TÉCNICO LISBOA

Geração de Indicadores para Processos de Negócio

Construção de um Data Warehouse para o motor de processo edoclink

João Miguel Gandum Ferreira

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática e de Computadores

Orientador: Prof. Pedro Manuel Moreira Vaz Antunes de Sousa

Júri

Presidente: Prof. Francisco António Chaves Saraiva de Melo
Orientador: Prof. Pedro Manuel Moreira Vaz Antunes de Sousa
Vogal: Prof. Mário Jorge Costa Gaspar da Silva

Janeiro 2021

Resumo

Os Indicadores são essenciais para qualquer sistema em que é importante uma constante monitorização e avaliação, dando-nos informação sobre desempenho, realização e responsabilidade.

No contexto da empresa edocLink, foram definidos alguns indicadores que permitissem fazer uma análise aos dados gerados pelo seu produto (edoc - ferramenta de gestão documental), útil para, por exemplo, automatizar o processo de atribuição de tarefas ou para simplesmente ver a evolução do volume de trabalho ao longo do tempo.

Desenvolveu-se uma solução que permite extrair e transformar os dados de uma base de dados do ambiente do edoc, para depois carregar para um Data Warehouse. Esse Data Warehouse é depois conectado com o Power BI onde será feita a análise através de dashboards e cumpridos os objetivos definidos em conjunto com colaboradores da empresa.

No final, a solução e os resultados obtidos são avaliados em 4 vertentes - extensibilidade, produtização, escalabilidade e usabilidade dos Dashboards. Conclui-se que a solução é extensível e produtizável, mas não escalável. Por sua vez, os dashboards criados concluíram-se ser simples, interativos e de compreensão imediata, mesmo para um utilizador que não tenha um conhecimento vasto dos conceitos do edoc.

Palavras-chave: Data Warehouse, Processos de Negócio, Indicador, Bases de Dados, Esquema em estrela, Power BI, ETL

Abstract

Indicators are essential for any system where constant monitoring and evaluation is important, giving us information on performance, achievement and responsibility.

In the context of a company named edocLink, some indicators were designed to allow an analysis of the data generated by its product (edoc, a document management tool), useful for, for example, automating the task assignment process or to simply see the evolution of the workload over the time.

We devoleped a solution that allows extracting and transforming data from an edoc generated database and then loading it into a Data Warehouse. This Data Warehouse is then connected with Power BI where the analysis will be made through dashboards, fulfilling the objectives defined jointly with edocLink.

In the end, the solution and the results obtained are evaluated in 4 different ways - extensibility, production, scalability and Dashboards' usability. It is concluded that the solution is extensible and productive, but not scalable. In addition to that, the dashboards created were concluded to be simple, interactive and of immediate understanding, even for a user who does not have a vast knowledge of the edoc concepts.

Keywords: Data Warehouse, Business Process engine, Business Process KPI, Database, Star schema, Power BI , ETL

Agradecimentos

Em primeiro lugar, queria agradecer ao Prof. Pedro Sousa pelo tema e orientação, pela rapidez e clareza com que sempre se disponibilizou para ajudar.

De seguida, à Carolina Marques e ao João Guilherme pelo apoio inigualável que me deram como co-orientadores. Desde perguntarem diariamente se precisava de ajuda a partilharem comigo ideias, sugestões e dicas. Foram cruciais neste projeto.

Ao Filipe Correia, pelo apoio técnico que me deu.

À edocLink por me ter recebido, ainda que por causas maiores, não tenha sido possível estar fisicamente presente.

À minha família, por me ter proporcionado todos os meios necessários para ser bem sucedido e por me motivar a dar sempre o meu melhor e a não desistir.

À Rita Sobral por ser um exemplo de esforço e dedicação tremendo e me fazer querer dar sempre mais.

Muito obrigado,
João Gandum Ferreira

Conteúdo

Lista de Figuras

Acrónimos

ETL - Extract, Transform, Load
DW - Data Warehouse
BD - Base de Dados
PDI - Pentaho Data Integration
PRD - Pentaho Report Designer
SCD - Slowly Changing Dimension

Capítulo 1

Introdução

1.1 Contexto

Este projeto surge no contexto da empresa edocLink Enterprise, mais especificamente, do edocLink - uma ferramenta de gestão documental. Pensou-se que seria interessante automatizar o processo de alocação das tarefas aos colaboradores. Para tal, é primeiro preciso definir de que forma se quer fazer a atribuição. Será que se atribui uma tarefa à pessoa com menos tarefas realizadas no último mês? Ou à pessoa que mais tarefas desse tipo entregou? Ou ao colaborador que menos tempo as demora a executar?

