

UNIVERSIDADE FEDERAL DE SERGIPE
NÚCLEO DE SISTEMAS DE INFORMAÇÃO
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

RELATÓRIO DO ESTÁGIO

SUPERVISIONADO

IGOR IURE SANTOS MACIEL
MARCUS TÚLIO DE ARAUJO MACHADO

ITABAIANA

2010

RELATÓRIO DO ESTÁGIO SUPERVISIONADO

**IGOR IURE SANTOS MACIEL
MARCUS TÚLIO DE ARAUJO MACHADO**

Relatório de estágio apresentado ao Curso de Sistemas de Informação como parte das exigências da disciplina Estágio Supervisionado, sob a orientação do Prof. Marcos Barbosa Dósea e orientação técnica do Analista de Centro de Processamento de Dados da Universidade Federal de Sergipe Diego Vasconcelos e Carmo.

**ITABAIANA – SERGIPE
JUNHO DE 2010**

RESUMO

Sistemas Integrados de Informação têm ganhado importância no tocante ao gerenciamento de controle de processos organizacionais. Com eles organizações passam a ter crescimento planejado e estruturado essenciais para o desenvolvimento das mesmas. A Universidade Federal de Sergipe (UFS) a fim de integralizar seus sistemas computacionais adquiriu no ano de 2010, através de um projeto de cooperação interinstitucional, o sistema utilizado pela Universidade Federal do Rio Grande do Norte (UFRN) para substituir um conjunto de sistemas que trabalhavam praticamente sem nenhuma integração. Entretanto, os dados existentes nesses sistemas legados precisam ser migrados para o sistema recém-adquirido, sendo este uma dos principais projetos desenvolvidos atualmente pelo Centro de Processamento de Dados da UFS. Este relatório tem por objetivo apresentar as atividades realizadas para migração dos dados existentes em um dos principais sistemas que serão implantados pela UFS: o sistema de processos. O sistema foi totalmente convertido e as dificuldades e alternativas encontradas nesse processo de conversão são aqui apresentadas.

PALAVRAS CHAVES: Sistemas Integrados, migração de dados.

SUMÁRIO

1 Introdução	6
1.1 Local do estágio.....	6
1.2 Objetivos do estágio	7
1.3 Estrutura do Relatório	7
2 Referencial Teórico.....	8
2.1 Modelagem de dados.....	8
2.2 Talend.....	9
2.3 Extração de dados	12
2.4 Postgres.....	13
3 Atividades do Estágio.....	14
3.1 Sistema legado de processo da UFS	14
3.2 Módulo protocolo - SIPAC.....	15
3.3 Migração de dados	15
3.3.1 Job Status	16
3.3.2 Job Tipo Processo.....	16
3.3.3 Job Interessado	17
3.3.4 Job Processo.....	18
3.3.5 Job Processo Interessado	20
3.3.6 Job Movimento	21
3.3.7 Job Update Processo	23
3.3.8 Job Principal.....	24
4 Conclusão	25
Referências Bibliográficas.....	26

Índice de Figuras

Figura 1 DER – SIPAC Módulo Protocolo	9
Figura 2. Modelagem de negócio – Talend	10
Figura 3. Job – Talend	11
Figura 4. Tmap – Talend	11
Figura 5. Job Status	16
Figura 6. tMap Job Tipo Processo.....	17
Figura 7. tMap Job Interessado	17
Figura 8. Job Processo	18
Figura 9. tMap_1 Job Processo	20
Figura 10. Job Movimento	21
Figura 11. tMap_1 Job Movimento	23
Figura 12. Job Update Processo.....	23
Figura 13. Job Principal	24

1 Introdução

A utilização de Sistemas Integrados de Informação destinados ao gerenciamento e controle de todos os processos organizacionais tem se mostrado essencial para o crescimento planejado e estruturado das organizações. Consciente dessa tendência e das dificuldades advindas de um contexto onde ainda não há metodologias e recursos tecnológicos disponíveis, a Universidade Federal de Sergipe (UFS) percebeu que para concretizar e controlar adequadamente o seu processo de expansão seria essencial a adoção de um sistema que proporcionasse sua evolução sustentável.

Com isso, a UFS optou por investir num sistema que já provesse a grande maioria das funcionalidades requeridas. Isso foi possível através de uma parceria, através de um contrato de colaboração interinstitucional com a Universidade Federal do Rio Grande do Norte (UFRN).

O sistema adquirido já foi implantado em outras Universidades, o que demonstra certo amadurecimento do processo de implantação e notável aceitação e aplicabilidade da tecnologia no contexto das Instituições Federais de Ensino. Além disso, esse produto utiliza um vasto conjunto de funcionalidades através das mais modernas tecnologias existentes no mercado, o que o torna bastante adequado para o atendimento da demanda da instituição.

1.1. Local do estágio

O estágio foi realizado no CPD do Campus Universitário Professor Alberto Carvalho – UFS (Universidade Federal de Sergipe). Sendo utilizadas 20 (vinte) horas semanais durante a manhã, compreendendo o período de 15 de março de 2010 a 30 de julho de 2010. Foram disponibilizados dois computadores Desktop para a realização dos trabalhos propostos para estágio.

1.2. Objetivos do estágio

O objetivo geral do estágio foi a migração dos dados do sistema de processos da UFS para o novo sistema adotado pela mesma, o SIPAC - Módulo de protocolo. E ainda, adaptações necessárias para a migração das informações.

Os objetivos específicos do Estágio Supervisionado foram:

- Restaurar o Banco de Dados do Sistema de Processos da UFS utilizando o SGBD IBM - DB2[5] e criação do ambiente dos SGBD IBM - DB2 e Postgres[4];
- Restaurar o Banco de Dados do SIPAC - Módulo Protocolo utilizando o SGBD Postgres;
- Migrar os dados do Sistema de Processos da UFS para o SIPAC módulo protocolo;
- Fazer acréscimos ou alterações necessárias para a utilização do SIPAC módulo protocolo;
- Realização de testes exploratórios[1] nas funcionalidades convertidas e desenvolvidas;
- Treinamento dos usuários.

1.3. Estrutura do relatório

Este relatório está organizado como descrito a seguir. Na Seção 2 é apresentado o referencial teórico para o desenvolvimento do projeto. As atividades do estágio, bem como os resultados alcançados são detalhados na Seção 3. Finalmente na Seção 4 são discutidas as principais conclusões das atividades do estágio.

2 Referencial Teórico

Para o desenvolvimento do estágio, foi necessário realizar o estudo de alguns referencias teóricos. Nesta seção são abordados aqueles fundamentais para compreensão e realização das atividades realizadas durante o estágio.

Na Seção 2.1 uma breve introdução sobre modelagem de dados é apresentada. Na Seção 2.2 é apresentada a ferramenta Talend Open Studio utilizada para a migração dos dados. Já a Seção 2.3 discute sobre extração de dados em sistemas legados. Por fim, na Seção 2.4 é feita uma breve apresentação do SGBD PostgreSQL, utilizado no sistema adquirido pela UFS.

