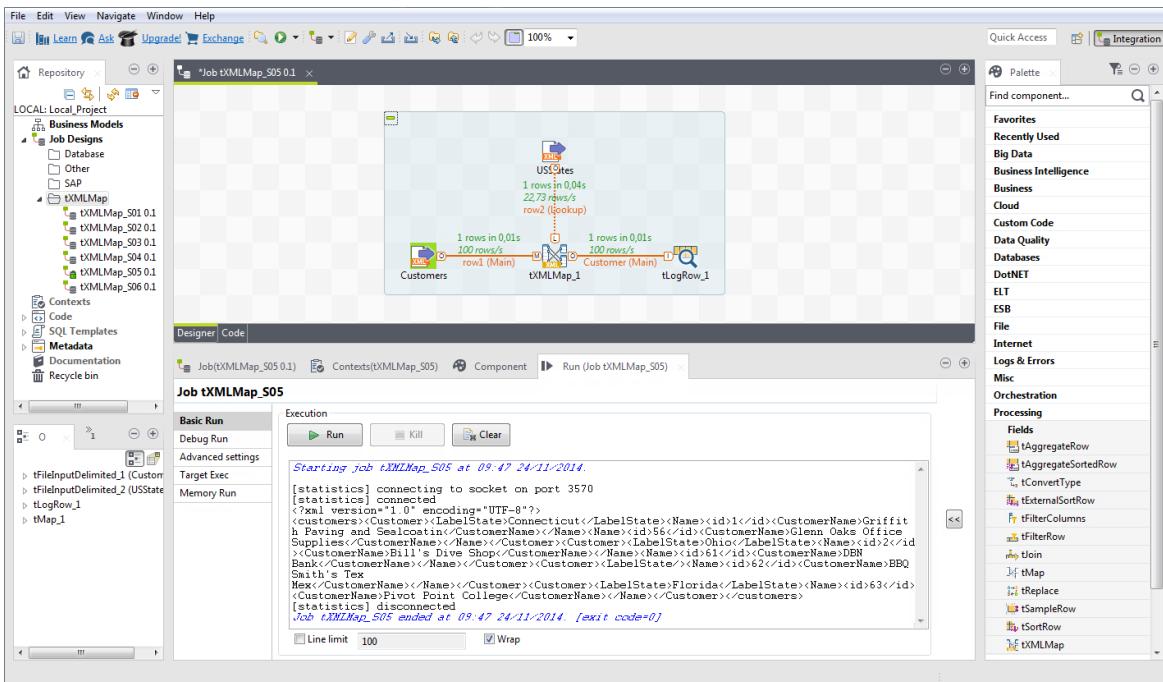


7- Pronto você já pode começar a trabalhar.



Estrutura de funcionamento do Talend Data Integration

O Talend Data Integration, possui uma janela multi painel que permite o gerenciamento das cargas em execução, bem como o controle de log's e debug's real-time. A medida que é executado, todas as ações são exibidas. São divididas em menu **bar**, **toolbar**, **repositório em árvore**, **design workspace**, **paleta**, e a **code view**.



File	Close	Fecha a corrente janela do Studio design workspace.
	Close All	Fecha todas as janelas abertas do Studio design workspace.
	Save	Salva as mudanças feitas na janela em aberto.
	Save as	Salva todas as mudanças sem fazer mudanças na janela em aberto.
	Save All	Salva todas as mudanças de todas as janelas abertas.
	Print	Imprime dados da janela disponível.
	Switch Project or Workspace	Fecha todos os trabalhos em aberto e permite abrir um diferente projeto em uma workspace diferente.
	Edit project properties	Edita as propriedades do corrente projeto.
	Import	Abri um wizard, permitindo que sejam importados diferentes arquivos de projetos de versões diferentes.
	Export	Abri o wizard e permite exportar o projeto para diferentes formatos (arquivos, XML catalogs, etc.)
	Exit	Fecha o Talend Data integration.
Edit	Undo	Desfaz a última ação feita.
	Redo	Refaz a última ação feita.
	Cut	Recorta o objeto selecionado.
	Copy	Copia o objeto selecionado.
	Paste	Cola o objeto selecionado.
	Delete	Deleta o objeto selecionado.
	Select All	Seleciona todos os componentes
View	Zoom In	Aumenta a imagem.
	Zoom Out	Diminui a imagem.
	Grid	Exibe uma grade no workspace.
	Snap to Geometry	Habilita o recurso Snap to Geometry.
Window	Perspective	Abre diferentes perspectivas ou janelas em diferentes view do GUI.
	Show View...	Mostra o [Show View] que habilita a exibição de diferentes view do GUI.
	Maximize Active View or Editor...	Maximiza as diferentes view.
	Preferences	Abre a caixa de diálogo de preferências.
Help	Welcome	Abre a documentação e oferece o bem-vindo ao Talend Data integration.
	Help Contents	Abre a ajuda.

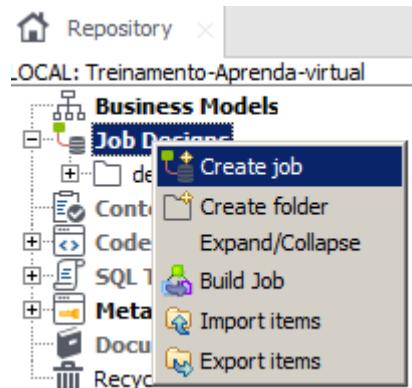
	Install Additional Packages...	Abre o [Additional Talend packages] onde você pode adicionar pacotes extras ao Talend Data Integration.
	About Talend Studio	Exibe: <ul style="list-style-type: none"> • A versão do software, • Detalha sobre a configuração e os problemas, • Detalha sobre plug-in(s), • Detalha sobre futuras funcionalidades.
	Support Logs	Abre o wizard para que possa ser exportado todos os log's gerados para um arquivo de dados.
	Studio Quick Tour	Abre passo-a-passo a apresentação ao Repository, ao Design Workspace, as configurações em geral e paleta.

Primeiros passos com o Talend Data Integration

Vamos iniciar nosso curso, trabalhando com os comandos básicos do Talend Data Integration e no decorrer do curso, vamos nos aprofundar nos assuntos com um grau de dificuldade maior.

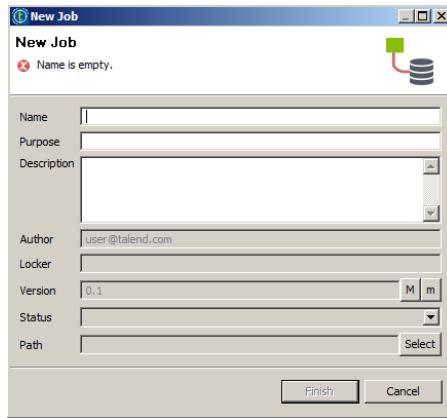
Para iniciar seu primeiro trabalho vamos criar nosso primeiro **Job**, ou seja, criar uma carga de dados. Para isso, vamos utilizar uma planilha chamada **notas fiscais.xls**, que será gravada em um banco de dados ORACLE.

Você irá ao repositório e escolherá a opção **create job**.



Clique em **create**, do seu notebook ou desktop.

A tela abaixo irá ser aberta:



Digite as informações sobre o JOB:

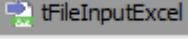
Name: Carga_Produto

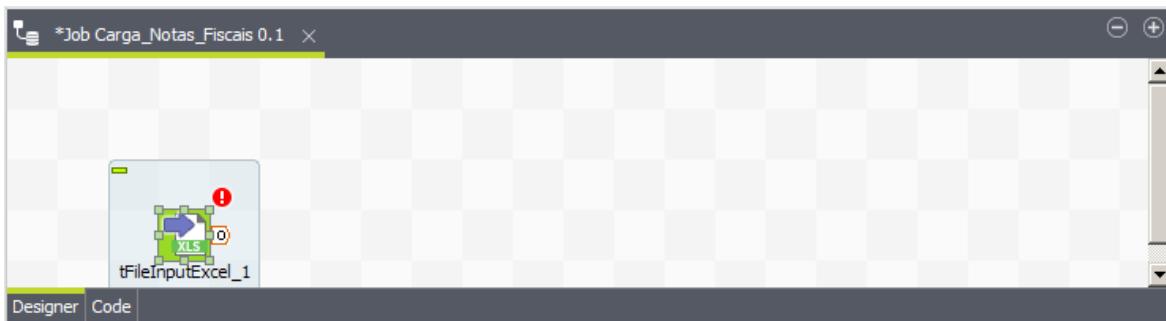
Purpose: a Finalidade é efetuar carga no banco de dados.

Description: carga de arquivo de produtos, xls de Notas Fiscais.

Clique em **Finish**.

Pronto, a área de carga está liberada para a leitura do arquivo **.xlsx** e para ser carregado no banco de dados

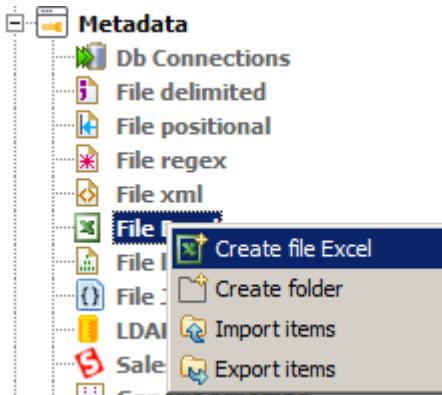
Vamos à paleta e trazer o objeto  na aba **Input**. Vamos arrastar para a área de construção de cargas, chamada de **design workspace**.



Agora para **lermos** o arquivo, necessitaremos criar um **metadados** para determinar qual será a aba do Excel que será aberta, qual é a estrutura do arquivo, quais campos serão

carregados, etc. O metadados facilita a comunicação entre o Talend Data Integration e os arquivos de leitura e destino.

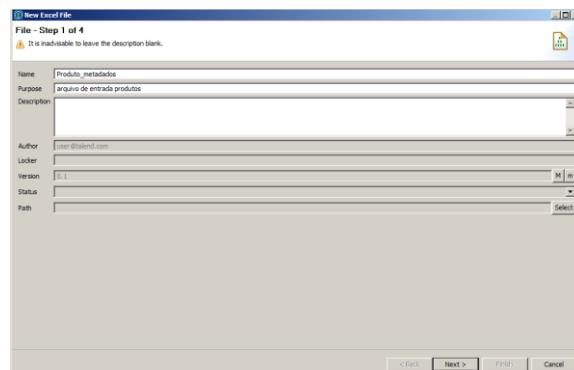
Vá até a aba do Repositório e crie o metadados para a planilha produto, dentro do arquivo Notas Fiscais.xlsx



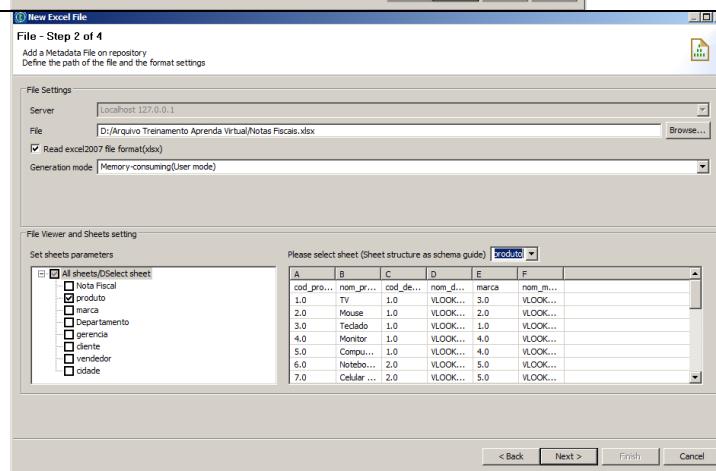
Vamos passar por algumas etapas para a geração do metadados:

Passos:

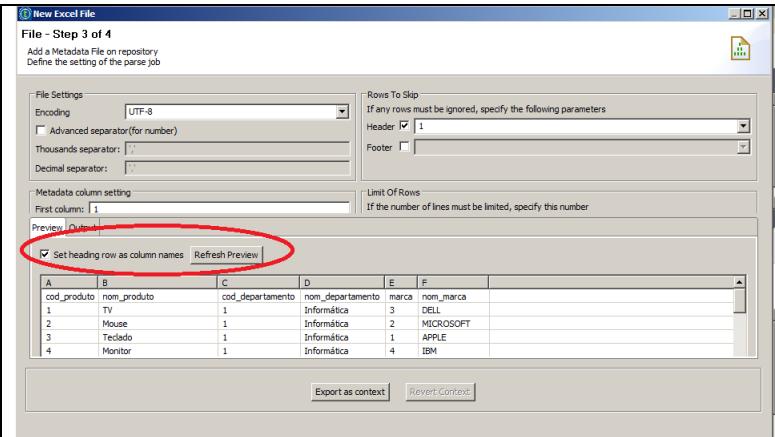
- Preencha os dados do arquivo que será carregado.
Produto_metadados



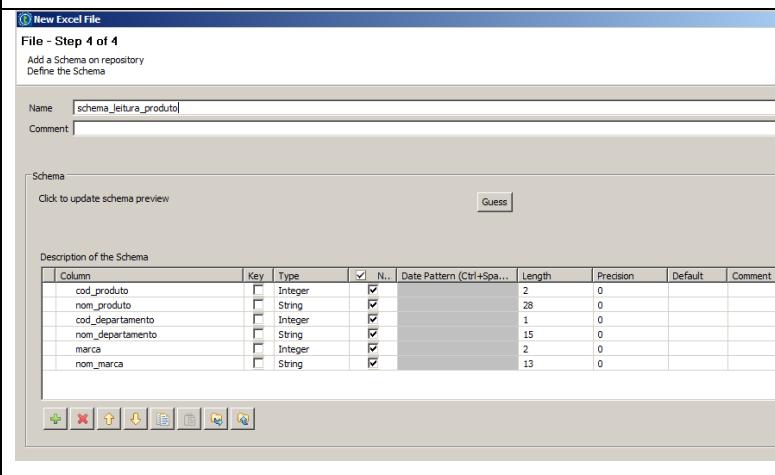
- Escolha a aba **produto**, dentro da planilha **notas fiscais.xlsx**.



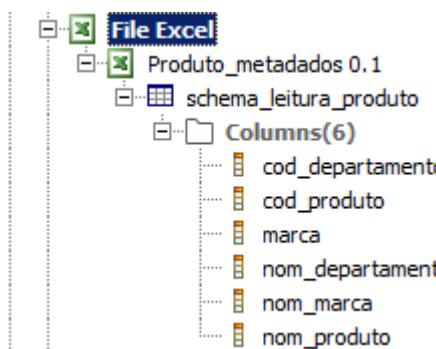
- 3- Marque a opção **set heading row as column name**. Para identificar a primeira linha como cabeçalho do arquivo de dados.



- 4- Defina o nome do schema para o arquivo de leitura. **Schema_leitura_produto**.

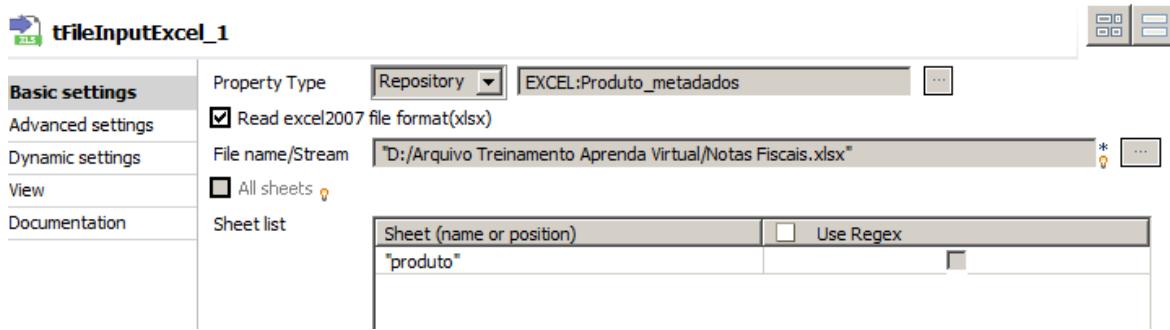


Pronto, o metadados que vai lhe ajudar a carregar os dados está disponível:

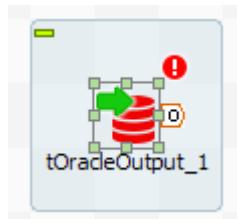


Para que você possa utilizá-lo, basta arrastá-lo sobre o componente **t_FileInputExcel-1**

Note que o arquivo está corretamente disponibilizado para a carga.



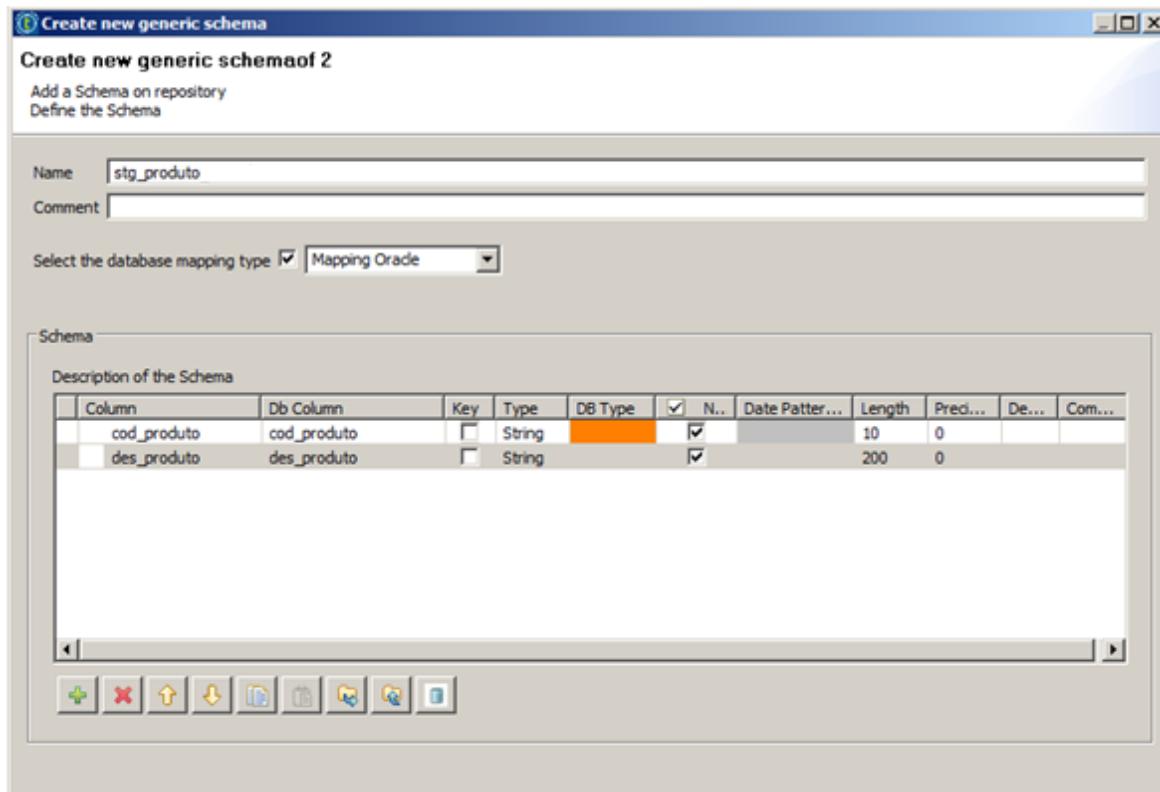
Agora utilizaremos um componente chamado de **TOutputOracle** que será utilizado para guardar os dados carregados.



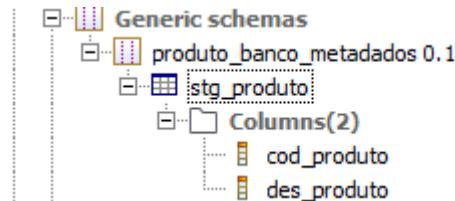
Ao clicar na aba **Basic Settings**, você talvez não tenha a conexão **ojdbc.jar** instalada, basta instalar para utilizar, veja a mensagem:



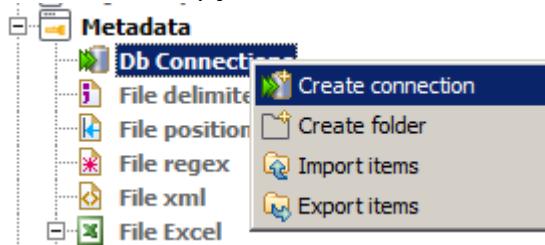
Vamos criar mais um metadados para o arquivo de saída do banco dados, pois caso contrário você fará a configuração manualmente. Utilizaremos a criação do **schema genérico**, lá escolheremos a opção **select the database mapping type** a opção **mapping oracle**.



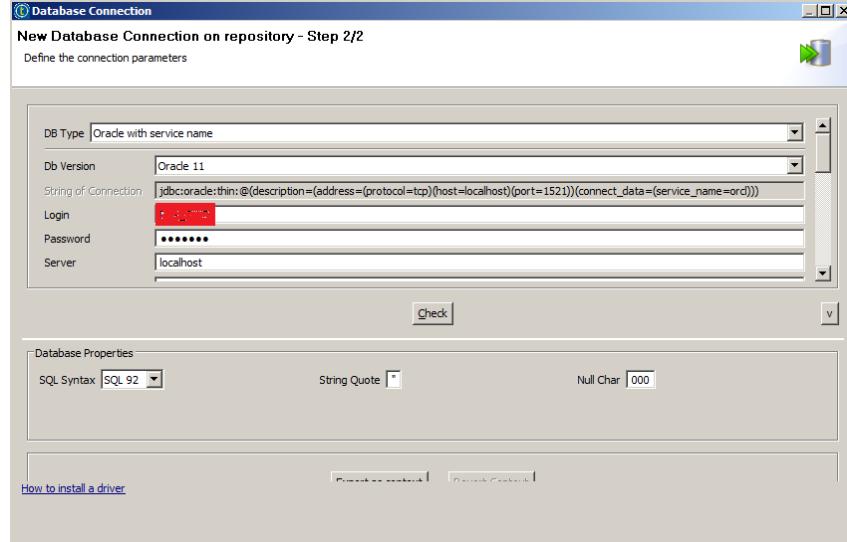
Ficará da seguinte forma:



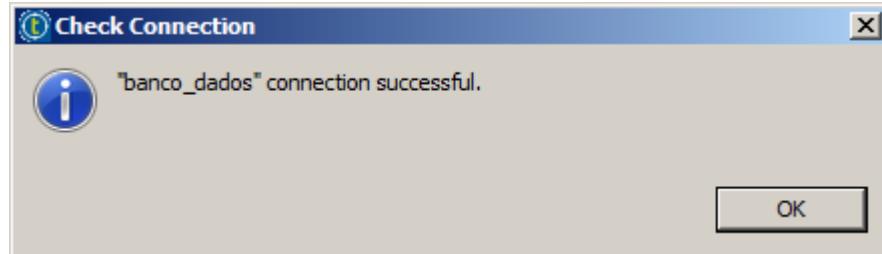
Agora vamos criar a conexão com o banco de dados, escolha em **Repositório -> Metadados** a opção **DbConnection**.



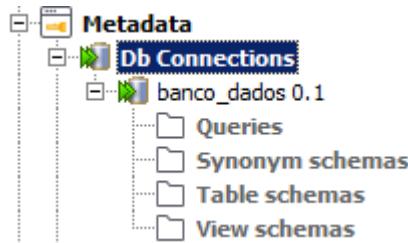
Você deve criar a conexão com seu banco de dados, basta digitar as informações de conexão.



Clique em **check** e verifique a mensagem abaixo:

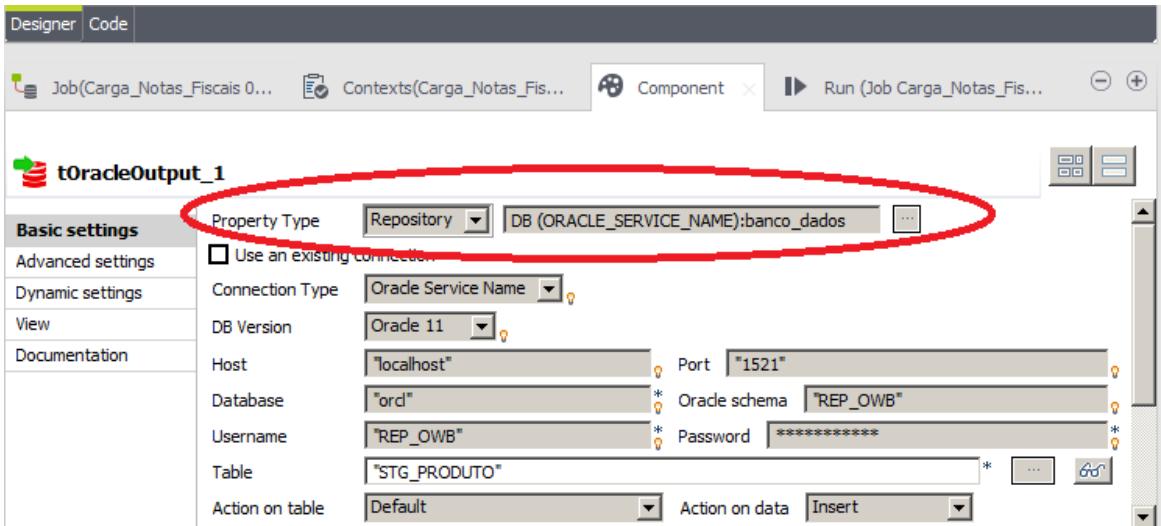


Deverá aparecer na árvore do repositório a conexão:



Agora, vamos escolher a conexão e colocar a tabela do banco de dados, chamada **STG_PRODUTO**. Isto nas opções do componente **TOutputOracle**.

Na opção **Repository (repositório)** escolha a conexão que você acabou de criar, caso contrário você terá que digitar manualmente.



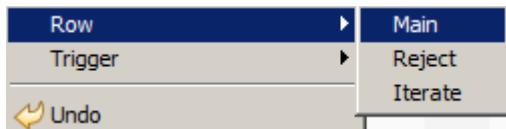
Em **action on Table**, escolha a opção **CREATE TABLE**.

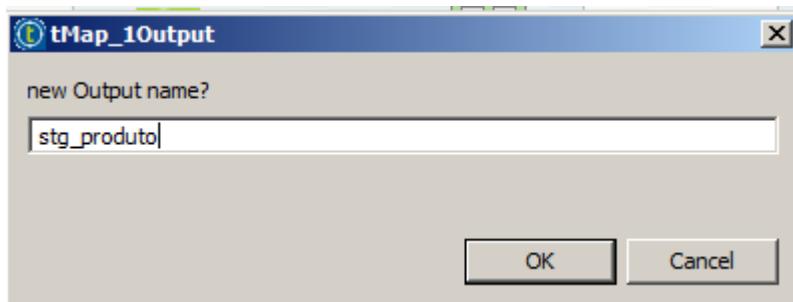
Em schema selecione o **Generic Schema** produto_banco_metadados.