Este projeto pretende desenvolver uma solução que consiga, com base numa base de dados com dados gerados pelo edoc, fazer uma análise e dar resposta a um conjunto de indicadores definidos juntamente com a edocLink.

Para tal, inicialmente foi necessário compreender-se a organização e funcionamento da empresa, o seu ambiente e conceitos. Depois, prepararam-se os dados e carregaram-se para um Data Warehouse, para que então pudesse ser feita uma análise e apresentados os resultados.

1.2 Objetivos

O objetivo do projeto é desenvolver um sistema capaz de dar resposta a um conjunto de indicadores definidos diretamente com a ajuda da edocLink. Os indicadores são os seguintes:

- **I. Tarefas Pendentes por Tipo de Distribuição, por Interveniente, por Executante**
- **II. Tarefas Entregues por Tipo de Distribuição, por Interveniente, por Executante**
- **III. Tempo Médio de Aceitação por Etapa, por Tipo de Distribuição**
- **IV. Tempo Médio Total por Tipo de Distribuição**
- **V. Tempo Médio Total por Etapa, por Tipo de Distribuição**
- **VI. Tempo Médio Total por Fase, por Tipo de Distribuição**

- **VII. Volume de distribuições por Tipo de Distribuição, por Ano e por Mês**

1.3 Metodologia

Para que fosse possível dar resposta aos indicadores definidos, foram necessários os seguintes passos:

- Analisar a BD de origem e perceber quais as tabelas necessárias para o projeto;
- Criar um Data Warehouse, definindo tabelas de dimensão e de facto;
- Implementar o processo de ETL usando o Pentaho Data Integration, que extraiu os dados da BD do edoc, transformou e carregou para o Data Warehouse;
- Desenvolver dashboards que permitissem uma visualização direta dos dados necessários para responder aos indicadores definidos.

1.4 Resultados

A solução obtida foi um sistema com 4 componentes - BD do edoc, Data Warehouse, Pentaho Data Integration e Power BI.

Concluiu-se que a solução é extensível, produtizável, interativa e de fácil compreensão para o utilizador. No entanto, concluiu-se também que não é escalável, devido à grande diferença no tempo de execução quando a solução é executada com um grande volume de dados (10 vezes superior à da base de dados original).

1.5 Estrutura do Documento

O documento apresenta 5 capítulos. No Capítulo 1 é feita uma introdução ao tema, onde se explica o contexto em que se insere o projeto desenvolvido e quais os objetivos que se pretendem alcançar com a solução desenvolvida.

O Capítulo 2 trata sobre o Estada da Arte - faz referência a várias definições de conceitos importantes para o tema, expõe diferentes tipos de modelos de dados e introduz várias ferramentas relacionadas, detalhando vantagens e desvantagens de cada uma.

A solução implementada é depois detalhada ao pormenor no Capítulo 3, no qual são demonstrados como foram desenvolvidos todos os componentes da mesma. No início do capítulo é introduzida a empresa para qual o projeto foi implementado e explicado o seu ambiente - conceitos, funcionamento, organização. De seguida, a arquitetura e a tecnologia usada na solução são apresentados para depois se começar a detalhar cada decisão tomada durante o processo de desenvolvimento.

No Capítulo 4, a solução é avaliada com o objetivo de verificar se os objetivos definidos no capítulo 1 foram cumpridos - tanto em termos de desempenho como funcionais.

Por fim, o Capítulo 5 conclui o documento, fazendo uma breve conclusão sobre todo o projeto e sugerindo algumas propostas para um trabalho futuro.

Capítulo 2

Estado da Arte

Neste capítulo serão apresentados alguns conceitos cruciais para o entendimento do tema, como também algumas ferramentas e diferentes estruturas de dados. Esta pesquisa servirá como base da solução a implementar. Em primeiro lugar, é importante definir o que é um indicador e de que forma é que poderá ser essencial.

De seguida, serão apresentados diferentes tipos de modelos de dados e algumas ferramentas relacionadas, algumas das quais serão utilizadas posteriormente para implementar a solução proposta.

2.1 Indicadores

Indicadores são um componente essencial de qualquer sistema de monitorização e avaliação eficaz, fornecendo informação crucial acerca de desempenho, realização e responsabilidade. Por exemplo, é com base nestes que são tomadas decisões de otimização de desempenho numa empresa. Também são estes que influenciam a estratégia a seguir pelo governo de modo a melhor responder perante uma pandemia [14].

