UNIVERSIDADE FEDERAL DA PARAÍBA

CENTRO DE CIÊNCIAS APLICADAS A EDUCAÇÃO

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

ANÁLISE COMPARATIVA DAS FERRAMENTAS DE ETL - KETTLE E TALEND

HERMANNY ALEXANDRE DOS SANTOS LIRA FILHO

Orientadora: Profª. Msc. Ana Liz Souto Oliveira de Araújo

Co-orientadora: Profª. Msc. Renata Fernandes Viegas

RIO TINTO – PB

2013

HERMANNY ALEXANDRE DOS SANTOS LIRA FILHO

ANÁLISE COMPARATIVA DAS FERRAMENTAS DE ETL - KETTLE E TALEND

Monografia apresentada para obtenção do título de Bacharel à banca examinadora no Curso de Bacharelado em Sistemas de Informação do Centro de Ciências Aplicadas e Educação (CCAE), Campus IV da Universidade Federal da Paraíba.

Orientadora: Profª. M.Sc Ana Liz S. O. de Araújo

Co-orientadora: Profª. M.Sc Renata F. Viegas

RIO TINTO – PB
2013

L768a Lira Filho, Hermanny Alexandre dos Santos.
Análise comparativa das Ferramentas de *ETL – Kettle* e *Talend* / Hermanny Alexandre dos Santos Lira Filho. – Rio Tinto: [s.n.], 2013.
79f.: il. –
Orientadora: Ana Liz S. O. de Araújo.
Coorientadora: Renata F. Viegas.
Monografia (Graduação) – UFPB/CCAE.

1. Dados. 2.Ferramentas ETL. 3. ETL.

UFPB/BS-CCAE CDU: 004.6(043.2)

HERMANNY ALEXANDRE DOS SANTOS LIRA FILHO

ANÁLISE COMPARATIVA DAS FERRAMENTAS DE ETL - KETTLE E TALEND

Trabalho de Conclusão de Curso submetido ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal da Paraíba – Campus IV, como parte dos requisitos necessários para obtenção do grau de **BACHAREL EM SISTEMAS DE INFORMAÇÃO.**

Assinatura do autor:_____

APROVADO POR:

Orientador: Profª. M.Sc Ana Liz Souto Oliveira de Araújo

Universidade Federal da Paraíba – Campus IV

Co-orientador: Profª. M.Sc Renata Fernandes Viegas

Universidade Federal da Paraíba – Campus IV

Profª. Drª. Adriana Zenaide Clericuzi

Universidade Federal da Paraíba – Campus IV

Prof. M.Sc José Jorge Lima Dias Junior

Universidade Federal da Paraíba – Campus IV

RIO TINTO – PB

2013

AGRADECIMENTOS

Agradeço a Deus por ter me iluminado e dado todas as condições para poder realizar mais um sonho na minha vida.

A minha querida e amada mãe, pessoa na qual sempre me deu forças e condições para superar todos os obstáculos na minha vida.

A minha tia Susana e demais familiares, que sempre me apoiaram.

A minha amada Tárcia, pessoa essencial na minha vida, onde sempre esteve comigo tanto nos momentos difíceis como alegres.

Aos meus amigos que foram bastante importantes durante todo o curso.

A Ana Liz e Renata Viegas, minha orientadora e co-orientadora respectivamente, que foram essenciais para o desenvolvimento da minha monografia.

Agradeço também aos professores da UFPB por todo o conhecimento compartilhado, no qual contribuíram bastante para minha formação e desenvolvimento profissional.

“Quando não se sabe para onde se vai, nunca se vai muito longe.”

Johann Goethe

RESUMO

ETL é o acrônimo de *Extract*, *Transform* e *Load* (Extração, Transformação e Carga) e trata-se de um processo de extração de dados de fontes de origem, transformação para atender as necessidades de negócio e carga dos dados em fontes de destino. Há diversas ferramentas de ETL *open-source* disponíveis no mercado. Entre estas, podemos destacar duas: Kettle e Talend. Estas ferramentas livres podem ter as mesmas características e oferecer os mesmos recursos entre si, mas podem se diferenciar no tocante a desempenho, ambiente de trabalho, linguagem na qual foi desenvolvida, na forma de desenvolver as migrações dos dados, na forma de exibição dos erros, entre outras. Este trabalho tem como objetivo comparar as ferramentas de ETL *open-source* Kettle e Talend, tendo como base alguns critérios relacionados quanto a forma de desenvolver transformações, as funcionalidades disponibilizadas pelas ferramentas e ao desempenho das transformações.

Palavras chave: Ferramentas ETL, ETL, Dados.

ABSTRACT

ETL is acronym of Extract, Transform and Load is a process of extracting data from source databases, transforming to meet the business needs and load the data into the target sources. There are several tools open-source ETL available market. Among these, we highlight two: Kettle and Talend. These tools can have the same features among themselves, but may differ in regard to performance, work environment, in which language was developed, in form to perform data migrations, display of errors, and others. This paper aims to compare the open-source tools Kettle and Talend ETL, based on some criteria related as how to develop transformations, the functionality provided by the tools and the performance transformations.

Keywords : ETL tools, ETL, data.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

BD	Banco de Dados
BI	*Business Intelligence*
CPU	*Central Processing Unit*
CSV	*Comma Separated Values*
DM	*Data Mart*
DSA	*Data Staging Area*
DW	*Data Warehouse*
ETL	*Extract Transform Load*
GUI	*Graphical User Interface*
IDE	*Integrated Development Environment*
KDD	*Knowledge Discovery in Databases*
PDI	*Pentaho Data Integration*
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	*Structured Query Language*
XML	*Extensible Markup Language*

SUMÁRIO

1. INTRODUÇÃO

Com a difusão da internet e a evolução da tecnologia da informação, a maioria das empresas utilizam sistemas informatizados para realizar seus processos diários. Com o passar do tempo, as empresas percebem a grande quantidade de dados gerados relacionados aos negócios, como por exemplo, pedidos, vendas, preços, custos, máquinas, entre outros. Porém, no ambiente competitivo que as empresas estão inseridas hoje em dia, é preciso lidar com essa massa de dados como uma matéria-prima para, ao relacionarem entre si, gerar informações úteis para a gestão do negócio (KOCSKA et al., 2009).

A análise destes dados é requerida a todo instante por executivos e gerentes, a fim de adaptarem rapidamente a empresa às tensões do mercado. Entretanto, segundo Fayyad et al. (1996a), o homem não está preparado para interpretar uma grande quantidade de dados. Uma das alternativas para o gestor analisar e interpretar essa grande quantidade de dados é a utilização de técnicas e ferramentas, destacando-se assim, o processo de descoberta de conhecimento – *Knowledge Discovery in Databases* (KDD).

Na maioria das empresas esses dados são provenientes de diversos sistemas ao mesmo tempo, e em bases de dados diferentes. Nesses casos, uma análise integrada dos dados de todos esses sistemas se torna custoso e de difícil solução. Portanto, para a solução desse contexto, a utilização de um *Data Warehouse* (DW) é fundamental. Um *Data Warehouse* é um conjunto de dados integrados que extrai e reúne informações de diversas fontes.

No decorrer do procedimento de KDD ocorre um processo chamado de *Extract, Transform, Load* (ETL), onde sua principal funcionalidade é integrar todos estes dados de forma consistente em um único repositório de dados como um DW. Porém, o processo de ETL não é apenas um sub-processo na construção de um DW (Abreu, 2008), sua utilização abrange outros cenários como: migrar dados entre bases transacionais diferentes, exportar dados, carregar *Data Mart* (DM), entre outros.

Diante do contexto apresentado, este trabalho propõe realizar uma análise comparativa entre as ferramentas de ETL Kettle e Talend, tendo como base alguns critérios estabelecidos. Esses critérios são subdivididos em três grupos como: a forma de desenvolver transformações ou *jobs*, as funcionalidades disponibilizadas pelas ferramentas e ao desempenho das transformações ou *jobs*.

1.1. Objetivos, metodologia e questões de pesquisa

1.1.1. Objetivo Geral

O objetivo geral do trabalho consiste em uma análise comparativa entre as ferramentas de ETL Kettle e Talend baseada em critérios estabelecidos e, a partir disso, com os resultados obtidos, contribuir como um documento que pode ser utilizado como fonte para auxiliar na escolha da ferramenta que melhor se aplique ao contexto.

1.1.2. Objetivos Específicos

Os objetivos específicos do trabalho são:

- Descrever sobre as ferramentas de ETL *open-source* Kettle e Talend;
- Definir base de dados a ser utilizada para a comparação entre as ferramentas;
- Definir critérios para a comparação entre as ferramentas;
- Realizar a análise comparativa entre as ferramentas de ETL Kettle e Talend, através dos critérios estabelecidos.

1.1.3. Questões de Pesquisa

- **QP01.** Quais os principais conceitos e características das ferramentas Kettle e Talend?
- **QP02.** Quais critérios que podem ser usados para comparar as ferramentas de ETL Kettle e Talend?
- **QP03.** Que ferramenta melhor se adequa aos grupos de critérios estabelecidos tendo como base o cenário utilizado na análise comparativa?

1.1.4. Metodologia

Nesta monografia, foi realizado um trabalho exploratório e descritivo. Assim, o mesmo foi dividido nas seguintes etapas:

- **Etapa I – Revisão Bibliográfica:** análise de artigos, livros, revistas científicas sobre DW, DM, KDD, ETL, ferramentas de ETL, entre outros;

- **Etapa II – Coleta de dados:** levantamento de dados e ferramentas necessárias para aplicar o processo de ETL;
- **Etapa III – Análise de metodologia:** definir metodologia para comparar as ferramentas de ETL;
- **Etapa IV – Análise e apresentação dos resultados:** analisar e apresentar os resultados da comparação das ferramentas de ETL *open-source*.

1.1.5. Organização do Trabalho

Este trabalho encontra-se estruturado da seguinte forma:

- O capítulo dois apresenta a fundamentação teórica, abordando temas como *Data Warehouse*, *Data Mart*, KDD e ETL;
- O capítulo três aborda conceitos e características das ferramentas de ETL *open-source* Kettle e Talend;
- O capítulo quatro apresenta a análise comparativa entre as ferramentas de ETL, onde compõe a metodologia de comparação das ferramentas como também a própria análise comparativa entre elas;
- O capítulo cinco aborda os trabalhos relacionados ao tema em questão, mostrando os diferenciais;
- O capítulo seis explora os resultados obtidos das comparações feitas entre as ferramentas de ETL;
- O capítulo sete apresenta algumas considerações finais sobre o trabalho e perspectivas para futuros trabalhos;
- Por fim, são apresentadas as referências bibliográficas utilizadas neste trabalho.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo será apresentada a fundamentação teórica usada como base para a preparação deste trabalho, abrangendo assuntos como *Data Warehouse* e *Data Mart*, o processo de KDD e por fim o processo de ETL, que é o foco do trabalho.

2.1. *Data Warehouse*

Atualmente as empresas almejam cada vez mais competitividade no mercado. A capacidade para agir rapidamente e decisivamente em um mercado em crescente competitividade passou a ser um fator crítico para o sucesso (TAKAOKA, 2004). Assim, por esse fator estar estritamente ligada à tomada de decisões, não se questiona a importância de obter um ambiente como o *Data Warehouse* para análise inteligente dos dados essenciais da organização.

Segundo Inmon (1997), um DW é uma coleção de dados orientada por assuntos, integrada, variante no tempo e não volátil, que tem por objetivo dar suporte aos processos de tomada de decisão. As principais características de um DW são:

- Orientado por assunto: Os dados são organizados de acordo com os assuntos de interesse da empresa, como por exemplo: produto, cliente, loja;

- Integrado: Todo dado trazido dos sistemas transacionais para o DW, tem que passar por uma "limpeza", ou seja, consolida-los de forma que passem a terem significado único. Por exemplo, não deixar que o atributo "sexo" seja tratado de várias maneiras, como: "m/f", "h/m", "1/0";

- Variante no tempo: Os dados são precisos em relação ao tempo e representam resultados operacionais do momento em que foram capturados para poderem ser utilizados em comparações, tendências e previsões;

- Não Volátil: Os dados não são atualizáveis. A cada mudança ocorrida no dado, uma nova entrada no DW é criada e não atualizada. Em outras palavras, DWs são carregados uma única vez e, a partir desse momento, só podem ser consultados.

Os dados em si, isolados, não têm muita importância por não trazerem nenhuma conclusão ou vantagem estratégica para a empresa. Entretanto, se possuírem um formato e/ou ligação com outros dados, podem fornecer informações úteis se cruzadas sob uma perspectiva de uma determinada regra de negócio. Como mostra a figura 1, os dados são advindos de fontes de origem, passa por um processo de ETL e são carregados em repositórios de dados como DW ou DM (*Data Mart*).

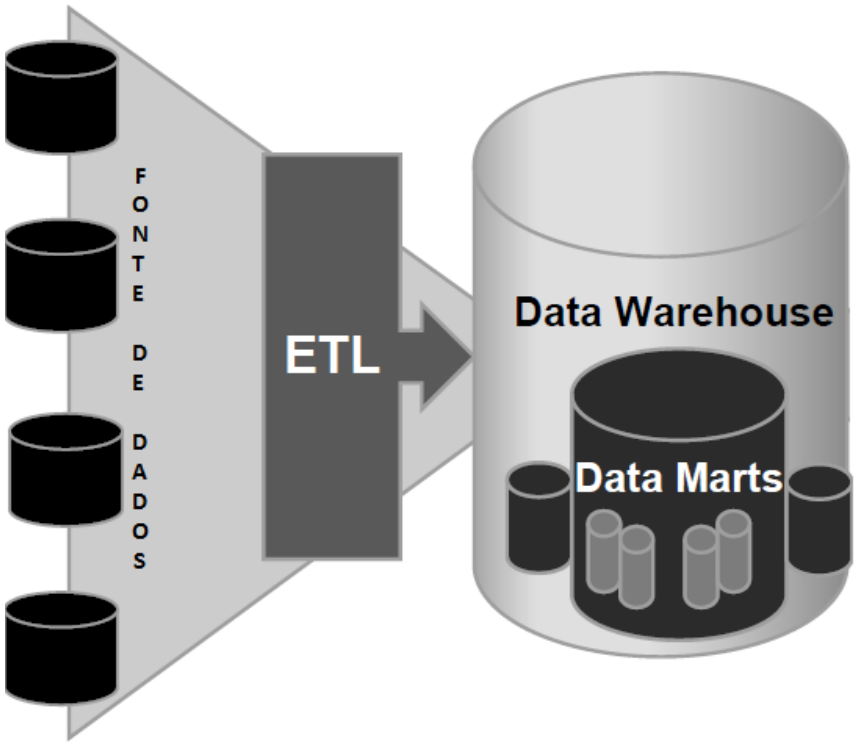


Figura 1 – Estrutura de um *Data Warehouse*, adaptada de Barbieri (2001)

Os dados carregados no DW possibilitam análises complexas por estarem relacionados a um ponto no eixo do tempo, resultando, portanto, em novos conhecimentos, comparações, previsões, aumentando assim a produtividade das empresas.

2.1.1. *Data Mart*

Um *Data Mart* pode ser considerado uma especialização, uma espécie de *Data Warehouse* com um assunto-foco, que atende a áreas específicas da empresa, porém voltado da mesma forma para os processos decisórios gerenciais (BARBIERI, 2001). Diferenciando um DM de um DW, essencialmente, ver-se que o primeiro é um DW departamental que fornece

informações de suporte à decisão não para uma organização de modo geral, mas sim para departamentos específicos como, por exemplo, finanças, estoque, vendas e marketing, como demonstra a figura 2.

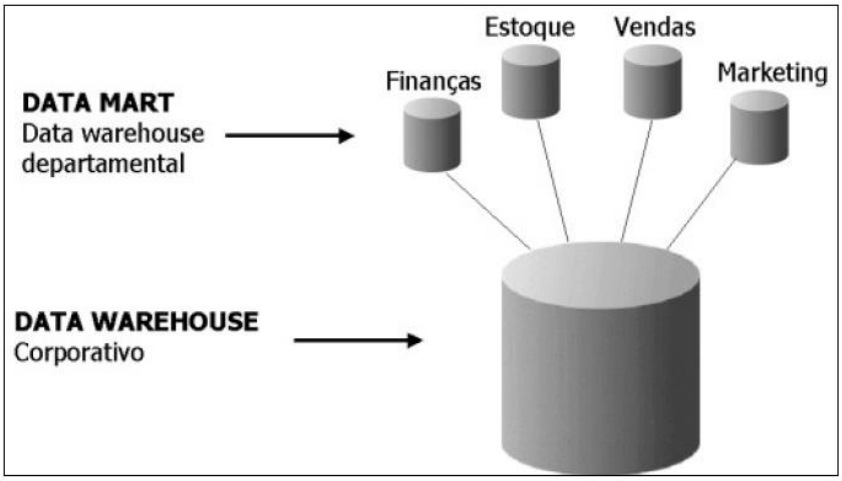


Figura 2 – *Data Warehouse* X *Data Mart*, adaptada de Machado (2000)

Os *Data Marts* são muito bem aceitos no campo empresarial, pois suas características exigem menos investimento de infraestrutura, produzem resultados mais rapidamente e são escaláveis até um *Data Warehouse* (SECO et al., 2000). A partir dos DMs, pode-se construir um DW em que esse último representará o conjunto de todos os departamentos das organizações.

2.2. O Processo de KDD (*Knowledge Discovery in Databases)*

Com o avanço da tecnologia, não tem sido difícil para as grandes empresas armazenar grandes volumes de dados (registros históricos) em seus computadores e resgatá-los quando necessário. Embora os dados armazenados sejam um bem valioso de uma organização, muitas se deparam com o problema de ter "muitos dados, mas pouco conhecimento" sobre eles (LU et al., 1995). Esta grande quantidade de dados supera muito as habilidades de uma mente humana a interpretá-los, criando assim a necessidade de obter técnicas que permitam sua automatização e análise inteligente, como o processo de KDD.

O Processo de KDD ou Descoberta de Conhecimento em Bases de Dados, é definido por Fayyad et al. (1996b) como sendo um processo não trivial de identificação de padrões válidos, novos, potencialmente úteis e compreensíveis, embutidos nos dados. A extração des-

se conhecimento é um processo no qual incorpora técnicas utilizadas em diversas áreas como Banco de Dados, Inteligência Artificial e Estatística.

Este processo é caracterizado como sendo um processo interativo e iterativo, composto por várias etapas interligadas (FAYYAD et al., 1996b). As etapas do processo de KDD, conforme mostra a figura 3, são:

1. Seleção: Uma vez definido e compreendido o domínio sobre o qual se pretende executar o processo de descoberta, o primeiro passo a ser realizado é selecionar um conjunto de dados que sejam relevantes para o processo de KDD. Nesta etapa pode ser necessário integrar bases de dados e compatibiliza-las;

2. Pré-processamento: Etapa onde ocorre a limpeza dos dados, podendo ocorrer a remoção de informações julgadas desnecessárias ou também um processo de padronização dos dados. Também adota-se estratégia para manusear dados ausentes e inconsistentes (DILLY, 1995; GONÇALVES, 2000);

3. Transformação: A transformação dos dados consiste em desenvolver um modelo sólido de dados de maneira que possam ser utilizados por um algoritmo de extração de conhecimento. As transformações são ditadas pela operação e técnica a ser adotada. São conversões de um tipo de dados para outro, definição de novos atributos, adequação de um valor que estar fora do contexto, entre outros (GONÇALVES, 2000; IBM, 1997);

4. *Data Mining*: Esta etapa é o núcleo do processo. É a mineração de dados no qual envolve um conjunto de técnicas e ferramentas computacionais usadas para a identificação de padrões (conhecimentos) embutidos em grande quantidade de dados;

5. Interpretação: Após identificar padrões do sistema, estes são interpretados, gerando assim, conhecimentos no qual darão suporte a tomada de decisões na empresa. Caso os resultados não forem satisfatórios, pode-se realizar todo o processo novamente ou parte do mesmo.

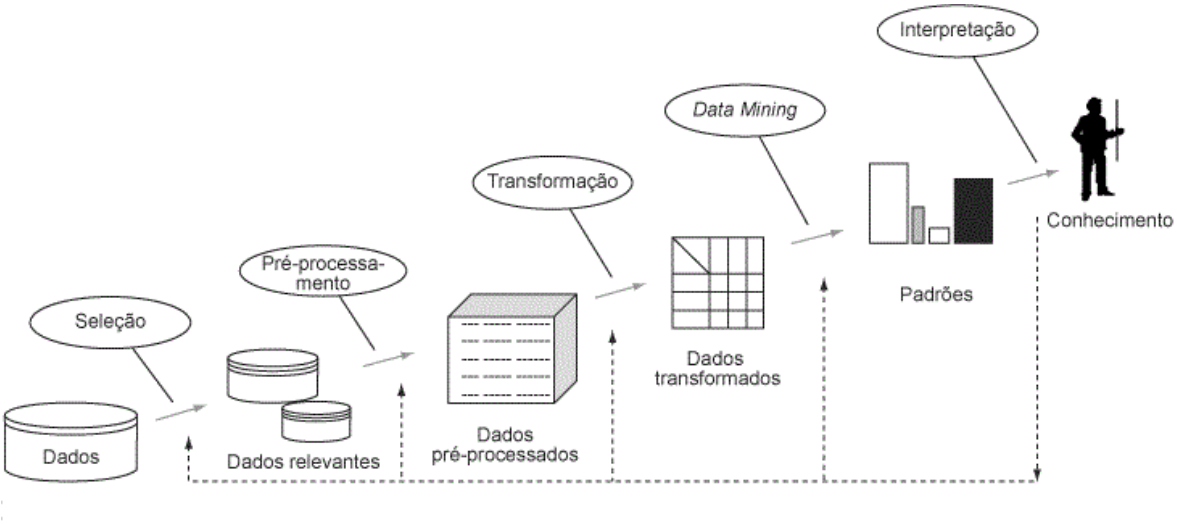


Figura 3 - Etapas do processo de KDD, adaptada de Fayyad et al. (1996b)

Observa-se que o processo de KDD é composto por três etapas iniciais, seleção, pré-processamento e transformação, e essas etapas compõem todo o processo de ETL que descreve toda a preparação dos dados para poderem posteriormente serem minerados e gerar descoberta de conhecimentos. Conforme dito anteriormente, este trabalho tem um foco maior no processo de ETL, no qual correspondem as etapas 1,2 e 3 descritas no processo de KDD.

2.3. ETL (*Extract, Transform e Load*)

ETL (ou Extração, Transformação e Carga), para Abreu (2008), é um processo que tem como objetivo a extração, transformação e carga dos dados de uma ou mais bases de dados de origem para uma ou mais bases de dados de destino.