Agora vamos utilizar um outro componente que interligará a entrada e a saída de dados, é o objeto **tMap**. O objeto **tMap** encontra-se em **Processing -> Fields -> tMap**, ou digite o nome do componente no **search** dos componentes.

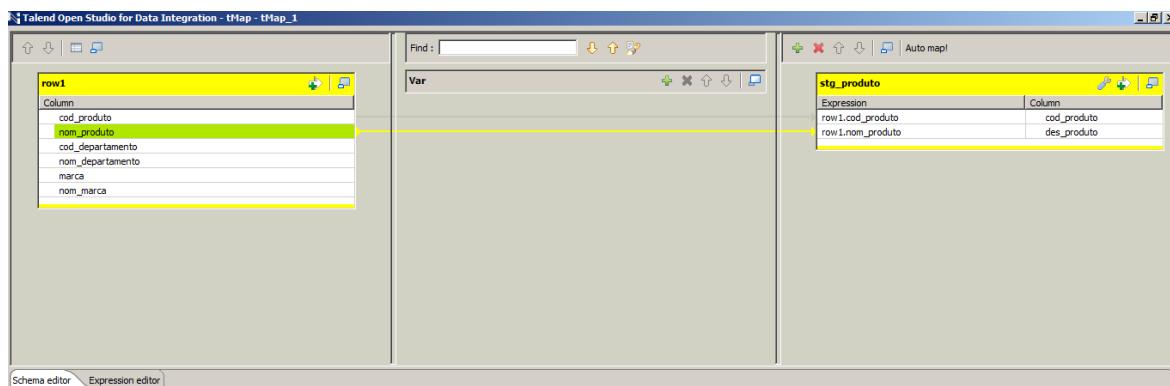
Agora para fazer a ligação entre os componentes **tFileInputExcel** e **TOutputOracle** é necessário mapear a transformação, primeiro crie links entre os três objetos dispostos no ambiente de design, para isso:

- 1- Clique com o botão direito no **objeto** -> **Row** -> **Main** e arraste até o objeto seguinte.



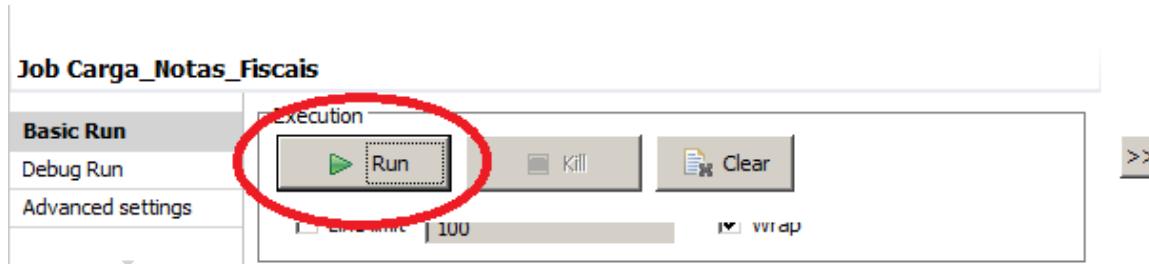


2- Em seguida dê dois clique no objeto **tMap** e clique em auto Map.



Note que o atributo **nom_produto** não encontrará seu correspondente, pois o nome do campo na tabela está diferente, para isso basta clicar e arrastar até o campo **des_produto**.

3- Depois vá à aba **run Job** e execute o Job clicando em **Run**.



Ao final esta mensagem deve aparecer, indicando **SUCESSO!**

The screenshot shows the 'Execution' tab of the Talend interface. At the top, there are three buttons: 'Run' (green), 'Kill' (grey), and 'Clear' (highlighted with a dotted line). Below the buttons is a scrollable log window containing the following text:

```
[statistics] connecting to socket on port 3481
[statistics] connected
[statistics] disconnected
Job Carga_Notas_Fiscais ended at 21:27
19/04/2017. [exit code=0]
```

At the bottom of the log window, there are two checkboxes: 'Line limit' (unchecked) and 'Wrap' (checked).

Pronto, agora que já demos nosso primeiro passo, vamos iniciar nosso projeto de construção de um DW com o Talend Data Integration.

Construção do DW

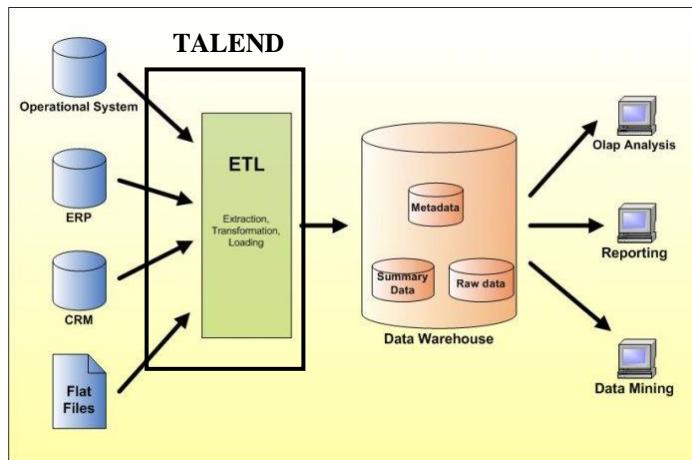
Business Intelligence

Business Intelligence (BI), é um termo criado pelo Gartner Group. Este conceito surgiu na década de 80 e descreve as habilidades das corporações para explorar informações que normalmente estão em um Data Warehouse/Data Mart. BI é um processo que auxilia no desenvolvimento Tático, Estratégico e Operacional da Corporação.

1. **BI Tático:** É a aplicação de ferramentas de BI para analisar tendências de negócios, comparando métricas em diferentes períodos a fim de descobrir tendências onde devemos ter mais atenção.
2. **BI Estratégico:** Apoia o alto comando da empresa na gestão de desempenho do negócio, apoiando-o em longo prazo metas e objetivos corporativos.
3. **BI Operacional:** é utilizado pela linha de frente do negócio, para orientar nas decisões operacionais da empresa. O uso do BI Operacional é feito para gerenciar e aperfeiçoar as operações de negócio.

Data Warehouse

O Data Warehouse (DW) é o banco de dados de suporte a tomada de decisão, alinhado a um programa para coletar, limpar, transformar e armazenar dados de diferentes fontes externas, essas duas partes combinadas são utilizadas para atender as exigências históricas e analíticas de um projeto de Business Intelligence.



TRABALHO PRÁTICO

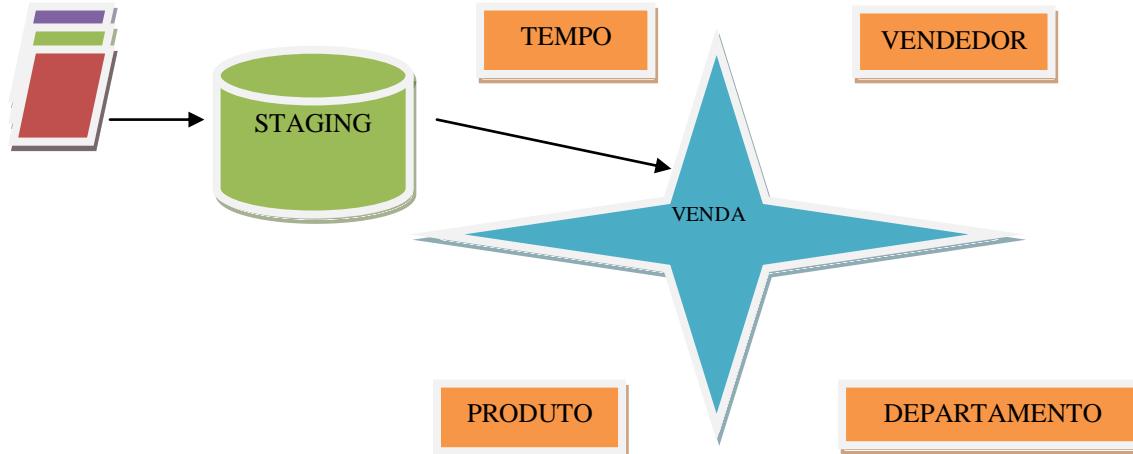
O Trabalho Prático consiste em construir um Data Mart de notas fiscais que são emitidas por uma loja, neste DW teremos as dimensões (**Vendedor, Produto, Tempo e Departamento**) e essas informações serão cruzadas com o valor da nota fiscal.

MODELO TRANSACIONAL

No desenvolvimento do trabalho prático, iremos utilizar como fonte de dados o Arquivo notas fiscais cujas abas possuem os arquivos no formato Excel (**Departamento, Vendedor, Produto, Vendas**), esses arquivos serão utilizados para a realização da carga no modelo multidimensional.

MODELO MULTIDIMENSIONAL

O Modelo Multidimensional criado contém 04 dimensões (**DIM_DEPARTAMENTO, DIM_VENDEDOR, DIM_PRODUTO, DIM_TEMPO**) e uma tabela Fato (**FATO_VENDA**).

Arquivo XLS**Criando Staging**

A área de *Staging* ou área de transição é o armazenamento de dados intermediário entre uma fonte de dados original e o repositório de dados centralizado. Toda purificação, reconciliação, transformação e relacionamentos necessários acontecem nessa área. Essa implementação é muito bem definida e progressiva. Ter uma área de *Staging* é uma estratégia comum e simples de obter informações de um sistema transacional sem a necessidade de realizar transformações complexas.

Vamos Criar 04 tabelas para receber os dados da planilha eletrônica.

STG_PRODUTO → Receberá os dados da aba Produto.

STG_VENDEDOR → Receberá os dados da aba Vendedor.

STG_DEPARTAMENTO → Receberá os dados da aba Departamento.

STG_VENDA → Receberá os dados da aba Nota Fiscal.

OBS: A dimensão tempo será executada com base em uma procedure para a carga dos dados.

As etapas de criação de nosso projeto vão obedecer de forma sequencial o seguinte fluxo de execução.

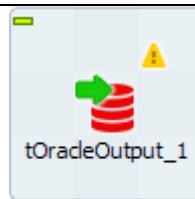
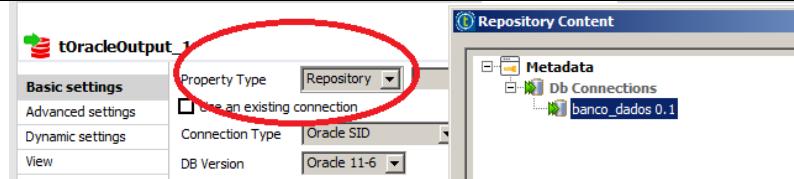
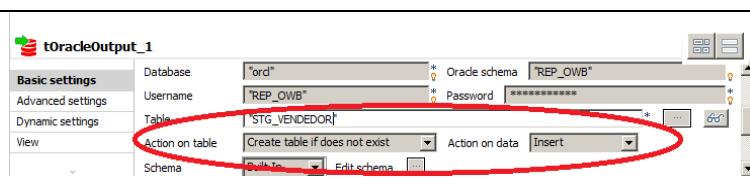
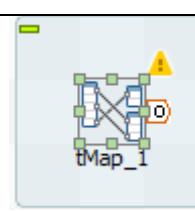
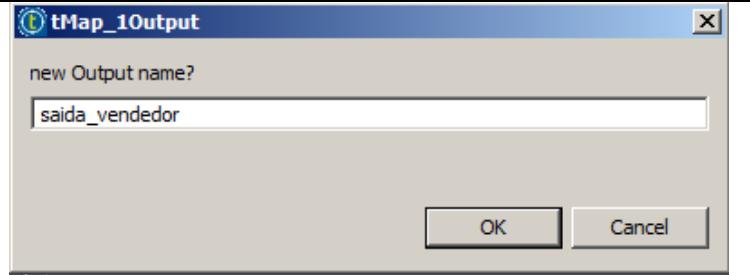
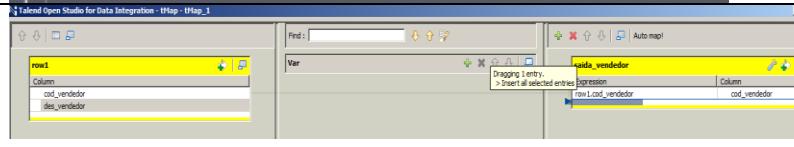
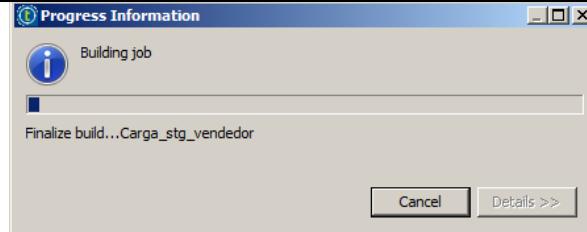


CARGA DA STAGING

Inicialmente começamos o projeto criando uma carga da **STG_PRODUTO**, chamada de **carga_produto**. Agora iremos criar as demais cargas. Como exemplificação, colocarei o passo a passo da criação da **carga_stg_vendedor**.

<p>1. Clique com o botão direito em Job. 2. Clique em Create Job.</p>	
<p>3. Vamos designar o nome do Job como carga_stg_vendedor. 4. Clique em OK.</p>	
<p>5. Crie o metadados vendedor_metadados. 6. Formate e selecione a aba da planilha vendedor.</p>	

	<p>File - Step 2 of 4 Add a Metadata file on repository Define the path of the file and the format settings</p> <p>File Settings</p> <p>Server: Localhost 127.0.0.1 File: D:\Arquivo Treinamento Aprenda Virtual\Notas Fiscais.xlsx <input type="checkbox"/> Read excel2007 file format(xlsx) Generation mode: Memory-consuming(User mode)</p> <p>File Viewer and Sheets setting</p> <p>Set sheets parameters Please select sheet (Sheet structure as schema guide): Indedor</p> <table border="1"> <thead> <tr> <th>A</th> <th>B</th> </tr> </thead> <tbody> <tr><td>cod_vendedor</td><td>Claudio Silva</td></tr> <tr><td>1</td><td>Márcio Barroso</td></tr> <tr><td>2</td><td>Isaac Palmeiras</td></tr> <tr><td>3</td><td>Daniela Matos</td></tr> <tr><td>4</td><td>Maria Braga</td></tr> <tr><td>5</td><td>Rodrigo Machado</td></tr> <tr><td>6</td><td>José Oliveira</td></tr> <tr><td>7</td><td></td></tr> </tbody> </table>	A	B	cod_vendedor	Claudio Silva	1	Márcio Barroso	2	Isaac Palmeiras	3	Daniela Matos	4	Maria Braga	5	Rodrigo Machado	6	José Oliveira	7										
A	B																											
cod_vendedor	Claudio Silva																											
1	Márcio Barroso																											
2	Isaac Palmeiras																											
3	Daniela Matos																											
4	Maria Braga																											
5	Rodrigo Machado																											
6	José Oliveira																											
7																												
<p>7. Selecione os campos cod_vendedor e des_vendedor. Troque o encoding para ISO-8859-1, isso irá fazer com que os sinais das grafias sejam gravados no banco de dados.</p>	<p>File - Step 3 of 4 Add a Metadata file on repository Define the setting of the parse job</p> <p>File Settings</p> <p>Encoding: ISO-8859-1 <input type="checkbox"/> Advanced separator (for number) Thousands separator: , Decimal separator: .</p> <p>Rows To Skip If any rows must be ignored, specify the following parameters: Header: 1 Footer:</p> <p>Metadata column setting</p> <p>First column: 1 Preview: Output</p> <p>Set heading row as column names cod_vendedor des_vendedor</p> <table border="1"> <thead> <tr> <th>cod_vendedor</th> <th>des_vendedor</th> </tr> </thead> <tbody> <tr><td>1</td><td>Claudio Silva</td></tr> <tr><td>2</td><td>Márcio Barroso</td></tr> <tr><td>3</td><td>Isaac Palmeiras</td></tr> <tr><td>4</td><td>Daniela Matos</td></tr> <tr><td>5</td><td>Maria Braga</td></tr> </tbody> </table> <p>Export as context Revert Context</p>	cod_vendedor	des_vendedor	1	Claudio Silva	2	Márcio Barroso	3	Isaac Palmeiras	4	Daniela Matos	5	Maria Braga															
cod_vendedor	des_vendedor																											
1	Claudio Silva																											
2	Márcio Barroso																											
3	Isaac Palmeiras																											
4	Daniela Matos																											
5	Maria Braga																											
<p>8. Troque o tipo do campo cod_vendedor para String. Coloque o nome do schema de: schema_leitura_vendedor.</p>	<p>File - Step 4 of 4 Add a Schema on repository Define the Schema</p> <p>Name: metadata Comment:</p> <p>Schema Click to update schema preview Guess</p> <p>Description of the Schema</p> <table border="1"> <thead> <tr> <th>Column</th> <th>Key</th> <th>Type</th> <th>N...</th> <th>Date Pattern (Ctrl+Space)</th> <th>Length</th> <th>Precision</th> <th>Default</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>cod_vendedor</td> <td></td> <td>String</td> <td><input checked="" type="checkbox"/></td> <td></td> <td>15</td> <td>0</td> <td></td> <td></td> </tr> <tr> <td>des_vendedor</td> <td></td> <td>String</td> <td><input checked="" type="checkbox"/></td> <td></td> <td></td> <td>0</td> <td></td> <td></td> </tr> </tbody> </table>	Column	Key	Type	N...	Date Pattern (Ctrl+Space)	Length	Precision	Default	Comment	cod_vendedor		String	<input checked="" type="checkbox"/>		15	0			des_vendedor		String	<input checked="" type="checkbox"/>			0		
Column	Key	Type	N...	Date Pattern (Ctrl+Space)	Length	Precision	Default	Comment																				
cod_vendedor		String	<input checked="" type="checkbox"/>		15	0																						
des_vendedor		String	<input checked="" type="checkbox"/>			0																						
<p>9. Escolha o componente tFileInputExcel_1 para ler o metadados que você acabou de criar.</p>	<p>tFileInputExcel_1</p>																											
<p>10. Na opção Property Type, escolha o metadados criado, vendedor_metadados.</p>	<p>tFileInputExcel_1</p> <p>Basic settings</p> <p>Property Type: Repository <input type="checkbox"/> Read excel2007 file format(xlsx) File name/Stream: *C:/Users/Administrador/Desktop/Notas Fiscais.xlsx View: <input type="checkbox"/> All sheets</p> <p>Repository Content</p> <ul style="list-style-type: none"> Metadata <ul style="list-style-type: none"> File Excel <ul style="list-style-type: none"> Produto_metadados 0.1 vendedor_metadados 0.1 																											

11. Vamos adicionar o componente tOracleOutput_1	
12. Clique em Property para escolher a opção Repository e a conexão de banco já criada.	
13. Crie a tabela STG_VENDEDOR , escolha as ações Create Table IF does not exist e a ação de dados INSERT .	
14. Vamos ligar os dois componentes, para que os dados possam ser gravados . Utilize o componente tMap_1 .	
15. Faça a ligação entre os componentes, simplesmente selecionando com a tecla shift . Chame a ligação de saída_vendedor .	
16. Arraste os campos da entrada para a saída , pois a tabela será criada.	
17. Clique em Run Job e execute a carga.	

18. Pronto,
carga
carregada.

		Colunas	Dados	Model	Constraints	Concessões	Estatísticas	Trigg
		COD_VENDEDOR	DES_VENDEDOR					
1	1		Claudio Silva					
2	2		Márcio Barroso					
3	3		Issac Palmeiras					
4	4		Daniela Matos					
5	5		Maria Braga					
6	6		Rodrigo Machado					
7	7		João Oliveira					
8	8		Márcia Brito					
9	9		Diego Santos					
10	10		Danilo Barreto					
11	11		Filipe Almeida					
12	12		Marilia Cardoso					

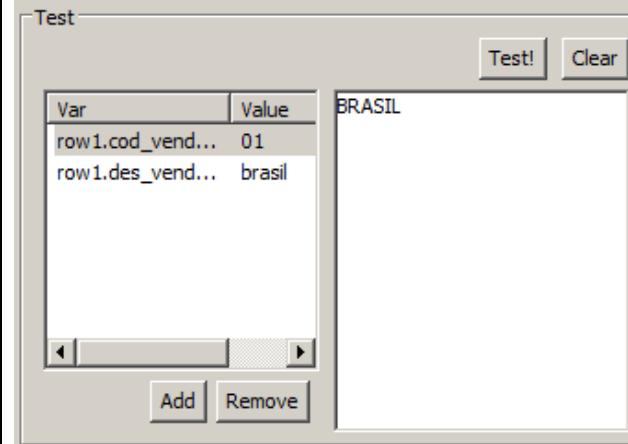
Transformação dos Dados

Antes de criarmos as cargas staging de **DEPARTAMENTO** e **VENDAS**, vamos ver como podemos utilizar as transformações em carga de DW.

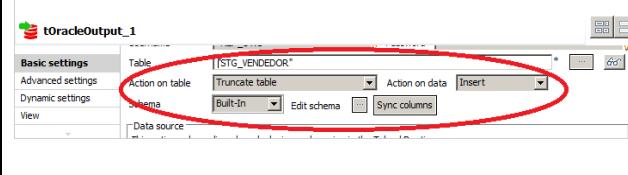
1. Abra o componente **tMap_1**.
Clique na opção Expressão.

2. Vamos colocar todos os valores do campo **des_vendedor** em maiúsculo. Para isso, acrescente o comando:
StringHandling.UPCASE(row1.des_vendedor).

3. Você pode fazer um **teste** e verificar se está tudo ok com a função, basta digitar os valores na aba **Test**. Vamos digitar o código **01** e a palavra **Brasil**.



4. Pronto, está tudo certo, vamos salvar. Caso a sua tabela não esteja na ação **TRUNCATE**, modifique conforme a figura e **re-execute** a carga.



5. Pronto, todos os nomes dos **vendedores** estão em **maiúsculo**.

	COD_VENDEDOR	DES_VENDEDOR
1	1	CLAUDIO SILVA
2	2	MÁRCIO BARROSO
3	3	ISSAC PALMEIRAS
4	4	DANIELA MATOS
5	5	MARIA BRAGA
6	6	RODRIGO MACHADO
7	7	JOÃO OLIVEIRA
8	8	MÁRCIA BRITO
9	9	DIEGO SANTOS
10	10	DANILO BARRETO
11	11	FILIPE ALMEIDA
12	12	MARILIA CARDOSO

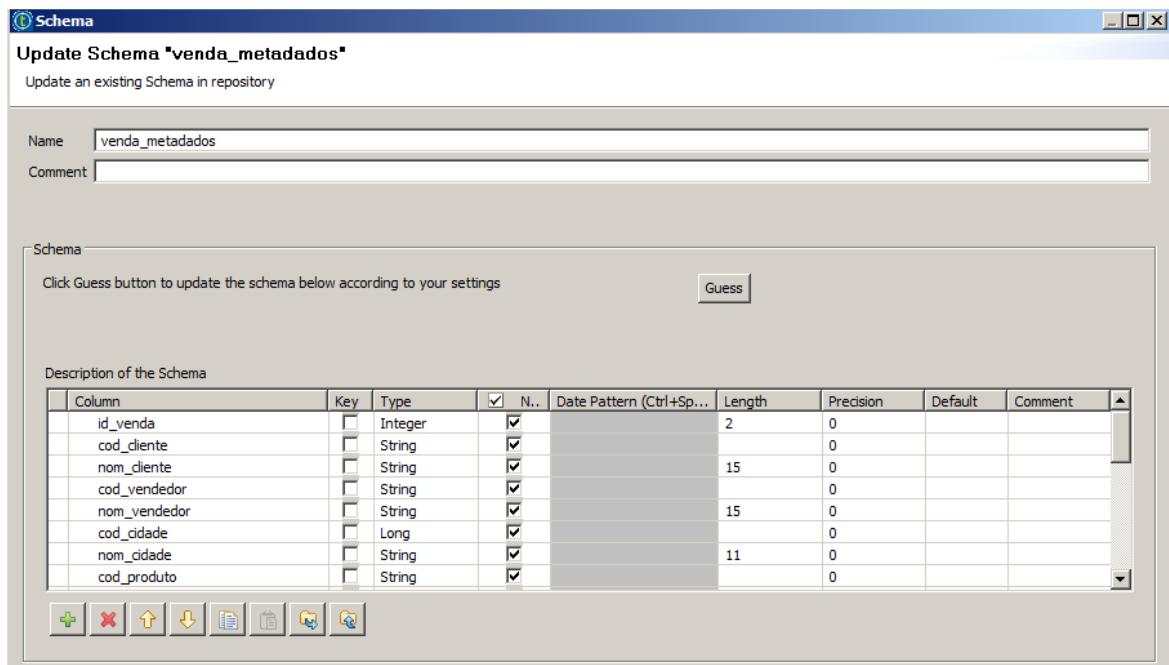
CRIE TODAS AS DEMAIS CARGAS STAGING.

Carga Tabela Staging Fato

Para a carga da tabela de **staging da fato** é importante salientar que é composta geralmente de muitos dados, portanto você deve criteriosamente estudá-la. Importante verificar se possui pelo menos um campo no formato de **data** para ser utilizada na ligação com a **dimensão Tempo** e se possui pelo menos uma **métrica**, claro que você não deve esquecer dos campos que farão uma ligação com a tabela de dimensão, geralmente estas ligações são feitas com as chaves das tabelas operacionais que você levou para a staging.

Agora que possuímos de forma mais clara as necessidades que compõe a tabela staging da fato, vamos ver dicas importantes para transferi-la com cuidado.

- 1- Crie um Schema de metadados com a **aba Notas Fiscal**, pois será nela que você lerá os dados. Converta os campos em **código** para string, **data** para string, **métricas** para string, pois vamos utilizar transformações nos dados para praticarmos.

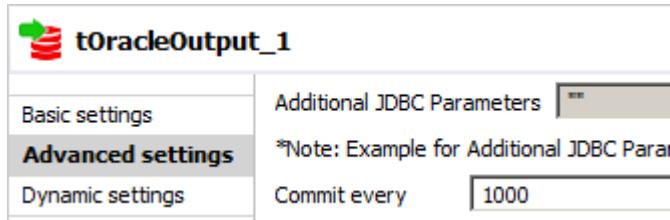


- 2- Crie conversões no **Tmap** para os campos **dtc_venda**, **qtd_venda** e **val_venda**.