2.1. Modelagem de dados

Um modelo é a simplificação da realidade. Construímos modelos para obtermos uma melhor compreensão do sistema que está sendo desenvolvido. Estes ajudam a visualizar um sistema como ele é ou como desejamos que ele seja, como, também, permitem especificar a estrutura ou comportamento de um sistema. Modelamos sistemas complexos pela impossibilidade de compreendê-los em sua totalidade.

Para o desenvolvimento com Banco de Dados relacionais é essencial o uso de modelagem de dados[2]. Esta tem por objetivo representar o ambiente observado, documentar, normatizar, fornecer processos de validação e observar processos de relacionamentos entre objetos. A construção de um modelo pode ser feita em três etapas:

- **Modelo Conceitual:** Representa as regras de negócio sem limitações tecnológicas ou de implementação sendo mais adequada para o usuário, pois não há a necessidade de conhecimento técnico
- **Modelo Lógico:** Este leva em conta alguns limites impostos pela tecnologia de Banco de Dados
- **Modelo Físico:** Leva em consideração limites impostos pelo SGBD[2] (Sistema de Gerenciamento de Banco de Dados) e pelos requisitos não funcionais pelo programas de acesso a dados.

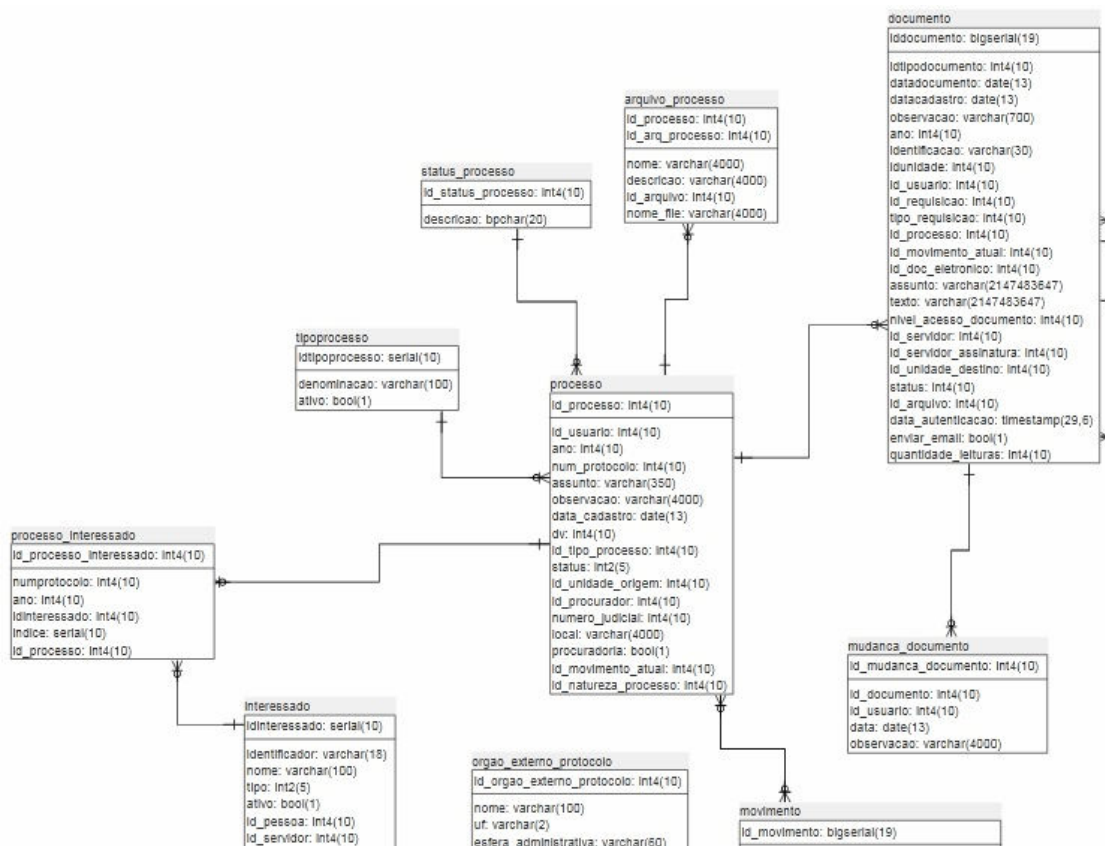


Figura 1 DER – SIPAC Módulo Protocolo

A Figura 1 mostra o DER do Módulo Protocolo do novo sistema adquirido pela UFS. Nela podemos ver que a entidade processo se relaciona com outras entidades, por exemplo, com as entidades documento, movimento, tipo_processo.

2.2. Talend

A ferramenta Talend Open Studio[3] é altamente reconhecida no mercado de integração de dados. Esta ferramenta provê recursos avançados para o aumento da produtividade do trabalho de integração de dados e, ainda, provê escalabilidade comprovada para garantir a execução desejada.

Sua modelagem de negócio é não-técnica e orientada a negócio visando uma construção usando uma biblioteca de formas e links.

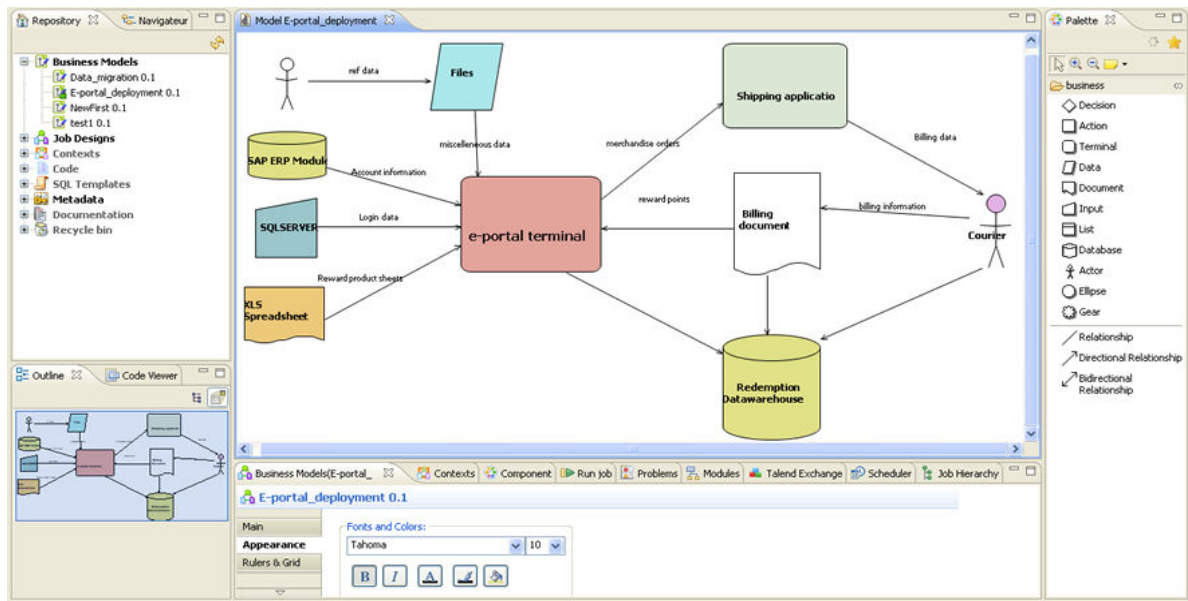


Figura 2. Modelagem de negócio – Talend

A Figura 2 mostra uma modelagem de negócio no *Talend Open Studio*.