De acordo com Vassiliadis (2002), a figura 4 descreve de forma geral o processo de ETL. A camada inferior representa todos os dados utilizados no processo. No lado esquerdo pode-se observar os dados brutos no qual são provenientes de fontes como base de dados, planilhas, arquivos textos. Os dados advindos destas fontes são obtidos, como é ilustrado na área superior esquerda da figura 4, por rotinas de extração que selecionam os dados relevantes para o processo. Posteriormente, esses dados são propagados para a *Data Staging Area* (DSA) onde são transformados e limpos antes de serem carregados em uma fonte de destino como DW, DM ou outras bases de dados. Por fim, na parte superior direita da figura, é onde ocorre o carregamento dos dados em fontes de destino através de atividades de carga programada. As três etapas (Extração,Transformação e Carga) serão mais detalhadas na seção a seguir.

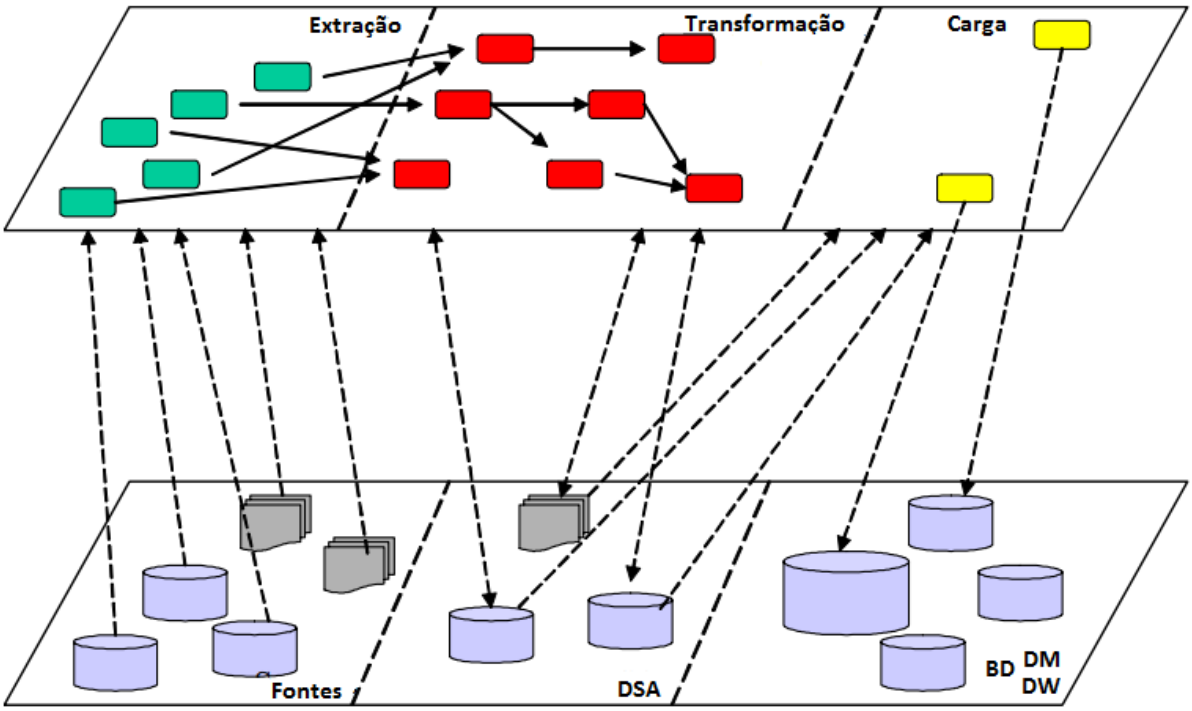


Figura 4 – Processo de ETL, adaptada de Vassiliadis et al. (2002)

2.3.1. Etapas de ETL

2.3.1.1. Extração

Nesta etapa inicial do processo de ETL, ocorre a extração dos dados de origem. Esses dados podem ser provenientes de diversas fontes como os SGBDs (Sistemas de Gerenciamento de Banco de Dados), planilhas eletrônicas, arquivos textos, entre outros.

A extração deve se basear na busca pelos dados necessários dos sistemas fontes ou externos e que estejam em conformidade com a modelagem do sistema de destino para que seja viabilizado o processo. Tal busca pode implicar em uma extração de dados inúteis ou até mesmo em um erro futuro ao carregar os dados devido à diferença de tipo, tamanho, estrutura em geral.

2.3.1.2. Transformação

Este é o processo responsável pelo tratamento e transformação dos dados. Esses dados podem ser decorrentes de fontes desconhecidas ou projetos com falhas de modelagem, por

isso é natural encontrar problemas de inconsistência como dados errôneos ou inválidos, falta de padronização, somatórios numéricos inconsistentes, falta de normalização e diversos outros problemas. Portanto, segundo Cielo (2013), todas as divergências encontradas devem passar por um processo de exclusão ou tratamento de acordo com as regras de negócio da aplicação de destino, solucionando-as para garantir confiabilidade ao processo de ETL.

Neste processo de transformação dos dados, no qual prover a integração dos mesmos, muitas divergências encontradas podem ser solucionadas ao serem submetidas a uma conversão, como por exemplo, padronizações de unidades de medida, padronizações de domínios e padronizações de tipos de dados, conforme mostra a figura 5.

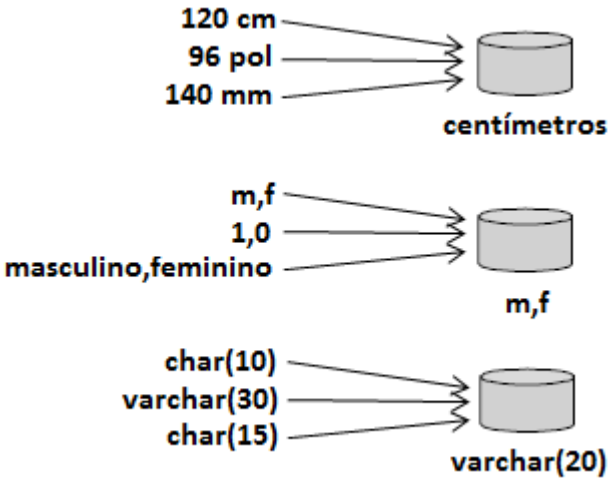


Figura 5 – Transformação dos dados advindos de origens diferentes, adaptada de Inmon, Terdeman, Imhoff (2001)

O tratamento dos dados ocorre sobre dados irrelevantes ou desnecessários dos sistemas legados que não afetam a funcionalidade destes sistemas e geralmente são inseridos somente para passar pela regra de negócio, sem acarretar valor algum para o sistema (MOSS, 1998). Resumindo, de acordo com Gonçalves (2003), o tratamento dos dados refere-se então à "limpeza" ou filtragem dos dados e a colocá-los em uma forma homogênea. Esses dois processos são descritos a seguir.

- Limpeza ou Filtragem dos dados – Identificar anomalias e garantir a integridade dos dados antes de serem carregados no seu destino final. Processo relacionado a correção

de erro de digitação, violações de integridade, substituição de caracteres desconhecidos;

- Homogeneização dos dados – Tratamento responsável por colocar os dados em uma forma homogênea, ou seja, definir um único formato sem que ocorram conflitos de modelagem. Esse tratamento é aplicado para dar precisão aos dados, padronizar expressões, tipos de dados, entre outros.

No subprocesso de homogeneização de dados podem ser encontrados vários conflitos de modelagem semântica e estrutural. Segundo Gonçalves (2003) e Abreu (2007), os conflitos semânticos são todos aqueles que envolvem o nome ou a palavra associada às estruturas de modelagem, por exemplo, mesmo nome para diferentes entidades ou diferentes nomes para a mesma entidade. Já os conflitos estruturais englobam os conflitos relativos às estruturas de modelagem escolhidas, tanto no nível de estrutura propriamente dita como no nível de domínios. Os principais tipos de conflitos estruturais são aqueles de domínio de atributo que se caracterizam pelo uso de diferentes tipos de dados para os mesmos campos.

2.3.1.3. Carga

Este processo consiste em gravar os dados, extraídos e tratados nas etapas anteriores, em uma fonte de destino. Para tal procedimento, as ferramentas de ETL proporcionam ao usuário uma funcionalidade na qual automatiza a migração dos dados entre todas as tabelas em um só processo. Essa funcionalidade ocorre na criação dos chamados *Jobs*, componentes funcionais das ferramentas de ETL. Com a criação do *Job*, também se pode programar para que a carga dos dados ocorra, dependendo da necessidade do processo montado, de uma única vez ou de forma periódica para atualização de dados. Esta última opção é a utilizada comumente em carga de dados em *data warehouse* (IBL, 2003).

2.4. Considerações Finais do Capítulo

Neste capítulo foram descritos assuntos como *Data Warehouse* e *Data Mart*, o processo de KDD e por fim o processo de ETL. No capítulo a seguir serão mostradas as características e conceitos das ferramentas de ETL que serão utilizadas na análise comparativa.

3. FERRAMENTAS ETL

Neste capítulo serão abordados alguns conceitos e características das ferramentas Kettle e Talend. Atualmente no mercado, existem diversas ferramentas de ETL comerciais como o PowerCenter, Data Stage, Oracle Enterprise Data Integrator, etc. Também existem as *open-sources* como Kettle da Pentaho, JasperETL, Talend Open Studio & Integration, CloverETL, entre outras.

As ferramentas de ETL *open-sources* foram criadas na década de 90. Desde então, elas encontram-se em evolução, se aperfeiçoando a cada nova versão lançada. Atualmente, elas já possuem um bom grau de maturidade para serem equiparadas às ferramentas proprietárias.

As principais características das ferramentas de ETL open-source são: suporte a diversas plataformas, conectividade com diversos bancos, facilidade de uso, suporte a *debugging*, reutilização de transformações, interface gráfica intuitiva, entre outras. Neste trabalho destacam-se as ferramentas Kettle e Talend descritas nas seções a seguir.

3.1. KETTLE

O Kettle, também chamado de Pentaho Data Integration (PDI), é uma ferramenta de código aberto voltado ao processo de ETL advindo da suíte Pentaho. O propósito de sua utilização está relacionado à migração de dados entre aplicações ou base de dados, exportação de dados, integração de aplicações, e também como parte de um processo de BI (*Business Intelligence*).

Segundo Bouman (2009), a arquitetura do Kettle é baseada na linguagem Java, e consiste de quatro componentes básicos:

- Spoon: ferramenta de modelagem gráfica direcionada ao usuário, onde se define a entrada, transformações e saída de dados;
- Pan: aplicativo de linha de comando para executar as transformações feitas no Spoon;
- Chef: ferramenta de modelagem gráfica direcionada para criação de *Jobs*, que consiste em tarefas como transformação, downloads, etc., no qual são colocados em um fluxo de controle;
- Kitchen: aplicativo de linha de comando para executar os *Jobs* criados no Chef.

Segundo a Pentaho (2013), no Kettle, todo processo é criado com uma ferramenta gráfica onde você especifica o que fazer sem escrever código para indicar como fazê-lo. Por isso pode-se dizer que o PDI/Kettle é orientado por metadados[1]. A figura 6 demonstra o ambiente de trabalho da ferramenta Kettle.

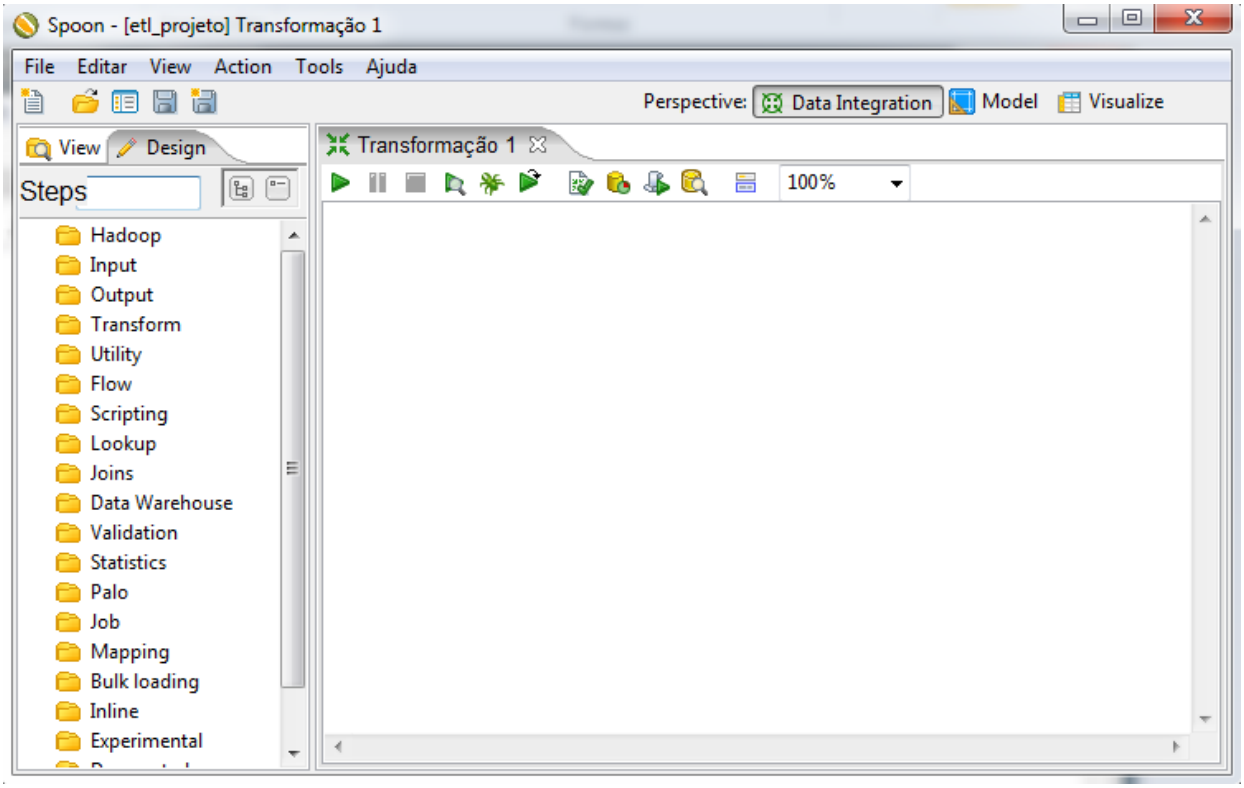


Figura 6 – Ambiente do Kettle

A ferramenta trabalha com dois tipos de modelagem como *jobs* e transformações. As transformações são rotinas formadas por passos interligados, onde, a princípio, é capturado a entrada de dados e por final, a saída dos mesmos. Os *jobs* são rotinas no qual servem para executar transformações ou até mesmo outros *jobs*.

Outros conceitos importantes são *steps* e *hops*. O *step* é uma unidade mínima do processo, onde executa uma tarefa específica, seja uma leitura ou transformação de algum dado. A ligação entre esses *steps* é chamada de *hop*, no qual são representados graficamente demonstrando o fluxo dos dados (PENTAHO, 2013).

[1] São dados que descrevem completamente os dados que representam (Almeida, 1999). Informações úteis para identificar, localizar, compreender e gerenciar os dados.

A figura 7 ilustra a estrutura de uma transformação no Kettle e seus componentes gráficos, *steps* e *hops*, conceituados anteriormente, nos quais são responsáveis pelo fluxo de dados da migração.

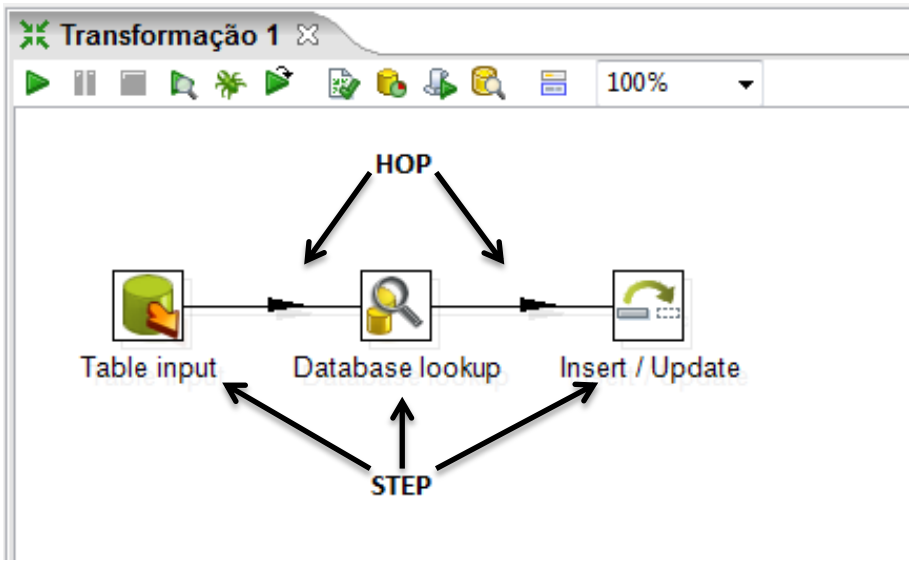


Figura 7 – Transformação no Kettle

3.2. TALEND

O Talend Open Studio é uma ferramenta de código aberto que oferece serviços de integração de dados. Possui suporte para a maioria dos tipos de fonte de dados além de vários componentes para integração, migração e operações de sincronização de dados (TALEND, 2009).

A ferramenta foi desenvolvida na linguagem Java e utiliza como base a Plataforma Eclipse, tendo como proveito todos os recursos que a IDE disponibiliza na hora de executar e depurar código. Esta ferramenta oferece um ambiente de desenvolvimento "amigável", baseado na composição de componentes visuais. Isso favorece que usuários não familiarizados com a linguagem Java possam utilizar a ferramenta. A figura 8 mostra a GUI (*Graphical User Interface*) do Talend.

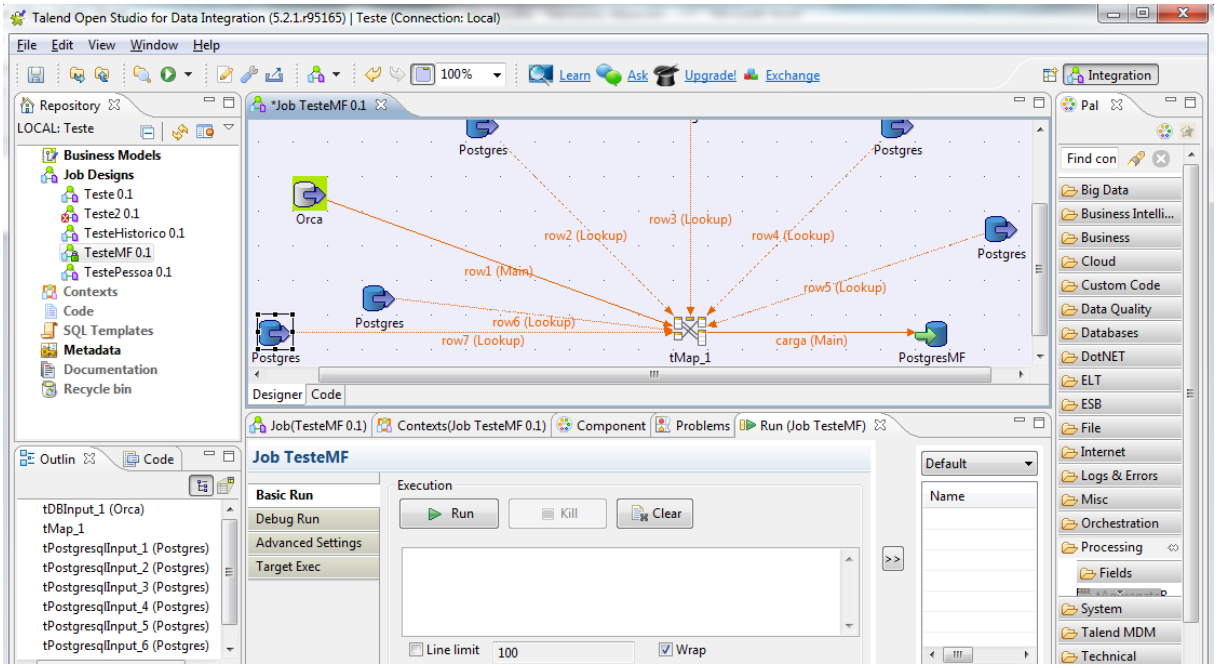


Figura 8 – GUI do Talend

O repositório local do Talend apresenta todos os elementos do projeto, organizados nas seguintes categorias:

- Business Models: Utilizado para fins documentacional. Contém elementos para criação de fluxograma de sequência, operações representadas por processos, e também anexar arquivo;
- *Job* Designs (Desenho de trabalho): Local onde ficam as tarefas. Tem suporte até a criação de pastas para organizar as tarefas do projeto;
- Contexts (Contextos): Local onde são definidas as constantes e seus respectivos valores, agrupados em contextos;
- Code (Código): Local onde pode escrever código java nas classes incorporadas ao projeto para resolver situações particulares;
- SQL Templates (Modelos de SQL): Local onde é possível definir comandos SQL para serem executados dentro do projeto;
- Metadata (Metadados): Conceito no qual permite definir e configurar os recursos que as tarefas vão necessitar no decorrer do projeto como: conexão com banco de dados, arquivos externos, etc;
- Documentation (Documentação): Local para fins de documentação do projeto;
- Recycle bin (Lixeira): Local onde ficam os elementos apagados, dando a possibilidade de recuperá-los.

Conforme mostra a figura 9, o Talend trabalha com o conceito de criação de *Jobs*, semelhante ao conceito de transformação no Kettle, onde são rotinas compostas por componentes gráficos, que ligados, compõem todo o fluxo dos dados no processo de ETL. Estes componentes gráficos são recursos no qual podem fazer leitura dos dados, tratamento dos dados, operações matemáticas, entre outras.

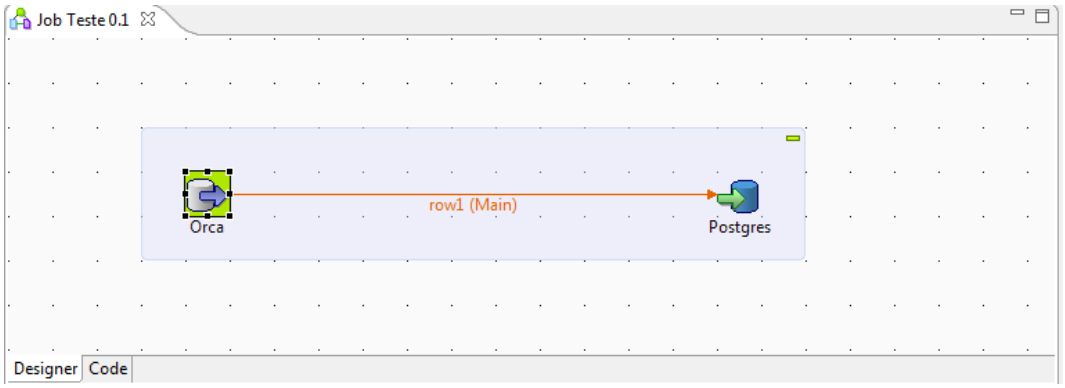


Figura 9 – *Job* no Talend

3.3. Considerações Finais do Capítulo

Neste capítulo foram descritas as ferramentas de ETL Kettle e Talend. No capítulo a seguir será apresentada a análise comparativa das ferramentas.