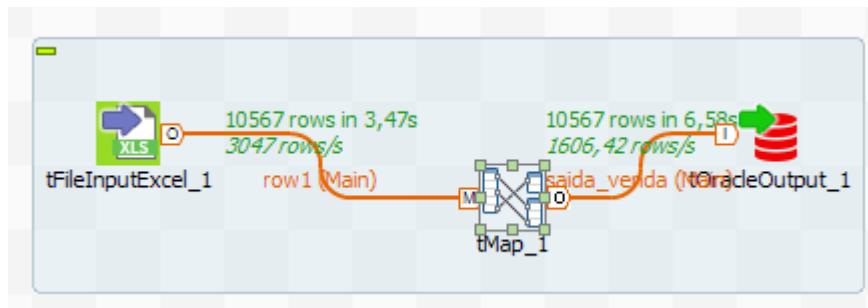
```

dtc_venda → TalendDate.parseDate( "dd/MM/yyyy",row1.dtc_venda)
qtd_venda → Integer.parseInt(row1.qtd_venda)
val_venda → Float.parseFloat(StringHandling.EREPLACE(row1.val_venda
,"","","."))
```

- 3- Como nós possuímos muitos dados para inserir, vamos alterar o **commit** do banco de dados para **1000 em 1000**. Atualmente está em **10000**. Isso evitará erros na carga.

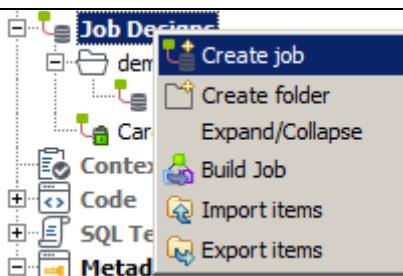
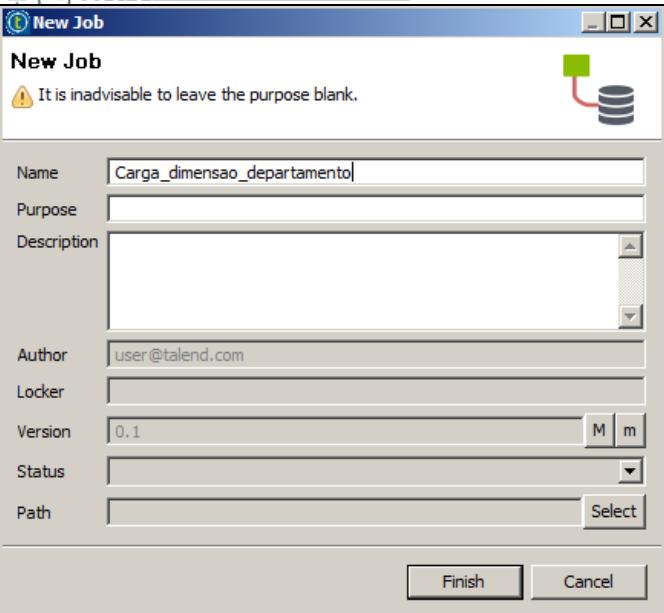
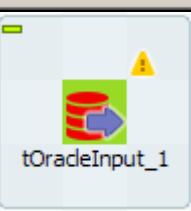


Pronto, agora é executar a carga.



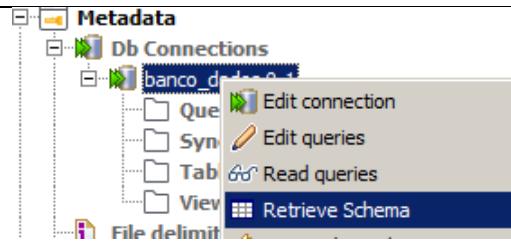
CARGA DA DIMENSÃO

A tabela dimensão representa os objetos importantes para a empresa e contém informações descritivas e textuais do negócio. As dimensões são as principais fontes de consultas nos relatórios, elas atuam como um link para as tabelas fatos, e seus dados fornecem grupos de relatórios.

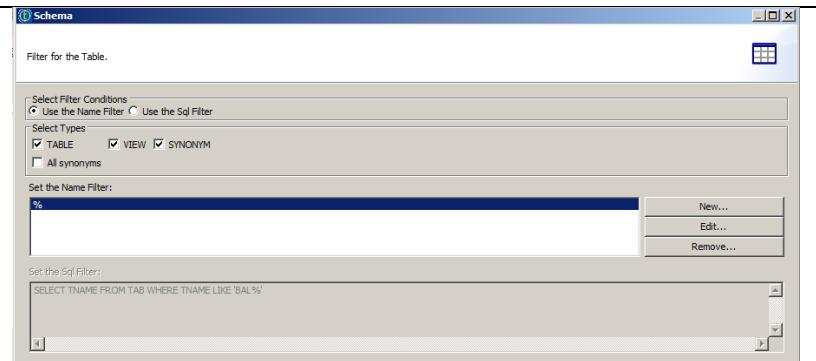
<p>1. Clique com o botão direito em Job.</p> <p>2. Clique em Create Job.</p>	
<p>3. Vamos designar o nome do Job como carga_dim_departamento</p> <p>4. Clique em OK.</p>	
<p>5. Vamos adicionar o objeto que chama a tabela staging de departamento e configurá-lo com a tabela stg_departamento.</p>	

6. Vamos adicionar a ligação com Tmap para selecionar os campos que irá popular nossa dimensão.	
7. Vamos adicionar o Objeto que iremos utilizar para popular a dimensão, para isso pesquise o componente TOracleSCD .	
8. Vamos criar a tabela que será a dimensão dim_departamento . Isso você fará no SQL developer .	<pre> CREATE TABLE "DIM_DEPARTAMENTO" ("SK_DEPARTAMENTO" NUMBER NOT NULL ENABLE, "COD_DEPARTAMENTO" VARCHAR2(10 BYTE), "DES_DEPARTAMENTO" VARCHAR2(200 BYTE), "DTC_INICIO" DATE, "DTC_FIM" DATE, CONSTRAINT "DIM_DEPARTAMENTO_PK" PRIMARY KEY ("SK_DEPARTAMENTO")); CREATE INDEX "DEPART_IDX" ON "DIM_DEPARTAMENTO" ("COD_DEPARTAMENTO"); </pre>

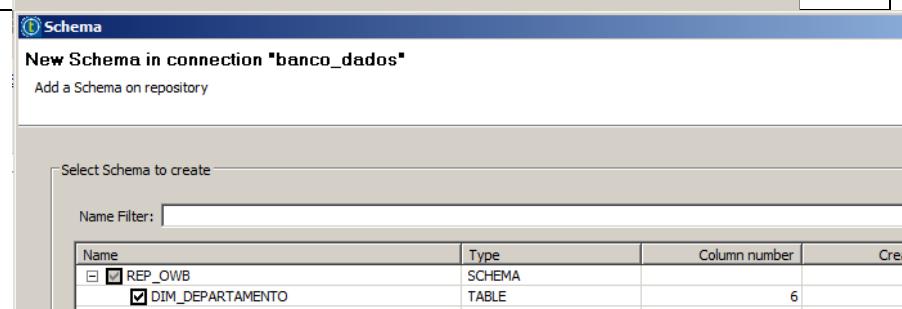
9. Vamos importar a estrutura da dimensão que você criou para o Talend Data Integration.



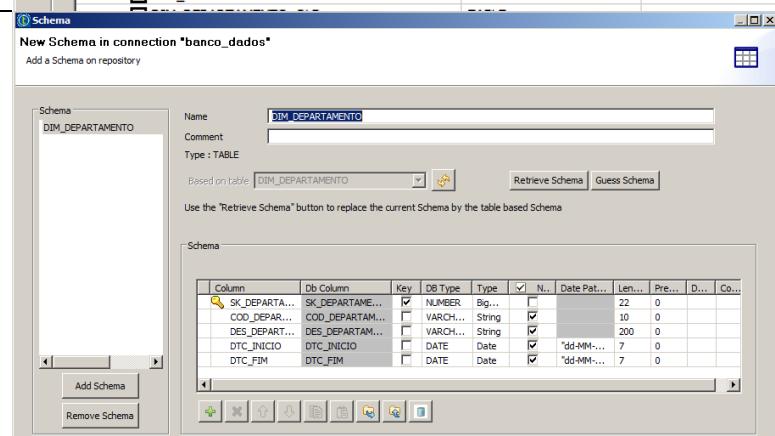
10. Aceite as opções.



11. Escolha a tabela **dim_departamento**.



12. Confirme apenas.



13. Vamos configurar o Objeto **TOracleSCD**. Primeiro vamos conectar a dimensão no banco de dados, lembre-se que criaremos pela primeira vez a tabela de dimensão **dim_departamento**. Podemos ver os dados que serão carregados da staging, clicando nos “óculos”.

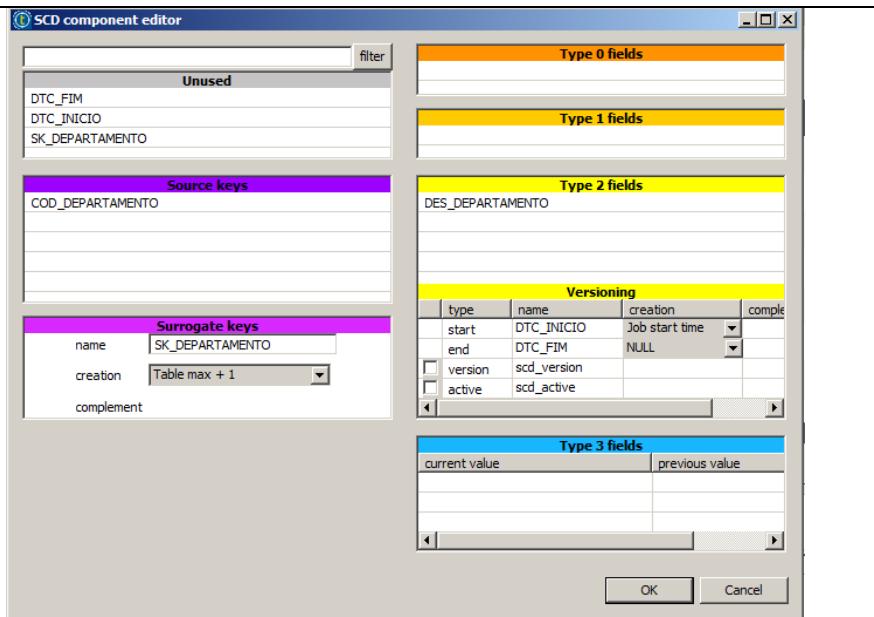
The screenshot shows the Talend Studio interface with two main windows. The top window is titled "tOracleSCD_1" and displays the configuration settings for connecting to an Oracle database. The "Basic settings" tab is selected, showing fields for Host ("localhost"), Database ("ord"), Username ("REP_OWB"), and Table ("dim_departamento"). A red circle highlights the "Test" button (a small icon with a checkmark) at the bottom right of the configuration panel. The bottom window is titled "SQL Builder [Component Mode] - JobCarga_dim_departamento - Component:tOracleSCD_1". It contains a SQL editor with the query "select * from stg_departamento" and a result grid showing data from the staging table:

COD_DEPARTAMENTO	DES_DEPARTAMENTO
1	INFORMÁTICA
2	TELEFÔNIA
3	PAPELARIA
4	COSMÉTICO
5	ELETRÔNICO
6	ELETRODOMÉSTICO

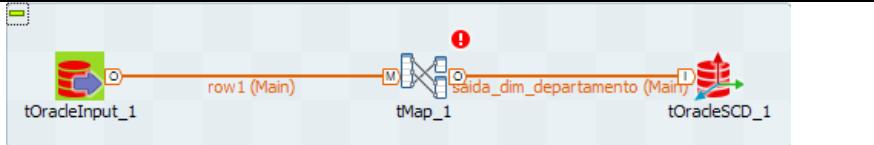
14. Vamos unir o **TMap** ao Objeto **TOracleSCD**. Registre a saída como **saída_dim_departamento**.

The screenshot shows a dialog box for the **tMap_1Output** component. The question "new Output name?" is displayed, and the input field contains the value "saída_dim_departamento". At the bottom right are the "OK" and "Cancel" buttons.

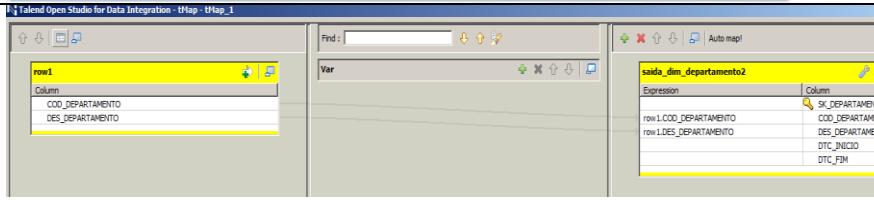
15. Agora clique na opção **SCD Editor**. Preencha conforme o quadro. A dimensão será **SCD 2**



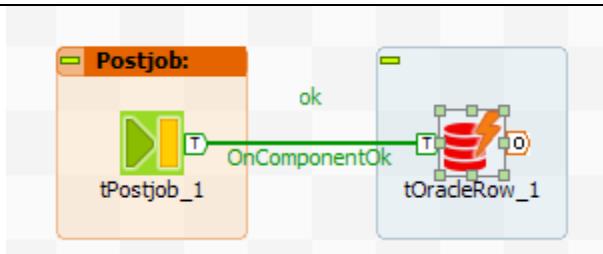
16. Una no Tmap.



17. Faça as ligações necessárias.



18. Para finalizar, todos os primeiros registros inseridos na dimensão, terão valor de **data_inicio** igual a 01/01/1900. Vamos colocar dois novos componentes **Prejob** e **TOracleRow**.



19. O componente **TOracleRow**

```
UPDATE DIM_DEPARTAMENTO DEP
SET DEP.DTC_INICIO=TO_DATE('01/01/1900' 00:00:00','DD/MM/YYYY
HH24:MI:SS')
```

<p>deve ter uma query em especial para fazer a modificação, após a carga da dimensão.</p>	<pre>WHERE EXISTS (SELECT COD_DEPARTAMENTO,COUNT(*) FROM DIM_DEPARTAMENTO DEP2 WHERE DEP.COD_DEPARTAMENTO=DEP2.COD_DEPARTAMENTO GROUP BY COD_DEPARTAMENTO HAVING(COUNT(*)=1))</pre>
<p>20. Ao final a dimensão ficará da seguinte forma.</p>	<p>The diagram illustrates a Talend job flow:</p> <ul style="list-style-type: none"> tOracleInput_1: An Oracle input component with a green arrow icon. It has a data preview window showing "9 rows in 0,05s" and "191,49 rows/s". tMap_1: A mapping component with a blue arrow icon. It receives data from the input and has a data preview window showing "9 rows in 1,47s" and "6,13 rows/s". Postjob: A component labeled "Postjob:" containing a tPostjob_1 component. It has a green arrow icon and an "ok" status indicator. tOracleRow_1: An Oracle output component with a red arrow icon, connected to the postjob component.

Controle das dimensões

Agora, chegou um momento importante, vamos antes entender os conceitos de modelagem de dados para DW, entenderemos a construção de dimensões do tipo **SCD1** e **SCD2**.

Dimensão Tipo SCD1: Permite que os dados inseridos na dimensão sejam substituídos, ou seja, não gerem histórico dos dados.

Dimensão Tipo SCD2: Permite que os dados inseridos na dimensão guardem o histórico dos dados, ou seja, haverá registros dentro da dimensão que serão versionados, possuem a mesma chave do operacional, mas com validades diferentes.

]

Um exemplo SCD 2:

STG_DEPARTAMENTO

COD_DEPARTAMENTO	DES_DEPARTAMENTO
1	INFORMÁTICA
2	PAPELARIA

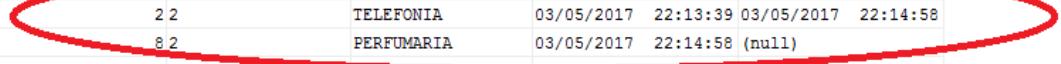
Digamos que o departamento 2, mudou de **PAPELARIA** para **PERFUMARIA**. Quando Executássemos a dimensão, os dados ficariam da seguinte forma:

DIM_DEPARTAMENTO

COD_DEPARTAMENTO	SK_DEPARTAMENTO	DES_DEPARTAMENTO	DTC_INICIO	DTC_FIM
1	1	INFORMÁTICA	10-05-2016	
4	2	PAPELARIA	20-07-2016	13-09-2016
4	3	PERFUMARIA	14-09-2016	

Perceba que ambos os registros foram conservados.

No nosso projeto, para que você perceba funcionando, vamos alterar a **stg_departamento** no **cod_departamento=2** a descrição de **TELEFONIA** para **PERFUMARIA**. Ao executar o JOB, o resultado ficará parecido no SQL Developer, da seguinte forma:



SK_DEPARTAMENTO	COD_DEPARTAMENTO	DES_DEPARTAMENTO	DTC_INICIO	DTC_FIM
1	1.1	INFORMÁTICA	03/05/2017 22:13:39 (null)	
2	2.2	TELEFONIA	03/05/2017 22:13:39	03/05/2017 22:14:58
3	8.2	PERFUMARIA	03/05/2017 22:14:58 (null)	
4	3.3	PAPELARIA	03/05/2017 22:13:39 (null)	
5	9.4	ACUPUTURA	03/05/2017 22:20:20 (null)	
6	5.5	ELETRÔNICO	03/05/2017 22:13:39 (null)	
7	6.6	ELETRODOMESTICO	03/05/2017 22:13:39 (null)	
8	7.7	MOVEIS	03/05/2017 22:13:39 (null)	

Vamos realizar o mesmo procedimento para as demais dimensões: vendedor e produto.

Os Scripts para criação das dimensões estão aqui:

DIM_PRODUTO

```
CREATE TABLE "DIM_PRODUTO"
(  "SK_PRODUTO" NUMBER NOT NULL ENABLE,
  "COD_PRODUTO" VARCHAR2(10 BYTE),
  "DES_PRODUTO" VARCHAR2(200 BYTE),
  "DTC_INICIO" DATE,
  "DTC_FIM" DATE,
CONSTRAINT "DIM_PRODU_DIM_KEY_PK" PRIMARY KEY ("SK_PRODUTO"));
CREATE INDEX "NPRODUT_IDX" ON "DIM_PRODUTO" ("COD_PRODUTO");
```

DIM_VENDEDOR

```
CREATE TABLE "DIM_VENDEDOR"
(  "SK_VENDEDOR" NUMBER NOT NULL ENABLE,
  "COD_VENDEDOR" VARCHAR2(10 BYTE),
  "DES_VENDEDOR" VARCHAR2(200 BYTE),
CONSTRAINT "DIM_VENDED_DIM_KEY_PK" PRIMARY KEY ("SK_VENDEDOR"));
CREATE INDEX "NVENDEDED_IDX" ON "DIM_VENDEDOR" ("COD_VENDEDOR");
```

Antes de encerrar, vamos criar um registro chamado de **sem informação (código = -1)**, isso devido à inconsistência de dados, pois podem ocorrer em projetos de DW, onde os dados do operacional não são íntegros. Por exemplo, **vendas de um determinado vendedor na tabela vendas do banco de dados**, cujo código de identificação não existe, ou não foi cadastrado, a colocação de uma identificação de inconsistência permitirá que a gestão da empresa veja que há dados inconsistentes na base.

Colocado em cada dimensão o código=-1, possibilitará que quando exibirmos os dados em uma ferramenta OLAP, Data Discovery ou Report qualquer, este seja identificado.

Execute os scripts abaixo, para cada dimensão:

DIM_DEPARTAMENTO:

```
Insert           into          DIM_DEPARTAMENTO
(SK_DEPARTAMENTO,COD_DEPARTAMENTO,DES_DEPARTAMENTO,DTC_INICIO,DTC_FIM)
values ('-1','-1','SEM INFORMACAO',to_date('01/01/1800 00:00:00','DD/MM/YYYY
HH24:MI:SS'),null);
```

DIM_VENDEDOR:

```
Insert into DIM_VENDEDOR (SK_VENDEDOR,COD_VENDEDOR,DES_VENDEDOR) values
('-1','-1','SEM INFORMACAO');
```

DIM_PRODUTO:

```
Insert           into          DIM_PRODUTO
(SK_PRODUTO,COD_PRODUTO,DES_PRODUTO,DTC_INICIO,DTC_FIM) values ('-1','-1',
'SEM INFORMACAO',to_date('01/01/1800 00:00:00','DD/MM/YYYY
HH24:MI:SS'),null);
```



CARGA DA DIMENSÃO TEMPO

A dimensão TEMPO é uma das mais importantes que existem em um projeto de Data Warehouse, sem ela, você não consegue estabelecer análises temporais. Com os dados da dimensão TEMPO você identifica quando uma determinada métrica foi inserida. Veja o exemplo:

TABELA VENDAS

COD_DEPARTAMENTO	COD_VENDEDOR	DTC_VENDA	VAL_VENDA
1	3	10/03/2016	250,00
2	4	15/05/2015	320,00

Note, as compras são identificadas conforme a data da venda. Sem a data da venda você não consegue fazer uma análise do tipo:

- Quais produtos são mais vendidos em determinado período.
- Se o primeiro trimestre obteve mais vendas que o segundo trimestre.
- Total de vendas por ano.

Dentre outras perguntas.

A nossa dimensão tempo será criada pela procedure: **UP_CARGA_DIMENSAO_TEMPO**

Está procedure está disponível no nosso site, basta baixar e executar dentro do banco de dados. **OBS.: Lembrando que ela está no formato de PL/SQL**

Antes, você deve criar a dimensão TEMPO, segue o script.

```
CREATE TABLE "DIM_TEMPO"
(
    "SK_TEMPO" NUMBER NOT NULL ENABLE,
    "DES_DATA_DIA" VARCHAR2(25 BYTE),
    "DTC_DATA" DATE,
    "NUM_ANO" NUMBER(4,0),
    "NUM_MES" NUMBER(3,0),
    "NUM_DIA" NUMBER(3,0),
    "NUM_QUADRIMESTRE" NUMBER(3,0),
    "NUM_TRIMESTRE" NUMBER(2,0),
```

```
"DES_QUINZENA" VARCHAR2(12 BYTE),
"DES_QUADRIMESTRE" VARCHAR2(12 BYTE),
"NUM_BIMESTRE" NUMBER(3,0),
"DES_BIMESTRE" VARCHAR2(12 BYTE),
"DES_ANO_MES" VARCHAR2(8 BYTE),
"DES_DIA" VARCHAR2(7 BYTE),
"NUM_SEMESTRE" NUMBER(3,0),
"DES_MES_ANO_NUMERICO" VARCHAR2(7 BYTE),
"DES_TRIMESTRE" VARCHAR2(12 BYTE),
"IND_FINAL_SEMANA" CHAR(1 BYTE),
"DES_MES" VARCHAR2(15 BYTE),
"NUM_QUINZENA" NUMBER(3,0),
"DES_MES_ANO_COMPLETO" VARCHAR2(30 BYTE),
"DES_SEMESTRE" VARCHAR2(12 BYTE),
"NUM_NIVEL" NUMBER(1,0),
"DES_MES_ANO" VARCHAR2(8 BYTE),
CONSTRAINT "DIM_TEMPO_DIM_KEY_PK" PRIMARY KEY ("SK_TEMPO"));
CREATE INDEX "TEMPO_IDX" ON "DIM_TEMPO" ("DES_DATA_DIA");
```



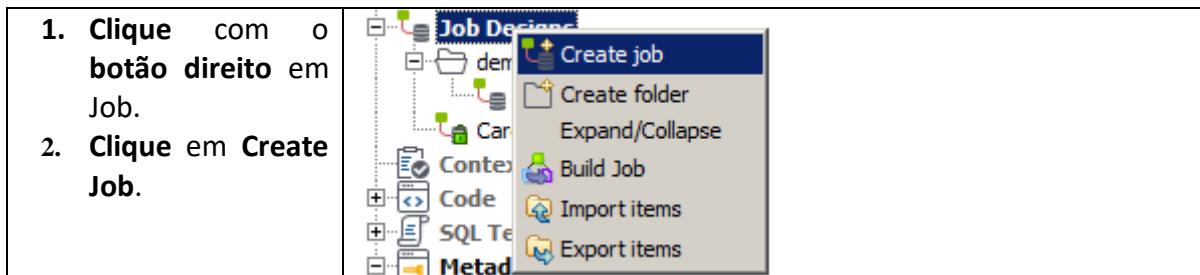
CARGA DA FATO

Estamos chegando ao final de nosso projeto de DW, para isso precisaremos criar a carga da tabela fato, que reunirá todas as chaves das dimensões e as métricas que utilizaremos. Neste momento, você deve imaginar que os dados serão pesquisados nas dimensões e as chaves artificiais gravadas na tabela FATO, ela deve ser da seguinte forma:

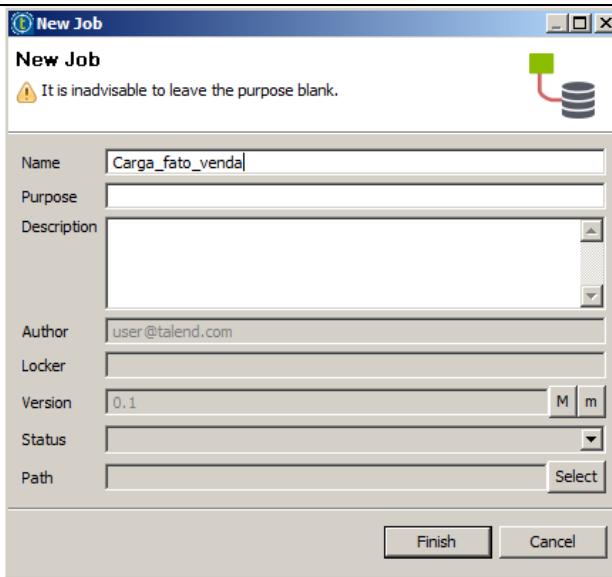
TABELA FATO_VENDAS

SK_DEPARTAMENTO	SK_VENDEDOR	SK_PRODUTO	SK_TEMPO	VAL_VENDA
1	3	1	420	250,00
2	4	1	435	320,00
2	1	2	437	500,00
1	2	2	456	470,00
3	2	4	457	580,00
1	1	7	498	720,00
2	3	1	501	630,00

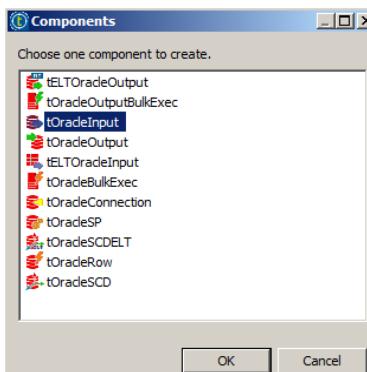
Note que a tabela fato é a maior de todas, contendo todas as informações das dimensões. O que faremos é criar uma carga de dados para a **FATO_VENDAS** que ao ler os dados da sua **STG_VENDAS**, identifiquem na dimensão quais são as chaves artificiais correspondentes (SK's). Utilizaremos componentes do Talend Data Integration que facilite esta pesquisa.