O desafio no desenvolvimento de indicadores está em assegurar a sua qualidade e integridade , de modo a garantir que este nos traz informação concisa e com valor.

De acordo com o DAC / OCDE3, um indicador é:
"Um fator ou variável quantitativa ou qualitativa que fornece um meio simples e confiável para medir a realização, para refletir as mudanças ligadas a uma intervenção, ou para ajudar a avaliar o desempenho de um ator do desenvolvimento."[4]

Um indicador é, portanto, útil porque é uma medida normalizada para fazer comparações ao longo do tempo, o que nos dá a capacidade de avaliar e interpretar a evolução de uma determinada variável [13].

2.2 Dados

De modo a se poder aplicar um indicador, é primeiro preciso haver uma coleção de dados. Estes dados são tipicamente gerados através da execução dos processos de uma organização/empresa - processos de negócio [3]. Quer a compra de um produto quer a entrega de encomendas num armazém são exemplos de eventos que ocorrem

durante estes processos de negócio e que produzem dados cruciais para poder ser feita uma análise posterior.

Os sistemas de informação são responsáveis por gerir a execução dos processos e guardar os dados gerados por cada passo do processo. Geralmente, processam um evento de cada vez, produzindo um registo de transação e identificando o estado corrente. De seguida, um outro evento poderá alterar o estado inicial, o que leva o sistema a atualizar o registo para tal se verificar. Por conseguinte, os modelos de dados destes sistemas de informação estão otimizados para haver uma escrita e atualização rápida dos dados.

Contudo, esta coleção de dados não é suficiente para poder ser feita uma análise e chegar a conclusões diretamente. Os dados provenientes dos processos são altamente normalizados para minimizar a ocupação de espaço e maximizar a velocidade com que estes são guardados, tornando mais difícil a compreensão da estrutura para os utlizadores [10]. Apesar de algumas fontes de dados permitirem uma análise superficial, através de dashboards, por exemplo, não são flexíveis o suficiente para um analista. Por esta necessidade, surgem então as bases de dados analíticas, conhecidas por Data Warehouses (DW). Segundo Kimball [9], um DW é uma cópia de registos informacionais gerados através de uma transação estruturada, que permite a análise e interrogação sobre a mesma. Os DW intregram dados de várias fontes num repositório central. Outra característica importante é o facto de toda a informação armazenada ser etiquetada temporalmente, permitindo uma análise tendo em conta o passado [1]. É importante perceber também que, antes do carregamento, os dados geralmente passam por um processo de processamento para apresentarem uma melhoria de qualidade relativamente aos dados operacionais.

2.3 Dimensões e factos

As dimensões são um conceito essencial para a arquitetura de um DW. São estas que permitem analisar a informação através de várias perspetivas. São utilizadas para selecionar e agrupar os dados conforme o nível de detalhe pretendido [1].

As tabelas de dimensão contêm atributos descritivos, utilizados para filtrar as queries. Cada tabela contém uma chave primária única que corresponde a uma das partes da chave composta da tabela de factos associada.

As tabelas de facto são outro componente importante, uma vez que armazenam as métricas a analisar. Cada facto é um objeto que representa algo que se pretende analisar. As tabelas de facto e de dimensão estão relacionadas, uma vez que nos modelos multidimensionais os factos são implicitamente definidos pela combinação das dimensões. As tabelas de factos têm uma chave primária com 2 ou mais chaves estrangeiras e apresentam uma relação de um para muitos com as tabelas de dimensão.

Parâmetros	Tabela de Facto	Tabela de Dimensões
Definição	Métricas, cálculos sobre um processo de negócio	Contém atributos descritivos para se poder utilizar como filtro nas queries
Localização	Localizada no centro do esquema em estrela ou floco de neve	Conectada à tabela de factos e localizada nas pontas
Objetivo	Proporcionar factos que permitam a análise e reporting	Coleção de informação acerca do negócio
Tipo de Dados	Contém informação como número de produtos vendidos	Contém atributos que descrevem os detalhes da dimensão. Por exemplo, dimensão de Produto poderá conter o ID do produto, nome do produto, etc
Key	Chave primária composta por chaves estrangeiras (das tabelas de dimensão)	Chave primária única que identifica cada dimensão
Hierarquia	Não apresenta	Apresenta hierarquia. Por exemplo, uma dimensão de Localização poderá conter o país, código postal, estado, cidade, etc.