4. ANÁLISE COMPARATIVA DAS FERRAMENTAS DE ETL - KETTLE E TALEND

Neste capítulo serão descritas e detalhadas as etapas de desenvolvimento da análise comparativa das ferramentas de ETL *open-source* Kettle e Talend através de critérios definidos. É utilizada como referência a base de dados usada no projeto de pesquisa chamado de Sistema de Informação Municipal (SIM) da Prefeitura Municipal de João Pessoa (PMJP). Este projeto tem como objetivo integrar módulos como recursos humanos, tributário, orçamentário, etc., para melhorar a comunicação entre eles, evitar retrabalho e integrar as informações.

O experimento é composto pelas seguintes etapas:

- Apresentação do ambiente;
- Modelagem Relacional;
- Definição dos métodos para avaliação/comparação das ferramentas;
- Desenvolvimento da avaliação/comparação das ferramentas através dos métodos pré-definidos.

4.1. Apresentação do ambiente

O ambiente utilizado para a comparação das ferramentas é composto por apenas uma máquina local tendo instalado todos os componentes necessários para a análise comparativa. A configuração da máquina e as ferramentas adicionais utilizadas estão descritas na Tabela 1 e 2 respectivamente.

Tabela 1 - Configuração da máquina

	Configuração da Máquina
Processador	Intel Core i5-2410M 2.30 GHz, 3 MB L3 Cache
Memória	4 GB (RAM) DDR3
Disco Rígido	500 GB, 7.200 RPM, 16MB DE BUFFER SATA II
Sistema Operacional	Windows 7 Ultimate 64 BIT

Tabela 2 - Ferramentas adicionais utilizadas

	Ferramentas Adicionais Utilizadas
SGBD Legado	Sql Server 2008 R2
SGBD Novo	PostgreSQL v8.4
Ferramenta de ETL	Kettle - Spoon v4.2.0
Ferramenta de ETL	Talend Open Studio For Data Integration v5.2.1
Ferramenta p/ Avaliação de Desempenho	Ferramenta do Windows 7 – Monitor de Recursos

As ferramentas escolhidas para realizar migrações de dados foram o Kettle e Talend. O Kettle foi escolhido por ser *open-source* e pela experiência adquirida em sua utilização no próprio projeto SIM. O Talend foi selecionado por também ser *open-source* e por ter mais materiais disponibilizados na internet.

Com a necessidade de comparar o desempenho das ferramentas ETL ao realizarem a migração de dados, foi feita uma pesquisa na internet para saber como comparar e também para escolher uma ferramenta que pudesse realizar essa comparação. Então através da pesquisa, foi definido que a comparação seria desenvolvida através do monitoramento dos processos (execuções de programas) individuais em tempo real mostrando o quanto cada um utiliza de memória e CPU. A ferramenta utilizada para essa tarefa foi o Monitor de Recursos, por ter um ambiente de fácil compreensão e por ser integrada ao Windows 7.

4.2. Modelo Relacional

Esta seção mostra parte do Modelo Relacional do projeto, tanto da base do sistema legado quanto do novo, onde há ênfase apenas nas tabelas que foram essenciais para comparar/avaliar as ferramentas de ETL.

A figura 10 mostra a modelagem das tabelas “Fonte_Pagadora”, tanto da base de dados do sistema legado quanto do novo, respectivamente. O conhecimento da tabela “Fonte_Pagadora” e seus campos servem para ajudar no desenvolvimento da transformação no qual irão migrar os dados correspondentes a ela.

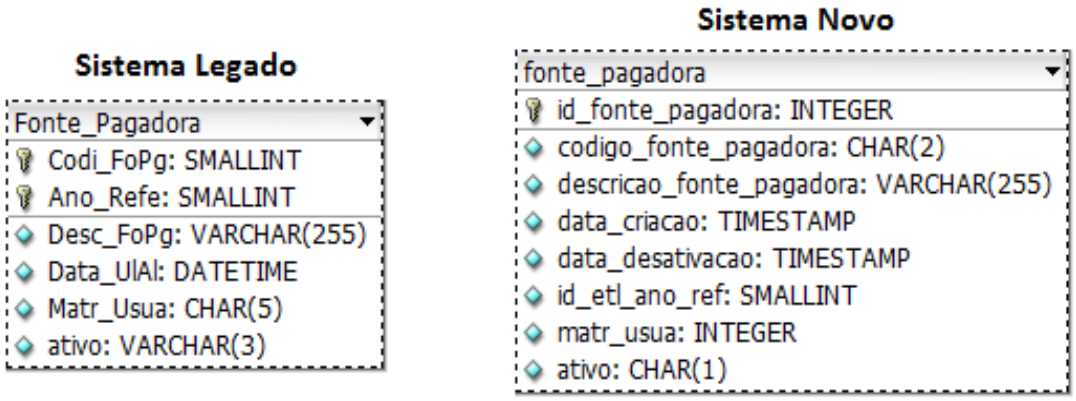


Figura 10 - Modelagem Relacional para migrar dados da tabela Fonte Pagadora

A figura 11 mostra parte da modelagem relacional da tabela "Produto" tanto da base legada quanto da base nova, evidenciando apenas o que é relevante para poder realizar a migração dos dados.

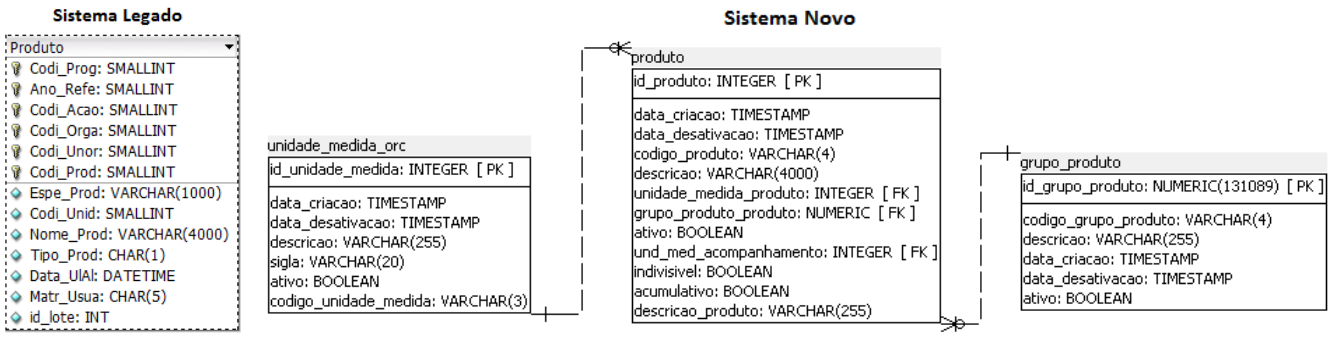


Figura 11 - Modelagem Relacional para migrar dados da tabela Produto

A figura 12 apresenta parte da modelagem das tabelas "Fornecedor", do sistema legado, e seu correspondente, a tabela "Pessoa" do sistema de destino. Apesar das tabelas terem modelagens diferentes, os conteúdos dos campos da tabela de origem irão se adaptar a modelagem da tabela de destino conforme o tamanho e tipo, para que a migração ocorra de forma consistente e sem ocorrer nenhum problema.

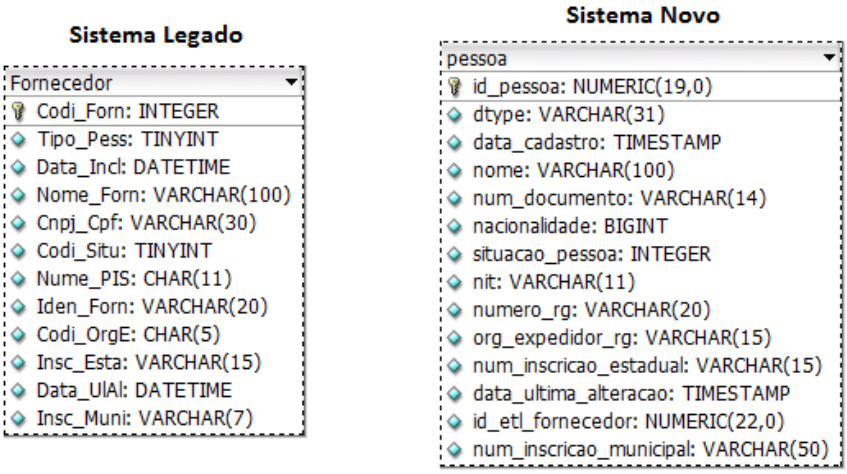


Figura 12 - Modelagem Relacional para migrar dados entre as tabelas Fornecedor e Pessoa

A figura 13 mostra a modelagem relacional das tabelas "Histórico Processos" do sistema legado e novo, respectivamente. Esse cenário passará por adaptação à modelagem de destino, para que ocorra a migração dos dados.

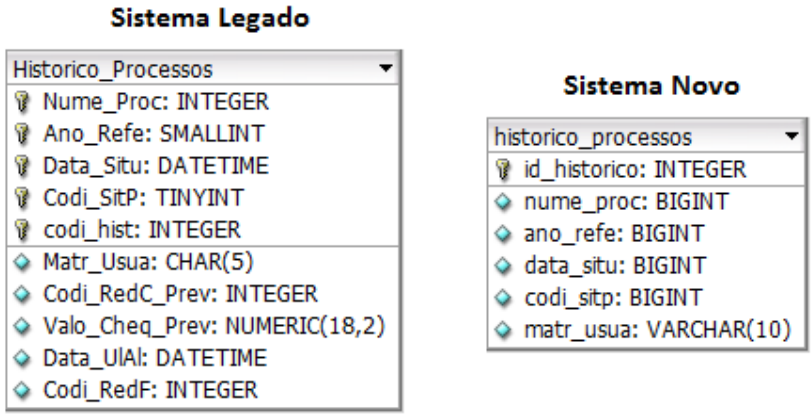


Figura 13 - Modelagem Relacional para migrar dados da tabela Historico Processos

4.3. Definição dos critérios para comparação das ferramentas

Nesta seção serão descritos os critérios que foram definidos para realizar a comparação entre as ferramentas *open-source* Kettle e Talend. Estes critérios, com suas respectivas fontes, são mostrados na tabela 3, e foram definidos através dos métodos a seguir:

- Pesquisa bibliográfica: Alguns critérios de comparação foram definidos através do esclarecimento dos principais requisitos em uma ferramenta de ETL, conforme Gonçalves (2003), através do livro Extração de Dados para *Data Warehouse*;

- Utilização das ferramentas: Alguns critérios foram definidos, para comparar as duas ferramentas, através da própria experiência ao utilizar tanto suas funcionalidades essenciais em um processo de ETL quanto suas funcionalidades diferenciais disponibilizadas por cada uma.

Tabela 3 - Critérios de comparação entre as ferramentas de ETL

Quanto a forma de desenvolver transformações ou *jobs*		
Código	**Critério**	**Fonte**
CRT 1	Mapeamento entre campos de tipos diferentes	Autoria Própria
CRT 2	Seleção e mapeamento dos campos em ordem	Autoria Própria
CRT 3	Mapeamento da forma que dois campos sejam preenchidos por dados de um campo	Autoria Própria
Quanto as funcionalidades disponibilizadas pelas ferramentas		
CRT 4	Extrair dados de diversas fontes	Gonçalves (2003)
CRT 5	Disponibilizar diagramas gráficos e/ou linguagem de programação para o desenvolvimento de transformações	Gonçalves (2003)
CRT 6	Conter repositório de metadados	Gonçalves (2003)
CRT 7	Permitir restauração de transformações ou demais elementos	Autoria Própria
CRT 8	Permitir verificação de transformações	Autoria Própria
CRT 9	Permitir pré-visualização das transformações	Autoria Própria
CRT 10	Permitir geração do SQL da extração dos dados através de um editor gráfico	Autoria Própria
Quanto ao desempenho das transformações ou *jobs*		
CRT 11	Velocidade	Autoria Própria
CRT 12	Tempo	Autoria Própria
CRT 13	Acesso a memória	Autoria Própria
CRT 14	Acesso a CPU	Autoria Própria

4.3.1. Quanto a forma de desenvolver transformações ou *jobs*

Nesta abordagem, as ferramentas de ETL são comparadas de acordo com a forma de desenvolver uma transformação ou *job*. Para tal experimento, foi necessário criar um cenário de migração de dados e em seguida desenvover a transformação relacionada a esta migração. O objetivo é identificar e demonstrar 3 (três) diferenças na forma de desenvolver essa transformação nas duas ferramentas. As comparações foram realizadas no tocante ao quanto à ferramenta é flexível para desenvolver as transformações. Os critérios de comparação relacionados a esta abordagem são descritos a seguir:

- CRT 1 - Mapeamento entre campos de tipos diferentes: se a ferramenta permite o mapeamento entre campos de tipos diferentes, como por exemplo, definir que um campo “codigo” do tipo “bigint(5)” receba valores do campo “codigo” do tipo “varchar(5)”;

- CRT 2 - Seleção e mapeamento dos campos em ordem: se para desenvolver uma transformação na ferramenta é necessário que, a relação da seleção dos campos para extração dos dados e o mapeamento dos campos que irá receber a carga dos dados, tem que estar em ordem. Um exemplo é que se colocado no código SQL de extração de dados que irá extrair primeiro o campo “nome” da base de origem, no mapeamento dos campos é necessário que seja feito primeiro o mapeamento deste campo “nome”, e não, por exemplo, um outro campo como “cpf”, pois pode acarretar em um erro determinando que são campos de tipos diferentes;

- CRT 3 - Mapeamento da forma que dois campos sejam preenchidos por dados de um campo: se a ferramenta permite que, ao realizar o mapeamento dos campos, dois campos sejam preenchidos com dados de apenas um campo.

4.3.2. Quanto as funcionalidades disponibilizadas pelas ferramentas

Nesta abordagem, as ferramentas de ETL são comparadas através das funcionalidades disponibilizadas por cada uma. Para o densenvolvimento deste experimento, foi necessário desenvolver uma transformação, com objetivo idêntico, nas duas ferramentas e analisar quais

as funcionalidades nas quais as ferramentas dispõem para tal transformação. Os critérios de comparação relacionados a esta abordagem são descritos a seguir:

- CRT 4 - Extrair dados de diversas fontes: capacidade da ferramenta extrair dados de diversas fontes, como arquivos textos, planilhas, arquivos XML, aplicações, diversos bancos de dados, entre outros;

- CRT 5 - Disponibilizar diagramas gráficos e/ou linguagem de programação para o desenvolvimento de transformações: capacidade da ferramenta disponibilizar de componentes gráficos e uma linguagem de programação para desenvolver as transformações;

- CRT 6 - Conter repositório de metadados: se a ferramenta possui um repositório de metadados para auxiliar no desenvolvimento do projeto;

- CRT 7 - Permitir restauração de transformações ou demais elementos: capacidade da ferramenta permitir restaurar transformações ou outros elementos apagados, como conexões de banco de dados, componentes gráficos, entre outros;

- CRT 8 - Permitir verificação de transformações: capacidade da ferramenta permitir verificar a transformação antes de ser executada. Esta funcionalidade aponta e descreve os erros e/ou acertos relacionados a cada *step* ou a transformação em geral;

- CRT 9 - Permitir pré-visualização das transformações: capacidade da ferramenta permitir uma pre-visualização da execução de transformações, mostrando assim todos os dados que compõem o fluxo da migração ou erros que poderão acontecer;

- CRT 10 - Permitir geração do SQL da extração dos dados através de um editor gráfico: capacidade da ferramenta permitir a geração do código SQL, no qual é responsável pela extração dos dados, através do uso de um editor gráfico representado pelas tabelas e seus respectivos campos da base de dados de origem.

4.3.3. Quanto ao desempenho das transformações ou *jobs*

Nesta abordagem, a comparação entre as ferramentas de ETL é feita através do desempenho da execução de uma transformação ou *job*. Para este experimento foi necessário desenvolver transformações com migrações de dados entre as mesmas tabelas nas duas ferramentas de ETL. Então, após o planejamento, foram executadas as transformações nas duas ferramentas para analisar os critérios de comparação descritos a seguir:

- CRT 11 - Velocidade: comparar, entre as duas ferramentas, a velocidade da execução de uma transformação;

- CRT 12 - Tempo: comparar, entre as duas ferramentas, o tempo necessário para executar uma transformação;

- CRT 13 - Acesso a memória: comparar, entre as duas ferramentas, o quanto uma transformação necessita de memória para ser executada;

- CRT 14 - Acesso a CPU: comparar, entre as duas ferramentas, o quanto uma transformação necessita de CPU para ser executada.

4.4. Desenvolvimento da análise comparativa das ferramentas através dos critérios definidos

Nesta seção, será descrito todo o desenvolvimento da comparação das ferramentas de ETL de acordo com os critérios definidos na seção 4.3.

4.4.1. Quanto a forma de desenvolver transformações ou *jobs*

Para essa avaliação foi definido um cenário, conforme mostra a figura 14, para que seja necessário desenvolver uma transformação que realize a migração dos dados entre as tabelas “Fonte Pagadora” do sistema legado e do sistema de destino. Então, a comparação das duas ferramentas, quanto a forma de desenvolver transformações, será feita através dos critérios 1, 2 e 3. Essas diferenças serão mostradas por simulações que acarretam no resultado da comparação.

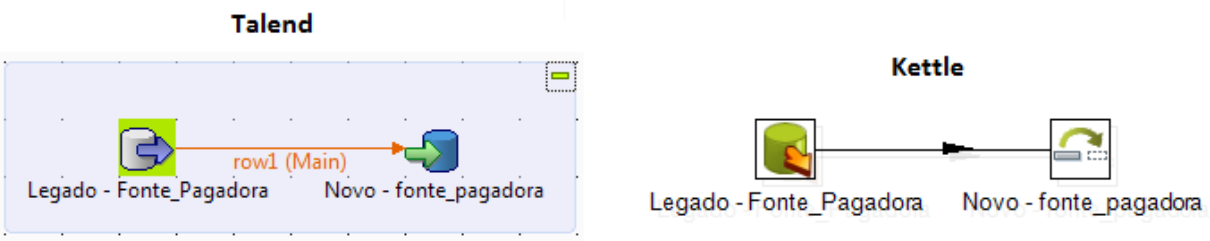


Figura 14 - Cenário para comparar transformações ou *jobs*, no Kettle e Talend

- **CRT 1 – Mapeamento entre campos de tipos diferentes**

  - **SML1 (Simulação 1)**

Nesta simulação o propósito é, ao desenvolver a transformação ou *job* do cenário da figura 14, demonstrar se as ferramentas permitem o mapeamento entre campos de tipos diferentes. Os passos para simulação foram:

1. Com o *step de table input*, foi feita a leitura/seleção dos dados através do mesmo código SQL no Kettle e no Talend, para extrair dados. Os campos que obtém os dados a extrair que não são correspondentes quanto ao tipo e tamanho nos dois sistemas são: código da fonte pagadora, sendo "smallint" no sistema legado e "char(2)" no sistema de destino, matricula do usuário onde o tipo é "char(5)" no legado e "integer" no novo, e a referência "ativo" no qual é "varchar(3)" no legado e "char(1)" no sistema novo;

    - **KETTLE:**

A figura 15 representa a estrutura do *step* do tipo *table input*, no Kettle, onde mostra alguns campos essenciais para a transformação preenchidos, como: nome dado ao *step*, a conexão com o banco de dados e o código SQL responsável por fazer a leitura/seleção dos dados que serão migrados para a fonte de destino.

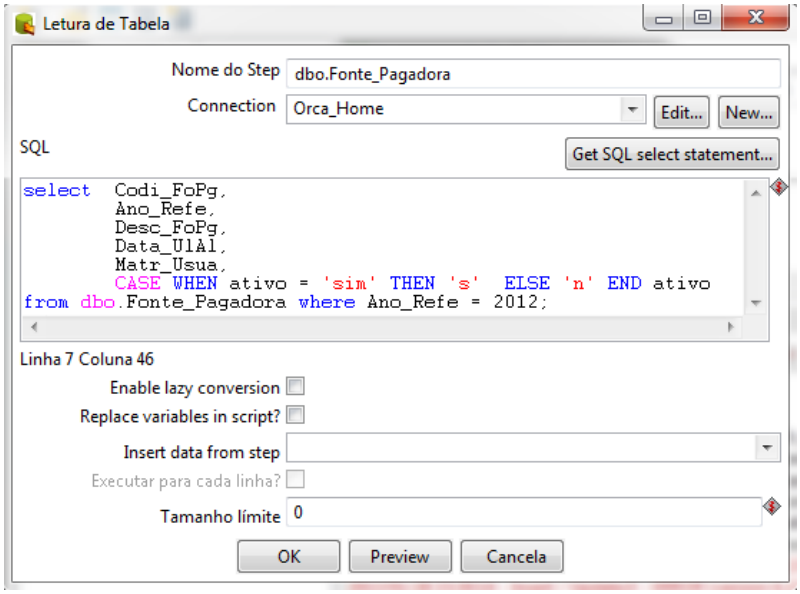


Figura 15 - *Step* para leitura/seleção dos dados no Kettle conforme a SML1

- **TALEND:**

No Talend, conforme mostra a figura 16, também se mostra o *step* com campos preenchidos como a conexão com o banco de dados e o código SQL da leitura/seleção dos dados.

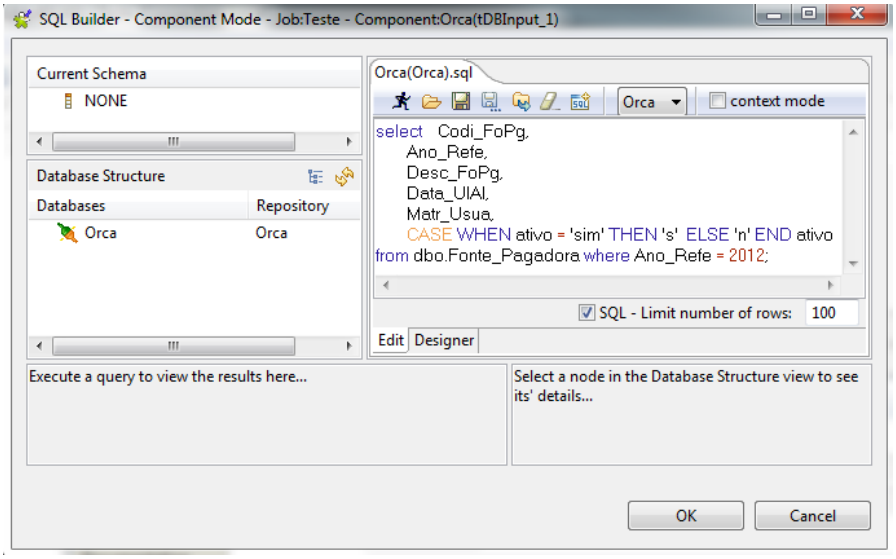


Figura 16 – Parte da estrutura do *step* para leitura/seleção dos dados no Talend conforme a SML1

2. No *step* de inserção/atualização dos dados é feito o mapeamento dos campos de origem com o de destino para que ocorra a migração dos dados;

- **KETTLE:**

A figura 17 representa a estrutura do *step* de inserção/atualização no Kettle onde se preenche campos como a conexão com o banco de dados de destino, *schema* e tabela de destino, campos chaves para comparar os registros e não duplicá-los, e o mapeamento dos campos de origem com os campos de destino.