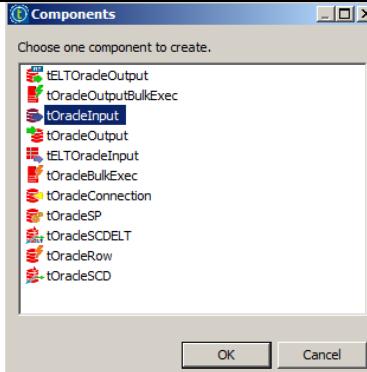
3. Vamos designar o nome do Job como **carga_fato_venda**.
4. Clique em OK.

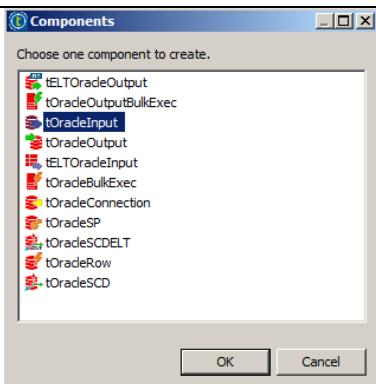
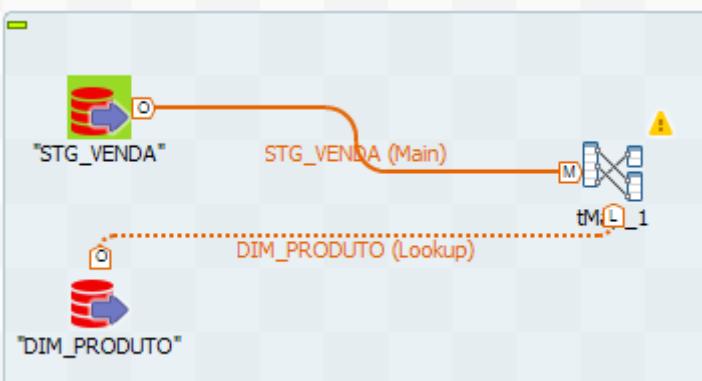


5. Vamos adicionar o objeto que chama a tabela staging de venda no metadados, basta **arrastar** a tabela e soltar.



6. Vamos adicionar também o objeto **dim_produto**, que será utilizado para pesquisar a sk_produto correspondente.



	
7. Ambas devem estar da seguinte forma.	
8. Agora vamos começar a criar a ligação entre ambas e efetuar uma pesquisa entre as tabelas, este procedimento é conhecido como JOIN. Utilizaremos o objeto Tmap para isso.	
9. Renomeia as linhas row1 e row2 para respectivamente stg_venda e dim_produto	

10. Agora vamos abrir o **Tmap** e criar a união entre as tabelas e preparar para a busca dos dados na **dimensão produto**. Arraste o campo **cod_produto** de **stg_venda** e solte sobre o campo **cod_produto** de **dim_produto**, pronto o **JOIN** está criado.

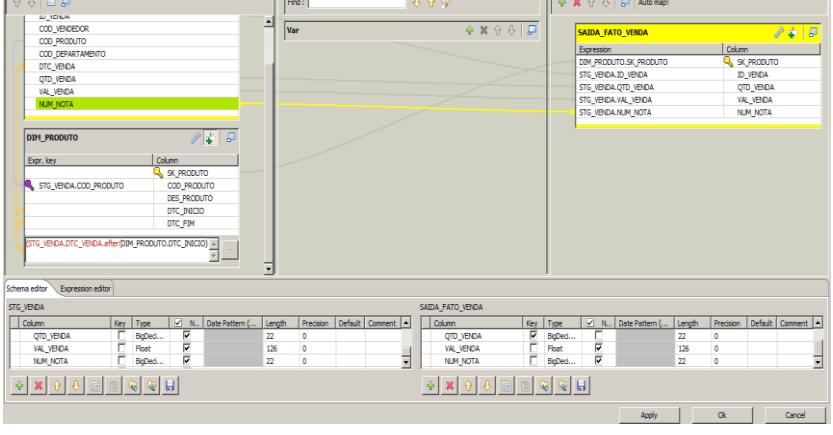
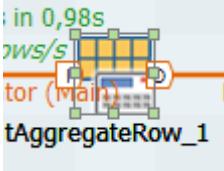
The screenshot shows the Talend Tmap component interface. On the left, the **STG_VENDA** table has columns: ID_VENDA, COD_VENDEDOR, COD_PRODUTO, COD_DEPARTAMENTO, DTC_VENDA, QTD_VENDA, VAL_VENDA, and NUM_NOTA. The **COD_PRODUTO** column is highlighted in yellow. On the right, the **DIM_PRODUTO** table has columns: Expr. key, SK_PRODUTO, COD_PRODUTO, DES_PRODUTO, DTC_INICIO, and DTC_FIM. The **STG_VENDA.COD_PRODUTO** column is mapped to the **SK_PRODUTO** column. A purple arrow points from the **COD_PRODUTO** column in the **STG_VENDA** table to the **SK_PRODUTO** column in the **DIM_PRODUTO** table, indicating the join relationship.

11. Vamos agora criar um filtro para buscar a chave artificial exata na dimensão **dim_produto**. Para isso clique no ícone e digite o código em Java.

```
(STG_VENDA.DTC_VENDA.after(DIM_PRODUTO.DTC_INICIO) ||
STG_VENDA.DTC_VENDA.equals(DIM_PRODUTO.DTC_INICIO)) &&
(STG_VENDA.DTC_VENDA.before((DIM_PRODUTO.DTC_FIM==null)?
TalendDate.parseDate("dd-MM-yyyy", "31-12-2050"):DIM_PRODUTO.DTC_FIM))
```

12. Vamos fazer uma alteração em **Match Model** (encontro de registros) da pesquisa na dimensão, colocaremos a opção **All Match**. Quando a dimensão versionar está mudança permitirá trazer

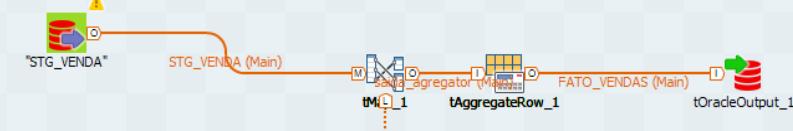
The screenshot shows the **Options** dialog box for the **DIM_PRODUTO** component. In the **Property** tab, the **Match Model** is set to **All matches**. In the **Expr. key** tab, the **STG_VENDA.COD_PRODUTO** column is mapped to the **SK_PRODUTO** column. Below the dialog, the component configuration shows the **Join Model** as **Left Outer Join**. An expression editor window is open, displaying the Java code for filtering rows based on date ranges.

todos os registros da dimensão e selecionaremos a correta.	
13. Vamos fazer a ligação do objeto Tmap com o componente tAggregateRow , para criarmos posteriormente a tabela FATO_VENDAS .	
14. Agora vamos arrastar os campos: ID_VENDA (pk) , VAL_VENDA , QTD_VENDA , NUM_NOTA , SK_PRODUTO para a saída. Estes campos criaram a tabela FATO_VENDAS .	
15. Vamos criar um código no campo SK_PRODUTO , que caso não encontre uma chave correspondente, este será gravado com o valor -1. Lembre-se que criamos um registro com esta informação.	<p>(DIM_PRODUTO.SK_PRODUTO==0)?-1:DIM_PRODUTO.SK_PRODUTO</p> 
16. O TaggregateRow , fará o papel de filtro dos registros	

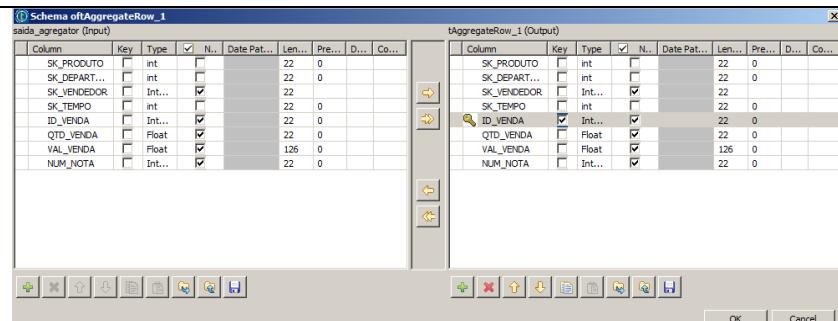
a serem gravados na FATO_VENDAS	
17. Você colocará a SK_PRODUTO , NUM_NOTA com a opção MAX , QTD_VENDA e NUM_NOTA como SUM . O ID_VENDA que é a chave da nossa fato, vamos utilizar no group by . A medida que for inserindo novas dimensões, você vai acrescentando na aba Operations .	
18. Agora, insira o componente TOracleOutput e configure a carga no TOracleOutput , para que a mesma não cancele, veja ao lado. Coloque Commit every 100	
19. Configure e coloque o nome da tabela FATO_VENDAS .	
20. Antes de iniciar,	

converta os tipos de campos **BigDecimal** para **int** ou **Decimal**, de acordo com a natureza do campo. Isso porque, alguns bancos de dados não suportam este tipo de dado. Abra o **Tmap** e altere

- 21.** Execute a carga e verifique o resultado.



- 22.** Vamos colocar a chave na tabela **FATO_VENDAS**, dê um clique em **tAggregateRow** e clique em **Edit Schema**.



- 23.** A tabela fato deve conter todos os registros que existem na **STG_VENDA**, você notará que há sk com -1 e outras que foram encontradas, isso de acordo com o que foi especificado.

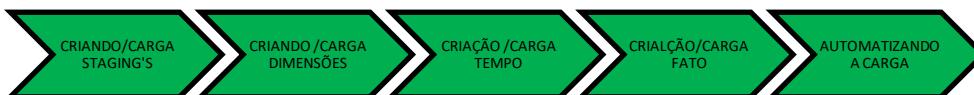
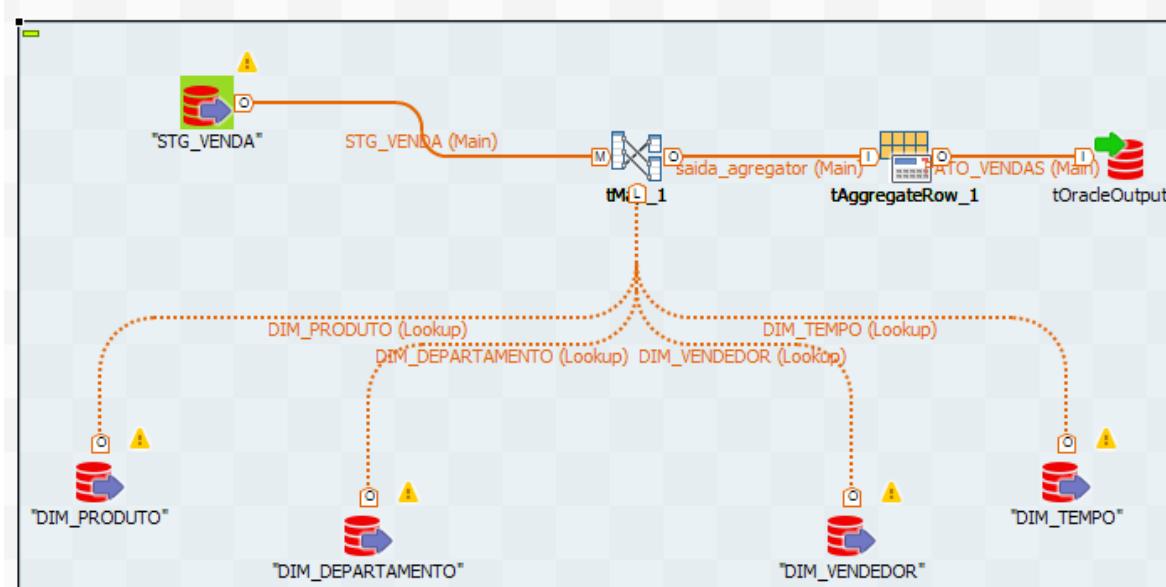
	SK_PRODUTO	ID_VENDA	QTD_VENDA	VAL_VENDA	NUM_NOTA
40	-1	124	3	1600	435
41	1	125	1	25,15	49

Uma observação importante. Devido ao nosso banco ser ORACLE quando for criar a ligação no tMap entre a **STG_VENDA** e a **DIM_TEMPO**, mude a máscara da data da dimensão tempo para **dd/MM/yyyy**.

Schema editor										
Expression editor										
DIM_TEMPO										
Column	Key	Type	N..	Date Pattern (...)	Length	Precision	Default	Comment		
SK_TEMPO	<input checked="" type="checkbox"/>	int	<input type="checkbox"/>		22	0				
DES_DATA_DIA	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		25	0				
DTC_DATA	<input type="checkbox"/>	Date	<input checked="" type="checkbox"/>	"dd/MM/yyyy"	7	0				

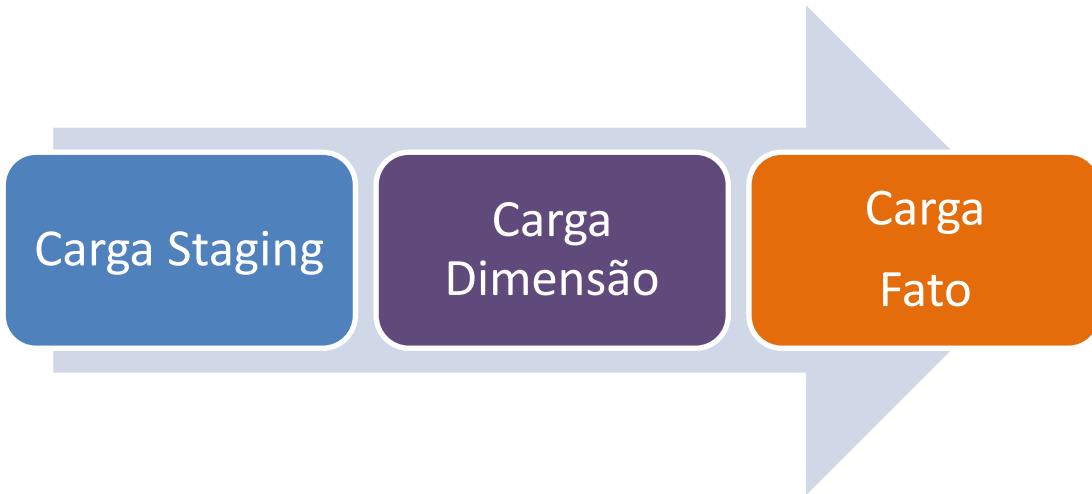
Agora, você deve completar o **tMap** com as outras dimensões: **DIM_DEPARTAMENTO**, **DIM_VENDEDOR**, **DIM_TEMPO** e executar novamente a carga de forma definitiva.

Ao final, sua carga da tabela **Fato Vendas** deve estar da seguinte forma:



AUTOMATIZAÇÃO DAS CARGAS

Vamos automatizar os job's criados, iremos seguir a sequência de execução do projeto de DW.



Antes de Tudo, vamos criar um Job que elimina os registros das staging's ao carregar o DW.

1. Clique com o botão direito em Job.	
2. Clique em Create Job.	
3. Crie o job Carga_trunca_staging	
4. Arraste o componente tOracleRow	

5. Configura com o comando
TRUNCATE TABLE <TABELA>

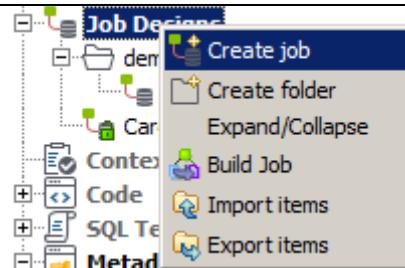


6. Para cada tabela crie uma conexão e crie a ligação entre os componentes, utilize o **MAIN**

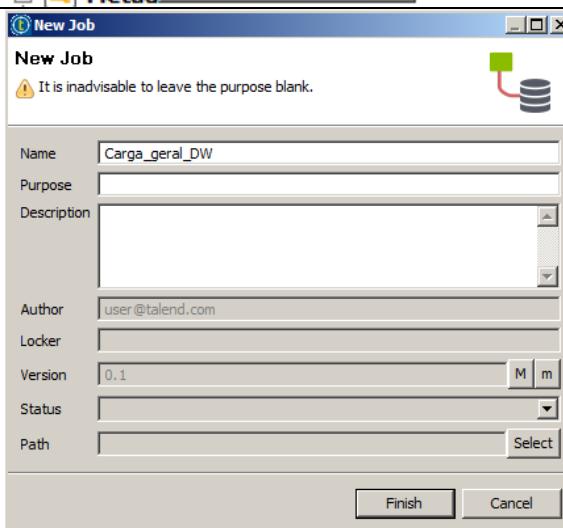


Agora, podemos criar nosso JOB de **Carga Geral DW**

7. Clique com o botão direito em Job.
 8. Clique em Create Job.



9. Crie o job **Carga_geral_DW**



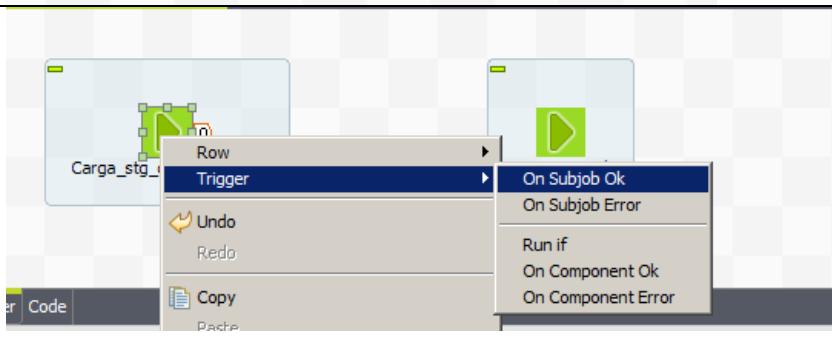
- 10.** Arraste o Job **Carga_trunca_staging** para iniciar.



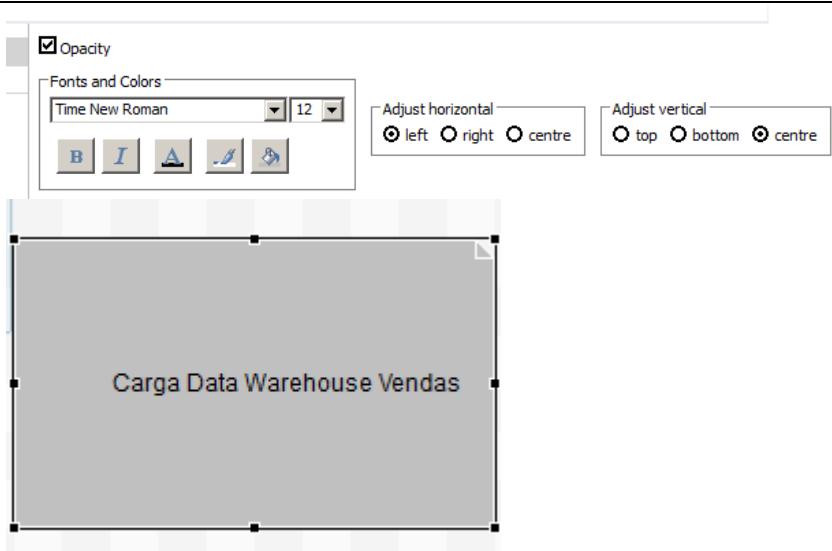
- 11.** Vamos arrastar, **job a job** e interligá-los para seguir uma sequência de execução.



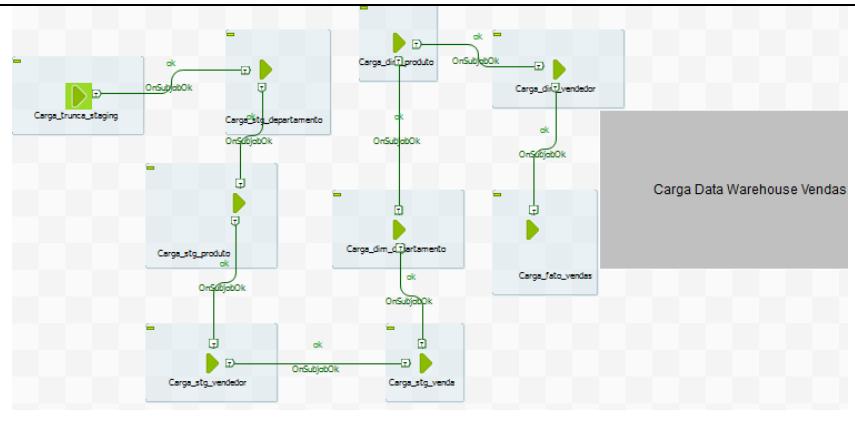
- 12.** Fazendo a Ligação com a trigger **ON Subjob OK**. Crie a ligação Job a Job.



- 13.** Vamos acrescentar uma Anotação para sinalizarmos que é a carga do DW. Puxe o componente **Note**.



14. Pronto, agora é só executar.



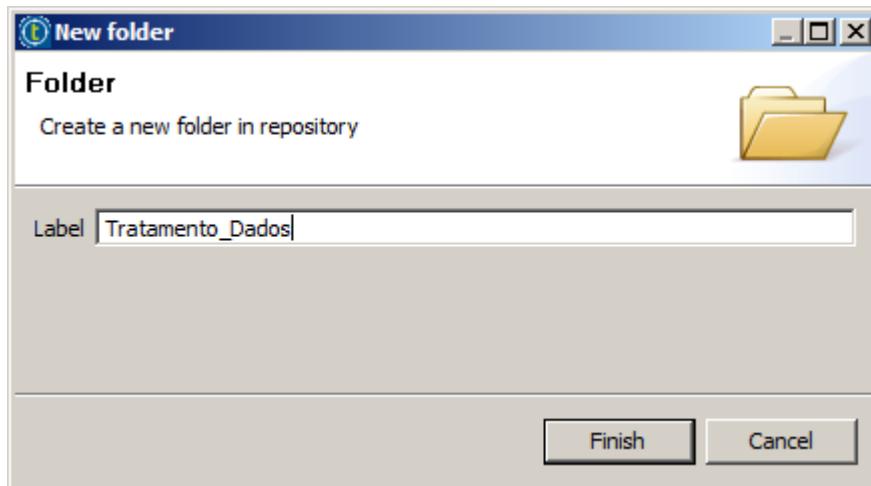
Chegamos ao final de como criar cargas de um projeto de DW do Zero!

Principais componentes do Talend Data Integration

O uso da paleta do Talend Data Integration é **livre** por se tratar de uma ferramenta de ETL, agora as suas necessidades vão sendo adequadas de acordo com sua criatividade e de como você pensa no seu projeto.

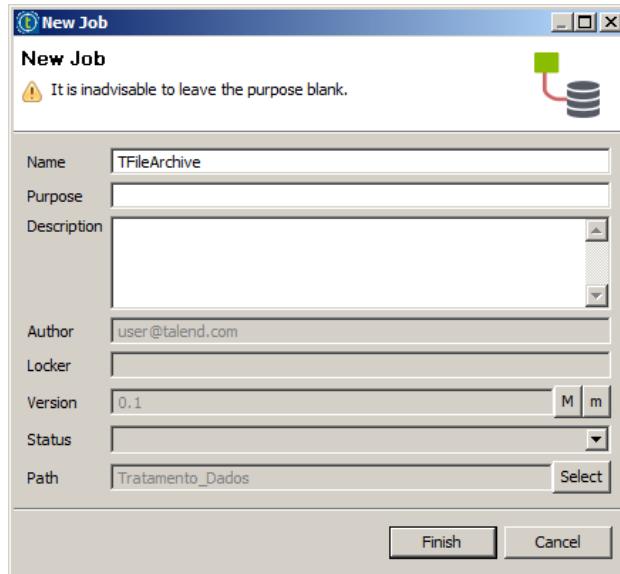
A paleta de objetos é muito rica, vamos começar criando uma extração de dados com componentes sobre **Componentes de tratamento de arquivos**:

Antes disso, crie um folder para esta finalidade, chamado **Tratamento_Dados**.

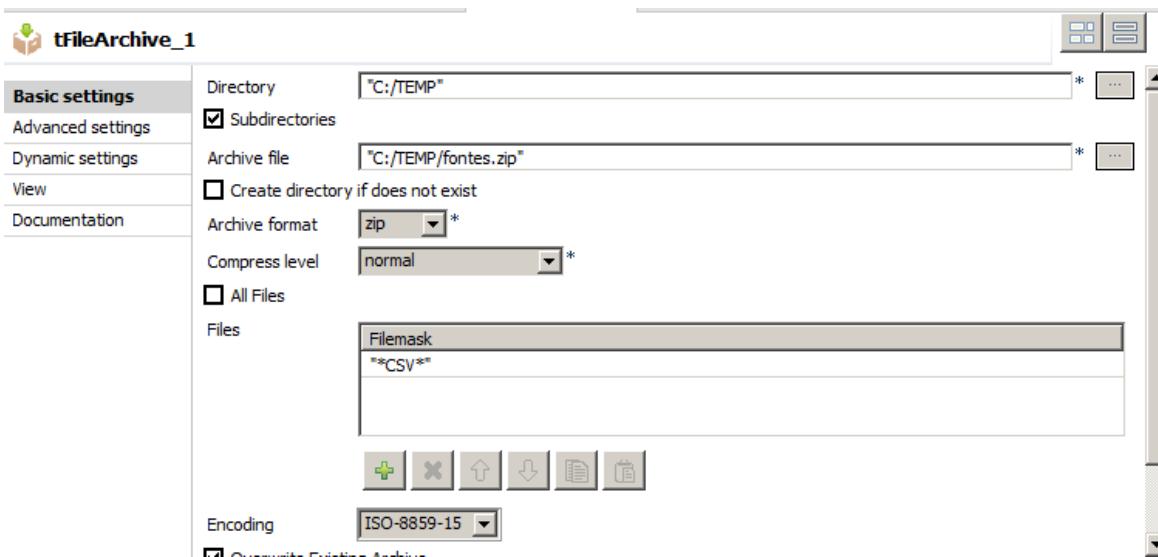


tFileArchive

tFileArchive: Componente criado para compactar um ou mais arquivos no formato **ZIP**.
Crie um Job com o nome do componente.



Arraste o componente e digite diretório, quais arquivos deseja “zipar”, complementando na aba componente. Na nossa atividade, serão os arquivos *.XLSX



Basta executar o JOB.