Tabela 2.1: Resumo características das tabelas de facto e dimensão

Como referido anteriormente, as tabelas de dimensão e de factos estão conectadas entre si e podem ter estruturas de maneira diferente. Os dois modelos mais utilizados são o esquema em estrela (star schema) e o floco de neve (snowflake schema).

2.3.1 Esquema em Estrela

O esquema em estrela é composto por uma tabela central (tabela de factos) rodeada de várias tabelas de dimensão. A grande vantagem desta organização é a velocidade de resposta, independentemente do volume de dados. Este comportamento é possível uma vez que o número máximo de operações join é igual ao número de tabelas de dimensões ligadas à tabela de factos. Neste esquema, as tabelas de dimensão são altamente desnormalizadas e, como tal, contêm muita informação redundante (em troca de apresentarem um melhor desempenho).

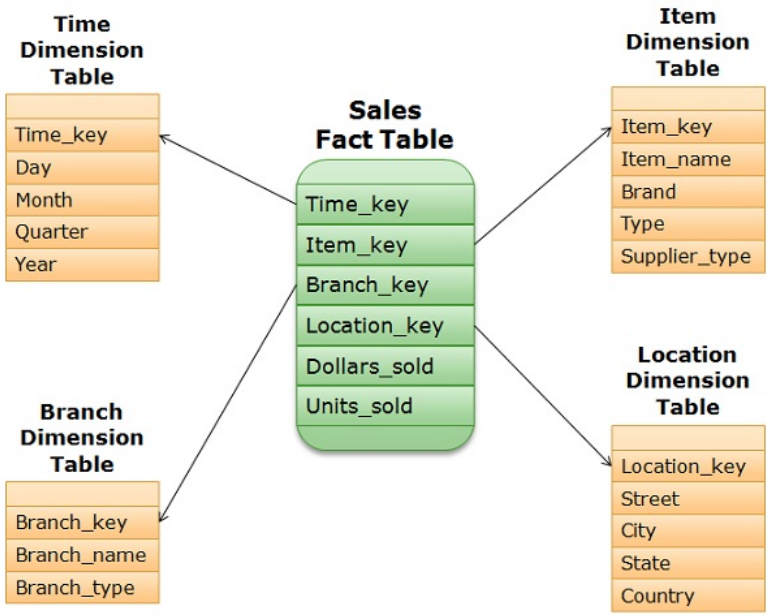


Figura 2.1: Esquema em Estrela [8]

2.3.2 Floco de Neve

Este esquema é semelhante ao esquema em estrela no sentido em que também tem uma tabela de factos central que está conectada a várias tabelas de dimensão. No entanto, as tabelas de dimensão estão normalizadas em várias tabelas (subdimensões), dividindo os dados para evitar redundância. Isto aumenta claramente o número de tabelas necessárias, aumentando o número de operações join para uma dada query, mas diminuindo o espaço necessário para armazenar a informação.

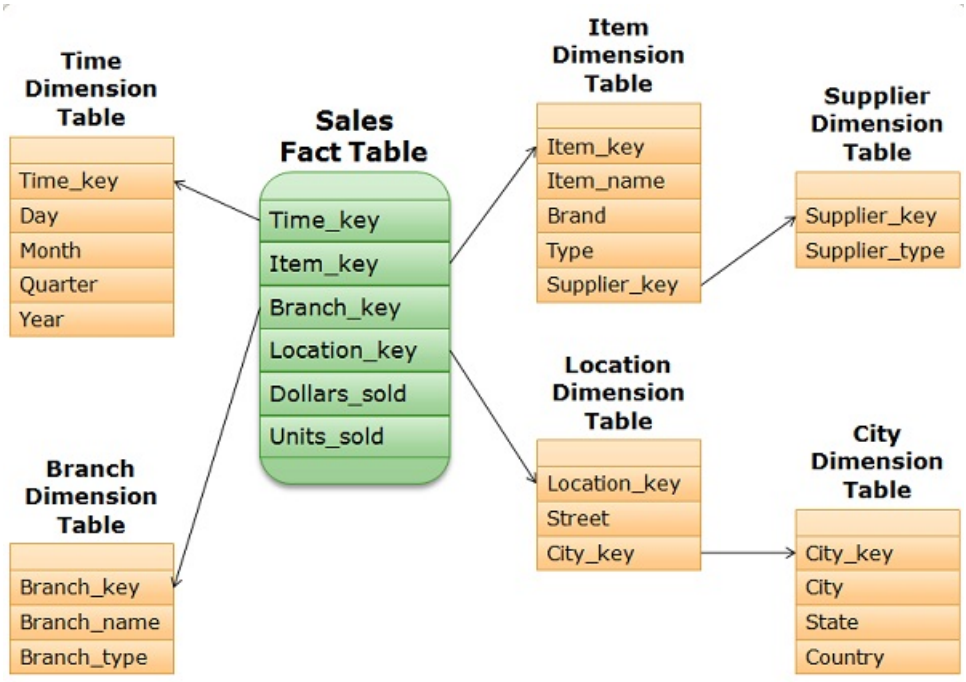


Figura 2.2: Esquema em Floco de Neve [8]