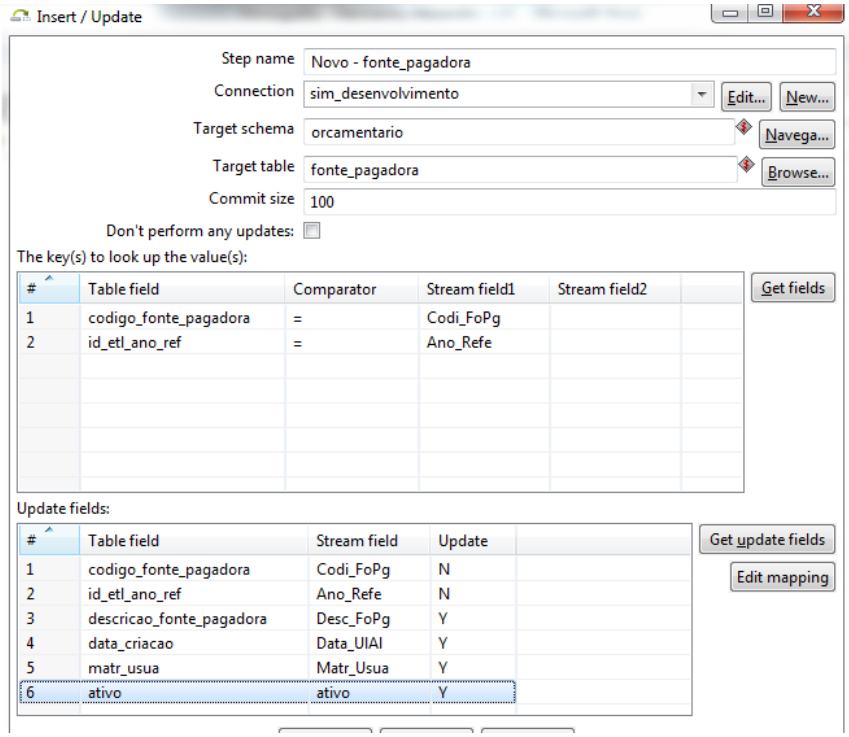


Figura 17 – *Step* de inserção/atualização com mapeamento de campos no Kettle conforme a SML1

- **TALEND:**

A figura 18 mostra o mapeamento dos campos de origem com o de destino no Talend. No Talend ao relacionar os campos das tabelas de origem com os campos das tabelas de destino, os tipos dos campos podem ser sincronizados automaticamente, de forma com que os campos fiquem com tipos compatíveis.

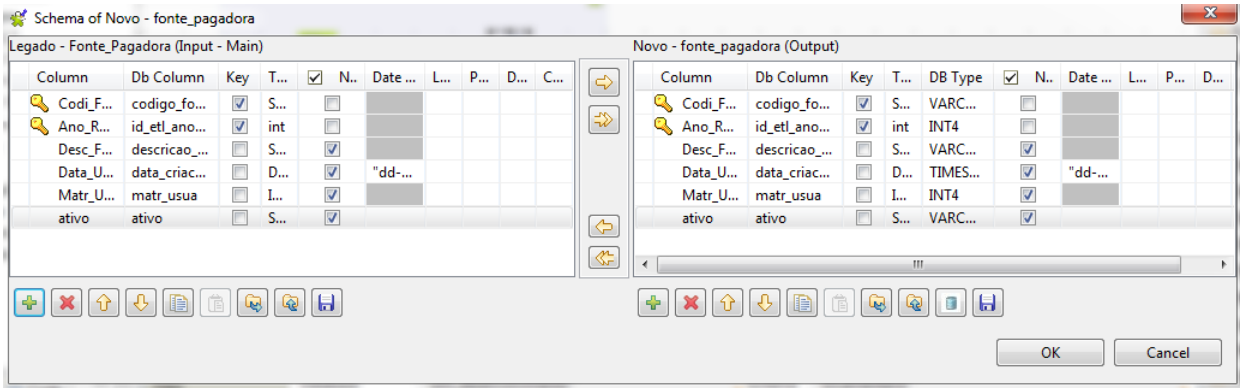


Figura 18 - Parte do *step* com o mapeamento dos campos no Talend conforme a SML1

3. Por fim, neste passo, as transformações são executadas para provocarem a diferença na construção da mesma nas duas ferramentas.

- **KETTLE:**

A figura 19 demonstra, devido ao problema de leitura/seleção dos dados, a execução da transformação "tr_orcamentario_fonte_pagadora" no Kettle apresentando um erro inesperado ao deixar o *step* de inserção/atualização de dados com a borda vermelha e alertando ao usuário, de maneira clara no *Logging*[2], que o erro está diretamente relacionado ao seguinte motivo: "operador não existe: character = bigint". O Kettle não permite fazer o mapeamento de campos com tipos diferentes, mesmo que o tamanho e o tipo de dados correspondam à regra de negócio e modelagem. Um exemplo disso, está nesta SML1, onde os dados advindos do campo "Codi_FoPg" do legado são de duas casas decimais e do tipo "smallint", enquanto que o campo "codigo_fonte_pagadora" do sistema novo que irá receber os dados é do tipo "char(2)", por isso, como mostra na figura posterior, o Talend executa o seu *job* sem acarretar problema algum na migração dos dados.

[2] Janela onde mostra detalhe sobre toda a execução da transformação. Recurso no qual auxilia bastante no tratamento de erros ao apontá-los.

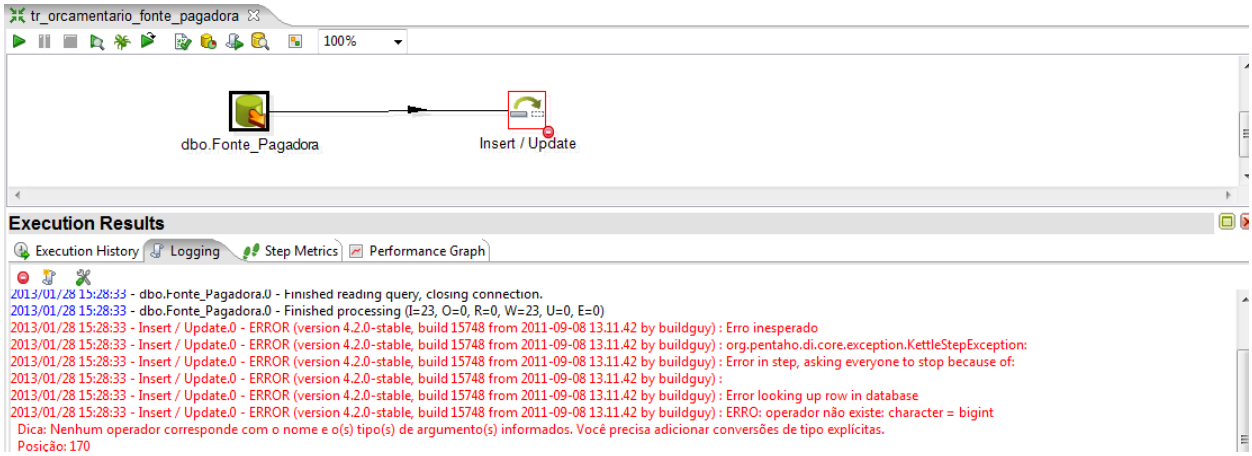


Figura 19 - Transformação executada no Kettle conforme SML1

- **TALEND:**

Conforme mostra a figura 20, foi executado no Talend o *Job* Teste no qual faz a migração dos dados conforme descrito na seção 4.4.1. Diferentemente do Kettle, o Talend não apresentou erro e suportou o mapeamento de dados entre campos de tipos diferentes.

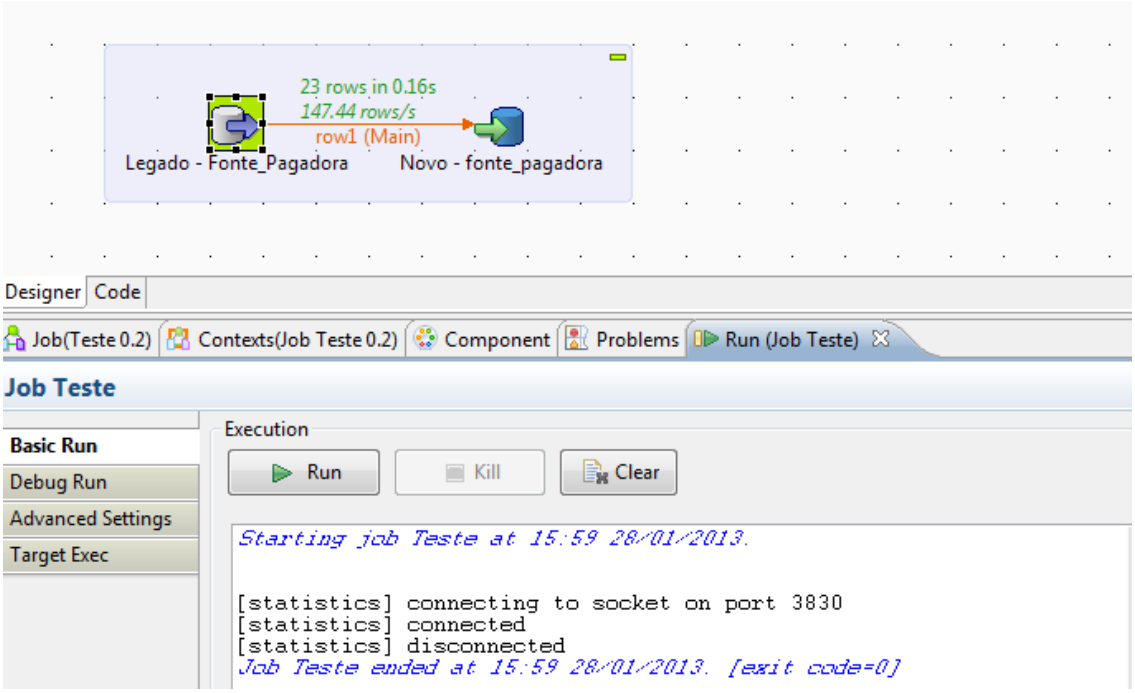


Figura 20 - *Job* executado no Talend conforme a SML1

➢ **CRT 2 – Seleção e mapeamento dos campos em ordem**

- **SML2 (Simulação 2)**

O objetivo desta simulação é, ao desenvolver uma transformação ou *job*, demonstrar se nas ferramentas é necessário que a seleção dos campos de extração de dados e o mapeamento dos campos para a carga dos dados é necessário estar na mesma ordem. Os passos para simulação foram:

1. Com o *step de table input*, foi feita a leitura/seleção dos dados correspondentes à modelagem do sistema de destino através do código SQL, onde se diferencia da SML1 por ter tratado a diferença entre os tipos dos campos. Ou seja, os campos "Codi_FoPg" e "Matr_Usua" da tabela "Fonte_Pagadora" do sistema legado foi convertido de "smallint" para "char(2)" e de "char(5)" para "integer" respectivamente;

- **KETTLE:**

A figura 21 demonstra o código SQL de seleção dos dados do sistema de origem contendo conversões entre tipos de campos no qual condiz ao tratamento necessário para ocorrer a migração.

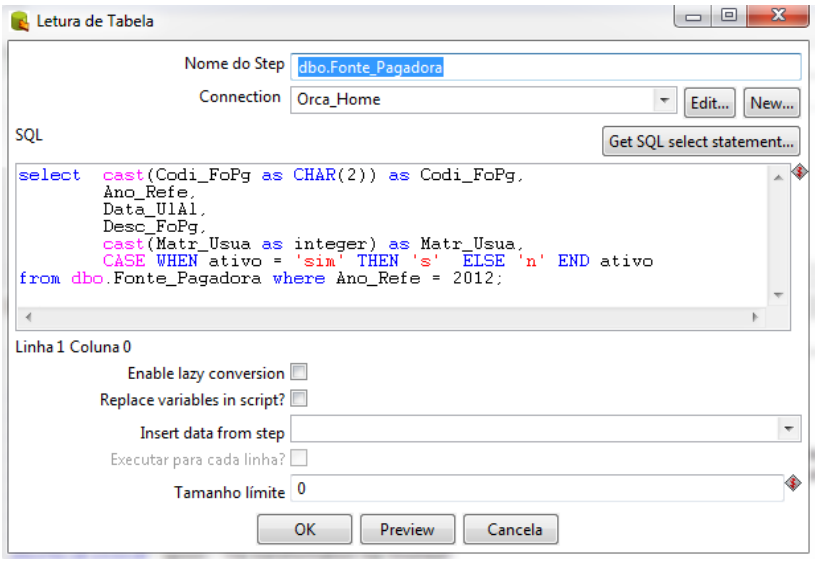


Figura 21 - *Step* para leitura/seleção dos dados no Kettle conforme a SML2

- **TALEND:**

A figura 22 mostra os detalhes do *step* de seleção de dados da tabela de origem, incluindo o código SQL responsável por essa funcionalidade. Estes dados já podem ser preparados, como mostra a figura, para serem migrados para a tabela de destino.

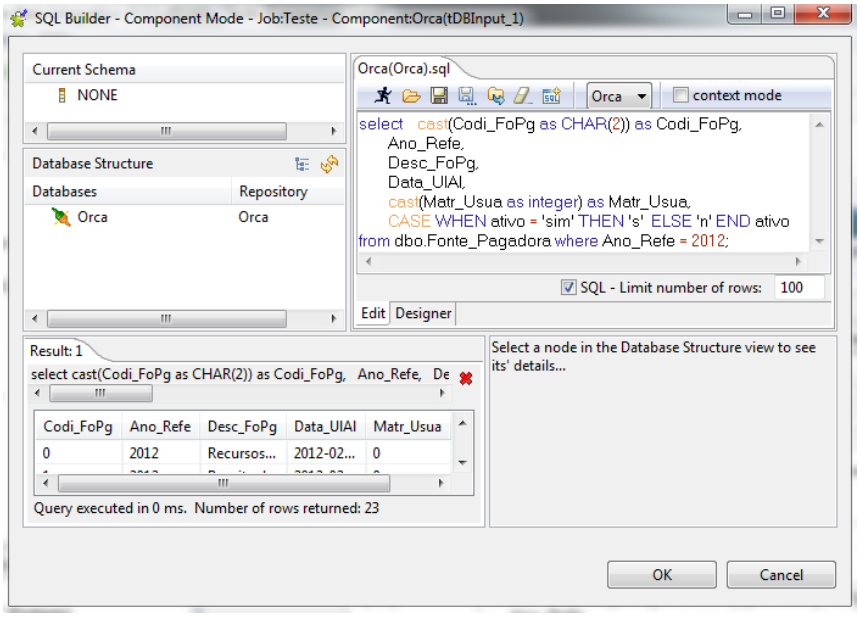


Figura 22 – Parte do *step* para leitura/seleção dos dados no Talend conforme a SML2

2. Com o *step* de inserção/atualização dos dados, foi feito o mapeamento dos campos de origem com o de destino, tendo como referência os campos utilizados no momento de extração/seleção dos dados. Para simular a diferença no desenvolvimento de transformações ou *jobs* nas ferramentas Kettle e Talend, não foi feito o mapeamento dos campos na mesma ordem dos campos contidos no código SQL de seleção dos dados;

- **KETTLE:**

A Figura 23 mostra o *step* de inserção/atualização com o mapeamento dos campos necessários para a migração dos dados.

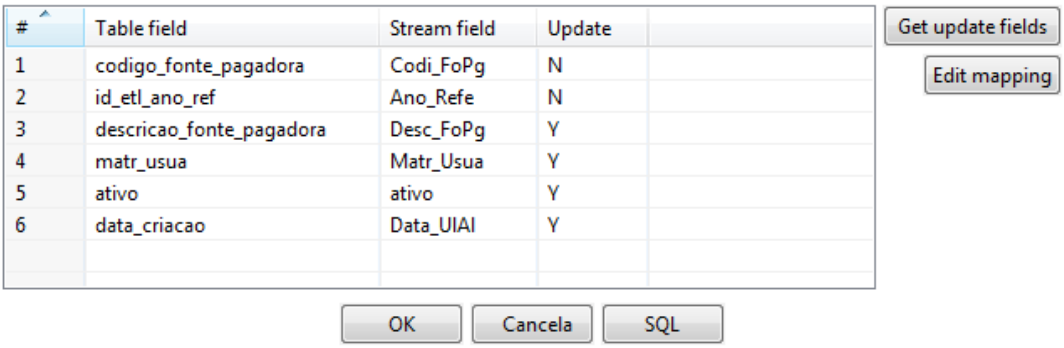


Figura 23 - *Step* de inserção/atualização com mapeamento de campos no Kettle conforme a SML2

- **TALEND:**

A figura 24 do *step* do Talend também mostra, conforme ocorrido no Kettle, o mapeamento dos campos necessários para executar a transformação ou *job*.

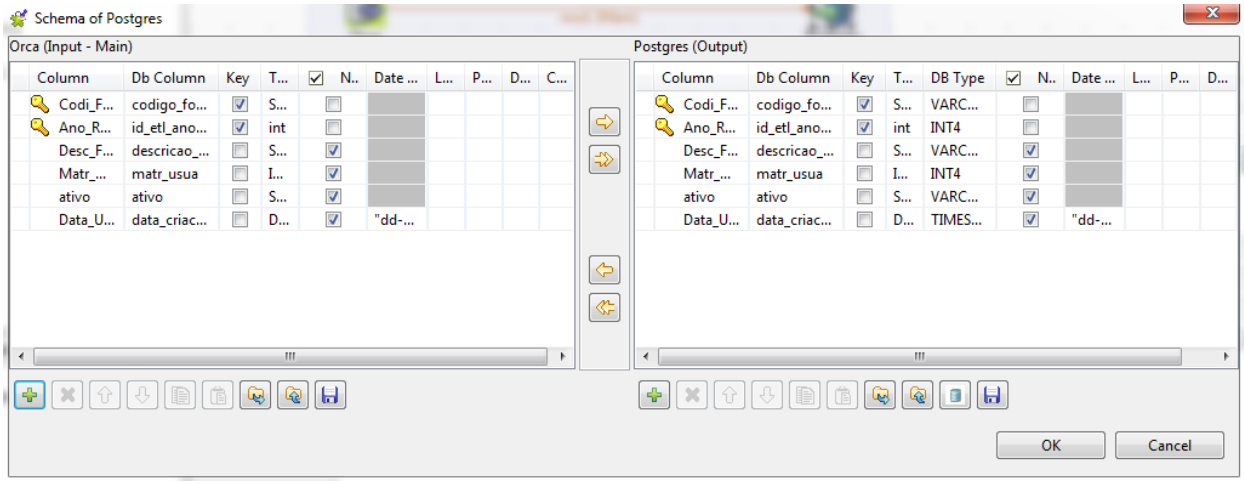


Figura 24 - Parte do *step* com o mapeamento dos campos no Talend conforme a SML2

3. Este passo mostra a transformação ou *job* sendo executada no ambiente da ferramenta, provocando assim, através da SML 2, a diferença na forma de desenvolver a migração nas duas ferramentas.

- **KETTLE:**

Conforme mostra a figura 25, o ambiente Kettle demonstra através do campo de *Logging* que a execução da transformação ocorreu com sucesso. Ou seja, mesmo que a seleção dos campos, que contém o conteúdo necessário a serem migrados, não esteja na mesma ordem do mapeamento dos campos correspondentes aos campos selecionados, o Kettle executa a transformação normalmente, pois no mapeamento ele já entende de que campo os dados irão sair e para que campo serão migrados.

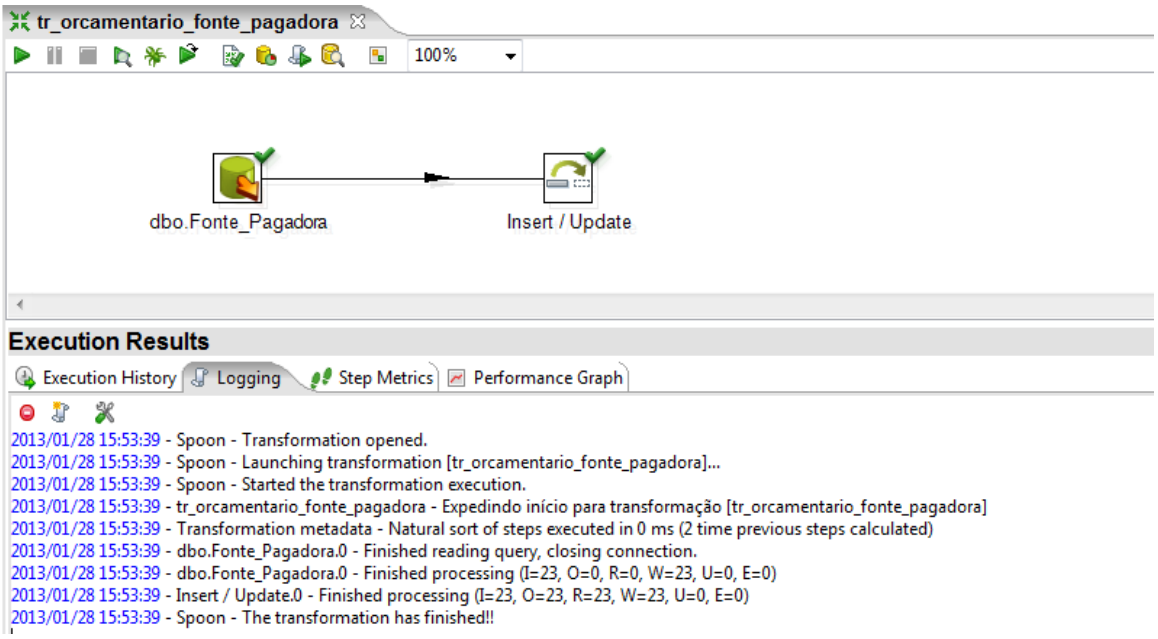


Figura 25 - Transformação executada no Kettle conforme a SML2

- **TALEND:**

A figura 26 mostra um erro ao executar o *job* no Talend. O ambiente de execução demonstra que o problema é devido a não poder inserir a “string” mostrada, no qual compõe uma data, em um campo do tipo “integer”. Mesmo não estando errado na seleção dos dados a ser migrado e na relação dos campos de origem com o de destino, o *job* causa erro por não ter feito o mapeamento dos campos na mesma ordem do código SQL que irá extrair os dados. Ou seja, na seleção dos dados da SML 2, o código SQL mostra que os primeiros dados são relacionados ao código da fonte pagadora, então no mapeamento dos campos, o primeiro a ser feito tem que ser o campo do código da fonte pagadora, pois diferente do Kettle, o Talend relaciona a extração e o mapeamento pela ordem.