Uma tarefa no *Talend Open Studio* é feita através de um *job*, um componente que agrupa vários outros componentes os quais vão executar o processo desejado. O *Job Designer* permite uma visão gráfica e funcional do processo de integração usando uma palheta gráfica de componentes e conectores. Os processos de integração são feitos pelo simples *dragging and dropping* (arrastar e soltar) dos componentes e conectores para a área de trabalho, desenhando conexões e relacionamentos entre eles.

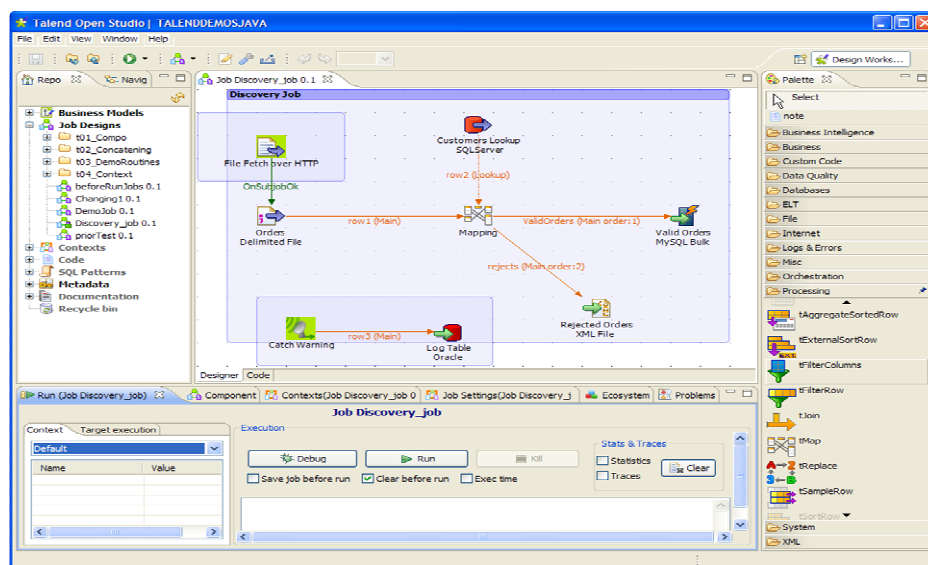


Figura 3. Job – Talend

A Figura 3 mostra um *job* no *Talend Open Studio*, onde a janela central é o *job*

em si, e todos os itens dentro dele são os componentes de entrada e saída de dados, mapeamento, entre outros.

Componentes e conectores abrangem todos os tipos de tarefas e operações em dados propriamente ditos ou em na sequência do fluxo de trabalho. Os conectores ajudam a acessar todos os dados de entrada e saída para migração de dados, integração de dados e sincronização de dados. As propriedades são configuradas principalmente em uma visão quando selecionamos cada componente em um *job* ou podem ser herdada de um gerenciador de metadados. Componentes complexos são equipados com uma interface gráfica intuitiva que ajudam os usuários a construir seus *jobs*.

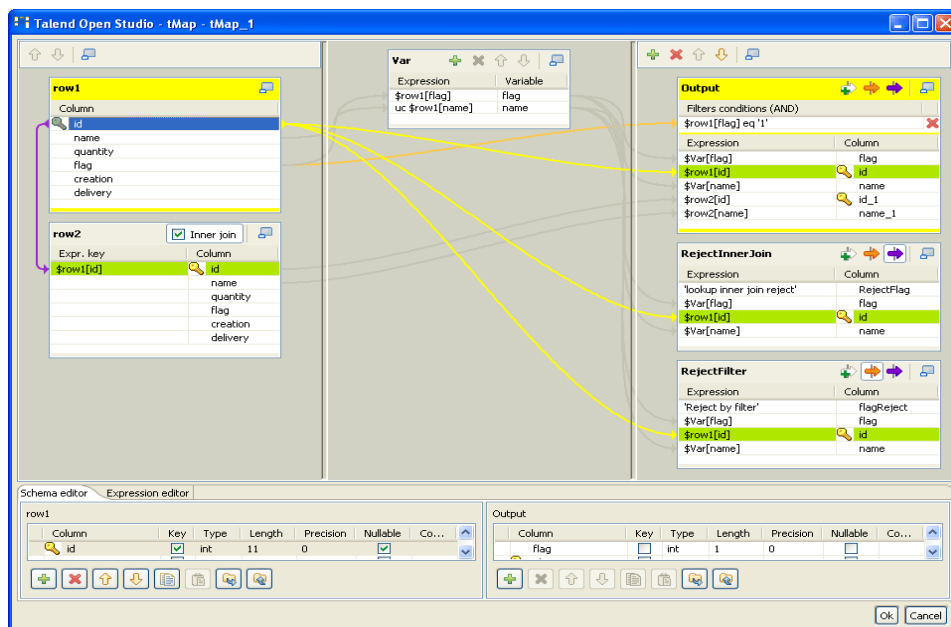


Figura 4. tmap – Talend

A Figura 4 exibe a interface gráfica do componente *tmap* o qual nos permite fazer várias ações de mapeamento entre os conectores (tabelas localizadas mais a esquerda são partes do conector de entrada *row1* 2 *row2*, já as mais a direita são as tabelas que compõem o conector de saída *Output*).

Todos os metadados são armazenados e gerenciados em um gerenciador de Metadados (O repositório) compartilhado para todos os módulos. Os repositórios centralizam todas as informações do projeto garantindo consistência entre todos os processos de integrações.

O Talend Open Studio oferece conectores técnicos e de negócio para todos os ambientes de tecnologia de informação. Os conectores são a chave do sucesso da

interoperabilidade de aplicações e banco de dados que permitem construir diversas e heterogêneas estruturas de dados com uma performance inigualável. Oferece conectividade para:

- Pacotes de aplicação (ERP, CRM,...), banco de dados, mainframes, arquivos, Web Services.
- Data warehouses, data marts, aplicações OLAP – para análise.
- Construção de componentes avançados para ETL, incluindo manipulação de *strings*, manipulação de pesquisa automática, etc.

2.3. Extração de dados

A atividade de extração dos dados incluem rotinas para leitura dos dados, conversão desses dados em um esquema intermediário e para movê-los para uma área de estágio comum, que se trata de uma área de trabalho temporária na qual os dados são mantidos em esquemas intermediários[1].

O processo de Extração dos dados subdivide-se em: Ler dados dos sistemas legados, determinar mudanças para identificar novos dados, generalizar chave e combinar registros de fontes múltiplas[1].

A leitura dos dados dos Sistemas Legados pode ser uma tarefa muito simples ou muito complexa. Quando o Sistema de origem dos dados for uma aplicação de legado aberta e bem documentada será uma tarefa muito fácil. Mas, como normalmente não é isto o que acontece, pois na grande maioria das vezes, a origem dos dados, provém de sistemas legados sem documentação sobre o significado dos dados e com uma estrutura interna de armazenamento de complexo entendimento. Entretanto, existem tarefas piores, que é quando os sistemas possuem uma estrutura proprietária, onde não se sabe o formato dos arquivos subjacentes; ficando o acesso às informações restrito a relatórios ou *jobs* pré-definidos de extração.