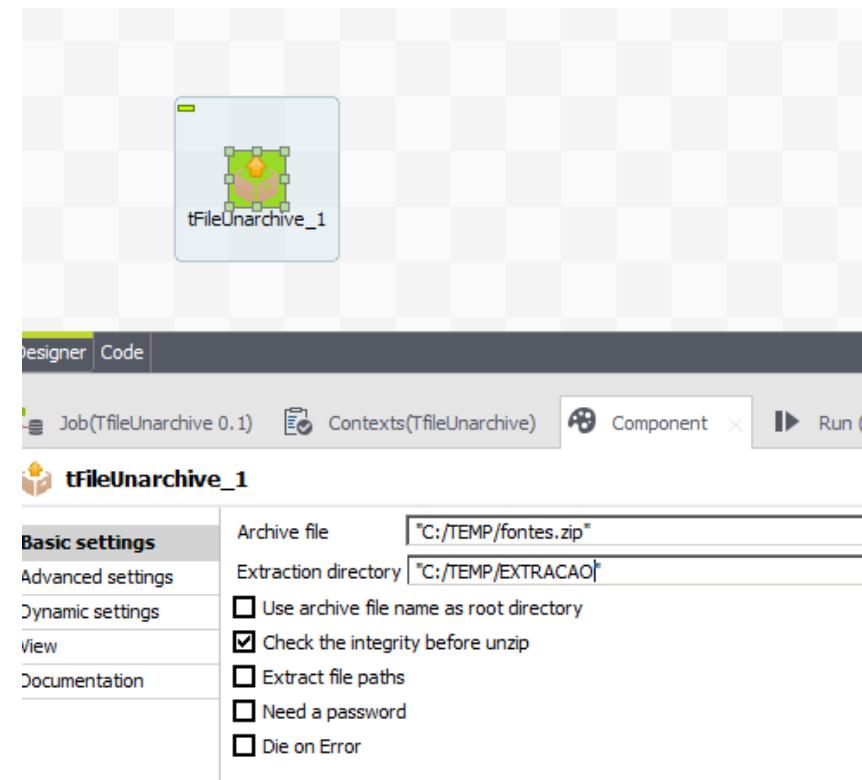
O Arquivo estará compactado.

Name	Date modified	Type
exportar.sql	13/05/2017 10:59	SQL File
Fonte de Dados 1 - Diarias.xlsx	22/05/2017 21:06	XLSX File
Fonte de Dados 2 - CPGF.xlsx	22/05/2017 21:06	XLSX File
fontes.zip	22/05/2017 21:18	Arquivo ZIP do Win...

tFileUnarchive

TfileUnarchive: Componente criado para descompactar um ou mais arquivos no formato ***.tar.gz , *.tgz, *.tar, *.gz and *.zip.**

Vamos criar um Job chamado TfileUnarchive.



Name	Date modified	Type
EXTRACAO	22/05/2017 21:29	File folder
exportar.sql	13/05/2017 10:59	SQL File
Fonte de Dados 1 - Diarias.xlsx	22/05/2017 21:06	XLSX File
Fonte de Dados 2 - CPGF.xlsx	22/05/2017 21:06	XLSX File
fontes.zip	22/05/2017 21:18	Arquivo ZIP do Win..

tFileCompare e tFileOutputDelimited

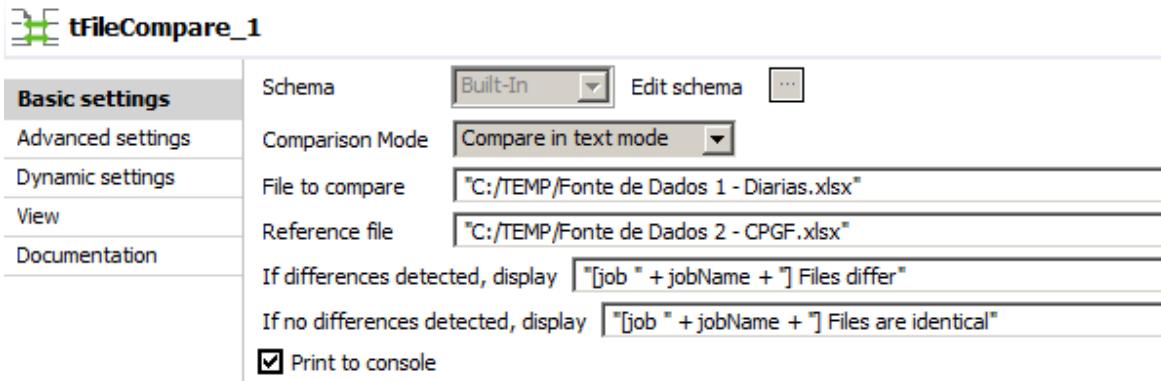
tFileCompare: Componente criado para comparar dois arquivos.

Vamos criar um Job chamado tFileCompare.

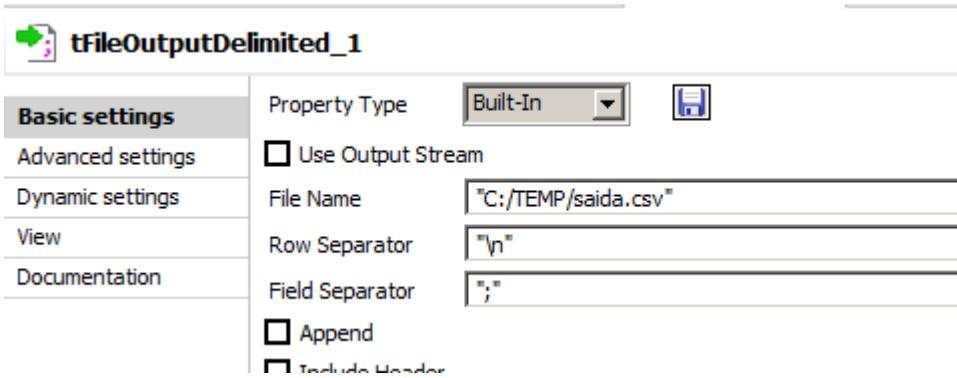
Arraste o componente tFilecompare.



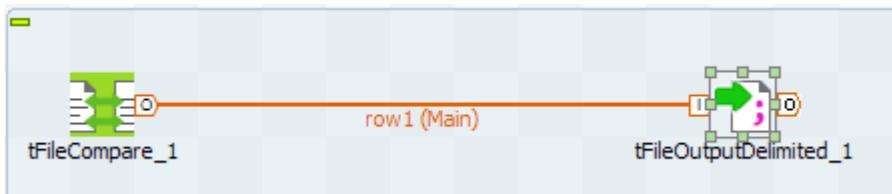
Inclua os dois arquivos a serem comparados:



Grave a saída em um arquivo **saida.csv**, colocando o componente **TfileOutputDelimited**.



Execute o job ligando os dois componentes.



Saída será:

```
P/Fonte de Dados 1 - Diarias.xlsx;C:/TEMP/Fonte de Dados 2 - CPGF.xlsx;2017-05-22 21:41:47;tFileCompare;tFileCompare_1;1;[job tFileCompare] Files differ
```

tFileCopy e tFileList

tFileCopy: Componente criado para copiar um ou mais arquivos entre diretórios.

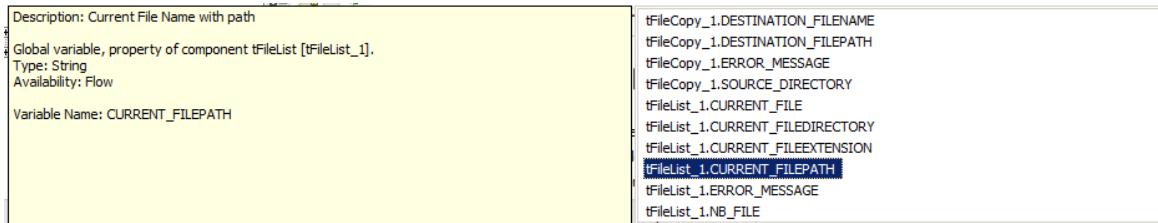
Vamos criar um Job chamado tFileCopy.

Para facilitar a copia de mais arquivos, vamos utilizar o componente **tFileList**

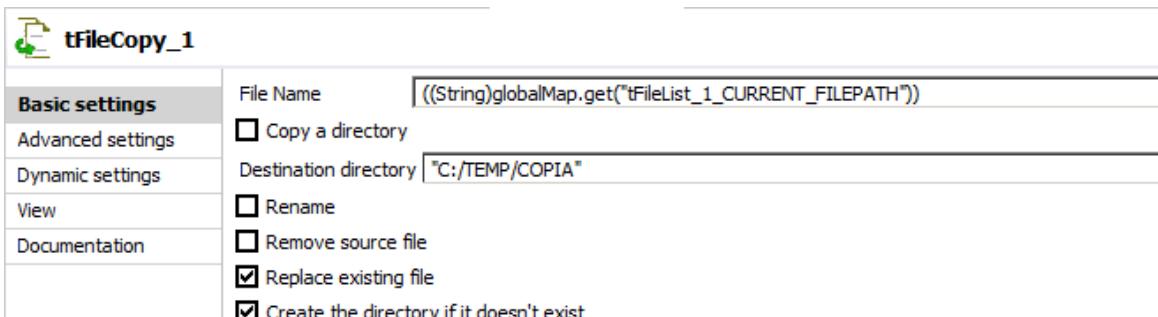
Vamos escolher o diretório aonde os arquivos serão copiados:

Basic settings	Directory: "C:/TEMP"
Advanced settings	FileList Type: Files
Dynamic settings	<input type="checkbox"/> Includes subdirectories
View	Case Sensitive: Yes
Documentation	<input type="checkbox"/> Generate Error if no file found
	<input checked="" type="checkbox"/> Use Glob Expressions as Filenames (Unchecked means Perl5 Regex Expressions)
Files	Filenames "*zip*" "*csv*"
	<input type="button" value="+"/> <input type="button" value="X"/> <input type="button" value="Up"/> <input type="button" value="Down"/> <input type="button" value="New"/> <input type="button" value="Delete"/>
Order by	<input checked="" type="radio"/> By default <input type="radio"/> ...

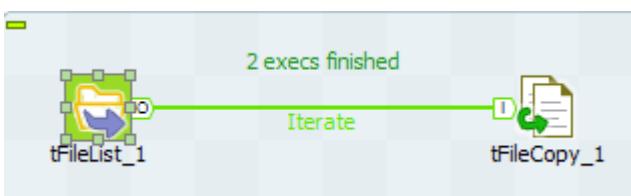
Agora vamos escolher o diretório que os arquivos serão copiados, para isso vamos para o componente **tFileCopy**, vamos aprender a trabalhar com variáveis de ambiente do Talend, pressione as teclas **CTRL + Space** e escolha **TFILE_LIST1:CURRENT_FILEPATH**:



A variável de ambiente em questão, permitira que busquemos os arquivos no diretório fornecido pelo componente **tfileList**, ficará da seguinte forma:



Faça a ligação entre os componentes e execute o Job.



O resultado será a criação do diretório **COPIA** com os arquivos.



tFileDelete e tJava

tFileDelete: Componente criado para deletar um ou mais arquivos entre diretórios.
Vamos criar um Job chamado tFileDelete.

Para facilitar o encontro dos arquivos vamos utilizar o componente **tFileList**.

Vamos escolher o diretório com os arquivos que serão copiados:



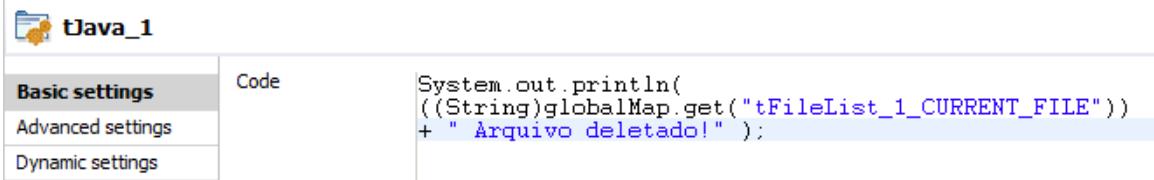
Vamos deletar os arquivos ***.zip**

Agora vamos inserir o componente **tFileDelete** e selecionar o diretório corrente que estamos utilizando, para isso vamos para o componente **TfileDelete**, utilizaremos variável de ambiente do Talend, pressione as teclas **CTRL + Space** e escolha **TFILE_LIST1:CURRENT_FILEPATH**:

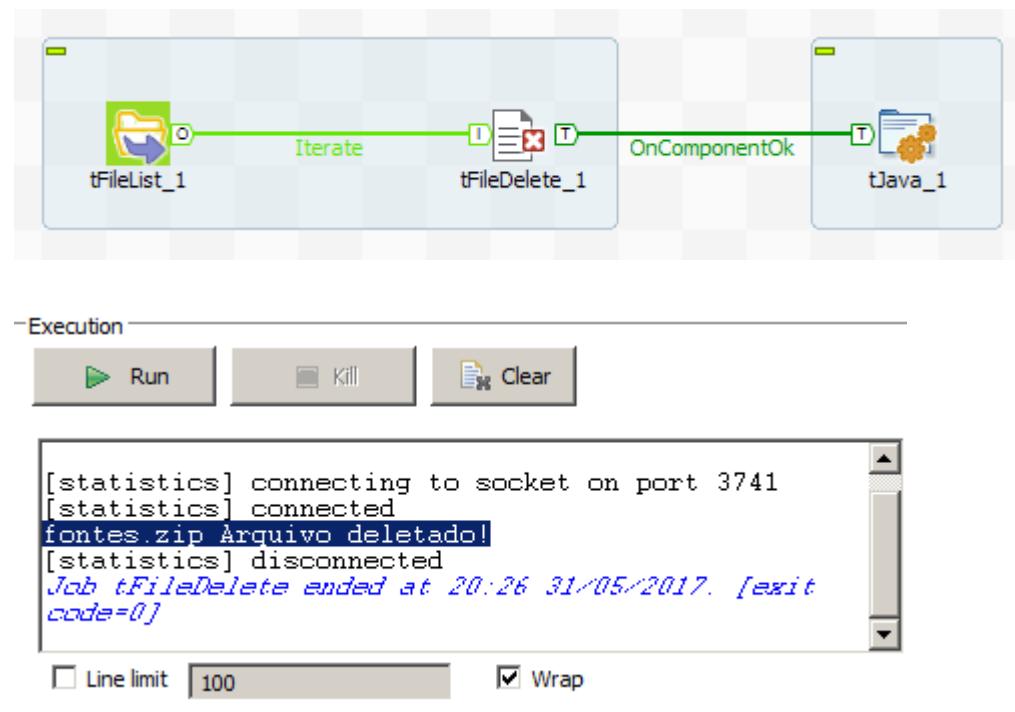


Em seguida, utilizaremos o componente **tJava**, para exibir uma mensagem na saída padrão (Executar console). Neste exemplo, digite o campo **Code**, o seguinte script:

```
System.out.println( ((String)globalMap.get("tFileList_1_CURRENT_FILE"))
+ " Arquivo deletado!" );
```



Ao final execute o Job, ele deve estar assim:



tFileExist e tMsgBox

tFileExist: Componente criado para verificar se um determinado arquivo existe.

Vamos criar um Job chamado tFileExist.

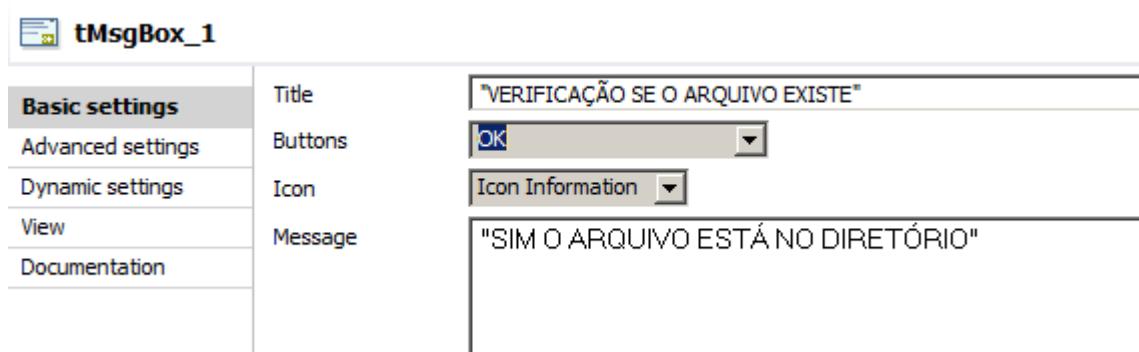


Para facilitar o encontro dos arquivos vamos pesquisar o arquivo **saída.csv** com o componente **tFileExist**

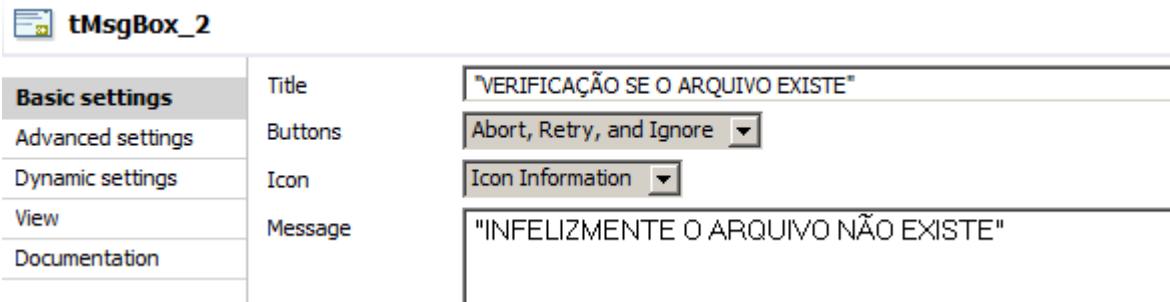


Vamos utilizar o componente **tMsgBox** para exibir uma mensagem na tela se o arquivo existe ou não existe. Vamos criar as duas situações para verificarmos a mensagem que será exibida.

Caso o **arquivo exista**, vamos exibir as seguintes informações:



Caso o **arquivo não exista**, vamos exibir as seguintes informações:



Perceba a opção **Buttons** a depender da situação, você pode prosseguir no job, interromper, ignorar, etc.

Você terá que trabalhar com a condição (IF) entre os componentes:

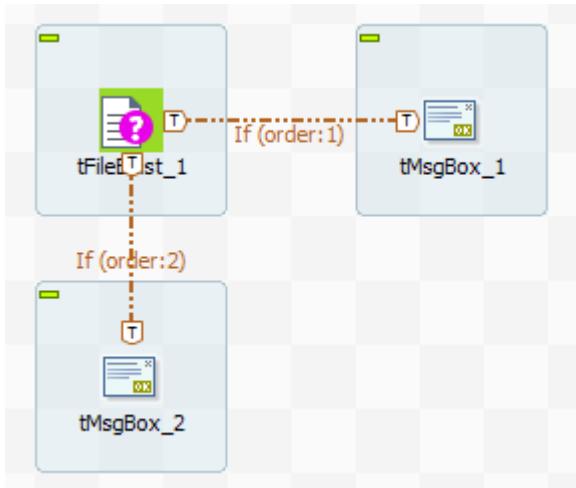
Caso o **arquivo exista**,

```
((Boolean)globalMap.get("tFileExist_1_EXISTS"))
```

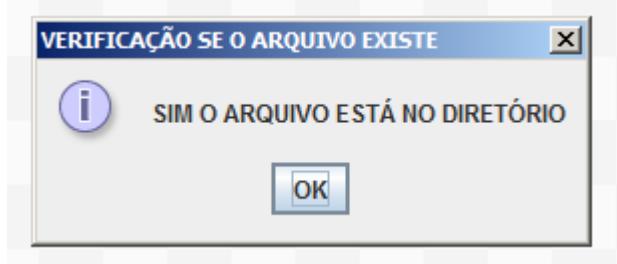
Caso o **arquivo não exista**,

```
!((Boolean)globalMap.get("tFileExist_1_EXISTS"))
```

Ao final o Job deve estar da seguinte forma:

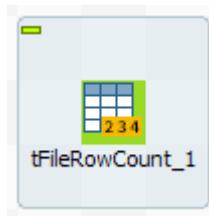


Após executar o Job, o resultado final será:

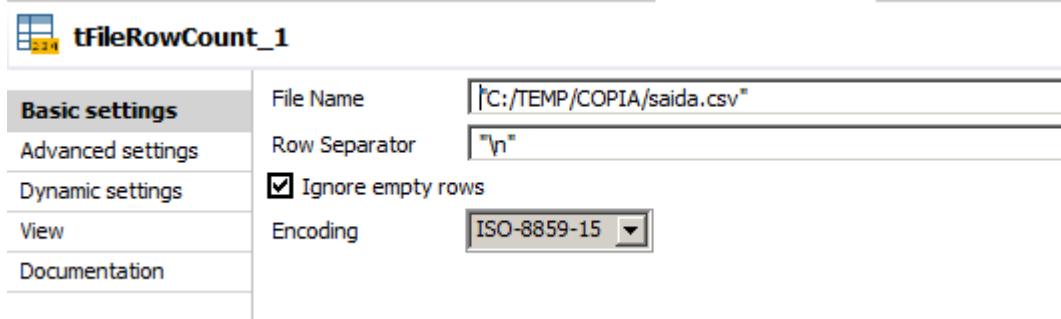


tFileRowCount

tFileRowCount: Componente criado para verificar a quantidade de linhas de um arquivo.
Vamos criar um Job chamado tFileRowCount.



Vamos ler o arquivo **saída.csv** e imprimir a quantidade de linhas na tela com o componente tJava;



Para imprimir a quantidade de linhas existentes, vamos utilizar o comando :

```
System.out.println( ((Integer)globalMap.get("tFileRowCount_1_COUNT"))
+ " Linha(s) encontrada!" );
```

The screenshot shows the configuration of a tJava_1 component. On the left, there is a sidebar with tabs: 'Basic settings' (selected), 'Advanced settings', and 'Dynamic settings'. The main area is titled 'Code' and contains the following Java code:

```
System.out.println(  
    ((Integer)globalMap.get("tFileRowCount_1_COUNT"))  
    + " Linha(s) encontrada!");
```

Ao final o job ficará da seguinte forma:



O resultado será:

The screenshot shows the execution log for the job 'tFileRowCount'. The log window displays the following output:

```
Starting job tFileRowCount at 20:56 31/05/2017.  
[statistics] connecting to socket on port 3688  
[statistics] connected  
1 Linha(s) encontrada!  
[statistics] disconnected  
Job tFileRowCount ended at 20:56 31/05/2017. [exit  
code=0]
```

Vamos trabalhar agora com componentes que executam todos os tipos de tarefas de processamento em **fluxos de dados**, incluindo **agregação**, mapeamento, **transformação**, **desnormalização**, **filtragem** e assim por diante.

Como base em nossas atividades vamos utilizar o arquivo **notas fiscais.xls**.

tAggregateRow, tSortRow e tLogRow

tAggregateRow: Componente criado para agregar dados em um arquivo ou tabela do banco de dados

Vamos criar um Job chamado tAggregateRow.

Antes, vamos ler o arquivo de vendas, utilizando o componente **venda_metadados**, que já possui os dados mapeados.

tFileInputExcel_1

Basic settings

- Property Type: Repository (EXCEL:venda_metadados)
- Read excel2007 file format(xlsx)
- File name/Stream: "D:/Arquivo Treinamento Aprenda Virtual/Notas Fiscais.xls"
- All sheets
- Sheet list: Sheet (name or position) "Nota Fiscal" Use Regex

Advanced settings

- Advanced separator (for numbers)
- Trim all columns

Column	Trim
id_venda	<input type="checkbox"/>
cod_cliente	<input type="checkbox"/>
nom_cliente	<input type="checkbox"/>
cod_vendedor	<input type="checkbox"/>
nom_vendedor	<input type="checkbox"/>
cod_cidade	<input type="checkbox"/>
nom_cidade	<input type="checkbox"/>

Vamos interligar ao componente tSortRow.

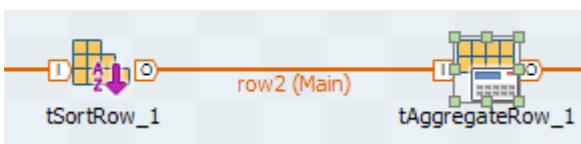


Clique no componente tSortRow, e escolha os campos **nom_gerencia** para ordenar, pois vamos sumarizar por este campo.

Você terá que modificar o tipo de dado do campo **qtd_venda**, pois faremos **qtd_venda(sum)**.

mes_venda	<input checked="" type="checkbox"/>	integer	<input checked="" type="checkbox"/>	2	0	mes_venda	<input checked="" type="checkbox"/>	integer
Dia_venda	<input checked="" type="checkbox"/>	Integer	<input checked="" type="checkbox"/>			Dia_venda	<input checked="" type="checkbox"/>	Integer
dtc_venda	<input checked="" type="checkbox"/>	String	<input checked="" type="checkbox"/>	10	0	dtc_venda	<input checked="" type="checkbox"/>	String
qtd_venda	<input checked="" type="checkbox"/>	Double	<input checked="" type="checkbox"/>	0		qtd_venda	<input checked="" type="checkbox"/>	Double
val_venda	<input checked="" type="checkbox"/>	String	<input checked="" type="checkbox"/>	2		var_venda	<input checked="" type="checkbox"/>	String

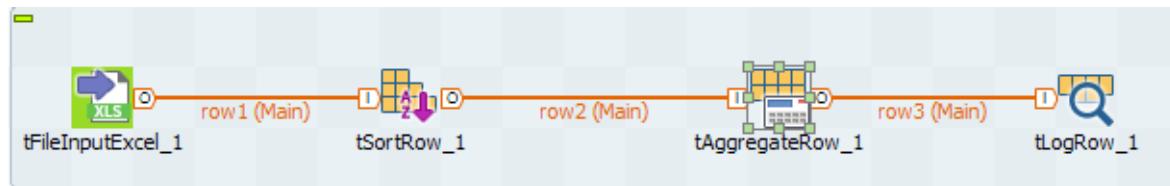
Agora utilize o componente **tAggregateRow** para summarizar as linhas por **nom_gerencia**, ligue **tSortRow** e **tAggregateRow**



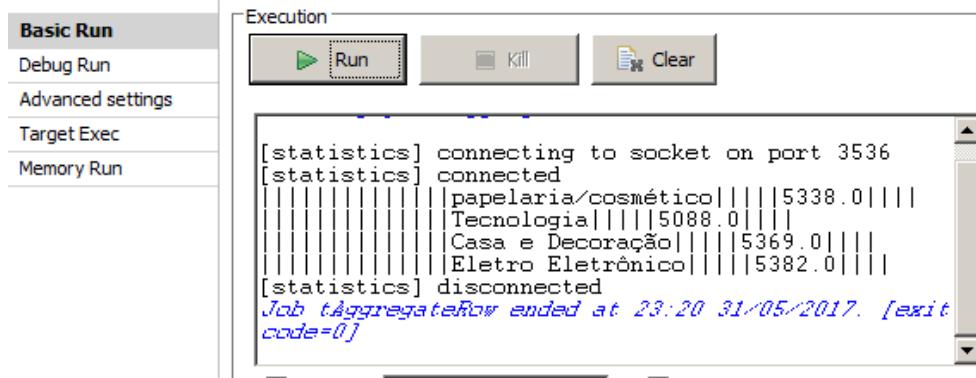
Vamos utilizar o componente **tLogRow** para exibir o resultado.