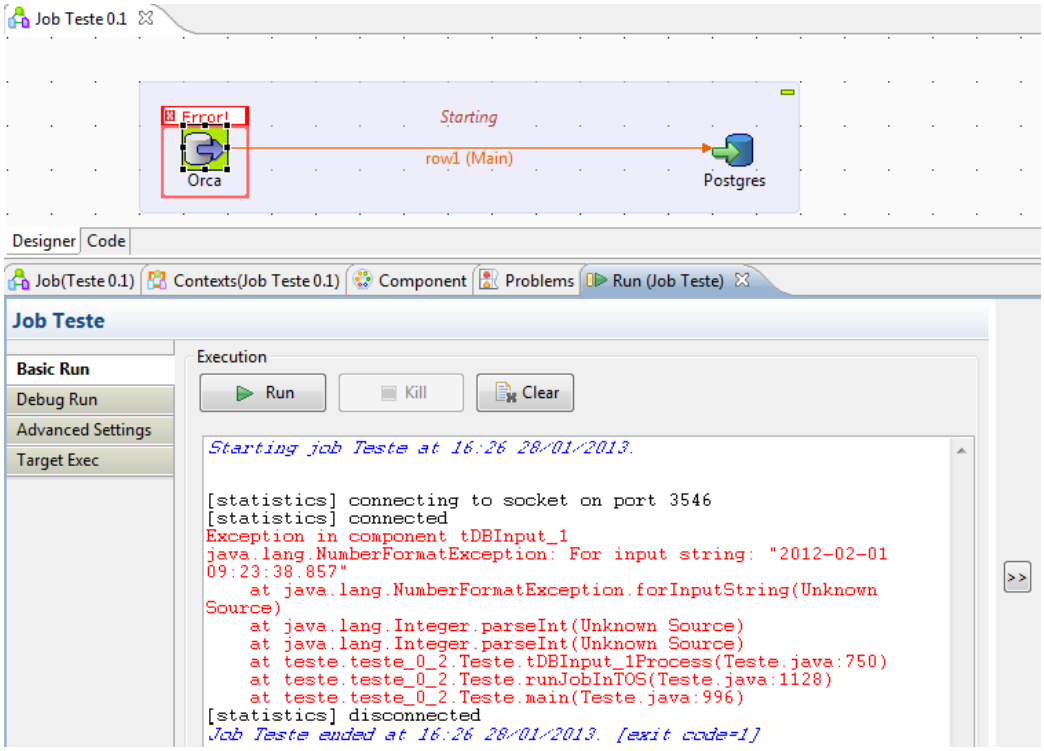


Figura 26 - *Job* executado no Talend conforme SML2

- **CRT 3 – Mapeamento da forma que dois campos sejam preenchidos por dados de um campo**

  - **SML3 (Simulação 3)**

Esta simulação tem por finalidade, ao desenvolver uma transformação ou *job*, demonstrar se as ferramentas permitem fazer o mapeamento dos campos da forma que dois campos sejam preenchidos por dados de apenas um campo. Os passos para simulação foram:

1. Com o *step de table input*, foi feita a leitura/seleção dos dados através do mesmo código SQL descritos na figura 21 do Kettle e a figura 22 do Talend, da SML 2;

2. No *step* de inserção/atualização dos dados é feito o mapeamento dos campos de origem com o de destino. Mas para essa simulação, foi necessário fazer o mapeamento de um campo novo do sistema de destino, “data_desativacao”. Então, para fazer o mapeamento foi definido que, por motivos circunstânciais, os dados do campo “Data_UlAl” do sistema

legado iria preencher os campos “data_criacao” e “data_desativacao” do sistema de destino;

- **KETTLE:**

A Figura 27 mostra parte do *step* de inserção/atualização com o mapeamento dos campos, inclusive com o campo novo “data_desativacao”, do sistema de destino, sendo preenchido também por dados do campo “Data_UlAl” do sistema legado.

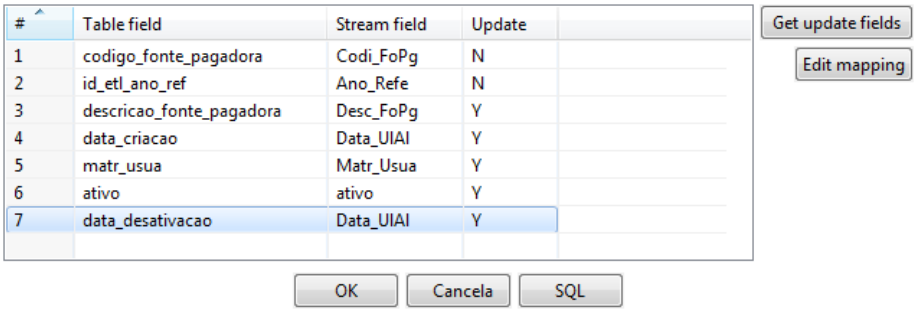


Figura 27 - Parte do *step* de inserção/atualização com mapeamento de campos no Kettle conforme a SML3

- **TALEND:**

A figura 28 mostra parte do *step* de inserção/atualização dos dados com o mapeamento dos campos. Mas, diferente da ferramenta Kettle, o Talend não dar a possibilidade de fazer o mapeamento de forma que 2 (dois) campos sejam relacionados com apenas 1 (um). A marcação vermelha, feita pela ferramenta, na figura, mostra o impedimento para descrever o nome do campo mais de uma vez.

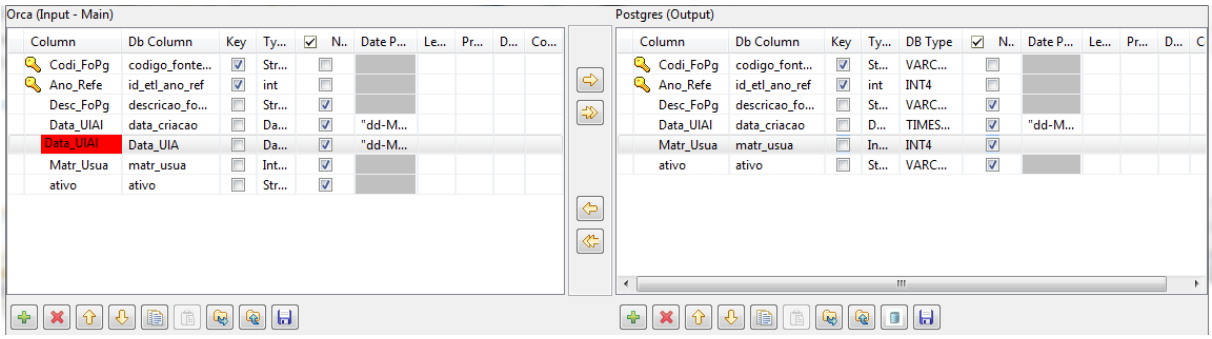


Figura 28 - Parte do *step* com o mapeamento dos campos no Talend conforme a SML3

3. Este passo mostra a transformação ou *job* sendo executada no ambiente da ferramenta Kettle, pois o Talend fica impedido no momento do passo anterior (mapeamento dos campos).

- **KETTLE:**

A figura 29 mostra a transformação sendo executada com sucesso no ambiente de desenvolvimento do Kettle. Então, diferente do Talend, o Kettle prover a possibilidade de fazer o mapeamento de campos de forma que 2 (dois) campos sejam preenchidos com dados de apenas 1 (um) campo.

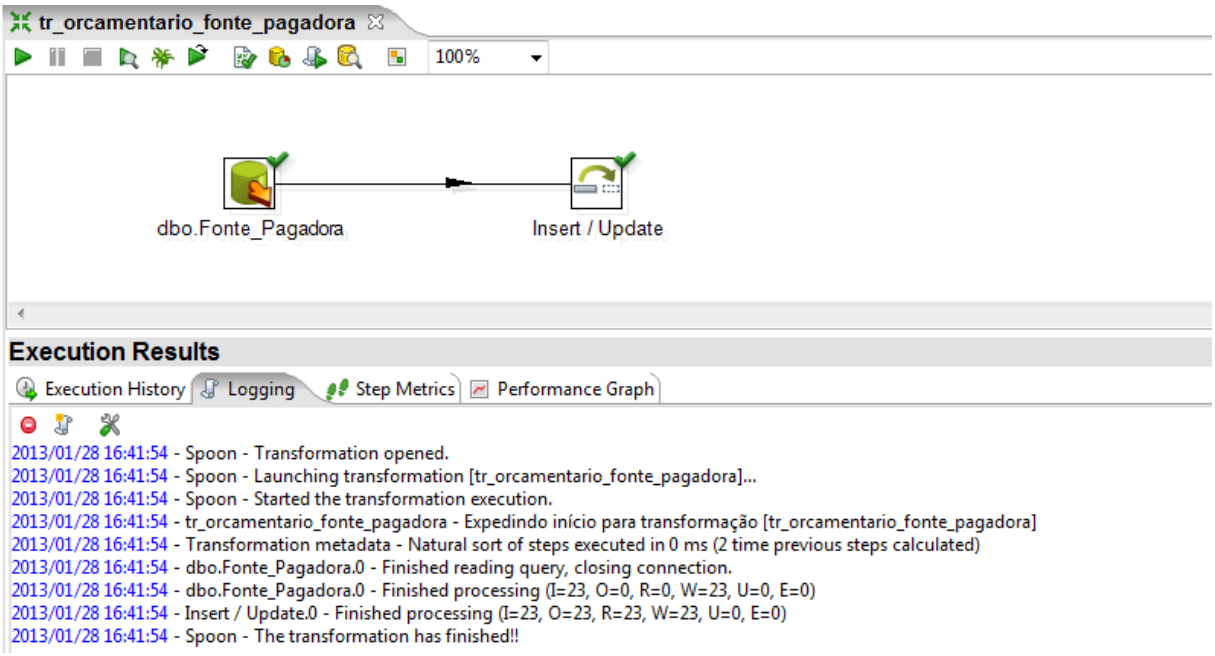


Figura 29 - Transformação executada no Kettle conforme SML3

4.4.2. Quanto as funcionalidades disponibilizadas pelas ferramentas

Para realizar a análise comparativa quanto as funcionalidades disponibilizadas pelas ferramentas foi necessário utilizar o cenário descrito na figura 14, onde ocorre todo o processo de desenvolvimento da transformação que faz a migração dos dados entre as tabelas “Fonte Pagadora” do sistema legado e novo. O objetivo deste experimento é comparar as ferramentas de acordo com os critérios relacionados a seguir.

- **CRT 4 - Extrair dados de diversas fontes**

  - **KETTLE:**

Na ferramenta Kettle, conforme mostra a figura 30, os dados podem ser extraídos de diversas fontes, como arquivo CSV, XML, arquivos textos, planilhas excel, base de dados (Oracle, MySQL, PostgreSQL, Firebird, DB2, etc), entre outros, sendo um total de 40 opções de tipos de conexões para base de dados.

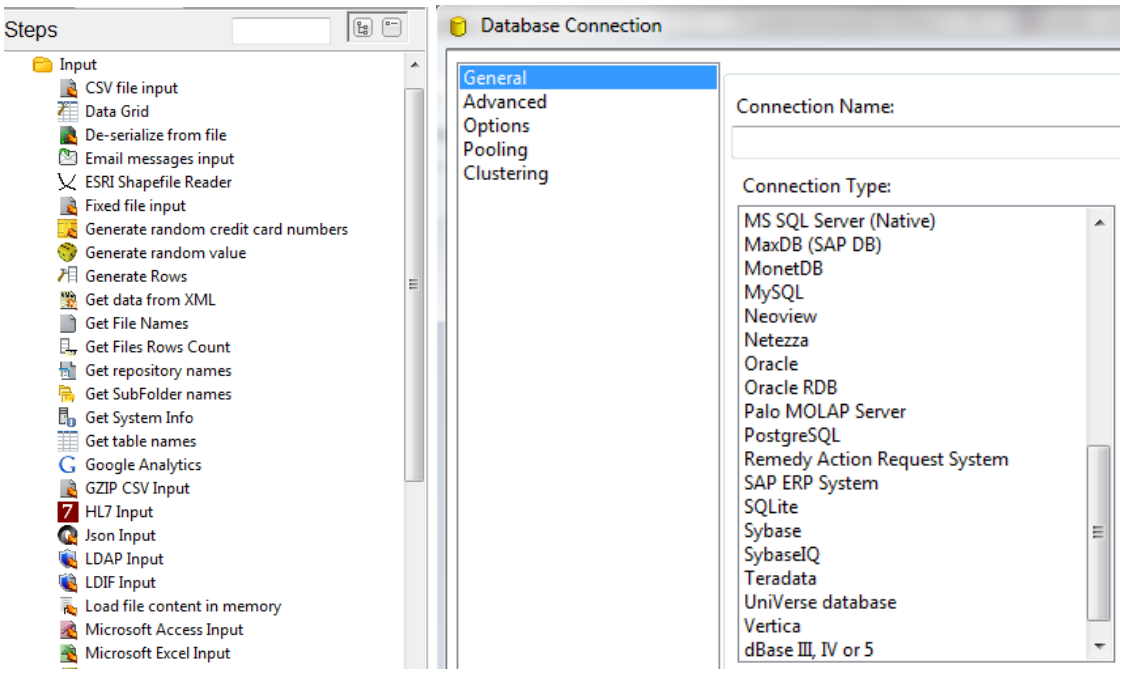


Figura 30 - Tela de *steps* de entrada e de conexões com BDs no Kettle

  - **TALEND:**

Na ferramenta Talend, conforme mostra a figura 31, os dados também tem a opção de serem extraídos de diversas fontes, como arquivo CSV, XML, arquivos textos, planilhas excel, base de dados (Oracle, MySQL, PostgreSQL, Firebird, DB2, etc), entre outros, sendo um total de 37 opções de tipos de conexões para base de dados.

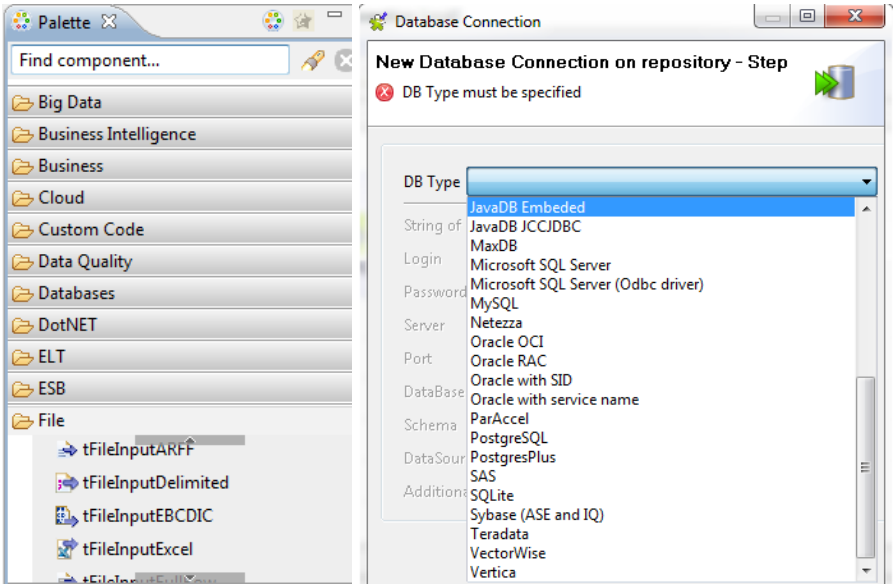


Figura 31 - Tela de *steps* de entrada e de conexões com BDs

- **CRT 5 - Disponibilizar diagramas gráficos e/ou linguagem de programação para o desenvolvimento de transformações**

  - **KETTLE:**

A ferramenta Kettle, por ter uma interface gráfica toda em *drag-and-drop* (arrastar e soltar) como mostra a figura 32, se torna bastante intuitiva. O desenvolvimento das transformações ocorre ao utilizar de diagrama gráficos (*steps*) onde compõem todo o fluxo dos dados. O que a ferramenta não dispõe é de uma linguagem de programação para que o usuário desenvolva transformações ou altere-as de maneira desejada para atender um caso específico.

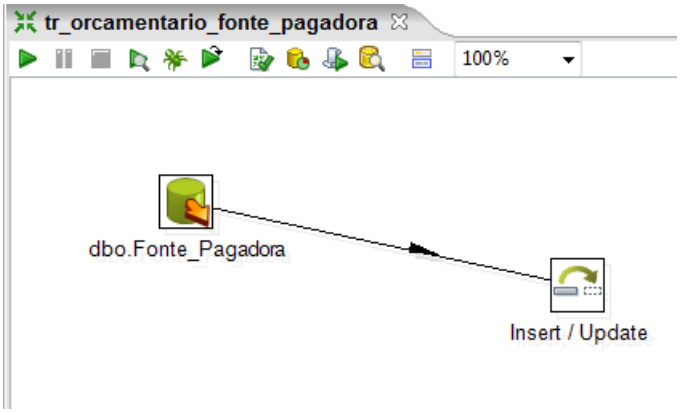


Figura 32 - Interface gráfica do Kettle

- **TALEND:**

A ferramenta Talend também dispõe de uma interface gráfica intuitiva com método *drag-and-drop* para desenvolver *jobs*. Conforme mostra a figura 33, os *jobs* são desenvolvidos através de diagramas gráficos (*steps*) e/ou de codificação, utilizando-se da linguagem de programação Java.

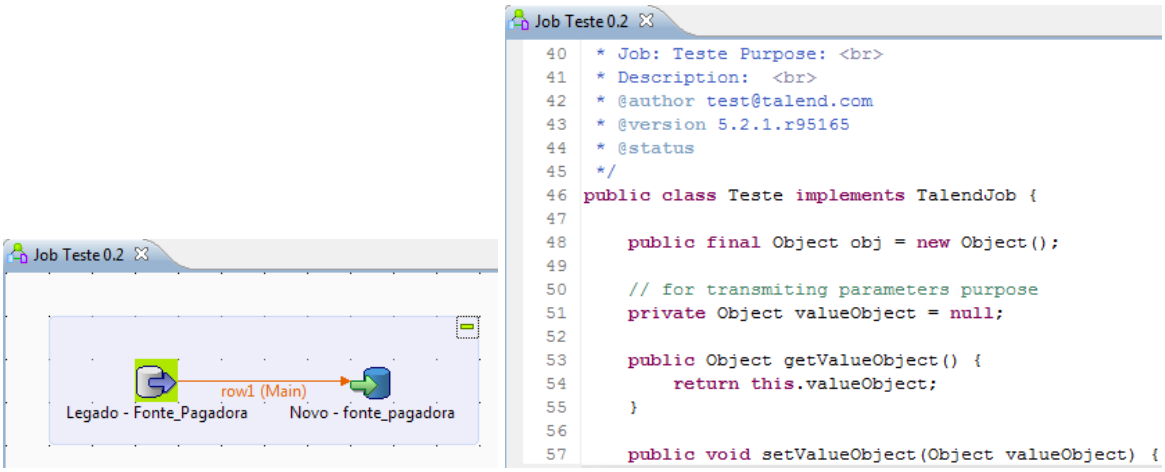


Figura 33 - Área de desenvolvimento de *jobs* por diagramas gráficos e codificação da ferramenta Talend

- **CRT 6 - Conter repositório de metadados**

- **KETTLE:**

A ferramenta Kettle não contém um repositório geral de metadados, mas sim um repositório para as transformações ou *jobs* e conexões com banco de dados.

- **TALEND:**

Conforme mostra a figura 34, a ferramenta Talend possui um repositório de metadados chamado Metadata, onde o usuário pode configurar e armazenar centralmente os *steps* responsáveis por todos os tipos de conexões (com arquivos em gerais, banco de dados, web services, etc.), para que sejam utilizados no *job* atual ou para reutilizá-los em outros *jobs*.

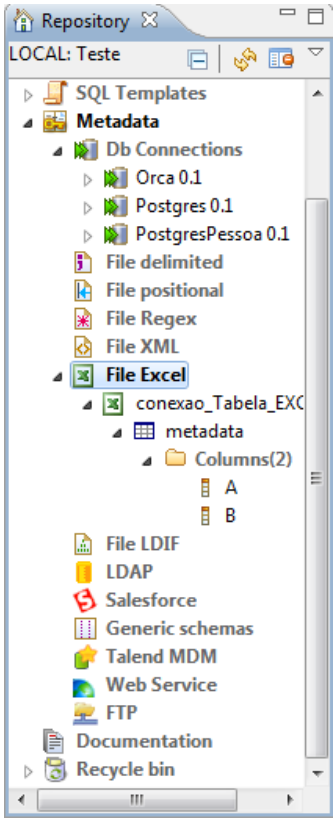


Figura 34 - Repositório de metadados no Talend

- **CRT 7 - Permitir restauração de transformações ou demais elementos**

  - **KETTLE:**

A ferramenta Kettle não permite a restauração de transformações ou demais elementos que a compõem. Caso seja deletado algo, será por definitivo, não havendo possibilidade de recuperação.

  - **TALEND:**

Conforme mostra a figura 35, a ferramenta Talend dispõe de um repositório chamado *Recycle bin* (Lixeira) onde ficam os elementos apagados e que podem serem restaurados/recuperados. Esses elementos podem ser jobs, conexões de banco de dados, planilhas, arquivos textos, entre outros.