2.4. Postgres

O Postgres é uma poderoso SGBD objeto-relacional de código aberto. Por possuir mais de 15 anos em atividade conquistou forte reputação e confiabilidade, integridade de dados e conformidade a padrões. É compatível com quase todos os grandes sistemas operacionais como o Windows, Linux, Unix (Mac OS, Solaris, etc).

Além disso, possui uma vasta documentação disponível para consulta [4].

Por ser um SGBD de nível corporativo possui funcionalidades sofisticadas como controle de concorrência multiversionado; recuperação em um ponto de tempo; replicação assíncrona; transações agrupadas; um *query* de consultas e tolerância a falhas. Suporta codificações de caracteres como multibyte, Unicode e sua ordenação por localidade, sensibilidade a caracteres maiúsculos e minúsculos e formatação.

Há, também, uma grande parte da documentação do Postgres já traduzida para o português pelos que compõem a comunidade PostGres[4] no Brasil.

3 Atividades de Estágio

As atividades deste estágio consistem na migração dos dados do sistema de processo da UFS para o novo sistema SIPAC – módulo protocolo. Foram feitas algumas reuniões com o setor responsável pelo sistema de processo da UFS. Para o desenvolvimento das atividades do estágio foram utilizados dois computadores para o desenvolvimento das tarefas propostas.

O ambiente de desenvolvimento foi preparado com a instalação do SGBD PostgreSQL, do SGBD IBM DB2 e do Talend Open Studio. Foi disponibilizado em seguida, o backup do banco de dados do sistema de processos da UFS em IBM DB2 como, também, o backup do banco de dados do Módulo de Protocolo do SIPAC. Ambos foram restaurados em seus respectivos SGBD's, possibilitando assim o início das atividades de migração de dados.

3.1. Sistema legado de processo da UFS

O Modelo de dados do sistema de processo da UFS encontra-se desnormalizado, não havendo relacionamento algum entre as tabelas, com isso, os dados nela armazenados são inconsistentes e redundantes, muitas vezes preenchidos manualmente. Abaixo, as tabelas mais importantes do sistema de processo da UFS.

- PROCESSO.ASSUNTO
 - Armazena os assuntos inerentes aos processos.
- PROCESSO.CADASTRO
 - Tabela com os dados do processo propriamente dito
- PROCESSO.CORRESPONCENCIA
 - Esta tabela conceitualmente não pertence ao negócio, pois ela armazena informações de correspondências que chegam aos Departamentos da UFS.
- PROCESSO.SETOR
 - Armazena dados sobre os setores da UFS.
- PROCESSO.PROCANEXO
 - Tabela responsável pela anexação de processos. Contém 15 colunas possibilitando a junção de no máximo 15 processos.

- PROCESSO.PROCARQ
 - Tabela responsável pela localização física dos processos arquivados.
- PROCESSO.PROCEDENCIA
 - Tabela de procedência do processo, podendo ser uma unidade ou pessoa (física/jurídica)
- PROCESSO.MOVIMENTO
 - Tabela responsável pelo trâmite do processo.

3.2. Módulo Protocolo – SIPAC

O modelo de dados do módulo protocolo do SIPAC contém vinte e quatro tabelas sendo que apenas cinco delas foram utilizadas no mapeamento durante a migração dos dados. São elas:

- protocolo.tipoprocesso
 - Tabela que contém os tipos de processo utilizados no cadastro de processos.
- protocolo.processo_interessado
 - Um processo possui um ou mais interessados. Essa tabela realiza a associação entre um interessado a um processo
- protocolo.interessado
 - Interessados em um processo. Esta tabela identifica os interessados que são incluídos no processo através da tabela processo_interessado.
- protocolo.processo
 - Armazena informações sobre um processo.
- protocolo.movimento
 - Armazena informações da tramitação dos processos.

3.3. Migração dos dados

A migração dos dados do sistema legado para o módulo de protocolo do SIPAC foi através da ferramenta Talend Open Studio. O primeiro passo foi a criação de conexão com os dois SGBD (IBM DB2 e PostgreSQL). Logo após, foram criados

jobs com o objetivo de migrar os dados em módulos e, posteriormente, um *Job* principal para executar todos eles.

Para uma melhor compreensão da descrição dos *jobs* é preciso entender que as tabelas descritas com caracteres em maiúsculo pertencem ao sistema legado de processo da UFS, já as tabelas escritas com letras minúsculas pertencem ao módulo protocolo do SIPAC.

3.3.1. Job Status

Este *job* faz a inserção dos dois estados básico que um processo pode se encontrar: ativo ou arquivado. Com os dados da tabela 'CADASTRO' (input conforme Figura abaixo), é feita uma consulta que retorna os dois estados referentes ao status do processo, sendo eles assim inseridos na tabela 'status_processo'.

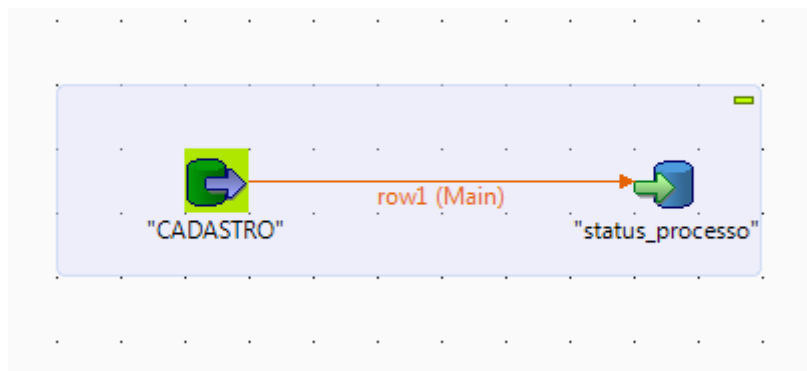


Figura 5. Job Status

A Figura 5 mostra a ligação entre um componente de input (que provê dados) para um componente de output (que recebe os dados e os persiste no banco de dados).

3.3.2. Job Tipo Processo

Este *job* é responsável por transcrever os dados da tabela 'ASSUNTO' para a tabela 'tipoprocesso'. Mapeando o campo 'ASSU_A' de 'ASSUNTO' para o 'idtipoprocesso' de 'tipoprocesso' e o campo 'DESC_A' de 'ASSUNTO' para o 'denominacao' de 'tipoprocesso'. A Figura abaixo mostra bem essa situação.