O resultado final, deve ficar da seguinte forma:



Execute o Job e terá o seguinte resultado:

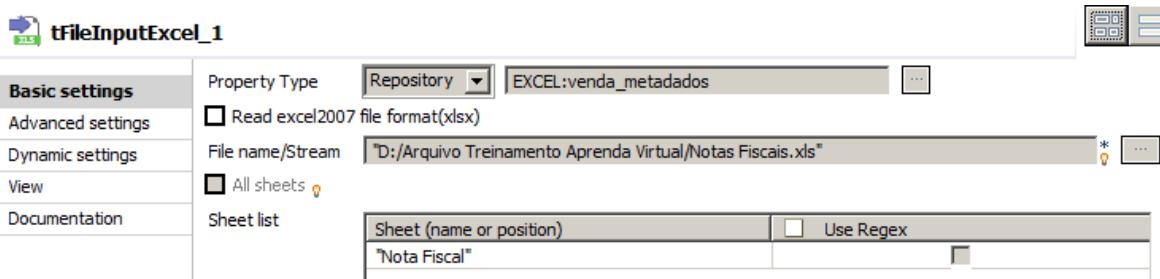


tReplace e tFilterColumn

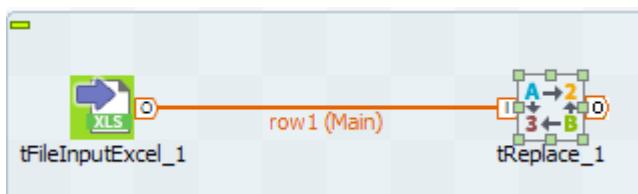
tReplace: Componente criado para executar uma operação de Pesquisa e Substituição nas colunas de um entrada definida.

Vamos criar um Job chamado tReplace.

Antes, vamos ler o arquivo de vendas, utilizando o componente **venda_metadados**, que já possui os dados mapeados.



Vamos interligar ao componente tReplace.



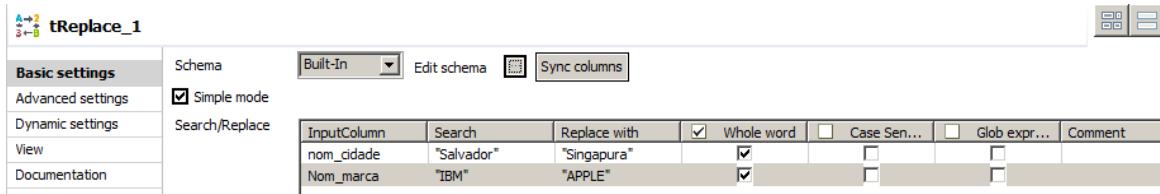
Ambos os componentes devem ter os campos:

Column	Key	Type	N...	Date Pa...	Le...	Pr...	D...	Co...
id_venda		Int...	✓		2	0		
cod_cliente		Str...	✓			0		
nom_cliente		Str...	✓		15	0		
cod_vende...		Str...	✓			0		
nom_vende...		Str...	✓		15	0		
cod_cidade		Long	✓			0		
nom_cidade		Str...	✓		11	0		
cod_produto		Str...	✓			0		
Nom_produto		Str...	✓		28	0		
cod_marca		Int...	✓		2	0		
Nom_marca		Str...	✓		13	0		
cod_depart...		Str...	✓			0		
nom_depart...		Str...	✓		15	0		
cod_gerencia		Int...	✓		1	0		
nom_gerencia		Str...	✓		19	0		

Column	Key	Type	✓	N...	Date Pa...	Le...	Pr...	D...	Co...
id_venda		Int...	✓			2	0		
cod_cliente		Str...	✓				0		
nom_cliente		Str...	✓			15	0		
cod_vende...		Str...	✓				0		
nom_vende...		Str...	✓			15	0		
cod_cidade		Long	✓				0		
nom_cidade		Str...	✓			11	0		
cod_produto		Str...	✓				0		
Nom_produto		Str...	✓			28	0		
cod_marca		Int...	✓			2	0		
Nom_marca		Str...	✓			13	0		
cod_depart...		Str...	✓				0		
nom_depart...		Str...	✓			15	0		
cod_gerencia		Int...	✓				1	0	
nom_gerencia		Str...	✓			19	0		

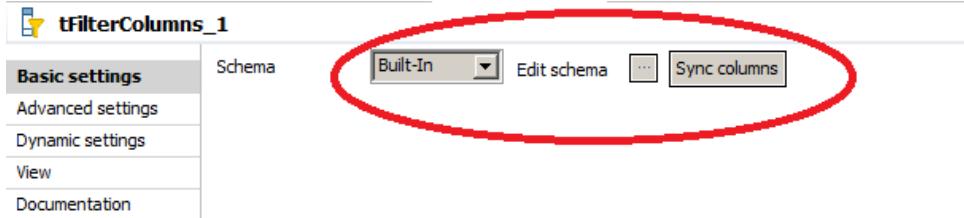
Clique no componente tReplace, e escolha os campos **nom_cidade** e **nom_marca**, vamos trocar os registros para as seguintes informações:

Cidade DE → Salvador PARA → Singapura
Marca DE → IBM PARA → APPLE



Vamos agora utilizar o componente **tFilterColumn** para mudar a ordem de apresentação dos resultados, ou seja, como desejamos que o arquivo de saída seja criado:

Clique em Editar Schema.



Altere de acordo com sua necessidade, no nosso caso, vamos colocar **Ano** e os campos de **Cidade** e **Marca** no inicio do arquivo, ele deve ficar no seguinte formato.

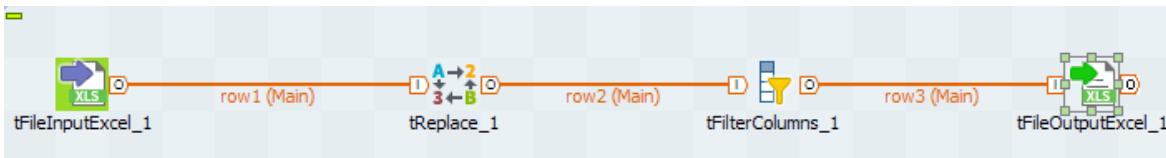
Column	Key	Type	N..	Date Pa...	Le...	Pr...	D...	Co...
id_venda		Int...	<input checked="" type="checkbox"/>		2	0		
cod_cliente		Str...	<input checked="" type="checkbox"/>			0		
nom_cliente		Str...	<input checked="" type="checkbox"/>		15	0		
cod_vende...		Str...	<input checked="" type="checkbox"/>			0		
nom_vende...		Str...	<input checked="" type="checkbox"/>		15	0		
cod_cidade		Long	<input checked="" type="checkbox"/>			0		
nom_cidade		Str...	<input checked="" type="checkbox"/>		11	0		
cod_produto		Str...	<input checked="" type="checkbox"/>			0		
Nom_produto		Str...	<input checked="" type="checkbox"/>		28	0		
cod_marca		Int...	<input checked="" type="checkbox"/>		2	0		
Nom_marca		Str...	<input checked="" type="checkbox"/>		13	0		
cod_depart...		Str...	<input checked="" type="checkbox"/>			0		
nom_depart...		Str...	<input checked="" type="checkbox"/>		15	0		
cod_gerencia		Int...	<input checked="" type="checkbox"/>		1	0		
nom_gerencia		Str...	<input checked="" type="checkbox"/>		19	0		

Column	Key	Type	N..	Date Pa...	Le...	Pr...	D...	Co...
Ano_venda		Int...	<input checked="" type="checkbox"/>		4	0		
cod_cidade		Long	<input checked="" type="checkbox"/>			0		
nom_cidade		Str...	<input checked="" type="checkbox"/>		11	0		
cod_marca		Int...	<input checked="" type="checkbox"/>			2	0	
Nom_marca		Str...	<input checked="" type="checkbox"/>		13	0		
cod_cliente		Str...	<input checked="" type="checkbox"/>			0		
nom_cliente		Str...	<input checked="" type="checkbox"/>		15	0		
cod_vende...		Str...	<input checked="" type="checkbox"/>			0		
nom_vende...		Str...	<input checked="" type="checkbox"/>		15	0		
cod_produto		Str...	<input checked="" type="checkbox"/>			0		
Nom_produto		Str...	<input checked="" type="checkbox"/>		28	0		
cod_depart...		Str...	<input checked="" type="checkbox"/>			0		
nom_depart...		Str...	<input checked="" type="checkbox"/>		15	0		
cod_gerencia		Int...	<input checked="" type="checkbox"/>		1	0		
nom_gerencia		Str...	<input checked="" type="checkbox"/>		19	0		

Utilize o componente **tOutputExcel** para saída dos dados

Basic settings	Property Type	Built-In
Advanced settings	<input type="checkbox"/> Write excel2007 file format(xlsx)	
Dynamic settings	<input type="checkbox"/> Use Output Stream	
View	File Name	[C:/TEMP/Replace.xls]
Documentation	Sheet name	[Sheet1]
	<input checked="" type="checkbox"/> Include header	The variable attached to this parameter is: <u>SHEETNAME</u>
	<input type="checkbox"/> Append existing file	
	<input type="checkbox"/> Is absolute Y pos.	
	Font	Default

O resultado final deve ficar da seguinte forma:



Execute o Job e terá o seguinte resultado:

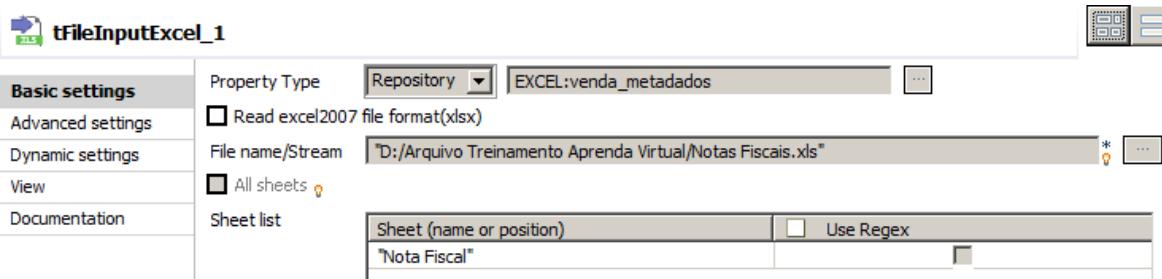
	A	B	C	D	E	F	G	H	I	J
1	Ano_venda	cod_cidade	nom_cidade	cod_marca	Nom_marca	cod_cliente	nom_cliente	cod_vende	nom_vende	cod_p
2	2015	3	Roma	4	APPLE	3	Renato Ru	3	Issac Palm	4
3	2014	1	Singapura	8	FABER CASTEL	4	Tom Jobim	3	Issac Palm	10
4	2014	3	Roma	8	FABER CASTEL	5	Edson Ara	5	Maria Braga	11
5	2014	2	Nova Iorque	13	BRASTEMP	5	Edson Ara	9	Diego Santos	19
6	2015	4	Bagdá	11	PHILIPS	6	Reginaldo	10	Danilo Bar	16
7	2014	3	Roma	8	FABER CASTEL	4	Tom Jobim	4	Daniela Ma	11
8	2015	1	Singapura	8	FABER CASTEL	3	Renato Ru	4	Daniela Ma	9
9	2014	5	Budapeste	13	BRASTEMP	2	Benito de	7	João Oliveira	19
10	2015	5	Budapeste	12	ELETROLUX	2	Benito de	9	Diego Santos	20
11	2014	1	Singapura	10	BOTICARIO	5	Edson Ara	9	Diego Santos	15
12	2014	4	Bagdá	11	PHILIPS	5	Edson Ara	3	Issac Palm	16
13	2014	3	Roma	12	ELETROLUX	5	Edson Ara	2	Márcio Bar	18
14	2015	3	Roma	8	FABER CASTEL	2	Benito de	4	Daniela Ma	9
15	2015	5	Budapeste	12	ELETROLUX	4	Tom Jobim	1	Claudio Sil	18
16	2014	4	Bagdá	5	SAMSUNG	2	Benito de	12	Marilia Car	8
17	2014	2	Nova Iorque	10	BOTICARIO	3	Renato Ru	8	Márcia Brito	15
18	2014	5	Budapeste	14	DA CASA	1	João do P	9	Diego Santos	21

tFilterRow

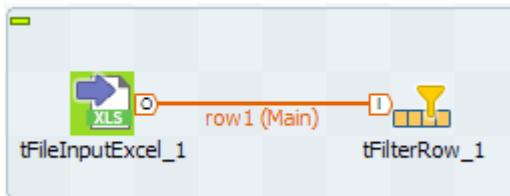
tFilterRow: Componente criado para filtrar registros dentro de um arquivo ou tabela de dados.

Vamos criar um Job chamado tFilterRow.

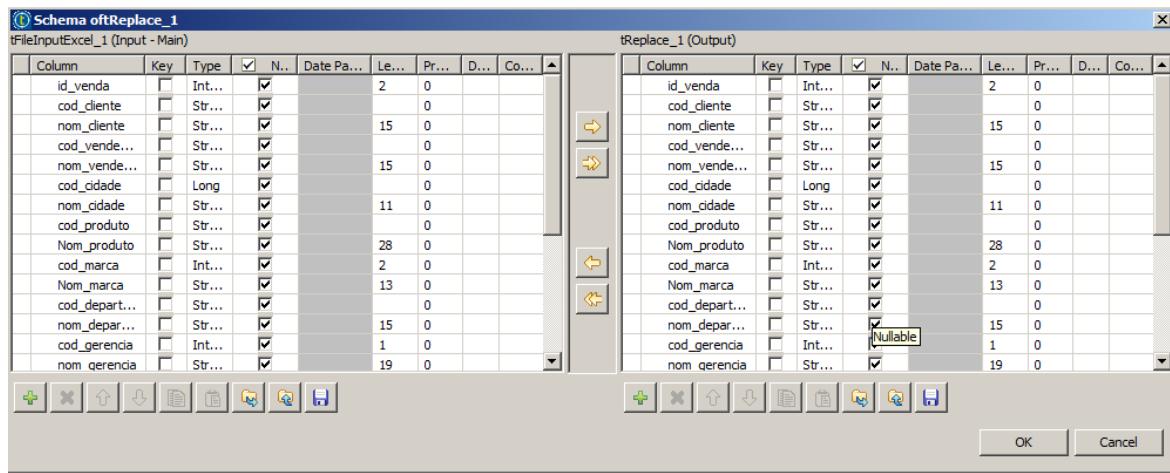
Antes, vamos ler o arquivo de vendas, utilizando o componente **venda_metadados**, que já possui os dados mapeados.



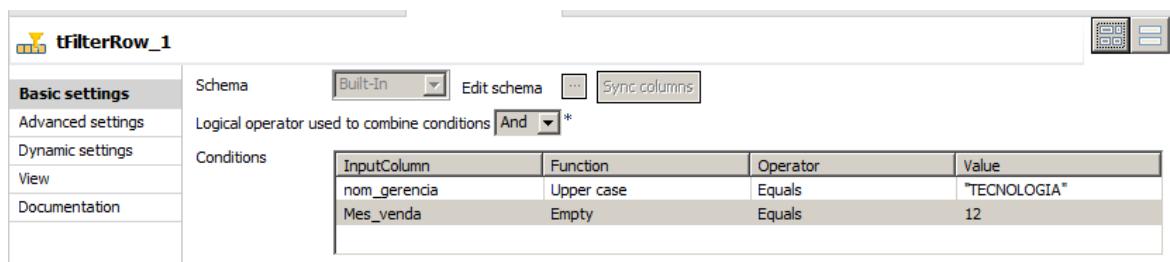
Vamos interligar ao componente tFilterRow.



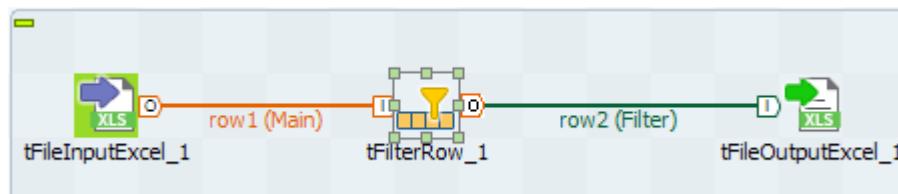
Ambos os componentes devem ter os campos:



No componente tFilterRow, coloque as condições:



O resultado final deve ficar da seguinte forma:



Execute o Job e terá o seguinte resultado:

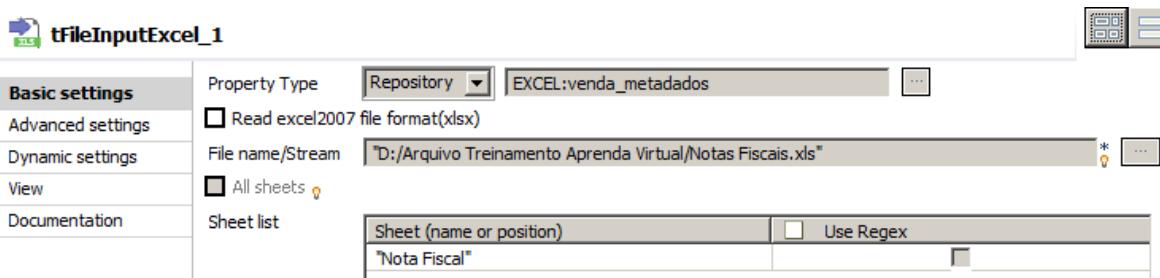
	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
142	Issac Paln	2 Nova Iorque	6	Telefone S	5 SAMSUNG	5	Eletrônico	1	Tecnologia	2014	9	09/12/2014	2			
143	Claudio Sil	5 Budapest	22	Mesa	14 DA CASA	3	Papelaria	1	Tecnologia	2015	14	14/12/2015	2			
144	Issac Paln	1 Salvador	1	TV	3 DELL	1	Informática	1	Tecnologia	2015	26	26/12/2015	3			
145	Issac Paln	4 Bagdá	22	Mesa	14 DA CASA	4	cosmético	1	Tecnologia	2014	8	08/12/2014	1			
146	Maria Brág	1 Salvador	9	Celular 80	8 FABER C	3	Papelaria	1	Tecnologia	2015	06/12/2015	1				
147	Maria Brág	5 Budapest	14	Classificad	9 AVON	1	Informática	1	Tecnologia	2014	26	26/12/2014	2			
148	Diego San	5 Budapest	6	Notebook	5 SAMSUNG	3	Papelaria	1	Tecnologia	2015	8	08/12/2015	2			
149	João Olivei	5 Budapest	2	Mouse	2 MICROSO	1	Informática	1	Tecnologia	2015	1	01/12/2015	2			
150	Filipe Alme	3 Roma	13	Estojo	7 TILIBRA	6	Eletrônico	1	Tecnologia	2014	11	11/12/2015	1			
151	Rodrigo M.	3 Roma	5	Computad	4 IBM	5	Eletrônico	1	Tecnologia	2015	13	13/12/2015	2			
152	João Olivei	2 Nova Iorque	7	Celular 4G	5 SAMSUNG	5	Eletrônico	1	Tecnologia	2014	26	26/12/2014	1			
153	Issac Paln	3 Roma	15	Baton	10 BOTICARI	4	cosmético	1	Tecnologia	2015	23	23/12/2015	1			
154	Filipe Alme	1 Salvador	9	Celular 80	8 FABER C	6	Eletrodom	1	Tecnologia	2014	18	18/12/2014	2			
155	Claudio Sil	4 Bagdá	19	Fogão	13 BRASTEM	6	Eletrodom	1	Tecnologia	2015	9	09/12/2015	2			
156	Rodrigo M.	5 Budapest	5	Computad	4 IBM	3	Papelaria	1	Tecnologia	2015	28	28/12/2015	1			

tSplitRow

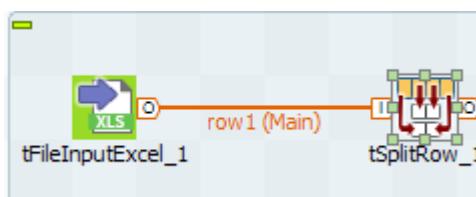
tSplitRow: Componente criado para transformar linhas em colunas dentro de um arquivo ou tabela de dados.

Vamos criar um Job chamado tSplitRow.

Antes, vamos ler o arquivo de vendas, utilizando o componente **venda_metadados**, que já possui os dados mapeados.



Vamos interligar ao componente tSplitRow.

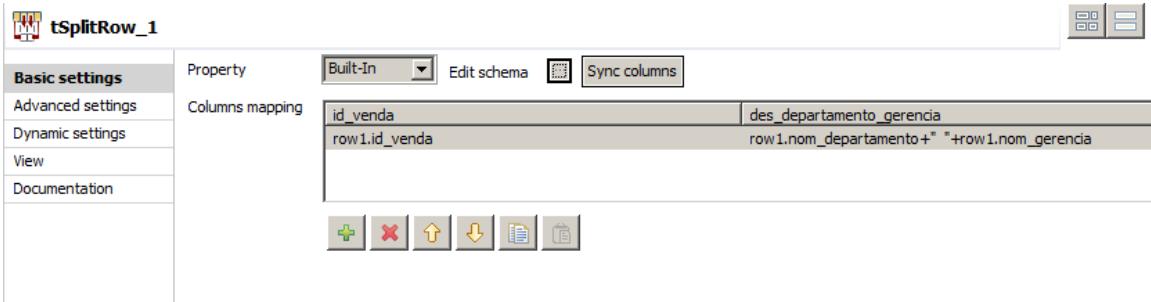


Os componentes devem ter os campos:

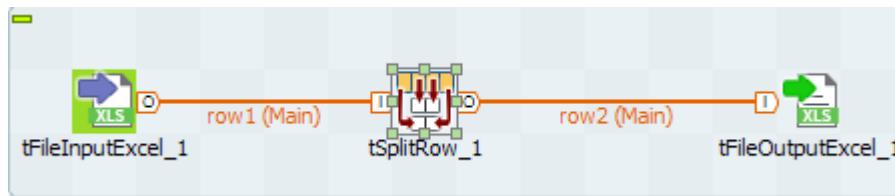
Column	Type	Key	N..	Date Pa...	Le...	Pr...	D...	Co...
id_venda	Int...		✓		2	0		
cod_cliente	Str...		✓			0		
nom_cliente	Str...		✓		15	0		
cod_vende...	Str...		✓			0		
nom_vende...	Str...		✓		15	0		
cod_cidade	Long		✓			0		
nom_cidade	Str...		✓		11	0		
cod_produto	Str...		✓			0		
Nom_produto	Str...		✓		28	0		
cod_marca	Int...		✓		2	0		
Nom_marca	Str...		✓		13	0		
cod_depart...	Str...		✓			0		
nom_depart...	Str...		✓		15	0		
cod_gerencia	Int...		✓		1	0		
nom_gerencia	Str...		✓		19	0		

Column	Type	Key	N..	Date Pat...	Len...	Pre...
id_venda	Int...		✓		2	0
des_departamento_gerencia	String		✓		100	

No componente tSplitRow, coloque as condições:



O resultado final deve ficar da seguinte forma:



Execute o Job e terá o seguinte resultado:

	A	B
1	id venda	des_departamento_gerencia
2	1	Informática Eletro Eletrônico
3	2	Informática Tecnologia
4	3	Informática Eletro Eletrônico
5	4	Informática Casa e Decoração
6	5	Eletrodoméstico Casa e Decoração
7	6	Papelaria Eletro Eletrônico
8	7	Eletrônico Casa e Decoração
9	8	cosmético Eletro Eletrônico
10	9	Papelaria papelaria/cosmético
11	10	Papelaria papelaria/cosmético
12	11	Eletrônico Eletro Eletrônico
13	12	Informática Casa e Decoração
14	13	Papelaria Tecnologia
15	14	Eletrônico Eletro Eletrônico
16	15	Telefonia Casa e Decoração

tFileInputXML

tFileInputXML: Componente criado para ler um arquivo XML e transformá-lo em um arquivo de dados por registros.

Vamos criar um Job chamado tFileInputXML.

Antes, vamos ler o arquivo **livraria.xml** ([baixe no site de aulas](#)). Vamos criar o metadados para leitura dos dados, de forma a facilitar a carga.

Crie o metadados do arquivo **livraria**.



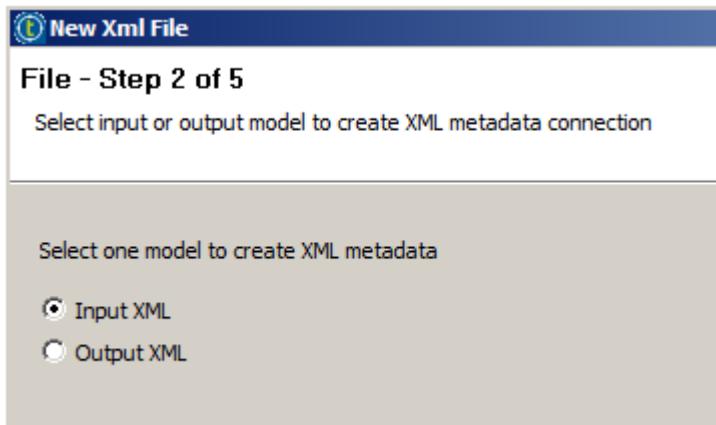
Coloque como metadados o descriptivo **livraria**.

A screenshot of the 'New Xml File' dialog box. It shows the following fields:

- Name: livraria
- Purpose: (empty)
- Description: (empty)
- Author: user@talend.com
- Version: 0.1
- Status: (empty)
- Path: Select

A warning message at the top left says: "⚠ It is inadvisable to leave the purpose blank." There is also a 'Select' button next to the Path field.

Em seguida, informe que o arquivo é de entrada de dados, ou seja, **INPUT**.