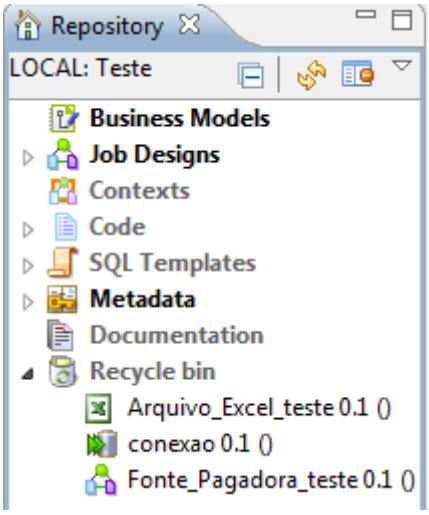
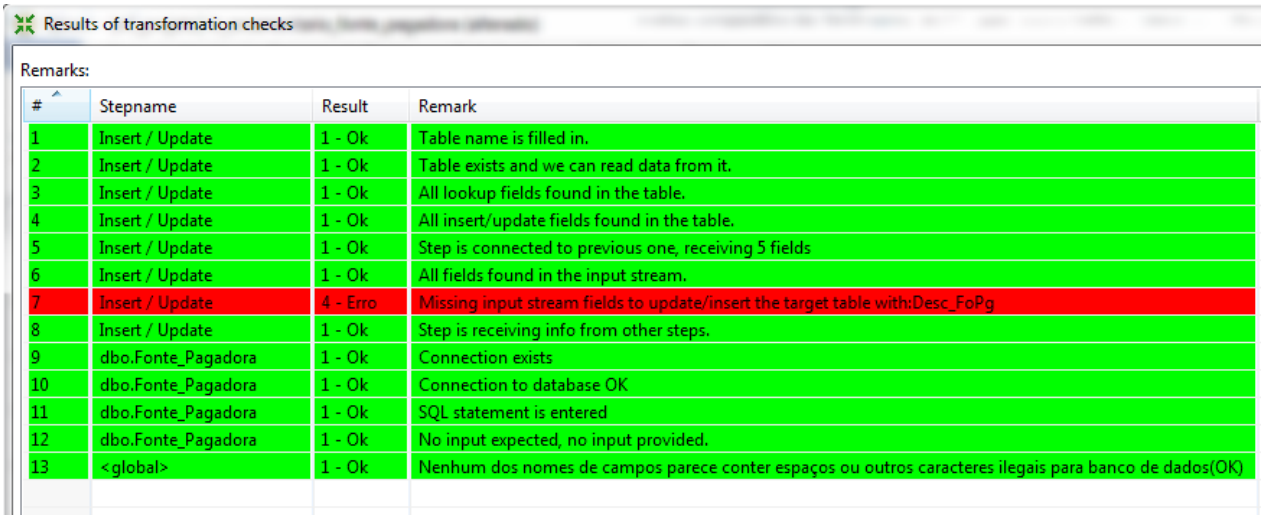


Figura 35 - Lixeira, repositório de restauração do Talend

- **CRT 8 - Permitir verificação de transformações**

  - **KETTLE:**

O Kettle, conforme mostra a figura 36, disponibiliza a funcionalidade de verificar as transformações antes de serem executadas. Estas verificações abrangem detalhes como o *step* utilizado, o resultado deste *step* – se contém erro ou não, e a observação, onde irá dar detalhes sobre o erro ou acerto dos *steps*.



#	Stepname	Result	Remark
1	Insert / Update	1 - Ok	Table name is filled in.
2	Insert / Update	1 - Ok	Table exists and we can read data from it.
3	Insert / Update	1 - Ok	All lookup fields found in the table.
4	Insert / Update	1 - Ok	All insert/update fields found in the table.
5	Insert / Update	1 - Ok	Step is connected to previous one, receiving 5 fields
6	Insert / Update	1 - Ok	All fields found in the input stream.
7	Insert / Update	4 - Erro	Missing input stream fields to update/insert the target table with:Desc_FoPg
8	Insert / Update	1 - Ok	Step is receiving info from other steps.
9	dbo.Fonte_Pagadora	1 - Ok	Connection exists
10	dbo.Fonte_Pagadora	1 - Ok	Connection to database OK
11	dbo.Fonte_Pagadora	1 - Ok	SQL statement is entered
12	dbo.Fonte_Pagadora	1 - Ok	No input expected, no input provided.
13	<global>	1 - Ok	Nenhum dos nomes de campos parece conter espaços ou outros caracteres ilegais para banco de dados(OK)

Figura 36 - Funcionalidade de verificação de transformações no Kettle

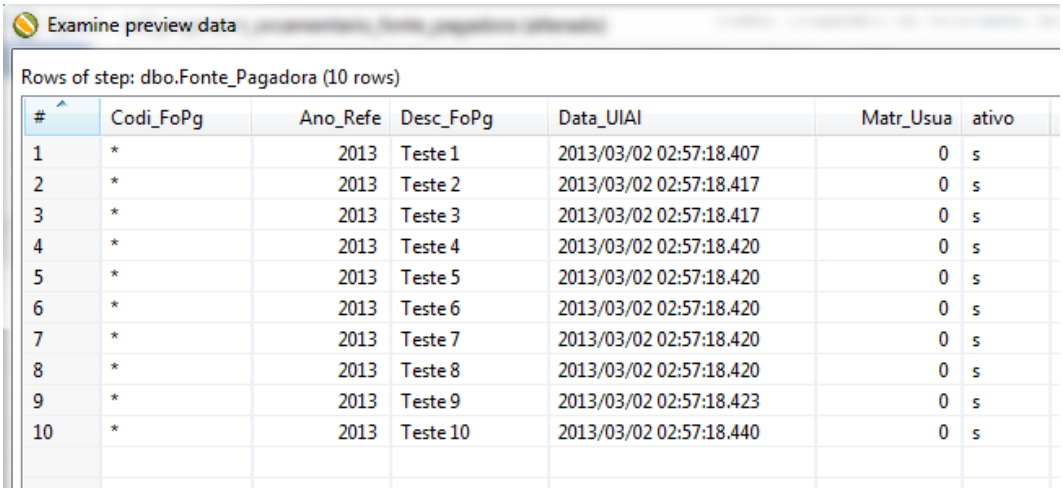
  - **TALEND:**

O Talend não dispõe de uma funcionalidade para verificar *jobs* antes de serem executados, fator que não agiliza a solução de possíveis erros.

- **CRT 9 - Permitir pré-visualização das transformações**

  - **KETTLE:**

O Kettle disponibiliza uma funcionalidade de pré-visualização dos dados que compõem todo o fluxo dos dados da transformação, sendo possível assim examiná-los sem precisar acessar a base de dados. A figura 37 mostra a pré-visualização dos dados de uma transformação teste, onde o asterisco ("*") representa os dados que geraram problema devido a modelagem da estrutura da fonte de destino, ou seja, os dados da fonte de origem eram compostos por três caracteres e o campo da fonte de destino suportava apenas dois caracteres.


Examine preview data

Rows of step: dbo.Fonte_Pagadora (10 rows)

#	Codi_FoPg	Ano_Refe	Desc_FoPg	Data_UlAl	Matr_Usua	ativo
1	*	2013	Teste 1	2013/03/02 02:57:18.407	0	s
2	*	2013	Teste 2	2013/03/02 02:57:18.417	0	s
3	*	2013	Teste 3	2013/03/02 02:57:18.417	0	s
4	*	2013	Teste 4	2013/03/02 02:57:18.420	0	s
5	*	2013	Teste 5	2013/03/02 02:57:18.420	0	s
6	*	2013	Teste 6	2013/03/02 02:57:18.420	0	s
7	*	2013	Teste 7	2013/03/02 02:57:18.420	0	s
8	*	2013	Teste 8	2013/03/02 02:57:18.420	0	s
9	*	2013	Teste 9	2013/03/02 02:57:18.423	0	s
10	*	2013	Teste 10	2013/03/02 02:57:18.440	0	s

Figura 37 - Pré-visualização de uma transformação no Kettle

  - **TALEND:**

O Talend não dispõe de uma funcionalidade para obter uma pré-visualização de jobs, não sendo possível ter uma amostragem dos dados correspondentes a migração sem ser pela base de dados.

- **CRT 10 - Permitir geração do SQL da extração dos dados através de um editor gráfico**

  - **KETTLE:**

A ferramenta Kettle não permite a geração de código SQL, da extração dos dados, através de um editor gráfico. Sendo assim, se torna importante a necessidade de o usuário conhecer a linguagem SQL.

  - **TALEND:**

O Talend além de permitir, ao usuário, extrair os dados através de codificação na linguagem SQL, também fornece a possibilidade de gerar código SQL através de um editor gráfico. A figura 38 mostra o código SQL de extração dos dados sendo gerado coforme é desenhado as tabelas nas quais são compostas pelos campos que contém o conteúdo a ser migrado. Este editor gráfico é semelhante a uma modelagem relacional.

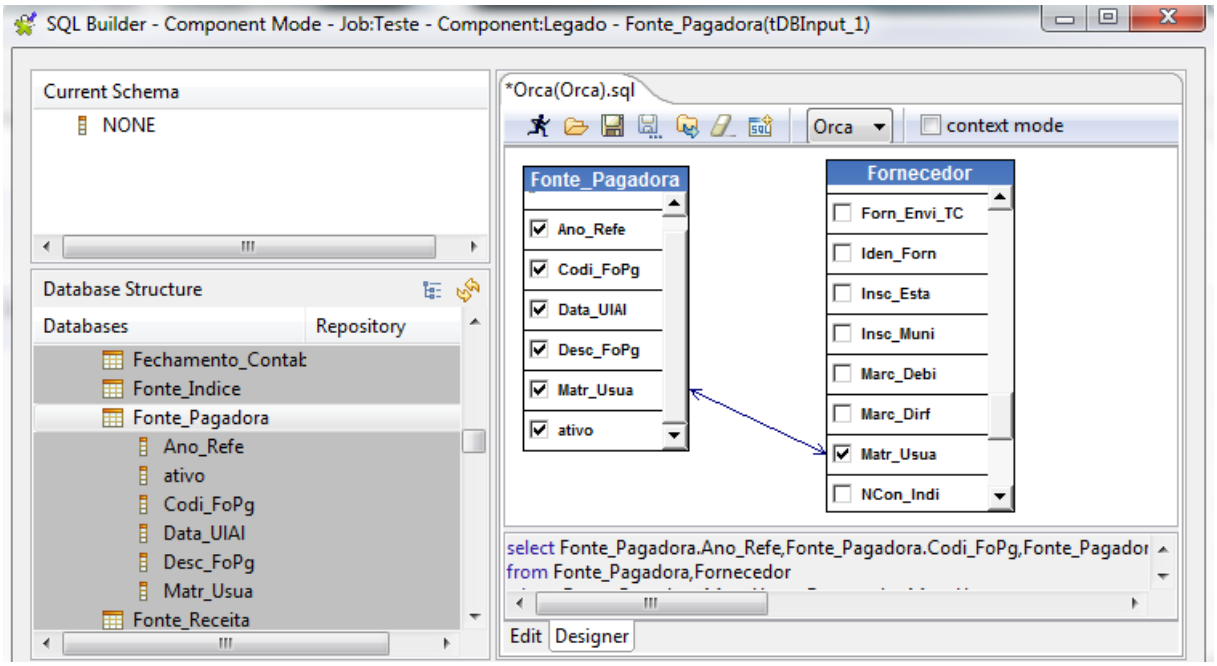


Figura 38 - Tela de geração do SQL de extração dos dados através de um editor gráfico no Talend

4.4.3. Quanto ao desempenho das transformações ou *jobs*

Com o objetivo de comparar o desempenho da execução de transformações ou *jobs* feitos nas duas ferramentas de ETL, foram desenvolvidas transformações as quais fazem a migração de dados entre as seguintes tabelas da base legada e da base nova, representadas na tabela 4:

Tabela 4 - Tabelas para avaliação de desempenho entre transformações

Tabela da Base legada (Origem)	**Tabela da Base Nova (Destino)**
Produto	Produto
Fornecedor	Pessoa
Historico_Processos	Historico_processos

Ao executar essas transformações, puderam-se obter dados como o tempo e a velocidade de linhas por segundo diretamente do ambiente das ferramentas de ETL. Além disso, em conjunto com a ferramenta Monitor de Recursos foram obtidos dados como o uso de CPU e memória através do monitoramento dos processos "javaw.exe" e "TOS_DI-Win32-x86.exe", do Kettle e Talend respectivamente.

O desenvolvimento da análise comparativa quanto ao desempenho das transformações nas duas ferramentas, tendo como base os critérios 4, 5, 6 e 7, é descrito a seguir:

- **CRT 11 - Velocidade / CRT 12 - Tempo / CRT 13 - Acesso a memória / CRT 14 - Acesso a CPU – Na ferramenta KETTLE**

    - TR1 (Transformação 1):

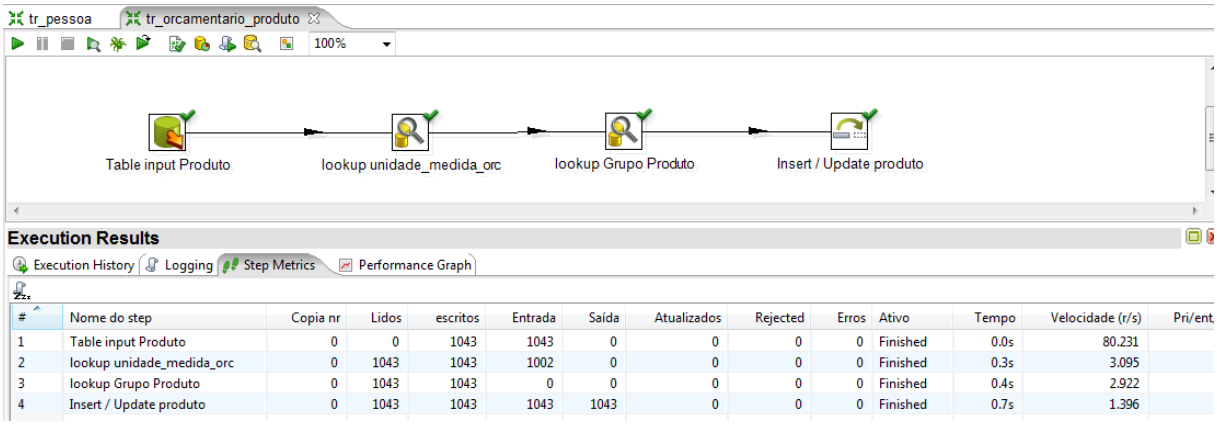
Conforme mostra a figura 39, a transformação executada no Kettle é composta pelos *steps*:

- Table input Produto – *Step* que ler/seleciona os dados da tabela "Produto" do sistema legado e adiciona-os no fluxo de dados da migração;

- Lookup unidade_medida_orc – *Step* que consulta os dados da tabela "unidade_medida_orc" para trazer valores tendo como referência uma chave comum;

- Lookup Grupo Produto – *Step* que consulta os dados da tabela "grupo_produto" para trazer valores tendo como referência uma chave comum;

- Insert/Update produto – *Step* onde teve por finalidade gravar todo o conteúdo do fluxo dos dados na tabela "Produto".



#	Nome do step	Copia nr	Lidos	escritos	Entrada	Saída	Atualizados	Rejected	Erros	Ativo	Tempo	Velocidade (r/s)	Pri/ent
1	Table input Produto	0	0	1043	1043	0	0	0	0	Finished	0.0s	80.231	
2	lookup unidade_medida_orc	0	1043	1043	1002	0	0	0	0	Finished	0.3s	3.095	
3	lookup Grupo Produto	0	1043	1043	0	0	0	0	0	Finished	0.4s	2.922	
4	Insert / Update produto	0	1043	1043	1043	1043	0	0	0	Finished	0.7s	1.396	

Figura 39 - Transformação da tabela Produto no Kettle

O ambiente do Kettle ao executar a transformação mostra uma janela de métricas e valores correspondentes para cada *step*. Esses números ajudam a analisar o desempenho da transformação em geral.

- Monitoramento da execução da TR1:

A figura 40 mostra, através de tabelas e gráficos, a ferramenta Monitor de Recursos filtrando o monitoramento pelo processo "javaw.exe", que corresponde à execução do programa Kettle, no qual foi rodado a TR1. Os dados apresentados foram selecionados de acordo com a relevância para análise comparativa, como o percentual médio do consumo de CPU, 5.40%, e a quantidade de memória consumida apenas pelo processo, 284.808 KB.

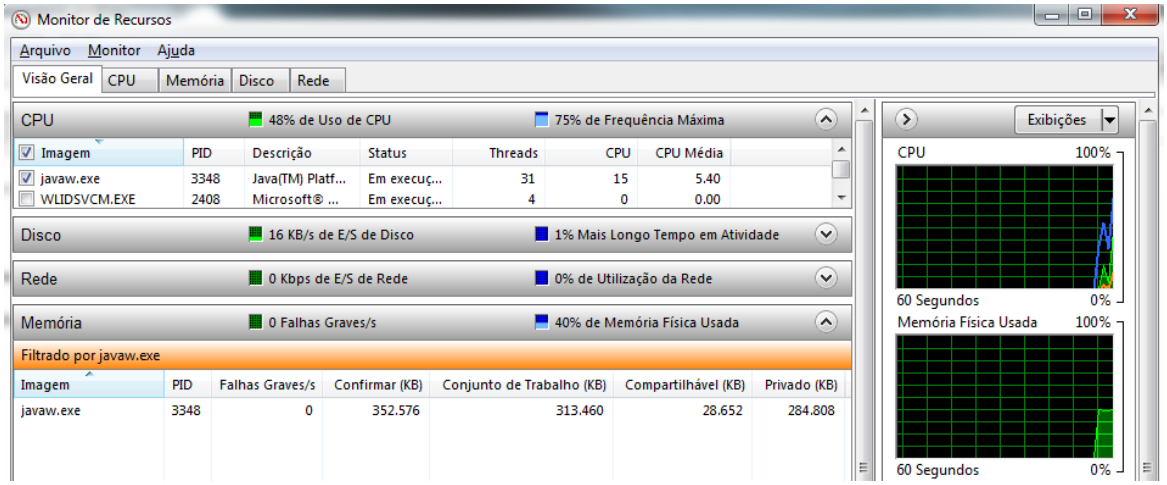


Figura 40 - Monitoramento da transformação da tabela Produto no Kettle, adaptada da ferramenta Monitor de Recursos

- TR2 (Transformação 2):

A figura 41 mostra uma transformação na qual utiliza-se de *steps* como *Table Input* e *Insert/Update* para selecionar dados da tabela "Fornecedor" do sistema legado, onde contém campos correspondentes aos campos da tabela "Pessoa" do sistema novo, e assim realizar a migração.

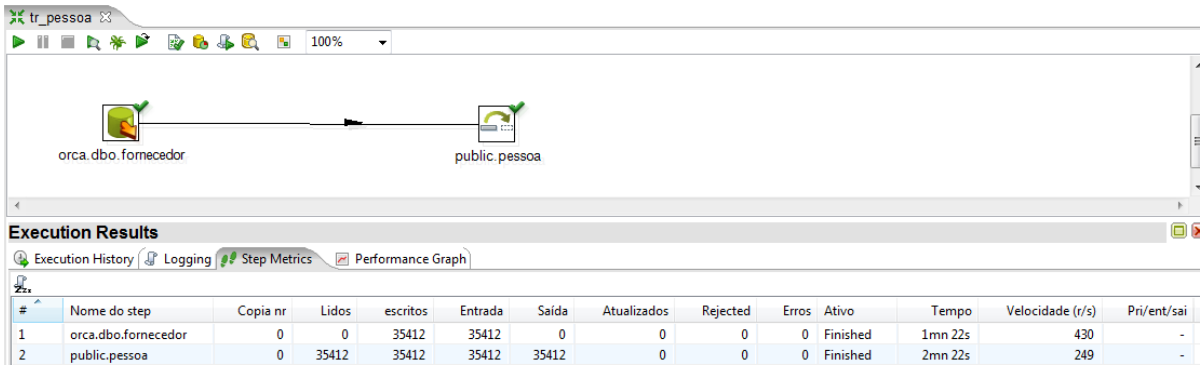


Figura 41 - Transformação entre as tabelas Fornecedor – Pessoa no Kettle

- Monitoramento da execução da TR2:

O Monitor de Recursos mostra, de acordo com a figura 42, o monitoramento da execução da TR2. Os dados apresentados foram o percentual médio do consumo da CPU, 0.88%, e a quantidade de memória acessada apenas pelo processo, 283.980 KB.

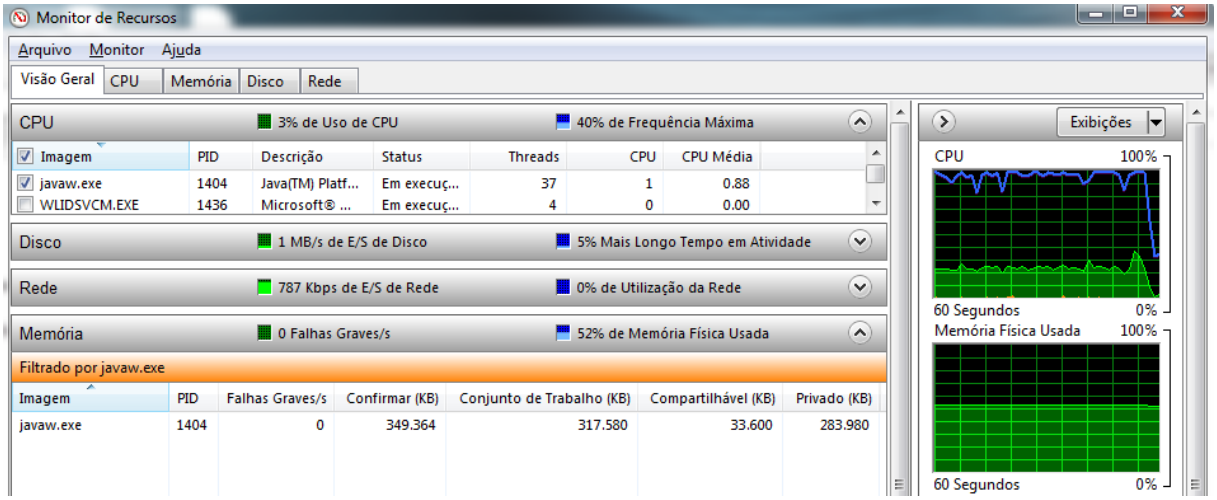


Figura 42 - Monitoramento da transformação da tabela Pessoa no Kettle, adaptada da ferramenta Monitor de Recursos

- TR3 (Transformação 3):

A figura 43 mostra uma transformação que utiliza-se de *steps* como *Table Input* e *Insert/Update* para selecionar e inserir ou atualizar dados entre as tabelas "Historico Processos" da base de dados do sistema legado para o novo, e assim realizar a migração.