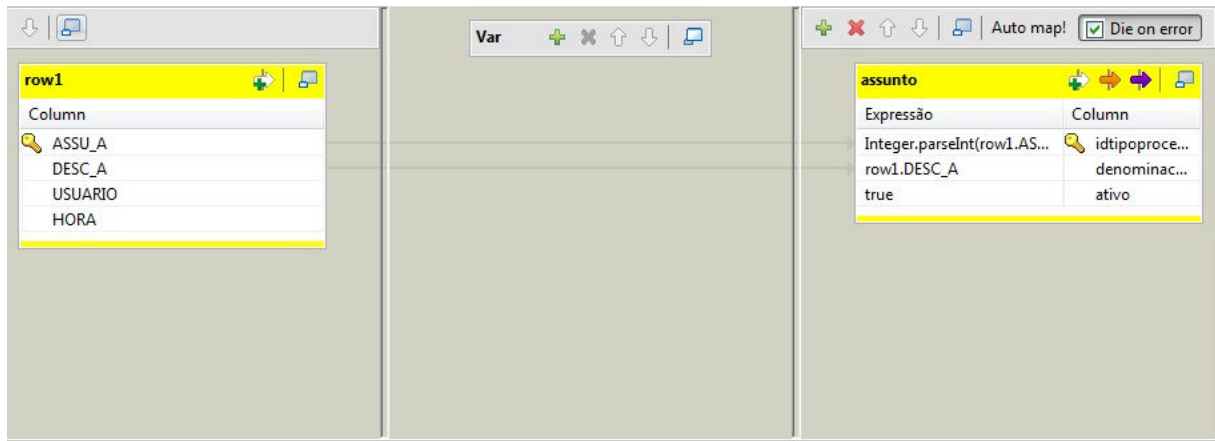


Figura 6. tMap Job Tipo Processo

A Figura 6 exibe o mapeamento entre duas tabelas, onde duas colunas da tabela de entrada estão alimentando à duas colunas da tabela de saída, sendo que o dado proveniente da primeira coluna está passando por um tratamento (a conversão para inteiro).

3.3.3. Job Interessado

No sistema legado só é possível que haja no máximo um interessado por processo, e isso é feito com um campo texto preenchido manualmente pelo usuário na aplicação. Com isso a redundância contida nos dados é muito grande. O mesmo interessado pode ser diferenciado inclusive pela inclusão de espaços em branco ou erros de grafia, já que foram digitados livremente pelos vários usuários que utilizam o sistema.

Esse *job* faz uma consulta na tabela 'CADASTRO', fazendo um tratamento no campo 'NOME_P', retirando o máximo de espaços em branco para igualar alguns dos interessados e diminuir o volume de dados redundantes e selecionando apenas os distintos após esse tratamento, o quê resultou na redução de 155090 registros para 88713 registros. Com essa redução dos diferentes interessados contidos no sistema legado, eles são, então, inseridos na tabela 'interessados' do SIPAC.

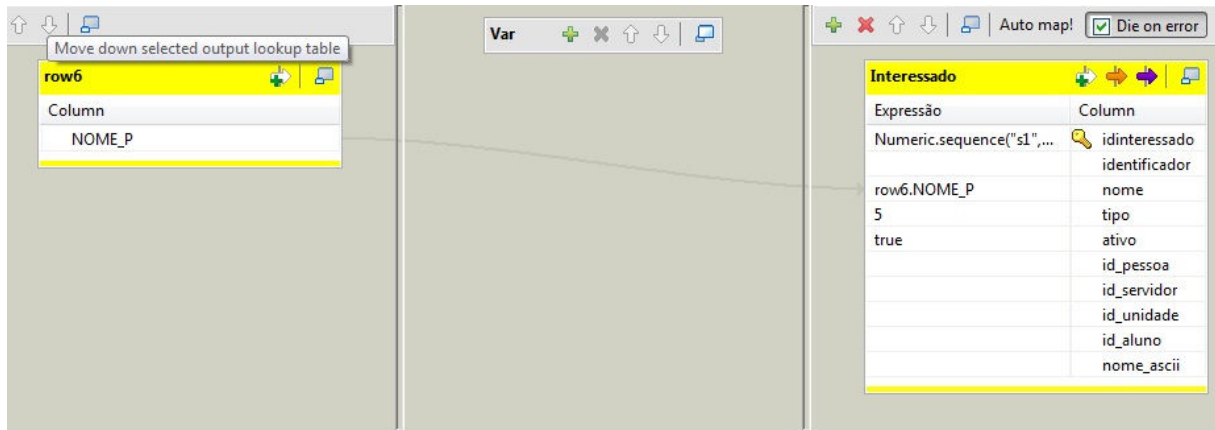


Figura 7. tMap Job Interessado

A Figura 7 mostra a ligação do resultado da consulta que seleciona apenas os interessados da tabela 'CADASTRO' e os armazena na tabela 'interessado'.

3.3.4. Job Processo

O Job processo é composto por várias tabelas e dois *tmaps*. Conforme trás a Figura 9. Nesta Figura pode-se visualizar várias entradas de dados (*inputs*) as quais são mapeadas através de dois *tmaps* para assim armazenarem os dados na tabela 'processo'.

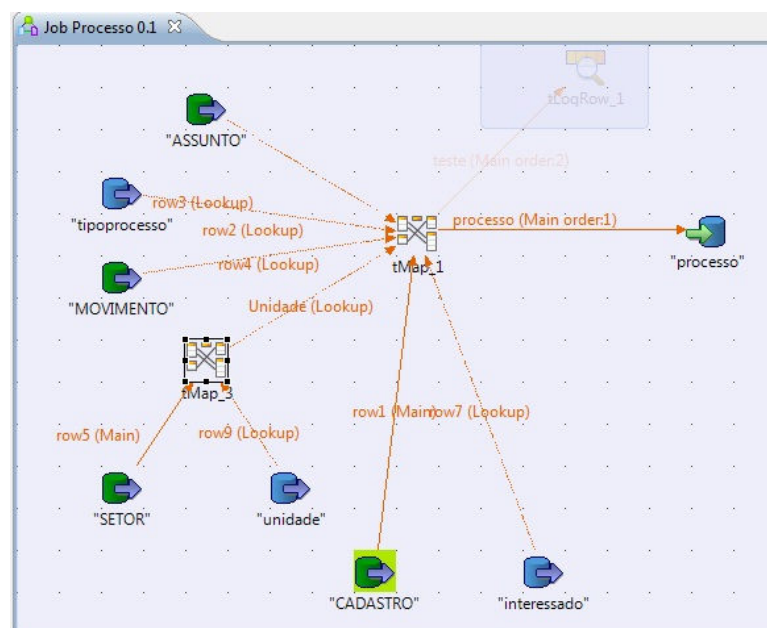


Figura 8. Job Processo

O *tMap_3* varre a tabela 'SETOR', e compara o campo 'DESC_S', que armazena o nome do setor, com o campo 'sigla' da tabela 'unidade'. A grande maioria dos setores do sistema legado difere das unidades contidas no novo sistema, pois até a implantação desse novo sistema, a UFS não continha nenhum tipo de hierarquia entre suas unidades. A comparação entre os campos das duas tabelas se dá com a cláusula SQL *LIKE*, onde todos os setores que combinarem com a sigla de alguma unidade será considerado como sendo daquela unidade.