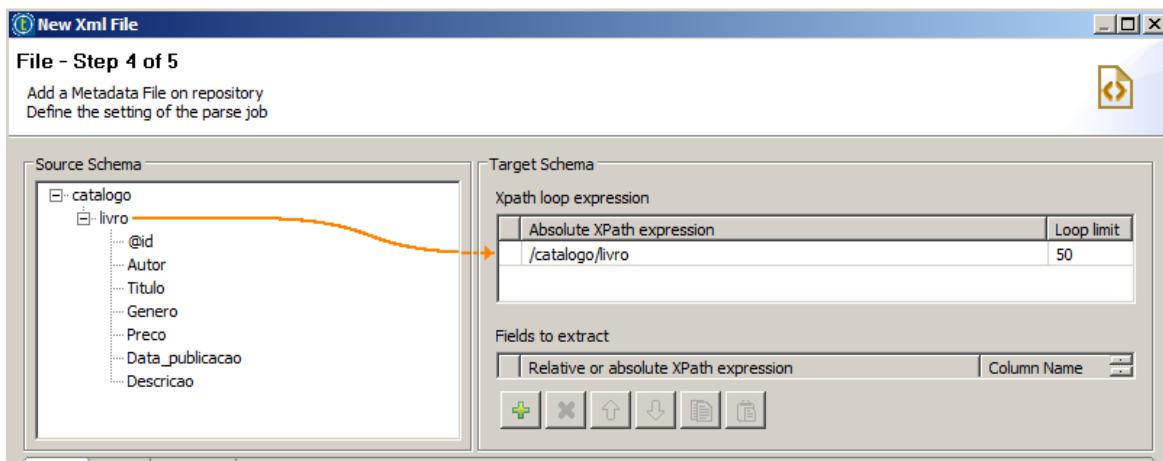


Pesquise o arquivo livraria.xml

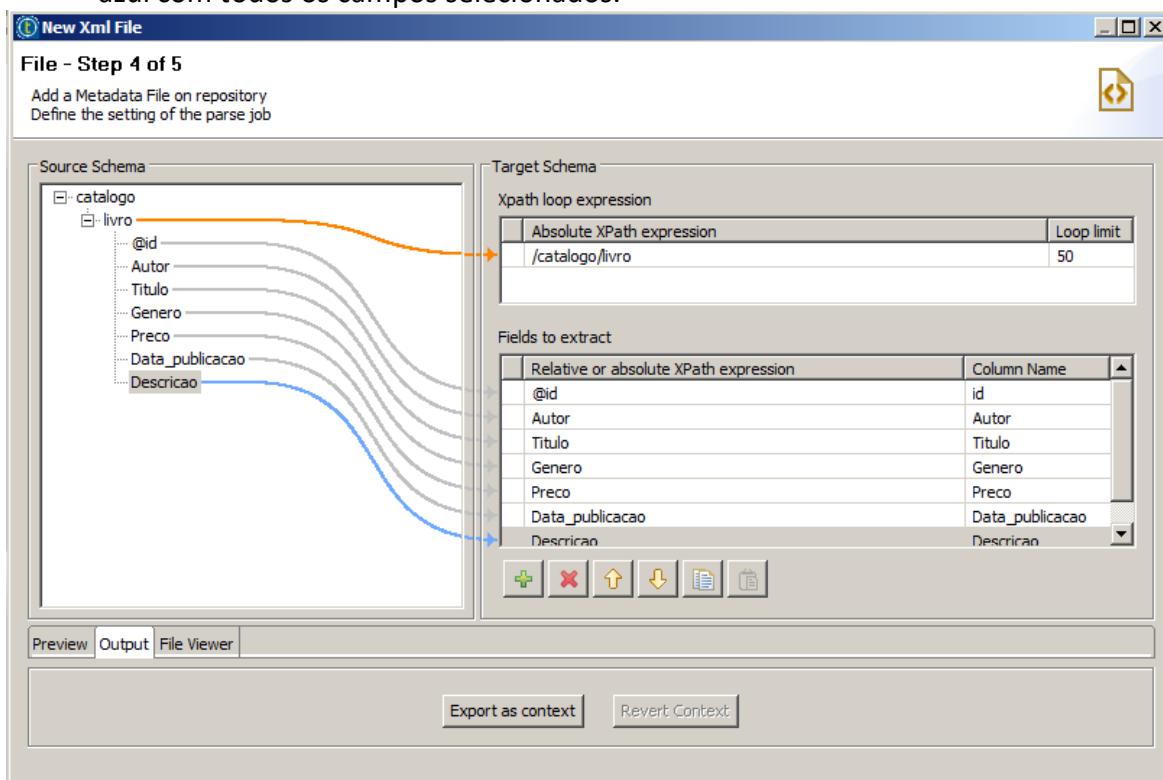
Note que os campos foram identificados. Agora vamos selecioná-los para a carga de dados.

Observe bem os passos para registrar no seu metadados.

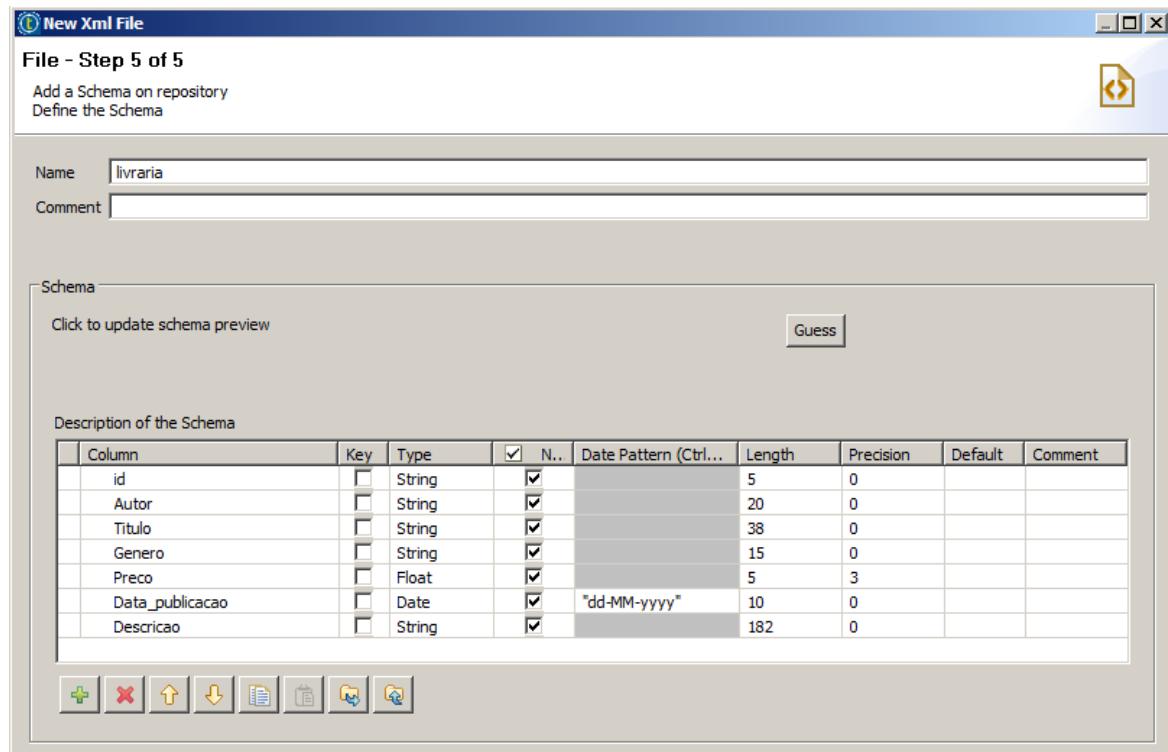
- 1- Arraste o campo livro para a sua direita, na aba **Target Schema**, note que ficará uma seta em laranja.



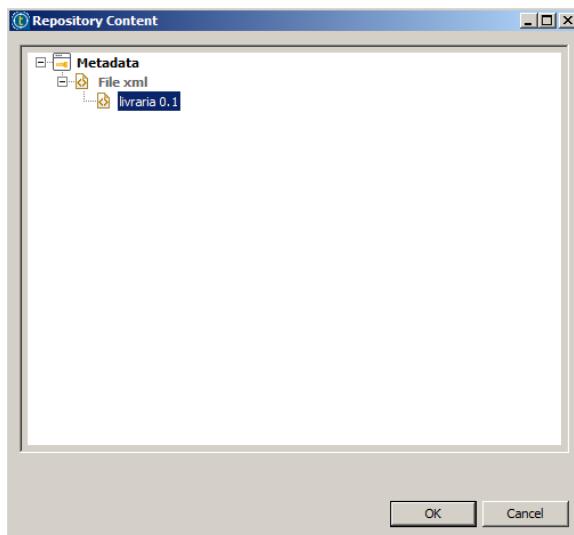
- 2- Arraste cada campo para sua direita na aba **Fields to extract**, ficará uma seta em azul com todos os campos selecionados.



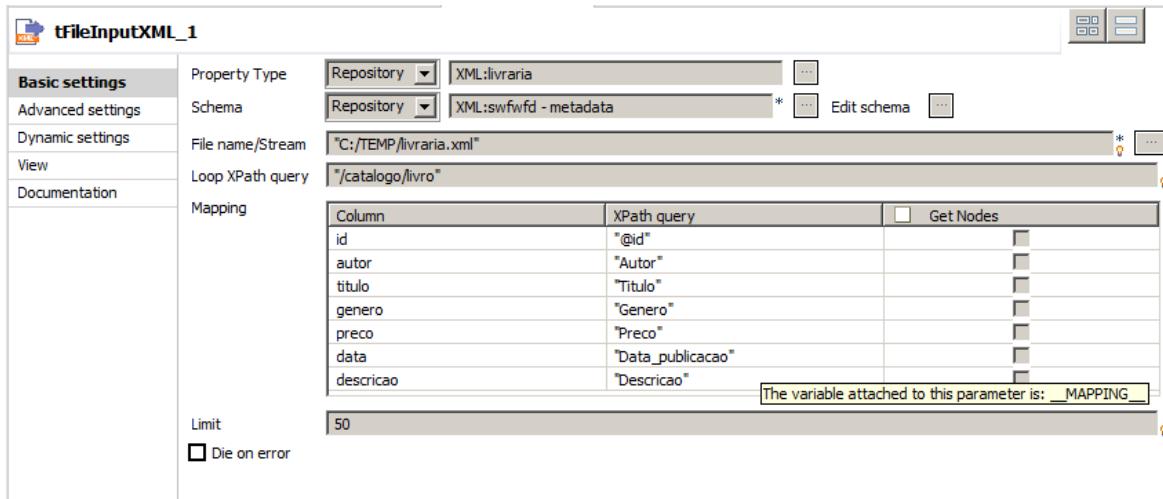
3- Coloque o nome do metadados como livraria e confirme.



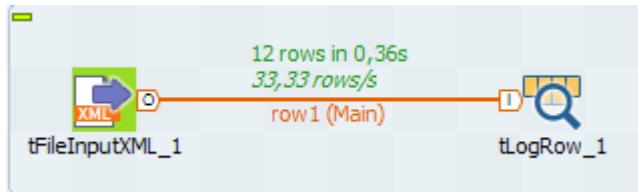
Vamos adicionar o componente tFileInputXML e tLogRow para entrada e saída dos dados. No componente tFileInputXML vamos clicar em **Property Type** para escolher o metadados que acabamos de criar, **livraria**.



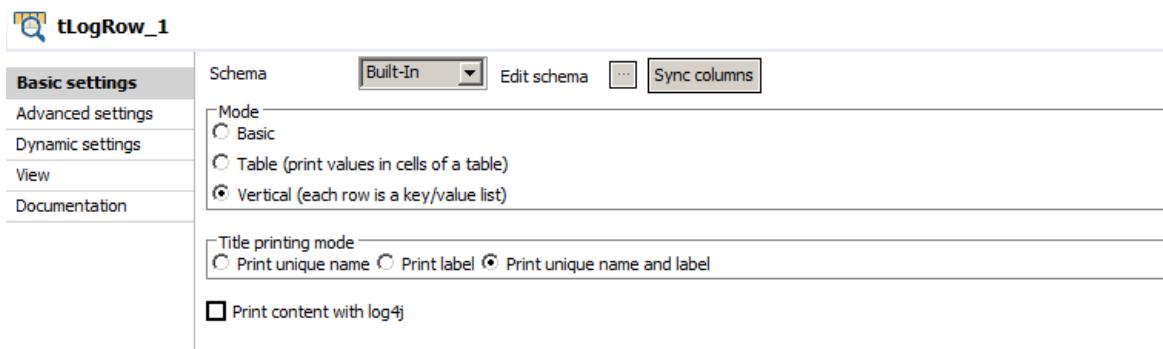
O componente vai estar formatado da seguinte maneira:



Pronto, una os componente tFileInputXML e tLogRow e execute.



Formate o componente tLogRow para que a saída fica mais fácil de entender.



Execute o Job e terá o seguinte resultado:

Job tFileInputXML

Basic Run

Run | Kill | Clear

key	value
id	bk101
Autor	Gambardella, Matthew
Titulo	XML Developer's Guide
Genero	Computer
Preco	44.95
Data_publicacao	23-03-0007
Descricao	An in-depth look at creating applications with XML.

Line limit Wrap

Vamos trabalhar agora com componentes que controlam logs e erros dos nossos Jobs.

tLogCatcher , tRowGenerator,tDie e tLogRow

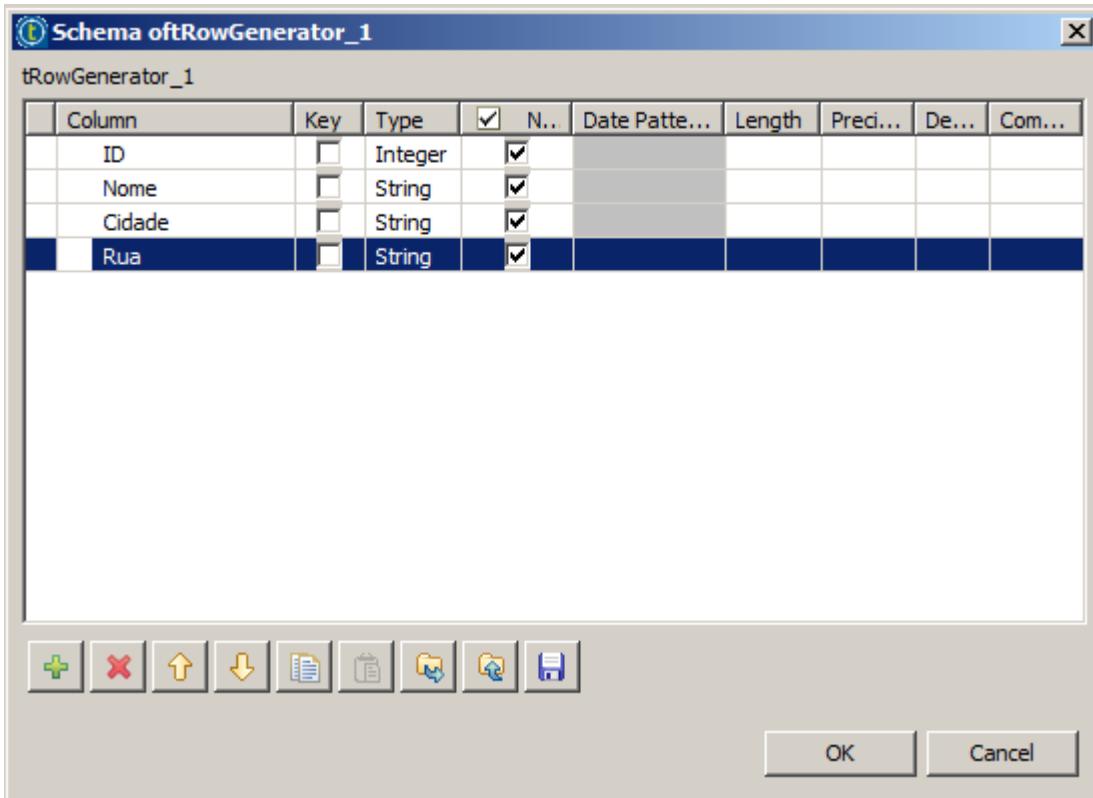
tLogCatcher: Componente criado para configurar os campos e as mensagens de Java Exception, tDie e / ou tWarn (faz sentido quando usado ao lado de um tLogCatcher para que os dados de registro coletados sejam encapsulados e transmitidos para a saída definida) e passa-os para o próximo componente.

tRowGenerator: Gera tantas linhas e campos conforme necessário e alimenta cada campo com um valor aleatório.

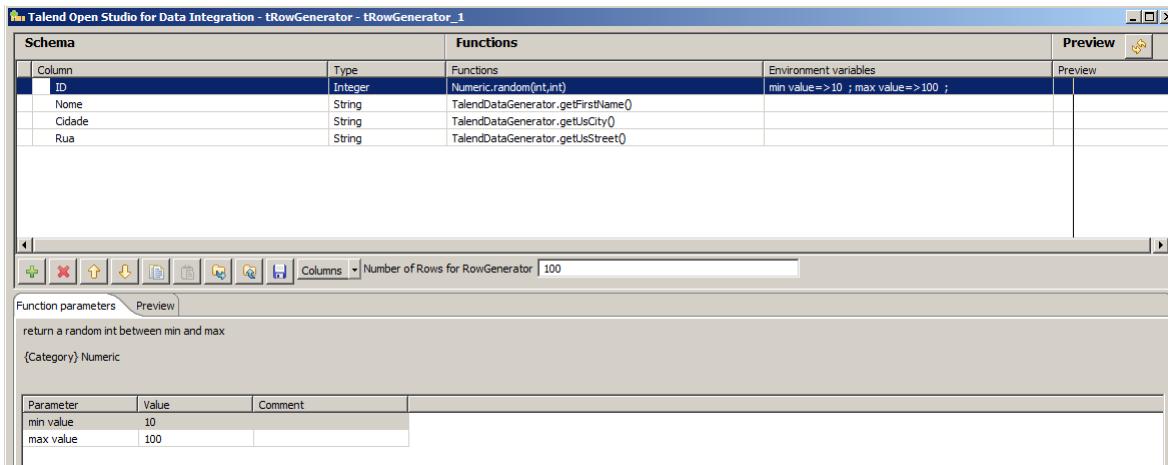
Vamos criar um Job chamado tLogCatcher.

Vamos utilizar o componente tRowGeneration para gerar dados fictícios.

Primeiramente, vamos criar alguns campos no schema para geração dos dados, veja abaixo, abra o componente e em **Edit Schema** crie os campos:



Próximo passo é clicar em **RowGenerator Editor** e acrescente as funções para gerar as informações:

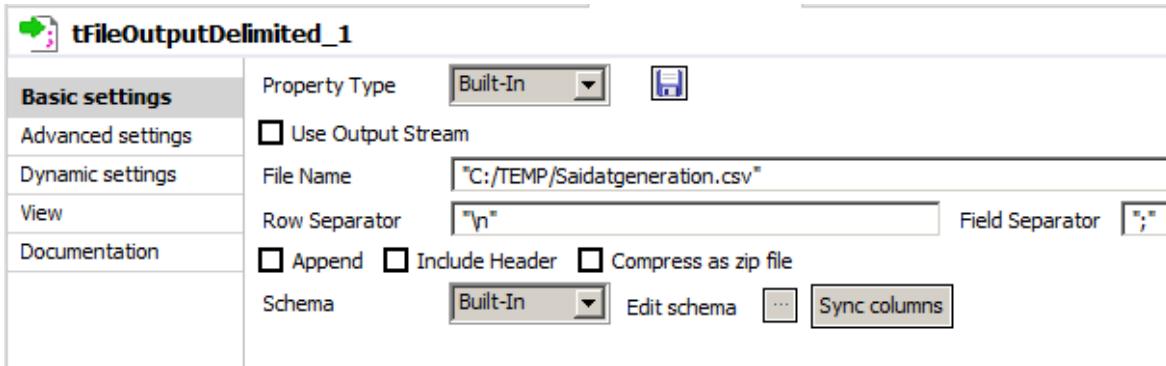


Colocamos um intervalo variando de 10 a 100 para o campo ID. Agora clique em **Preview** e os dados serão exibidos.

	ID	Nome	Cidade	Rua
1	49	Thomas	Oklahoma City	Erringer Road
2	28	Franklin	Little Rock	Hutchinson Rd
3	91	Harry	Des Moines	Lindbergh Blvd
4	68	Ulysses	Austin	Padre Boulevard
5	24	Zachary	Atlanta	S Rustle St
6	12	Gerald	Nashville	Pacific Hwy S
7	25	John	Springfield	Pacific Hwy S
8	95	Martin	Montgomery	N Harrison St
9	66	Herbert	Salem	Santa Rosa North
10	12	Bill	Charleston	N Harrison St

Vamos gravar os dados em um arquivo texto. Traga o componente **tFileOutputDelimited**. Para gravar os dados, vamos executar um teste após a gravação e forçar um erro.





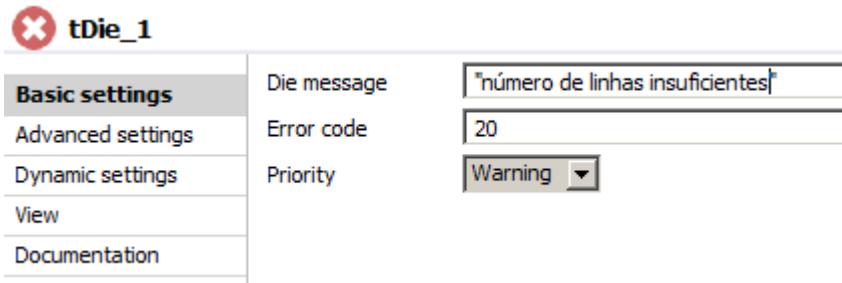
Traga o componente tDie e conecte o tFileOutputDelimited ao tDie, utilizando a trigger>IF, veja:

Acrescente a cláusula IF o comando:

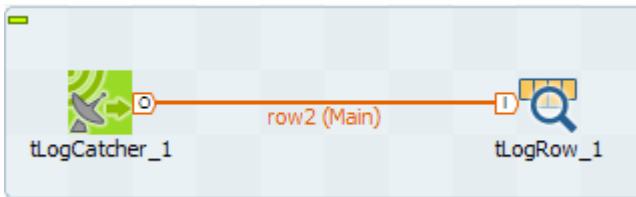
```
((Integer)globalMap.get("tRowGenerator_1_NB_LINE")) <=100
```

Será provocado um erro que será enviado ao tLogCartcher, estamos forçando o erro se a quantidade de linhas for menor que 100.

Para definirmos o erro, dê um duplo clique no componente tDie e acrescente a informação abaixo:



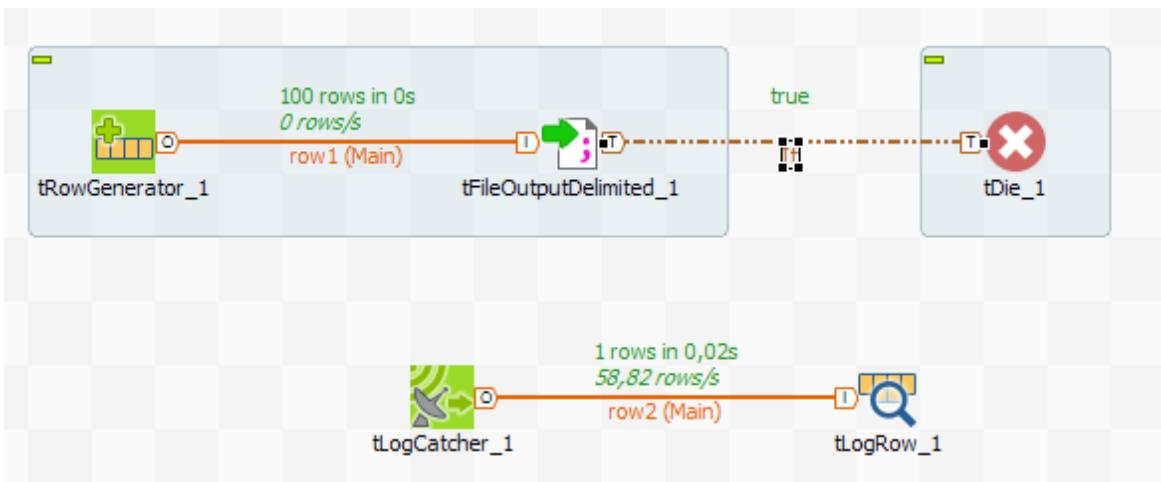
Acrescente os componentes tLogCatcher e TlogRow para receber a mensagem de erro (tLogCartcher) e para exibi-la (tLogRow).



Ao executar teremos o seguinte resultado:

```
Starting job tLogCatcher at 20:56 06/06/2017.  
[statistics] connecting to socket on port 4018  
[statistics] connected  
número de linhas insuficientes  
2017-06-06  
20:56:20|kVJFCi|kVJFCi|kVJFCi|TREINAMENTO_APRENDA_VIRTUAL|tL  
ogCatcher|Default|4|tDie|tDie_1|número de linhas  
insuficientes|20  
[statistics] disconnected  
Job tLogCatcher ended at 20:56 06/06/2017. [exit code=20]
```

Ao final todo o JOB terá a seguinte forma:



tSendMail

tSendMail: Componente criado para enviar email e caso seja necessário com arquivos em anexo.

Vamos criar um Job chamado tSendMail.

Vamos interligar ao componente tSendMail.



Vamos configurar o componente tSendMail.

The screenshot shows the configuration dialog for the 'tSendMail_1' component. The left sidebar has tabs for 'Basic settings', 'Advanced settings', 'Dynamic settings', 'View', and 'Documentation'. The 'Basic settings' tab is selected, showing the following fields:

- To: "grimaldol.lopes@gmail.com"*
- From: "grimaldo_lopes@hotmail.com"*
- Show sender's name?
- Cc: [empty]
- Bcc: [empty]
- Subject: "Arquivo de Dados sendo enviado"*
- Message: "Segue o arquivo de dados em anexo"

The 'Advanced settings' tab is expanded, showing:

- SMTP host: "smtp.gmail.com"*
- SMTP port: 587
- SSL Support
- STARTTLS Support
- Importance: High
- Need authentication?
- Username: "grimaldol.lopes"*
- Password: [redacted]*
- Die on error

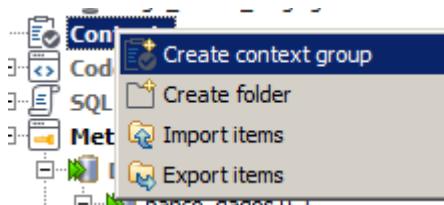
Variável de Contexto

Existe no TALEND a chamada variável de contexto, descreve os parâmetros definidos pelo usuário que são passados para seu job a ser executado. Variáveis de contexto podem armazenar diretórios, configuração de banco de dados, enfim, tipos de dados como string, integer, char, etc.

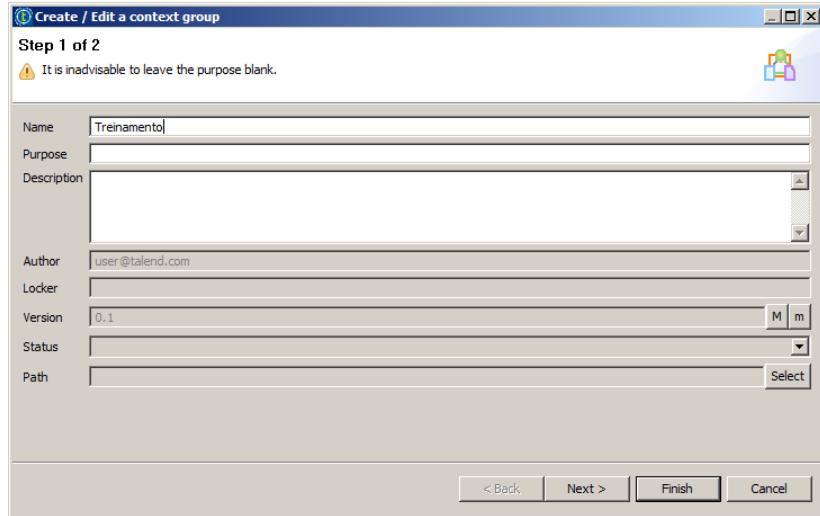
Como exemplificação do seu uso, utilizaremos uma variável de contexto que passa o nome de um diretório e o nome do arquivo que serão utilizados para serem copiados, por meio do componente **tFileDialog** e **tFileCopy**.