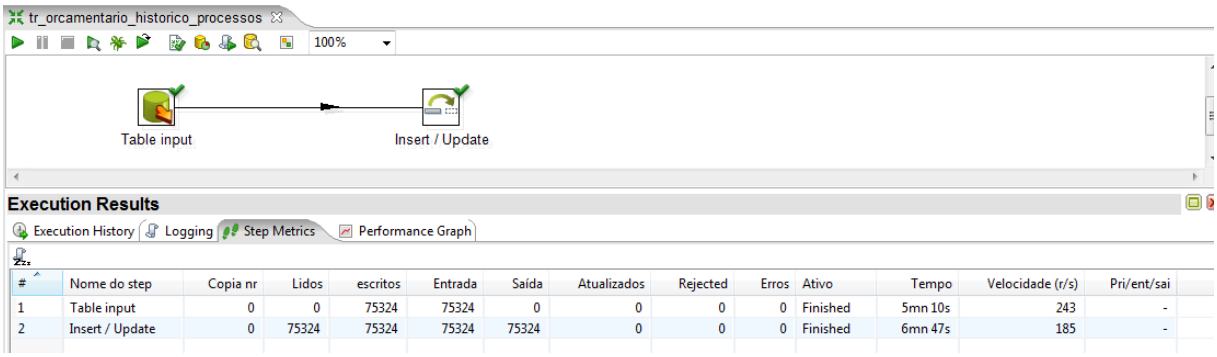


Figura 43 - Transformação da tabela Historico Processos no Kettle

- Monitoramento da execução da TR3:

A figura 44 mostra o monitoramento da execução da TR3 no qual apresenta dados como o percentual médio de consumo de CPU, 0.58%, e a quantidade de memória consumida apenas pelo processo, 271.864 KB.

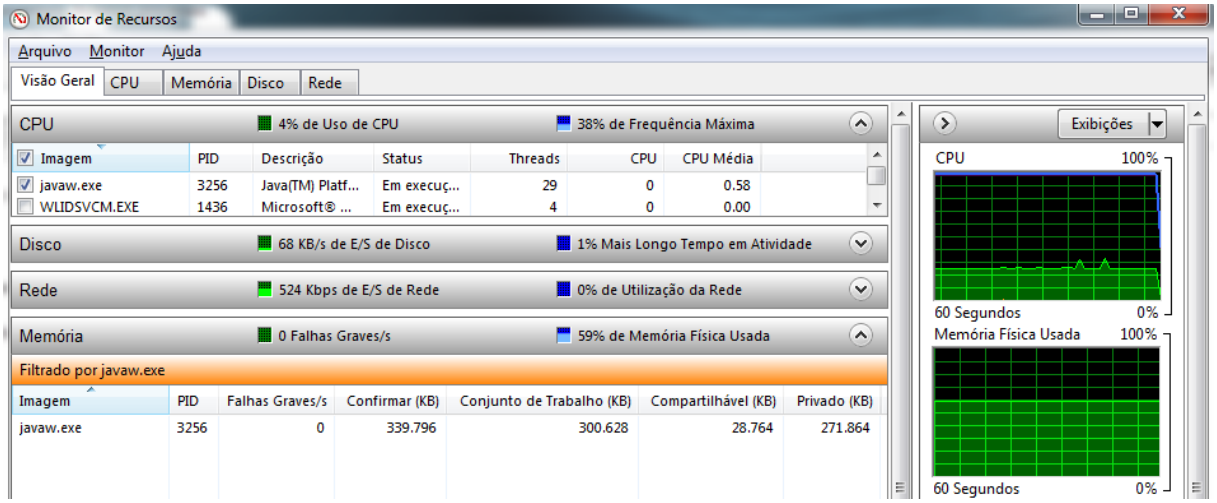


Figura 44 - Monitoramento da transformação da tabela Historico Processos no Kettle, adaptada da ferramenta Monitor de Recursos

- **CRT 11 – Velocidade / CRT 12 - Tempo / CRT 13 - Acesso a memória / CRT 14 - Acesso a CPU – Na ferramenta TALEND**

  - JB1 (*Job* 1):

A figura 45 mostra uma transformação executada no ambiente de trabalho do Talend composta pelos seguintes *steps*:

- Orca – *Step* de *Table Input* no qual também irá ler/selecionar os dados da tabela "Produto" do sistema legado e adicioná-los no fluxo de dados da migração;

- Postgres – *Steps* do tipo *Table Input* para selecionar dados da tabela "unidade_medida_orc" e "grupo_produto" para colocarem no fluxo dos dados;

- tMap_1 – *Step* no qual foi necessário para combinar vários fluxos de dados, sendo um principal (tabela "Produto") e outros secundários ("unidade_medida_orc" e "grupo_produto") até ter todos os dados necessários para colocar no fluxo dos dados;

- Postgres – *Step* final do tipo *Table Output*, de inserir ou atualizar, no qual faz o mapeamento dos campos de origem com o de destino e grava todo o conteúdo do fluxo dos dados.

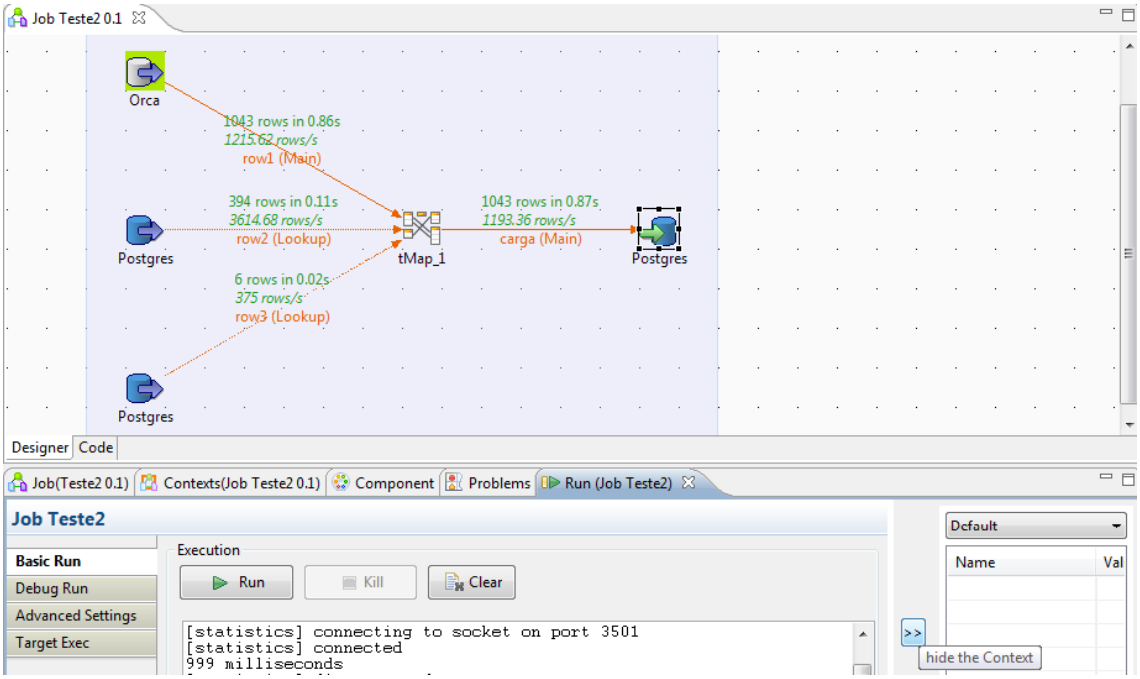


Figura 45 - *Job* da tabela Produto no Talend

O ambiente do Talend também mostra, ao executar um *job*, as métricas e valores correspondentes para cada relacionamento entre *steps*. Estas métricas são demonstradas na área de desenvolvimento dos *jobs*, onde é criado todo o fluxo dos dados através dos componentes gráficos necessários.

- Monitoramento da execução do JB1:

A figura 46 mostra a ferramenta integrada Monitor de Recursos apresentando o percentual médio de consumo de CPU, 4.98%, e a quantidade de memória acessada apenas pelo processo, 284.684 KB, ao monitorar a ferramenta Talend executando a migração de dados relacionado ao JB2.

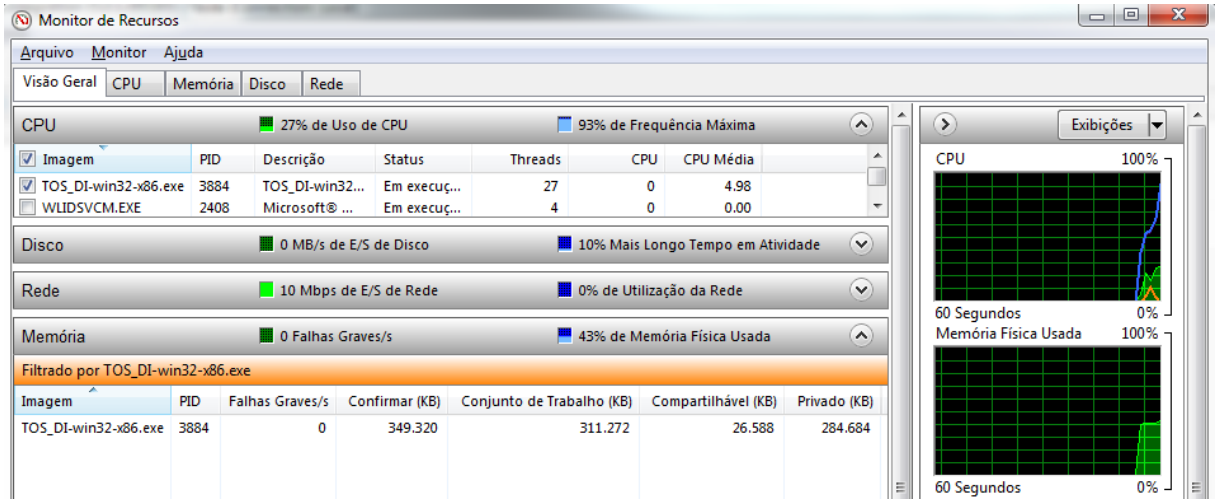


Figura 46 - Monitoramento do *job* da tabela Produto no Talend, adaptada da ferramenta Monitor de Recursos

- JB2 (*Job* 2):

A figura 47 demonstra o ambiente do Talend após a execução de um *job* no qual se utiliza de *steps* como *Table Input* e *Table Output de* inserir ou atualizar, onde seleciona dados da tabela "Fornecedor" do sistema legado e por fim mapeia os campos relevantes e migra os dados.

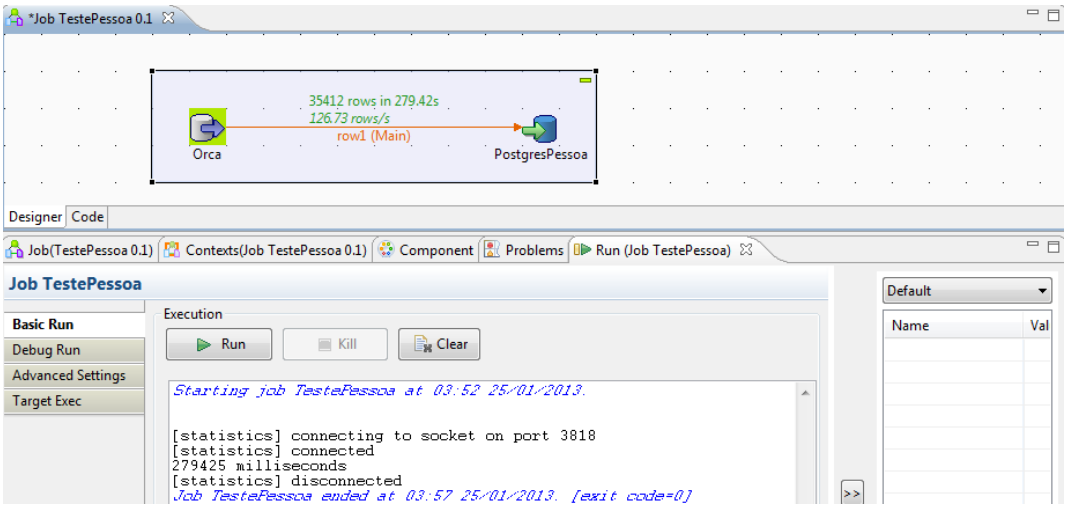


Figura 47 - *Job* entre as tabelas Fornecedor – Pessoa no Talend

- Monitoramento da execução do JB2:

A ferramenta Monitor de Recursos mostra o percentual médio de consumo de CPU, 0.11%, e a quantidade de memória acessada apenas pelo processo, 269.856 KB, ao monitorar a ferramenta Talend executando o JB2, conforme mostra a figura 48.

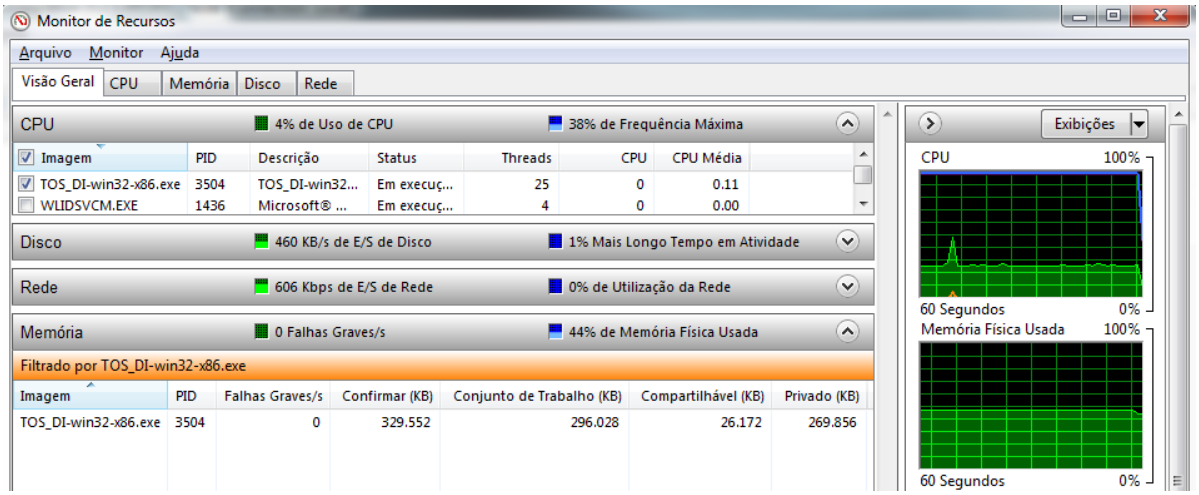


Figura 48 - Monitoramento do *job* da tabela Pessoa no Talend, adaptada da ferramenta Monitor de Recursos

- JB3 (*Job* 3):

A figura 49 mostra o ambiente do Talend após a execução de um *job* que utiliza *steps* como *Table Input* e *Table Output* de inserir ou atualizar, onde migram dados entre as tabelas "Historico Processos" da base de dados de origem para a de destino.

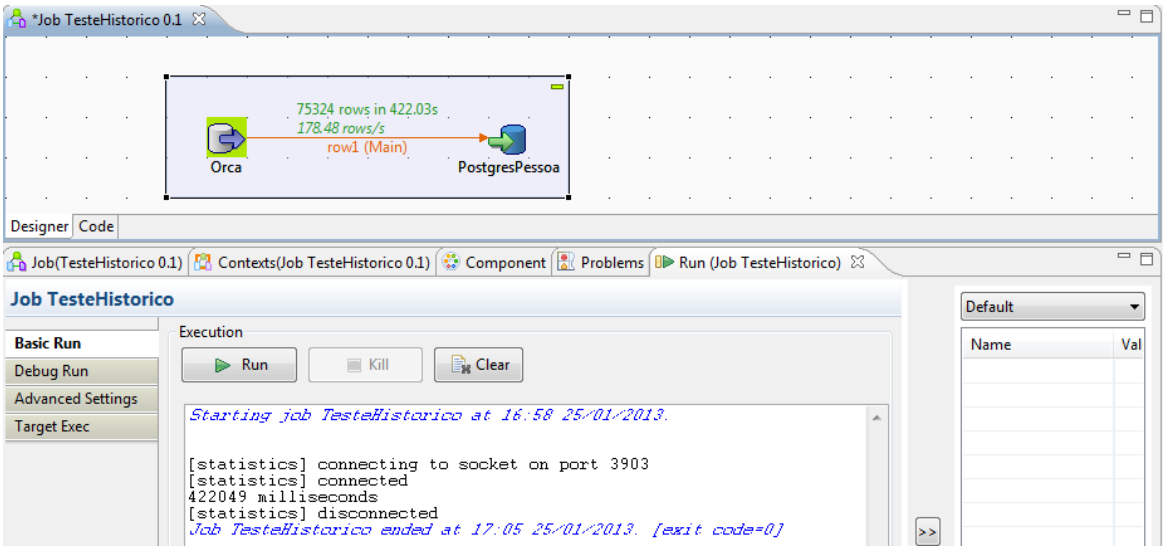


Figura 49 - *Job* da tabela Historico Processos no Talend

- Monitoramento da execução do JB3:

A figura 50 apresenta a ferramenta Monitor de Recursos monitorando a ferramenta Talend executando o JB3. Essa figura mostra o percentual médio de consumo de CPU, 0.16%, e a quantidade de memória acessada apenas pelo processo, 273.268 KB.

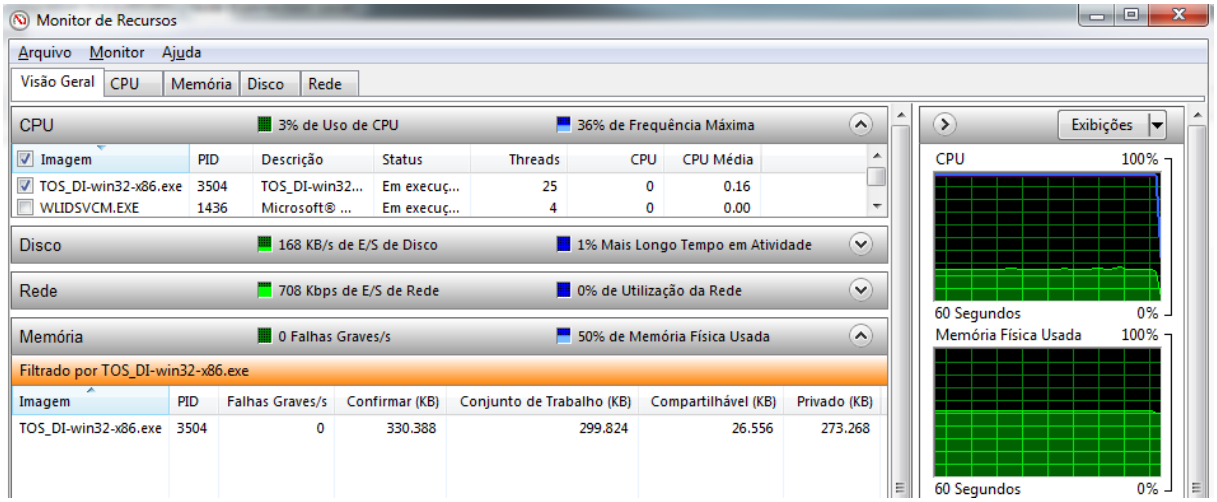


Figura 50 - Monitoramento do *job* da tabela Historico Processos no Talend, adaptada da ferramenta Monitor de Recursos

4.5. Considerações Finais do Capítulo

Neste capítulo foi feito todo o desenvolvimento da comparação das ferramentas de ETL de acordo com os critérios definidos na seção 4.3. No capítulo a seguir serão apresentados os trabalhos relacionados que auxiliaram no desenvolvimento do trabalho em questão.

5. TRABALHOS RELACIONADOS

Para auxiliar no desenvolvimento deste trabalho, foram realizadas pesquisas sobre trabalhos que tenham algum relacionamento com o tema em questão. Ao final da pesquisa, foram destacados dois trabalhos que tratavam de assuntos como: avaliação das ferramentas de ETL Talend e Kettle em projetos de DW em empresas de pequeno porte e uma comparação entre as ferramentas de ETL Apatar, Kettle e Talend.

O trabalho de graduação de Montassier Neto (2012), de título "Avaliação das ferramentas etl *open-source* Talend e Kettle para projetos de *data warehouse* em empresas de pequeno porte" objetiva desenvolver um método para avaliar as ferramentas ETL Talend e Kettle através de critérios relativos às suas características e funcionalidades. Após definir os critérios, foram realizadas comparações entre as ferramentas tendo como base o processo de ETL feito no estudo de caso prático realizado no âmbito de uma empresa de pequeno porte.

O trabalho de graduação de Santos (2009), de título "Uma comparação entre ferramentas ETL *open source*" objetiva comparar as ferramentas de ETL Apatar, Kettle e Talend. Estas comparações são feitas através dos requisitos, que foram definidos como principais em uma ferramenta de ETL, de acordo com Gonçalves (2003), e conhecimentos adquiridos de aplicações existentes, e através de testes de integridade e performance feitos em um estudo de caso.

Analisando os trabalhos apresentados e relacionando-os com o trabalho em questão, observa-se que diferentes análises comparativas entre ferramentas ETL vêm sendo abordadas. O trabalho de Montassier Neto (2012) e de Santos (2009) utilizam-se de critérios de comparação como algumas funcionalidades disponibilizadas pelas ferramentas e testes de performance. Entretanto, o teste de performance é composto apenas pela velocidade e tempo das transformações, onde podem ser considerado poucos dados para analisar a performance de qualquer ferramenta.

O trabalho em questão diferencia-se ao comparar as ferramentas de ETL Kettle e Talend através de critérios como funcionalidades não definidos nos trabalhos relacionados e outros critérios também não definidos como a forma de desenvolver transformações e o desempenho de transformações tendo como fatores a velocidade, o tempo e o uso de *hardware* (memória e CPU). O uso de *hardware* é um ponto de diferença em relação ao trabalho de Santos (2009) ao comparar as ferramentas no quesito performance, que se assemelha ao critério de desempenho do trabalho em questão.

5.1. Considerações Finais do Capítulo

Neste capítulo foram apresentados os trabalhos relacionados que auxiliaram no desenvolvimento deste trabalho. No capítulo a seguir serão apresentados os resultados obtidos ao realizar a análise comparativa entre as duas ferramentas.

6. RESULTADOS OBTIDOS

Neste capítulo serão descritos os resultados obtidos da análise comparativa feita na seção 4.4. Esses resultados são conclusões sobre comparações feitas entre as ferramentas de ETL Kettle e Talend tendo como base o cenário descrito na seção 4.4.

A análise comparativa entre as ferramentas foi realizada através de critérios definidos e divididos por categorias como a forma de desenvolver transformações ou *jobs*, o desempenho de transformações ou *jobs* e funcionalidades disponibilizadas pelas ferramentas. Os resultados obtidos e as considerações sobre os mesmos são apresentados nas seções 6.1 e 6.2.