O *tMap_1* fará o mapeamento de todos os dados contidos na tabela 'CADASTRO' que são relevantes à tabela 'processo'. A coluna 'NUME_P' da tabela 'CADASTRO', que, além de armazenar o número do processo, é também a chave primária da tabela. O número de um processo é contido por 10 dígitos, sendo os 6 primeiros o número do processo propriamente dito; os 2 dígitos seguintes correspondem aos dois últimos números do ano do processo (por exemplo, 02 para ano de 2002); e os 2 últimos dígitos são os dígitos verificadores do processo, Exemplo: 0234560256

- Número do processo: 023456
- Ano: 02 (2002)
- Dígitos verificadores: 56

Existem também nessa tabela muitos dados anômalos (espaços em branco entre os dígitos, caracteres especiais), alguns deles nem contendo os dados mínimos necessários para representar um processo. Para tratamento desses dados foram utilizadas variáveis temporárias, para retirar os caracteres especiais, espaços em branco e completando com zeros à esquerda, todos aqueles registros que tem menos de 10 dígitos. Com o uso de outras variáveis temporárias, esses dados formados por 10 dígitos são quebrados, se transformando em três números, para alimentar a tabela 'processo' que armazena esses três valores separadamente nas colunas 'num_protocolo', 'ano' e 'dv'.

Ainda no *tMap_1*, a coluna 'NOME_P' é mapeada com a tabela 'interessado' (previamente preenchida com os dados da própria tabela 'CADASTRO'), resgatando assim o id_interessado.

A tabela 'processo' possui ainda a coluna 'data_cadastro', dado que não existe na tabela 'CADASTRO'. É então feita uma busca na tabela 'MOVIMENTO' buscando a menor data em que o processo aparece, ou seja, a data da sua primeira tramitação.

A coluna 'ASSU_A' é mapeada com os registros da tabela 'ASSUNTO' que, por sua vez, é mapeada com a tabela 'tipoprocesso'.

A coluna 'id_unidade' da saída 'Unidade' do *tMap3* será mapeada para a coluna 'id_unidade' na tabela 'processo'. Os registros da tabela 'CADASTRO' que não contiverem um setor correspondente a uma unidade da tabela 'unidade', será alimentado com o id correspondente a unidade 'OUTRAS'.

Um detalhe importante que é feito nesse *job* é a criação de uma cópia da chave primária da tabela 'CADASTRO' para uma coluna da tabela 'processo' que não será usada na migração. Isso é necessário porque com a quebra da chave 'NUME_P' em três campos para a tabela de saída, a referência aos registros antigos seria perdida.

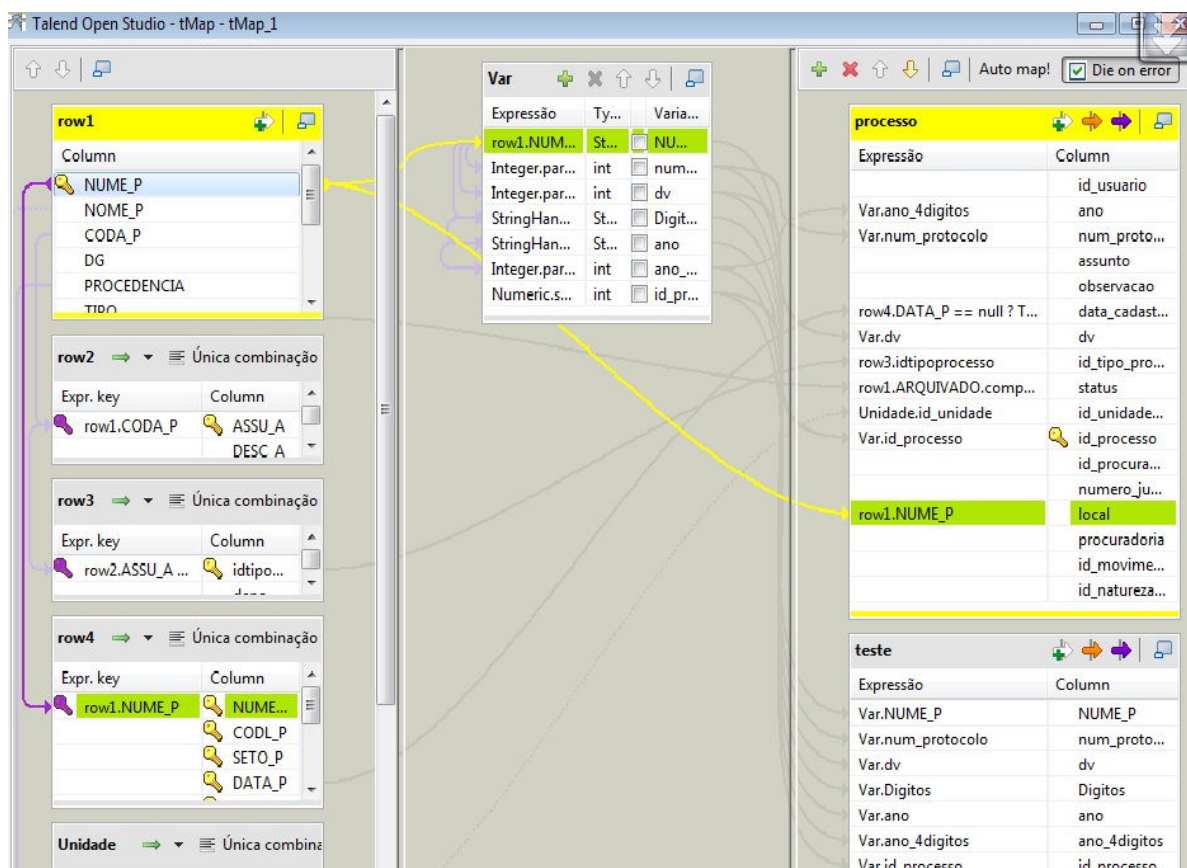


Figura 9. tMap_1 Job Processo

A Figura 9 mostra o mapeamento mais interno visto na Figura 8 onde além de algumas colunas de entrada se ligarem diretamente com as de saída, o dado que vem da coluna 'NUME_P' passa por uma série de tratamentos, com o uso de variáveis temporárias vistas na parte central da Figura.

3.3.5. Job Processo Interessado

Nesta tarefa iremos colher os dados para alimentar a tabela 'processo_interessado', que visa unir informações entre processos e seus interessados. No sistema legado isso não era possível, pois o interessado era um campo de texto para ser manualmente preenchido pelo usuário na aplicação. A coluna 'NUME_P' da tabela 'CADASTRO' será mapeada para a coluna 'local' da tabela 'processo' (que armazena uma cópia da chave primária da tabela 'CADASTRO') e, a seguir, o campo 'id_processo' de 'processo' será mapeado para a coluna 'id_processo' da tabela 'processo_interessado'. O campo 'NOME_P' de 'CADASTRO' será mapeado para a coluna 'nome' na tabela 'interessado'. Após isso, a coluna 'idinteressado' de 'interessado' será mapeada para o campo 'idinteressado' da tabela 'processo_interessado'. Como foi feito anteriormente no *job* processo, a coluna 'NUME_P' é quebrada novamente para assim alimentar as colunas 'num_processo' e 'ano'.