Primeira ação a fazer é criar as variáveis de contexto:

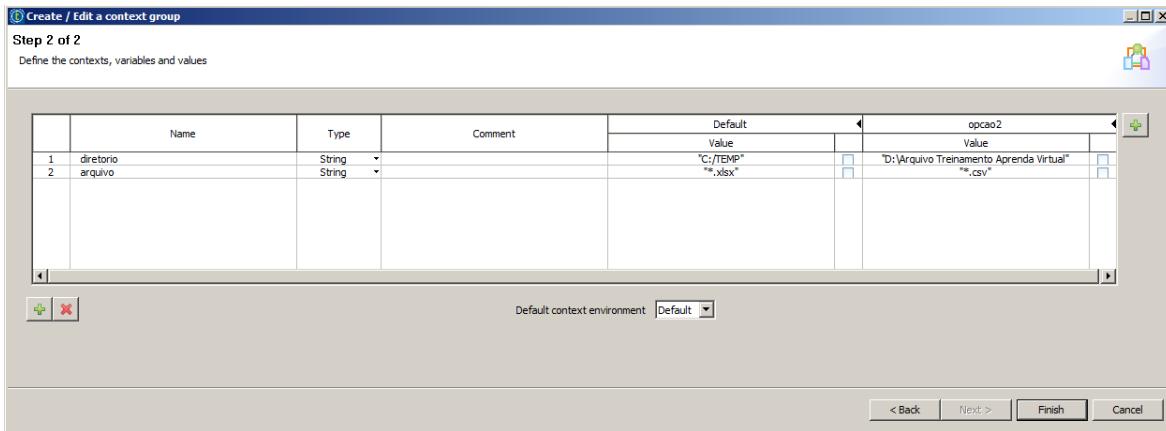
- 1- Vamos criar o contexto , clique com o botão direito em **Contexts** na aba lateral de repositório de trabalho.



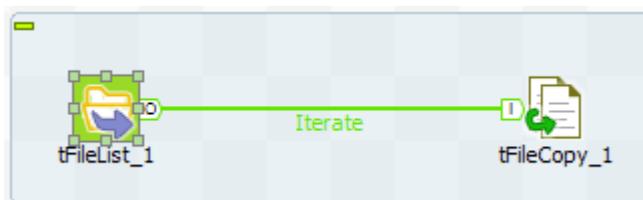
- 2- Em seguida, selecione **Create context group**, defina um nome ao seu grupo de contexto. Sugiro sempre usar um nome parecido ao do projeto. Colocaremos o nome de **Treinamento**.



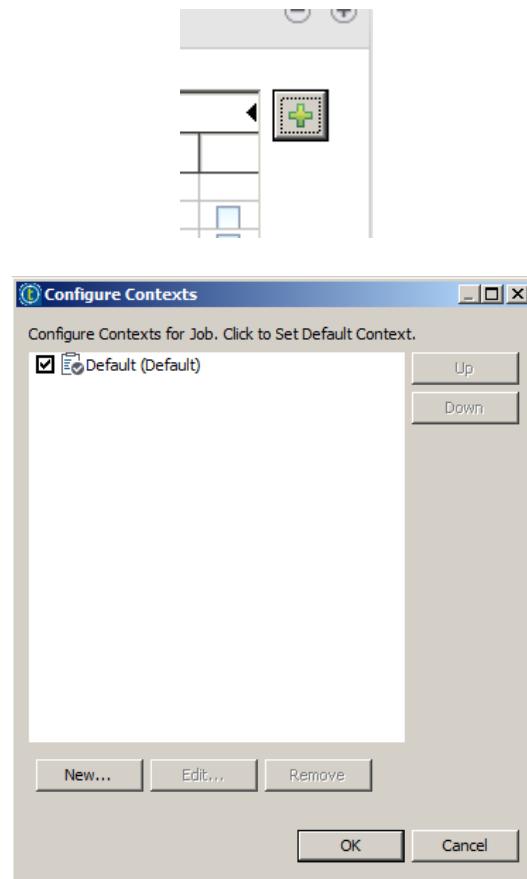
- 3- Acrescente as variáveis: **diretorio** e **arquivo**, ambas string. Você verá que pode criar varios cenários com estas mesmas variáveis.



- 4- Agora vamos copiar o JOB tFileCopy já criado por nós, deverá ficar assim:



- 5- Para acrescentar as variáveis de contexto, vamos clicr na aba **Context** ao lado de **component tFileCopy**. Clique no botão + do lado direito para incluir os campos **DEFAULT**.



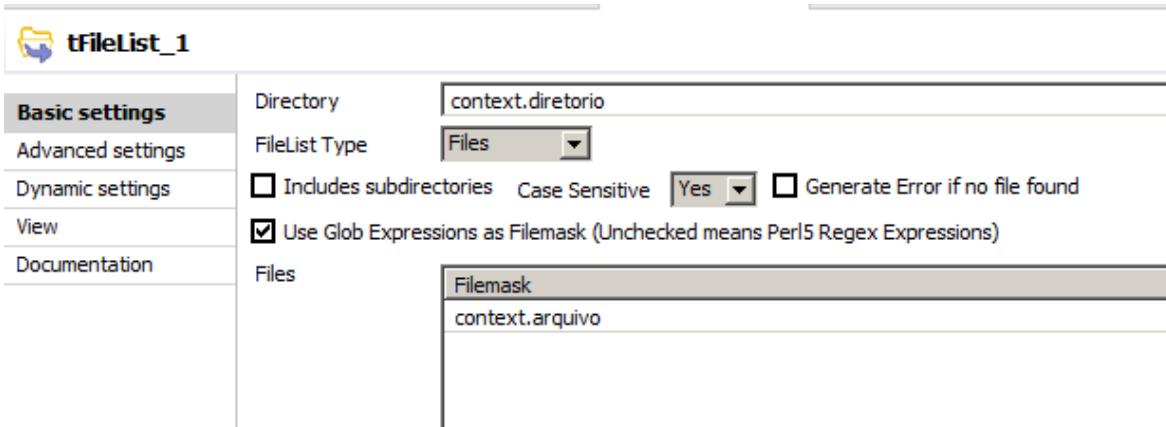
6- Pronto, as variáveis foram acrescentadas:

	Name	Type	Comment	Default	
				Value	
1	Treinamento (from repository context)				
2	arquivo	String		"*.xlsx"	
3	diretorio	String		"C:/TEMP"	

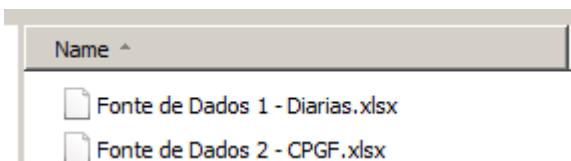
Default context environment

Buttons at the bottom: +, -, ↑, ↓, ↻.

7- Para utilizá-las, basta substituir no componente **Tlist**, pelo campos correspondentes:



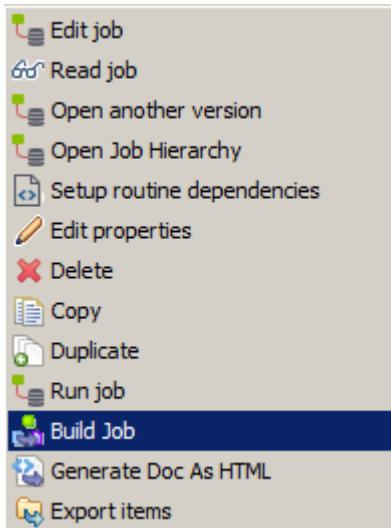
Ao executar, os arquivos serão copiados:



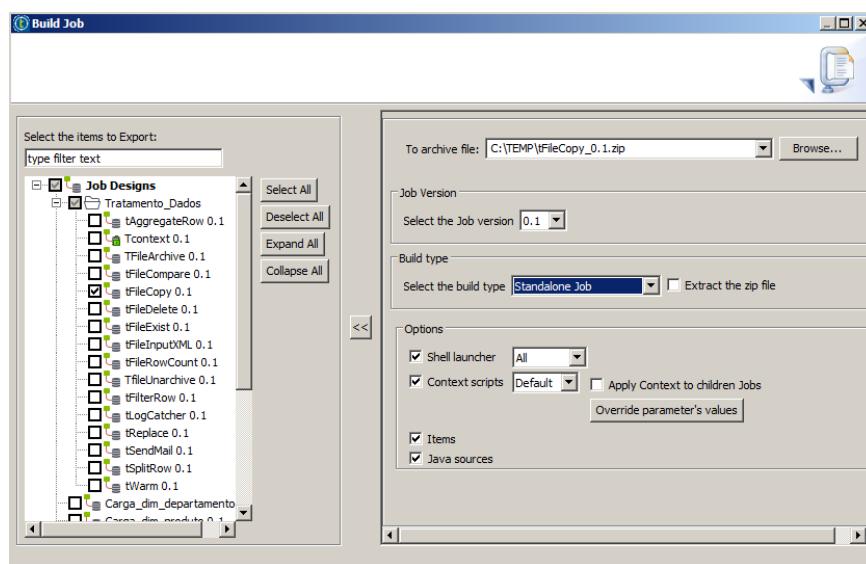
Exportar JOB

Podemos Exportar um JOB em específico para que este seja executado em qualquer máquina, geralmente sendo encapsulado em um arquivo .zip, para isso, basta clicar no JOB e escolher a opção **Build JOB**.

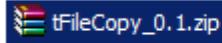
1- Vamos exportar o Job **tFileCopy**.



2- Abrirá uma janela, para a escolha do JOB e a geração em um arquivo .zip



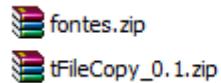
3-Agora vamos ao diretório e descompactar o arquivo, haverá o chamamento do job nos arquivos **.bat (windows)** ou **.sh (linux)**



Descompactando



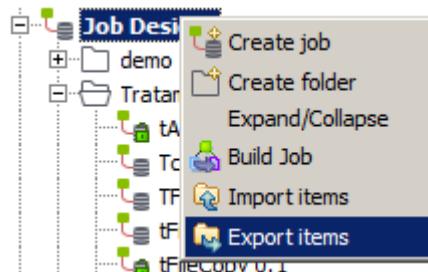
Pronto, os arquivos estão gerados, vamos executar o **tFileCopy.bat**, irá copiar os arquivos **.zip** ou **.csv** para o diretório copia.



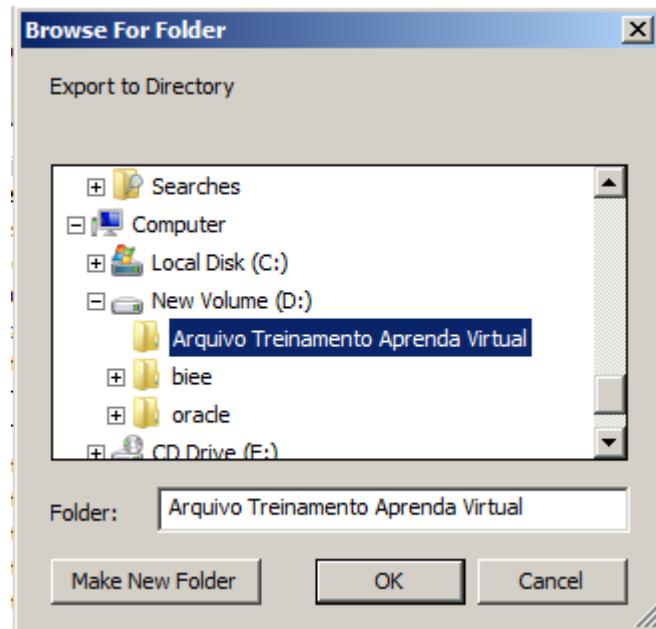
Exportar Projeto

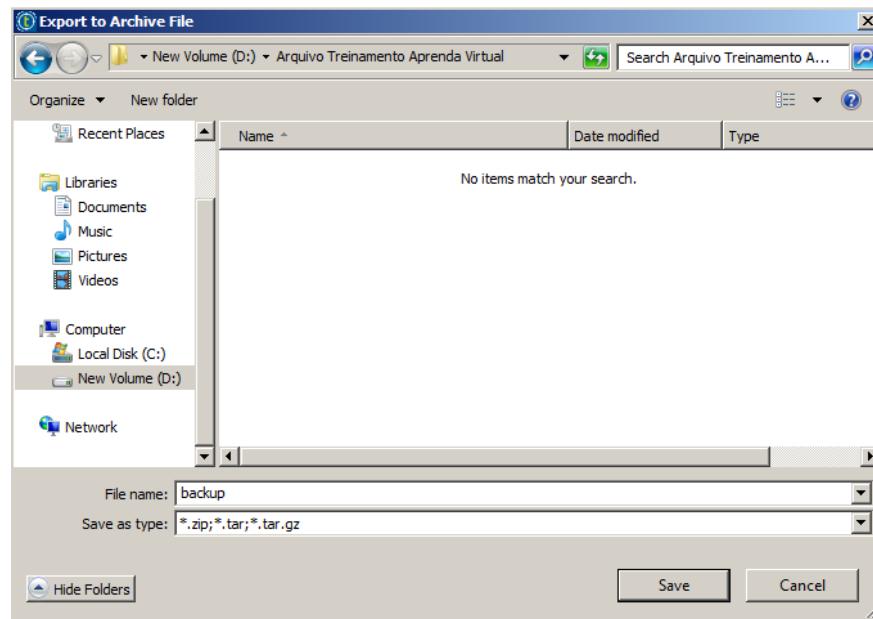
Podemos Exportar o projeto como um todo, ou seja, realizar um backup.

- 1- Clique em Job Design e escolha a opção **Export Items**.

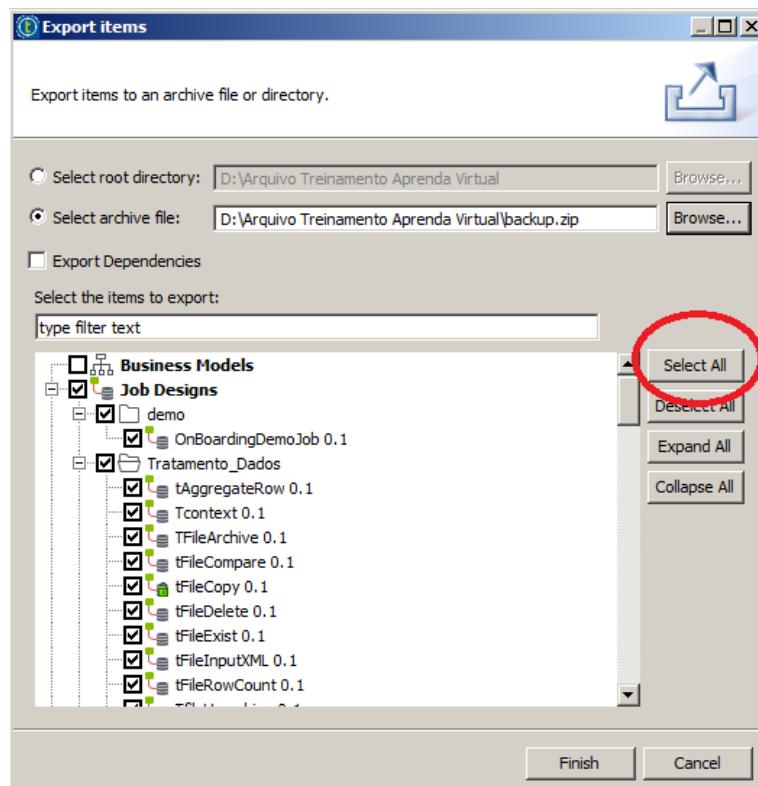


- 2- Escolha o diretório e o nome do arquivo.





3- Selecione todos os componentes para o backup e clique em **finish**.



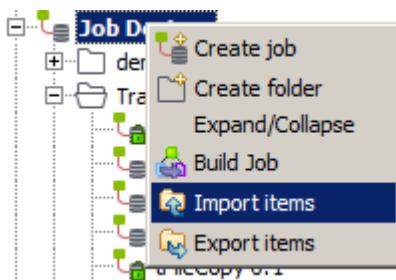
4- Pronto, o backup foi salvo.

Name	Date modified	Type
 backup.zip	17/06/2017 09:25	Arquivo ZIP do Win...

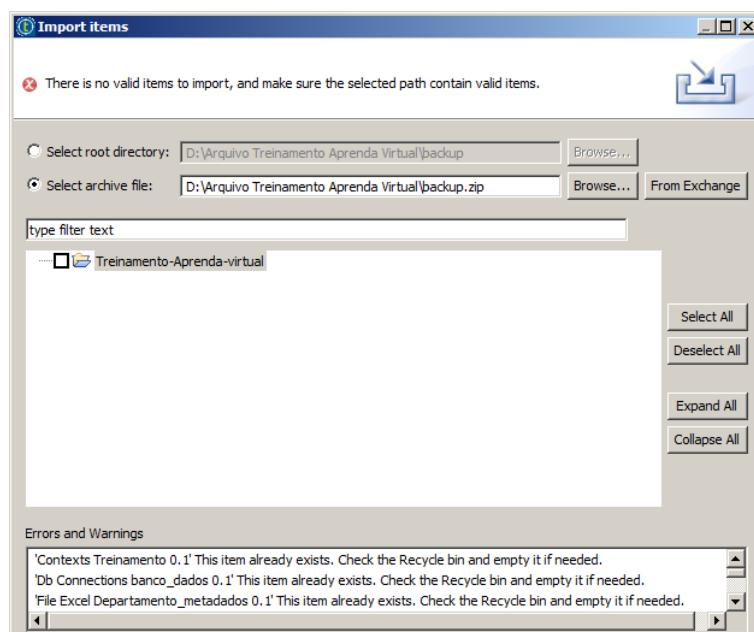
Importar Projeto

Podemos Importar um projeto como um todo, a partir de um backup realizado.

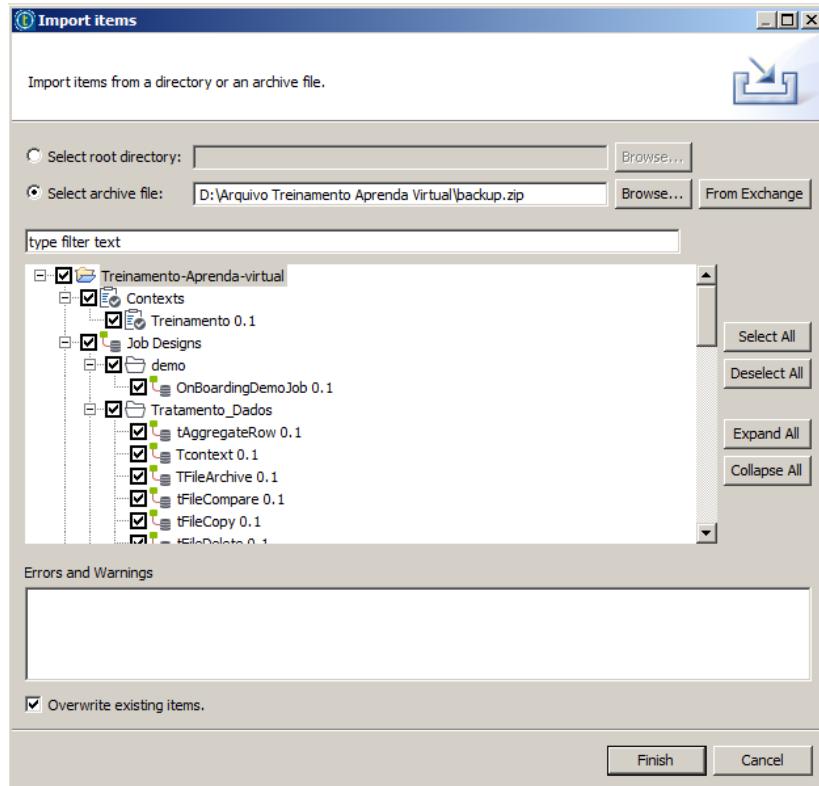
- 1- Descompacte o Backup do projeto salvo.
- 2- Clique em Job Design e importar itens.



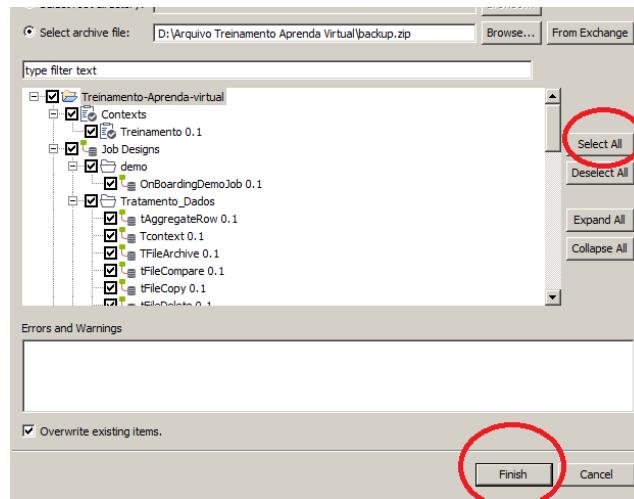
- 3- Escolha o Arquivo, no formato ZIP. Caso o projeto já exista, ele alertará sobre isso, você pode ignorar clicando na opção **Overwrite Existing Items**.



APÓS IGNORAR CLICANDO EM OVERWRITE EXISTING ITEMS



4- Selecione todos os componentes para o backup e clique em **finish**.



8. Exercícios Finais para Emissão do Certificado de Participação

Agora caro aluno, caso esteja apto, com todas as dúvidas retiradas e todas as vídeoaulas assistidas, inicie sua bateria de exercícios para que tenha direito ao certificado de participação.

O aluno deve realizar os exercícios e caso tenha dúvidas deve consultar o professor, mas o aluno deve tentar realizar **TODOS OS EXERCÍCIOS**, estejam corretos ou não.

O professor irá avaliar o seu grau de entendimento. Caso o professor não esteja satisfeito com o seu rendimento, este solicitará que você refaça os exercícios ou revise algumas aulas.

O professor tem total autonomia para ajudar e gerenciar as atividades dos alunos e decidir quando o aluno terá direito ao certificado de participação.

8.1 Primeira Bateria de Exercícios

De acordo com o texto abaixo, reproduza em seu computador e envie ao professor.

- O aluno deverá construir um projeto de DW do zero utilizando a ferramenta Talend Data Integration
- O projeto deve contemplar: tabelas Staging, tabelas Dimensão, tabela Fato, cargas Staging, cargas Dimensão e carga Fato;
- O aluno deve executar as cargas e popular as tabelas de Dimensão e Fato;
- O aluno deverá enviar o script de criação do banco de dados multidimensional do projeto ao professor e toda a estrutura de metadados do projeto através da exportação do projeto;
- O aluno utilizará os dados de uma Folha de Pagamento para realizar o projeto, o arquivo encontrasse na área do aluno no site do curso.

AO FINAL DEVEM SER CRIADOS:

DIMENSÃO: Cargo, Departamento, Divisão, Funcionário,Tempo

FATO: Folha de Pagamento

- Os arquivos de entrada que serviram para carga nas tabelas de Staging estão no formato texto (CSV).

8.2 Segunda Bateria de Exercícios

- 1) Carregue o arquivo **Fonte de Dados 1 - Diarias.xlsx** ou **Fonte de Dados 2 - CPGF**
- 2) Utilize pelo menos **4 componentes** vistos no curso para realizar tratamento de dados, ou qualidade dos dados. Explique qual a finalidade do seu JOB.

Considerações Importantes

“A alegria que se tem em pensar e aprender faz-nos pensar e aprender ainda mais”

Aristóteles (Início dos tempos)

9. Considerações Importantes

- Importante que você saiba que você pode retirar dúvidas com o professor no momento que desejar, para isso entre em contato via e-mail, o mesmo está na folha de rosto da apostila ou através do fórum dos alunos;
- Lembre-se que uma internet de banda larga ajudará na visualização dos vídeos, quando temos lentidão no acesso da internet isso influenciará na aprendizagem rápida do curso;
- Importante que você trabalhe com a ferramenta Talend Data Integration diariamente, ou um espaço de tempo de uma aula para outro pequeno, pois isso facilitará seu entendimento;
- Lembre-se o curso tem um custo baixo, para permitir que mais colegas possam realizar o curso e retire dúvidas com os professores e colegas, não compartilhe seu usuário e senha, pois prejudica uma cadeia de profissionais que trabalharam no curso.
- Qualquer dificuldade não hesite e entre em contato com o professor do curso, passe um e-mail.

10. Futuras atualizações

Toda necessidade de inclusão de novos comandos e exercícios, devem ser solicitados ao professor, lembre-se você pode melhorar e muito o curso informando problemas no acesso e sobre algum erro encontrado e identificado na apostila.

Apêndice

Glossário de Siglas e Termos

“Mesmo desacreditado e ignorado por todos, não posso desistir, pois para mim, vencer é nunca desistir..”

Albert Einstein (1879-1955)

I. Glossário de Siglas e Termos

A seguir estão disponíveis em ordem alfabética, a relação de siglas e termos frequentemente utilizados durante a criação do curso.

- A -

ALUNO – É o indivíduo que recebe formação e instrução de um ou vários professores ou mestres para adquirir ou ampliar seus conhecimentos.

projetos complexos e contínuos de sincronização de dados com milhares de trabalhos. Um repositório compartilhado com versões e capacidades de entrega contínua maximiza a produtividade e a colaboração.

- E -

EAD – É uma modalidade de educação mediada por tecnologias em que alunos e professores estão separados espacial e/ou temporalmente, ou seja, não estão fisicamente presentes em um ambiente presencial de ensino-aprendizagem.

- P -

PROFESSOR - É uma pessoa que ensina uma ciência, arte, técnica ou outro conhecimento.

PROGRAMAÇÃO - É um método padronizado para comunicar instruções para um computador. É um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador.

- T -

TALEND DATA INTEGRATION - Ferramenta e assistentes simples e gráficos, levam-no a funcionar rapidamente com mais de 900 conectores para conectar conectores nativos, aplicativos baseados em nuvem e muito mais. Gerencie e monitore facilmente projetos de software ETL simples e únicos, bem como