6.1. Resultado quanto a forma de desenvolver transformações ou *jobs* e quanto as funcionalidades disponibilizadas pelas feramentas

A tabela 5 mostra os resultados da comparação conforme a forma de desenvolver as transformações ou *jobs* e as funcionalidades disponibilizadas pelas ferramentas:

Tabela 5 – Análise comparativa quanto a forma de desenvolver transformações ou *jobs* e quanto as funcionalidades disponibilizadas pelas ferramentas

Quanto a forma de desenvolver transformações ou *jobs*		
Critério de comparação	**Kettle**	**Talend**
CRT 1 - Mapeamento entre campos de tipos diferentes	Não	Sim
CRT 2 - Seleção e mapeamento dos campos em ordem	Não	Sim
CRT 3 - Mapeamento da forma que dois campos sejam preenchidos por dados de um campo	Sim	Não
Quanto as funcionalidades disponibilizadas pelas ferramentas		
Critério de comparação	**Kettle**	**Talend**
CRT 4 - Extrair dados de diversas fontes	Sim	Sim
CRT 5 - Disponibilizar diagramas gráficos e/ou linguagem de programação para o desenvolvimento de transformações	Não	Sim
CRT 6 - Conter repositório de metadados	Não	Sim
CRT 7 - Permitir restauração de transformações ou demais elementos	Não	Sim
CRT 8 - Permitir verificação de transformações	Sim	Não
CRT 9 - Permitir pré-visualização das transformações	Sim	Não
CRT 10 - Permitir geração do SQL da extração dos dados através de um editor gráfico	Não	Sim

- Considerações sobre os critérios comparados:

- CRT 1 - Mapeamento entre campos de tipos diferentes: a ferramenta Kettle não possibilita de forma alguma, ao desenvolver uma transformação, fazer mapeamento entre campos de tipos diferentes, preservando assim a consistência dos dados. Porém, no Talend, pode-se desenvolver um *job* fazendo o mapeamento entre campos de tipos diferentes;

- CRT 2 - Seleção e mapeamento dos campos em ordem: o Kettle se torna mais flexível que o Talend quando, ao desenvolver uma transformação, dar a possibilidade de o usuário fazer a seleção dos dados a serem migrados e depois fazer o mapeamento dos campos em qualquer ordem, contudo que esteja certo. No Talend o *job* só executa de forma correta se os campos selecionados para extração estiver na mesma ordem na seção de mapeamento;

- CRT 3 - Mapeamento da forma que dois campos sejam preenchidos por dados de um campo: a ferramenta Kettle dar a possibilidade de, ao desenvolver a transformação, o usuário fazer mapeamento entre campos de forma que (2) dois campos sejam preenchidos por dados de (1) um campo apenas. Já o Talend não dar essa possibilidade, tendo que ser selecionado o mesmo campo duas vezes e com aliáses diferentes;

- CRT 4 - Extrair dados de diversas fontes: tanto a ferramenta Kettle quanto o Talend disponibilizam extrair dados de diversos tipos de fontes como arquivos CSV, planilhas, banco de dados, entre outros. Sendo o Kettle com mais opções que o Talend para conexões com os tipos de banco de dados;

- CRT 5 - Disponibilizar diagramas gráficos e/ou linguagem de programação para o desenvolvimento de transformações: o Talend dispõe de componentes gráficos e linguagem de programação Java para desenvolver as transformações, já o Kettle disponibiliza apenas os componentes gráficos;

- CRT 6 - Conter repositório de metadados: o Talend contém um repositório de metadados para que possibilite o usuário de armazenar *steps*, de conexões de fontes de origem, de forma centralizada. Obtendo vantagem assim, para a reutilização dos *steps* em outros *jobs*. Já o Kettle não dispõe de um repositório de metadados, apenas de conexões e transformações;

- CRT 7 - Permitir restauração de transformações ou demais elementos: o Talend permite que todas as transformações ou alguns elementos que foram apagados, sejam recuperados. Isto ocorre devido a ferramenta conter uma lixeira. O Kettle não disponibiliza dessa funcionalidade, sendo assim, todos os elementos que forem apagados são definitivamente exlcuídos;

- CRT 8 - Permitir verificação de transformações: a ferramenta Kettle permite que o usuário verifique se a transformação está correta quanto a sua estrutura antes de executá-la, mostrando assim detalhes sobre cada *step* utilizado e resultados de suas respectivas configurações. A ferramenta Talend não possibilita essa funcionalidade;

- CRT 9 - Permitir pré-visualização das transformações: a ferramenta Kettle permite o usuário visualizar todos os dados a serem migrados, mostrando assim os que geram problemas para uma possível análise e correção, caso seja necessário. Já a ferramenta Talend não disponibiliza dessa funcionalidade;

- CRT 10 - Permitir geração do SQL da extração dos dados através de um editor gráfico: a ferramenta Talend permite que o usuário possa gerar o código SQL de extração dos dados através de um editor gráfico no qual se assemelha ao desenho de tabelas e campos de uma modelagem relacional. A medida que vai adicionando tabelas e selecionando os campos que contém o conteúdo necessário para migração, o código SQL vai sendo gerado. Este método ajuda usuários nois quais tem dificuldade em codificar na linguagem SQL. Já na ferramenta Kettle o usuário não tem essa opção, sendo necessário codificar em SQL.

6.2. Resultado quanto ao desempenho das transformações ou *jobs*

Nesta seção, os resultados obtidos quanto ao desempenho das transformações ou *jobs* tendo como base os critérios 11, 12, 13 e 14, foram divididos em 3 (três) avaliações, demonstrando-as de forma descrita e em tabelas comparativas e gráficos.

Estes resultados obtidos podem ser influenciados caso haja alteração de hardware, sistema operacional, complemento de componentes gráficos nas transformações ou *jobs*, entre outros.

- **Avaliação 1:**

A tabela 6 mostra o resultado obtido após o desenvolvimento da avaliação de desempenho na migração dos dados da tabela "Produto":

Tabela 6 - Resultado da primeira Avaliação quanto ao Desempenho

Transformação	Ferramenta	Tempo	Velocidade (rows/s)	Qtd. Reg.	CPU (%)	Memória (KB)
TR1 – Tabela Produto	Kettle	0,7s	1.396	1043	5.40	284.808
JB1 – Tabela Produto	Talend	0,87s	1.193,36	1043	4.98	284.684

Tendo como base a execução da TR1 e JB1, percebe-se que o desempenho do processo de migração de dados quanto à velocidade (registros por segundo) e ao tempo (execução da transformação) na ferramenta Kettle é mais favorável que o Talend. Chega-se a essa conclusão observando, a velocidade média maior, de 1.396 registros por segundo, e um tempo médio de execução menor, de 0.7 segundos na ferramenta Kettle. A diferença é de aproximadamente 14,61% entre as ferramentas no tocante a velocidade. Já em função de uso de *hardware*, a ferramenta Talend tem mais vantagem por ter apenas um uso percentual médio de apenas 4.98% de CPU e 284.684 KB de memória.

- **Avaliação 2:**

A tabela 7 mostra o resultado obtido da próxima avaliação de desempenho feito na migração de dados da tabela "Pessoa":

Tabela 7 - Resultado da segunda Avaliação quanto ao Desempenho

Transformação	Ferramenta	Tempo	Velocidade (rows/s)	Qtd. Reg.	CPU (%)	Memória (KB)
TR2 – Tabela Pessoa	Kettle	2mn 22s	339,5	35412	0.88	283.980
JB2 – Tabela Pessoa	Talend	6mn39s	126,73	35412	0.11	269.856

Tendo como base a execução da TR2 e JB2, observa-se que o desempenho, no quesito tempo de execução da transformação e velocidade de registros por segundo, a ferramenta Kettle supera o Talend, pois obteve um tempo médio bem menor, de 2mn 22s, e uma velocidade média maior, de 339,5 r/s. A diferença é de aproximadamente 69,42% entre as ferramentas no tocante ao tempo e 62,67% a velocidade. Já em relação ao uso de recursos do *hardware* o Talend foi melhor, pois precisou de apenas uma média de 0.11% de CPU e 269.856 KB de memória. Percebe-se também que a diferença entre as ferramentas é de 87,5% em relação ao uso de CPU.

- **Avaliação 3:**

Na tabela 8 a seguir, mostra-se o resultado da avaliação de desempenho das ferramentas ETL ao migrar os dados da tabela “Historico Processos”:

Tabela 8 - Resultado da terceira Avaliação quanto ao Desempenho

Transformação	Ferramenta	Tempo	Velocidade (rows/s)	Qtd. Reg.	CPU (%)	Memória (KB)
TR3 – Tabela Historico Processos	Kettle	6mn 47s	214	75324	0.58	271.864
JB3 – Tabela Historico Processos	Talend	7mn 02s	178,48	75324	0.16	273.268

Nesta última avaliação de desempenho no quesito tempo e velocidade, a ferramenta Kettle novamente obtêm mais vantagem sobre o Talend, com um tempo menor de execução da transformação, de 6mn 47s, e uma velocidade maior de 214 registros por segundo. Obser-

va-se que a diferença entre as duas ferramentas é de aproximadamente 16,6% no tocante a velocidade. Quanto ao uso de *hardware*, o Talend ganhou no critério que corresponde ao menor uso de CPU, média de 0.16%, e perdeu no uso de memória por utilizar mais, 273.268 KB contra 271.864 KB do Kettle. Percebe-se também a diferença de aproximadamente 72,41% no uso de CPU entre as duas ferramentas.

6.2.1. Resultados Gerais das Avaliações

Para ter uma melhor visualização e em uma perspectiva geral contendo as três avaliações relacionadas, mostram-se então os gráficos 1 e 2 dos resultados subdivididos por critérios.

- **CRT 11 - Velocidade (registros por segundo) / CRT 12 - Tempo (minuto)**

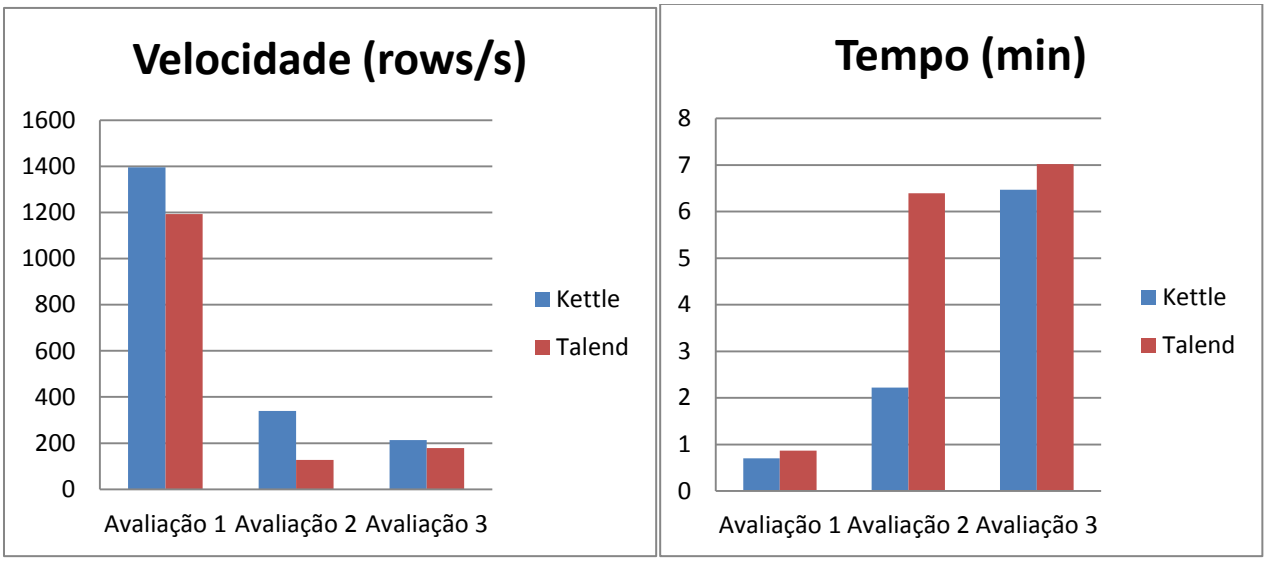


Gráfico 1 - Análise comparativa quanto ao desempenho, tendo como base a velocidade e tempo, respectivamente

- **CRT 13 – Acesso a memória (KB) / CRT 14 - Acesso a CPU (%)**

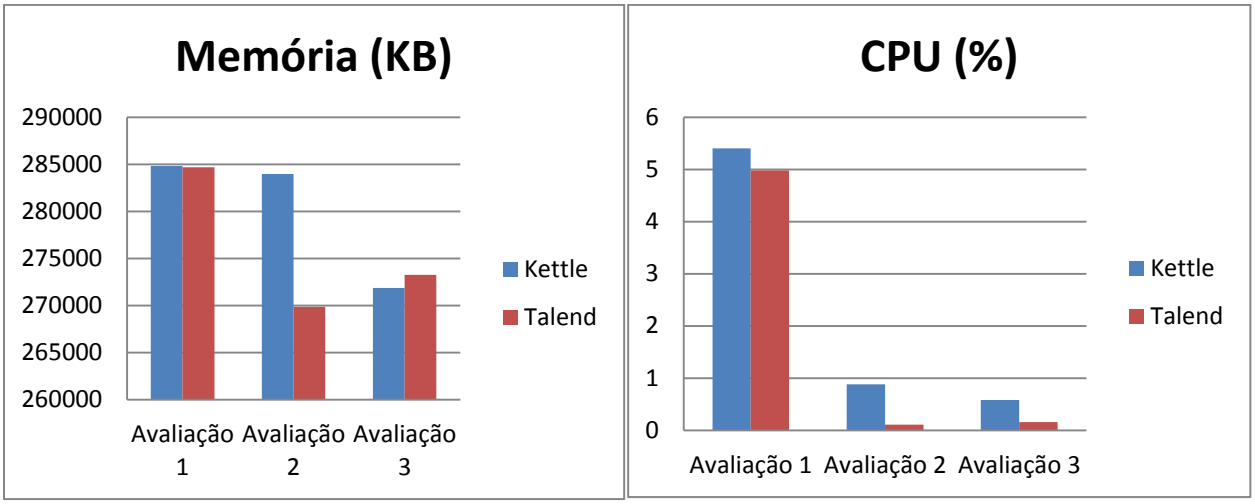


Gráfico 2 - Análise comparativa quanto ao desempenho, tendo como base o acesso a memória e CPU, respectivamente

6.3. Considerações Finais do Capítulo

Neste capítulo foram apresentados os resultados obtidos após realizar a análise comparativa entre as duas ferramentas de ETL. Esta análise comparativa teve como base alguns critérios de comparação, que foram subdivididos em categorias como: a forma de desenvolver transformações ou *jobs*, as funcionalidades disponibilizadas pelas ferramentas e o desempenho das transformações ou *jobs* nas duas ferramentas.

A ferramenta Kettle apresentou resultados superiores ao Talend quanto a flexibilidade de desenvolver as transformações ou jobs e quanto a velocidade e tempo para realizar as migrações de dados. Entretanto, o Talend se mostrou melhor ao determinar um menor acesso a memória e CPU para realizar as migrações de dados. Ao comparar as funcionalidades disponibilizadas pelas duas ferramentas, conclui-se que as mesmas possuem características diferentes para determinadas situações.

Seguem, no próximo capítulo, as considerações finais sobre este trabalho de conclusão de curso.

7. CONSIDERAÇÕES FINAIS

O presente trabalho teve como objetivo realizar uma análise comparativa entre as ferramentas de ETL Kettle e Talend tendo como base critérios estabelecidos e subdivididos em três categorias, como: a forma de desenvolver transformações ou *jobs*, as funcionalidades disponibilizadas pelas ferramentas e o desempenho das transformações ou *jobs*.

Para efetuar a análise comparativa entre as ferramentas, foi preciso estabelecer previamente alguns critérios de comparação. Estes critérios foram definidos através de dois métodos: uma pesquisa bibliográfica no qual retornou, conforme Gonçalves (2003), os principais requisitos de uma ferramenta ETL, servindo assim como alguns critérios de comparação, e através da experiência pessoal do uso das funcionalidades das ferramentas, onde também serviu como uma oportunidade para definir alguns critérios importantes.

Após o término das comparações realizadas e a análise dos resultados, foi possível identificar que as ferramentas possuem diferenças relevantes quanto a forma de desenvolver procedimentos de migração de dados, resultando assim, uma maior flexibilidade a ferramenta Kettle em relação ao Talend.

Observou-se também que as ferramentas, em termos de funcionalidades disponibilizadas, possuem características distintas. Entretanto, as funcionalidades relacionadas aos passos básicos em um processo de ETL existem em ambas.

Por fim, também foi possível identificar que a ferramenta Kettle obtém um melhor desempenho, comparado ao Talend, quando se trata de tempo e velocidade para migrar dados. Destaca-se o quesito velocidade, onde, em uma situação específica, obtivemos resultados com mais de 60% de diferença entre as duas ferramentas. Por outro lado, observa-se que a ferramenta Talend, no quesito uso de *hardware*, obtém um melhor desempenho. Ao destacar a diferença maior, as duas ferramentas chegam a diferenciar-se em mais de 80% quanto ao uso de CPU, tendo menos uso a ferramenta Talend.

Em trabalhos futuros, pretende-se realizar a mesma análise comparativa mas em diferentes situações, como:

- Comparar com novas características (máquina com configuração diferente, base de dados divergentes, plataforma Linux);
- Comparar com outras ferramentas livres (Apatar, CloverETL, ActiveWarehouse, etc.);
- Comparar com ferramentas proprietárias (Datastage, PowerCenter, OWB, etc.).

REFERÊNCIAS BIBLIOGRÁFICAS

ABREU, Fábio Silva Gomes da Gama e. **Desmistificando o Conceito de Etl**. Revista de Sistemas de Informação da FSMA n. 2, 2008. Disponível em: http://www.fsma.edu.br/si/Artigos/FSMA_SI_2008_2_Principal_1.html Acesso em: 08 de dez. 2012.

ABREU, Fábio Silva Gomes da Gama e. **Estudo de usabilidade do software Talend Open Studio como ferramenta padrão para ETL dos sistemas-clientes da aplicação PostGeoOlap**. Monografia de Graduação em Sistemas de Informação – Faculdade Salesiana Maria Auxiliadora, Macaé, 2007.

ALMEIDA, Luís Fernando Barbosa. **A Metodologia de Disseminação da Informação Geográfica e os Metadados**. Tese de doutorado. Centro de Ciências Matemáticas e da Natureza – UFRJ. Rio de Janeiro, 1999.

BARBIERI, Carlos. **Business Intelligence: Modelagem e Tecnologia**. Rio de Janeiro: Axcel Books, 2001.

BOUMAN, R. Roland. **Pentaho Data Integration: Kettle turns data into business**. Roland Bouman's blog. 2009. Disponível em: http://rpbouman.blogspot.com.br/2006/06/pentaho-data-integration-kettle-turns.html. Acesso em: 09 de jan. 2013.

CIELO, Ivã. **ETL – Extração, Transformação e Carga de Dados**. Disponível em: http://www.datawarehouse.inf.br/etl.htm. Acesso em: 06 de jan. 2013.

DILLY, Ruth. **Data Mining - an introduction**. Parallel Computer Centre - Queen's University of Belfast. Dezembro, 1995. Disponível em: http://www.pcc.qub.ac.uk/tec/coursers/ data-mining/stu_notes/dm_book_2.html. Acesso em: 04 de mar. 2013.

FAYYAD, U. M; PIATETSKY-SHAPIRO, G; SMYTH, P. **From Data Mining to Knowledge Discovery**. American Association for Artificial Intelligence. 1996a.

FAYYAD, U. M; PIATETSKY-SHAPIRO, G; SMYTH, P; UTHURUSAMY, R. **Advances in Knowledge Discovery & Data Mining**. 1 ed. American Association for Artificial Intelligence, Menlo Park, Califórnia, 1996b.

GONÇALVES, Marcio. **Extração de dados para Data Warehouse**. Rio de Janeiro: Axcel Books, 2003.

GONÇALVES, Alexandre Leopoldo. **Utilização de técnicas de mineração de dados na análise dos grupos de pesquisa no Brasil**. Florianópolis, 2000. Dissertação (Mestrado em Engenharia de Produção) - Engenharia de Produção e Sistemas, Universidade Federal de Santa Catarina.

IBL. **Extração, Transformação e Carga**. 2003. Disponível em: http://www.infobras.com.br/portugues/produtos_conceito_etl.asp. Acesso em: 08 de dez. 2012

IBM. **Intelligent Miner**. 1997. Disponível em: http://scanner-group.mit.edu/DATAMINING/Datamining/ibm.html. Acesso em: 04 de mar. 2013.

INMON, W.H. **Como Construir o Data Warehouse**. Rio de Janeiro: Campus, 1997.

INMON, Willian H; TERDEMAN, R. H; IMHOFF, Claudia. **Data Warehousing: como transformar informações em oportunidades de negócios**. São Paulo, SP. Berkely, 2001. 266 p.

KOCSKA, Fernanda; DINIZ, Tatiana Silva; BARATELLA, Michele; BARBOSA, Ronaldo Rodrigues; CARVALHO, Adriano Gueiros; GOULART, Elias Estavão. **Tecnologias para informações gerenciais: um estudo de caso**. Santo André, SP. 2009.

LU, H; SETIONO, R; LIU, H. **NeuroRule: A Connectionist Approach to Data Mining**. In: 21st. VERY LARGE DATA BASE CONFERENCE, (CD), Zurich, Switzerland, Proceedings. 1995.

MACHADO, Felipe N. R. **Projeto de Data Warehouse: uma visão multidimensional**. São

Paulo: Érica, 2000.

MONTASSIER NETO, Trajano C. **Avaliação das ferramentas ETL open-source Talend e Kettle para projetos de data warehouse em empresas de pequeno porte**. Monografia de Graduação em Sistemas de Informação – União Metropolitana de Educação e Cultura, Lauro de Freitas, 2012.

MOSS, L. **Data Cleasing: A Dichotomy of Data Warehouse?** DM Review Magazine, February. 1998.

PENTAHO. **Open Source Business Intelligence**. Disponível em: http://kettle.pentaho.com. Acesso em: 12 de jan. 2013.

SANTOS, Pablo da C. **Uma comparação entre ferramentas ETL open source**. Monografia de Graduação em Informática – Universidade Católica do Salvador, Salvador, 2009.

SECO, Antônio; SOUZA, Cláudio de; SILVA, Edilberto Magalhães; ARAÚJO, José Luís de & SOUSA, Paulo de Tarso Costa de. **Data Warehouse, Data Mart, Data Mining**. Brasília, 2000. Artigo (Mestrado em informática) – Universidade Católica de Brasília.

TAKAOKA, Hiroo. **Aplicação de Data Warehouse no varejo in: Marketing de Relacionamento no Varejo**. São Paulo: Saint Paul Institute of Finance, 2004.

TALEND. **Talend Open Studio: User Guide**. 2009. Disponível em: http://docs.huihoo.com/talend/TalendOpenProfiler_UG_32a_EN.pdf. Acesso em: 12 de jan. 2013.

VASSILIADIS, Panos. **Conceptual Modeling for ETL processes**. 2002. Disponível em: http://www.cs.uoi.gr/~pvassil/publications/2002_DOLAP/dolap02_CR.pdf. Acesso em: 04 de mar. 2013.