3.3.6. Job Movimento

Está é a maior tarefa e a que possui a maior quantidade de dados a serem migrados. É composta por várias tabelas e por três *tMaps* como segue.

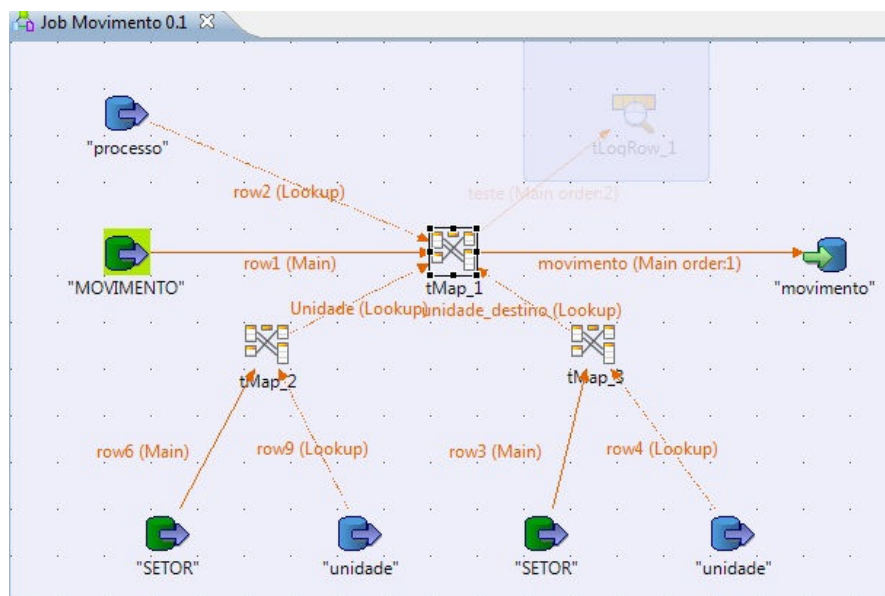


Figura 10. Job Movimento

A Figura 10 exibe todos os componentes utilizados no *job* movimento o qual possui três *tmaps*. O componente mais claro (tLogRow_1) é um componente que exibe toda a saída em texto no próprio Talend, facilitando a visualização antes de persistir os dados.

Nesse *job* os tMaps 2 e 3 recuperam todos os ids das unidades que correspondem a um setor no antigo sistema. Um para a unidade origem e outra para a unidade destino.

O *tMap_1* é responsável pelo mapeamento principal e gera a saída 'movimento'. A coluna 'NUME_P' da tabela 'MOVIMENTO' passa por um tratamento em algumas variáveis intermediárias antes de ser mapeado para 'movimento'. Isso é feito porque na tabela 'movimento' há colunas que necessitam de partes do número do processo. Com isso, a coluna 'NUME_P' alimenta os campos 'id_movimento', 'num_protocolo' e 'ano' da tabela 'movimento' após tratamento em variáveis temporárias. Ainda referente à coluna 'NUME_P' há um mapeamento com a coluna 'local' da tabela 'processo', que armazena uma cópia da chave primária de 'CADASTRO'.

Ainda há o mapeamento da coluna 'CODL_P' de 'MOVIMENTO' para a coluna 'CODI_S_Origem' de 'Unidade' (saída do *tMap_2*). Logo após um mapeamento da coluna 'SETO_P' de 'MOVIMENTO' para a coluna 'CODI_S_Destino' da saída 'unidade_destino' (saída do *tMap_3*), ambas alimentando respectivamente as colunas 'id_unidade_origem' e 'id_unidade_destino' da tabela 'movimento'. Um detalhe importante nesse ponto é que, como muitos setores não conseguem ser mapeados para as novas unidades, o sistema a ser implantado iria perder o histórico da tramitação desses processos. Sendo assim, são colocados na coluna 'observação', os setores de origem e destino da tramitação do processo no antigo sistema.

Seguindo na tabela 'MOVIMENTO' temos os mapeamentos das colunas 'DATA_P' e 'DATASAIIDA' para as colunas 'data_recebimento' e 'data_envio_origem', respectivamente, para a tabela 'movimento'.

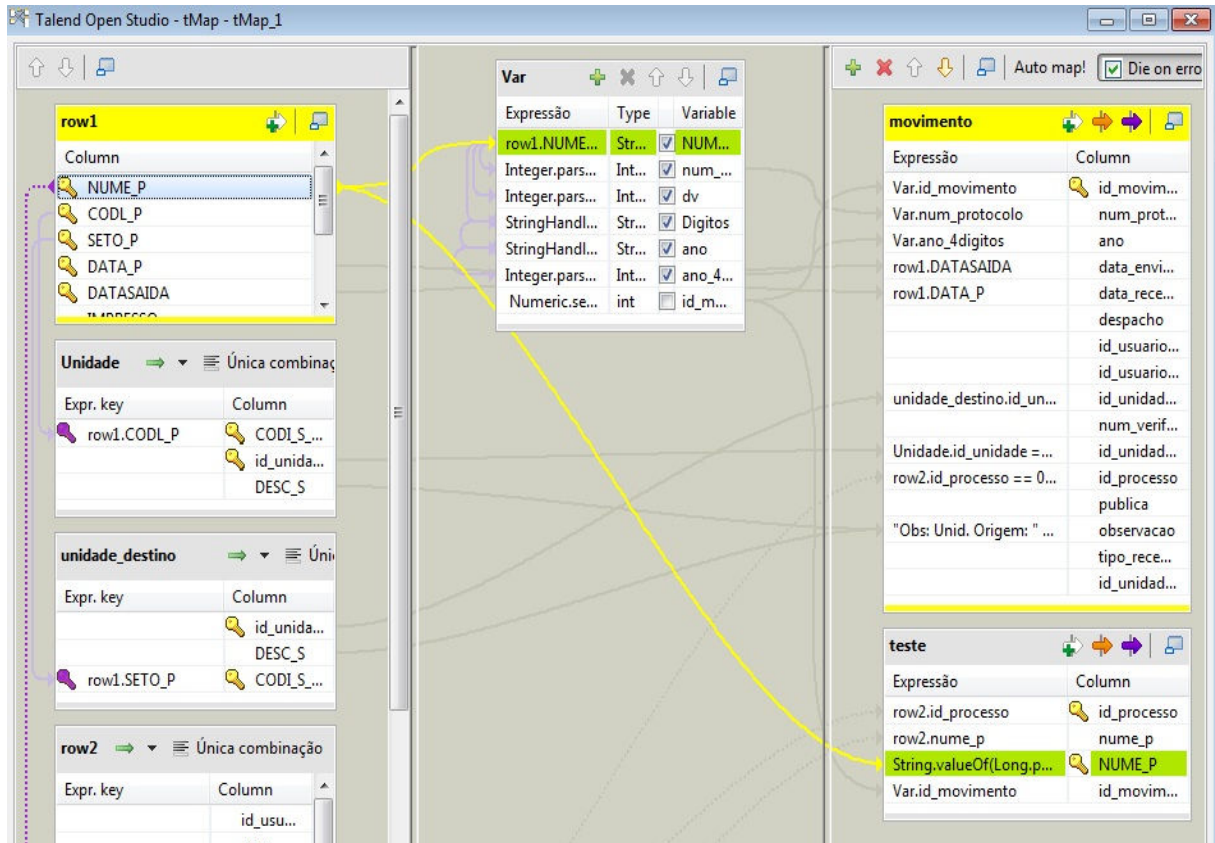


Figura 11. tMap_1 Job Movimento

A Figura 11 mostra o mapeamento principal do *job* movimento, que também, assim como na Figura 10 do *job* processo, faz o uso de variáveis temporárias.

3.3.7. Job Update Processo

Após todos os dados migrados, é feita uma limpeza na tabela 'processo', é executado um *update* para apagar a cópia da chave primária da tabela 'CADASTRO' que foi previamente feita e foi de muita importância para o sucesso de toda a operação.

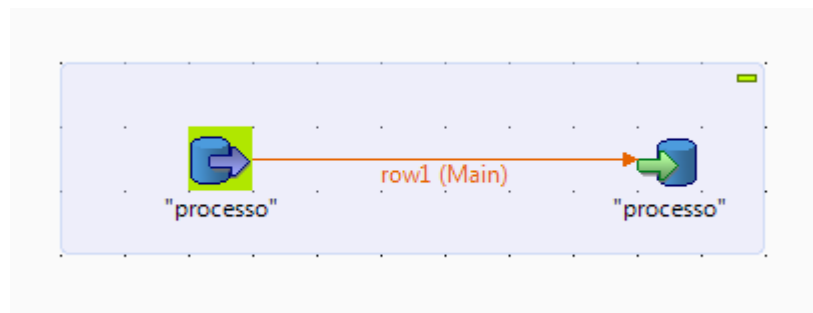


Figura 12. Job Update Processo

A Figura 12 mostra a ligação entre os componentes *input* e *output* que representam a mesma tabela, 'processo', a fim de realizar um update.

3.3.8. Job Principal

Esse *job* executa em sequencia todos os *jobs* descritos anteriormente, fazendo assim todo o processo do início ao fim.

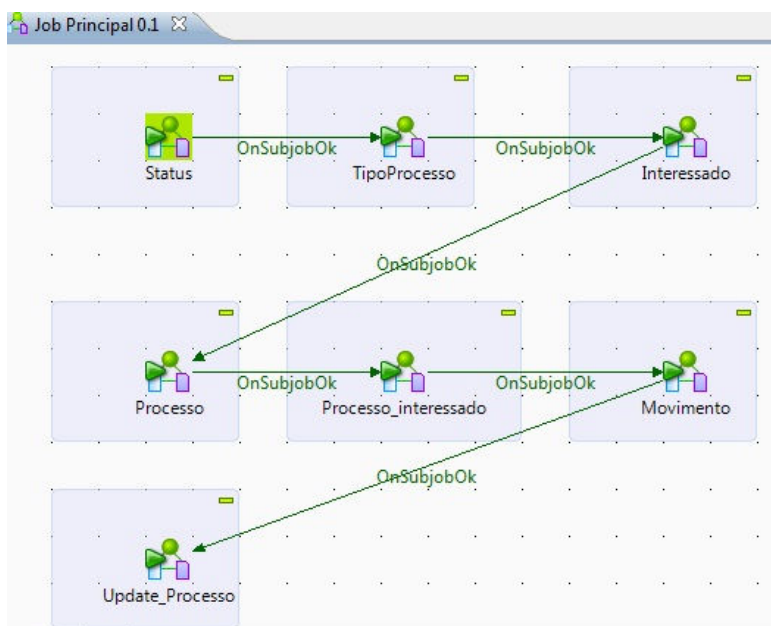


Figura 13. Job Principal

A Figura 13 mostra a ordem de execução desses processos. Inicialmente o *Job* 'Status' é executado para realizar as operações de inserção dos diferentes tipos de status. Em seguida o *job* TipoProcesso que executa as operações para inserção de todos os tipos de processo existentes no sistema legado. O *job* Interessado faz a limpeza e a inserção dos interessados. O *job* processo faz a inserção de todos os processos, seguido do *job* processo_interessado que faz a ligação entre as duas entidades. A seguir o *job* movimento é executado fazendo a migração de todas as tramitações dos processos. Por fim o *job* Update_processo é executado para efetuar a limpeza de uma coluna na tabela processo que foi usada para armazenar temporariamente a referencia da chave primária da tabela 'CADASTRO' do sistema legado.

4 Conclusão

A integração de sistemas tem se tornado cada vez mais importante para o crescimento das organizações. Instituições como a UFS necessitam destes para a integração e comunicação entre setores e departamentos, melhorando assim seus serviços e tarefas. Sendo assim optou por adquirir o sistema integrado da Universidade Federal do Rio Grande do Norte. Essa escolha levou à necessidade de migração dos dados dos sistemas legados existentes para o novo sistema adquirido.

O processo de extração de dados se tornou complexo, pois não houve documentação por parte do sistema legado. A complexidade só aumentou devido à falta de normalização do modelo de dados do Sistema de Processo da UFS. Contudo, com o auxílio das ferramentas adequadas para extração e realização dos testes as migrações necessárias foram concluídas com êxito.

Os dados do sistema legado foram passados para o novo sistema integrado (módulo protocolo – SIPAC). Para o sucesso das tarefas se fez necessário tratamento de dados, conversão de dados, tabelas temporárias e consultas de verificação nos dois SGBDs.

Algumas dificuldades foram encontradas, mas o contato constante com o pessoal do setor responsável através de reuniões presenciais foi essencial para o sucesso das atividades desenvolvidas no estágio. Vale ressaltar que o auxílio de ferramentas para gerência de projeto web como o Redmine[6] também teve grande importância para organização das tarefas realizadas no decorrer do estágio.

O estágio supervisionado foi de grande importância para a prática de grande parte do conhecimento teórico adquirido durante o curso. Disciplinas como Banco de Dados, Programação, Gerência de Projetos, Engenharia de Software, nos ajudaram com as atividades do estágio. Os conhecimentos da disciplina Banco de Dados foram os mais utilizados na tarefa de transição e migração, já os conhecimentos em programação, em especial Java, foram muito úteis para o uso da ferramenta Talend Open Studio[3]. Engenharia de software e Gerência de Projeto nos possibilitou a organização e fluxo das tarefas, assim como obter um melhor entendimento do negócio.

Referências Bibliográficas

1. Killball, R. The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data. 1. Internacional: John Wiley & Sons. 2004.
2. Heuser, Carlos A. Projeto de Banco de Dados. 4. Nacional: Sagra Luzzatto, 1998.
3. Talend Open Studio, Open Source. 2006-2010. Disponível em <http://www.talend.com/index.php>. Último acesso em julho de 2010.
4. SGBD PostgreSQL. Disponível em <http://www.postgresql.org.br>. Último acesso em julho de 2010.
5. SGBD IBM DB2. Disponível em <http://www-01.ibm.com/software/br/db2/lowerdatabasecosts/>. Último acesso em julho de 2010.
6. Redmine. Disponível em <HTTP://www.redmine.org>. Último acesso em julho de 2010.