2.4 ETL

Extract, Transform, Load é o nome que se dá ao processo de replicar dados de uma fonte origem para uma fonte destino (onde são organizados de forma distinta). Neste processo, os dados são extraídos da fonte origem, transformados conforme necessário e carregados para a fonte destino. Em primeiro lugar, a extração da informação depende do tipo da fonte. Caso seja uma base de dados, é um processo trivial, bastando apenas a conexão com a BD e, de seguida, copiar os dados. O processo apenas segue para a próxima fase quando a extração estiver completa. É na Transformação que são aplicadas um conjunto de regras e condições aos dados previamente copiados, de modo a mudar a sua estrutura ou conteúdo - podendo também integrar dados de várias fontes numa só. Finalmente, os dados transformados são carregados para o DW, permitindo que sejam feitas queries sobre o mesmo - o objetivo principal de um DW.

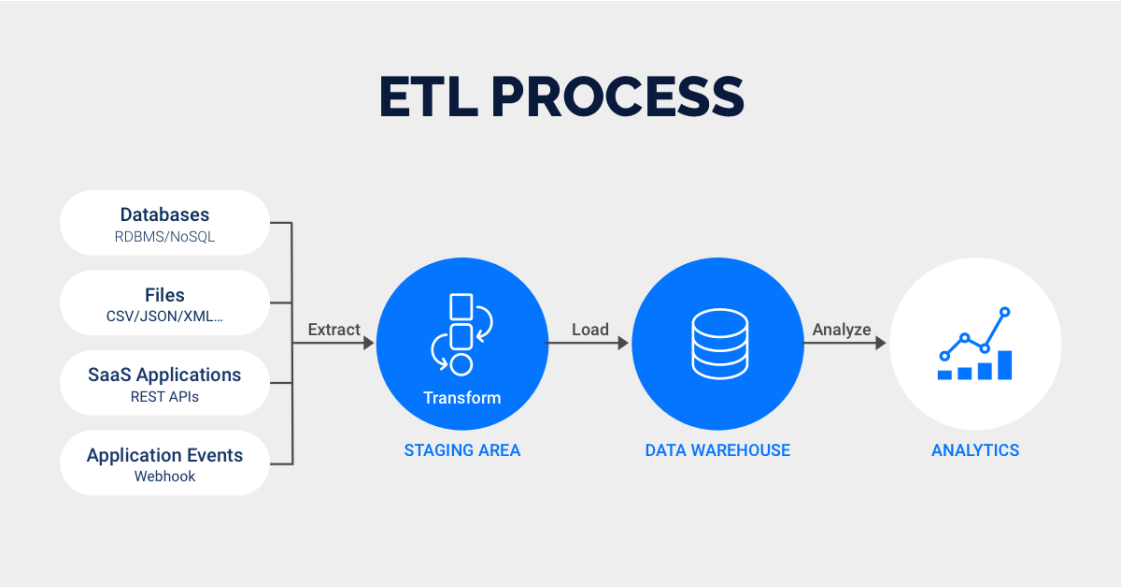


Figura 2.3: Processo de ETL [2]

De modo a facilitar o processo de desenvolvimento dos Data Warehouses, surgiram várias ferramentas que permitem a implementação do processo ETL sem que seja preciso escrever qualquer linha de código - tornando mais simples e intuitivo a compreensão para não programadores. Outra vantagem destes programas é o facto de apresentam interfaces gráficas que ajudam a acelerar o processo de mapeamento entre tabelas e colunas das fontes de origem e destino.

A Pentaho Data Integration (PDI) é um exemplo de uma ferramenta (open-source) com capacidades de ETL e que facilita o processo de capturar, limpar e armazenar dados de uma fonte para outra. Foi criada pela Pentaho e é um dos muitos produtos desenvolvidos por esta empresa. Permite criar transformações - em ficheiros onde são descritos os passos do processo ETL - e jobs - onde é definida a ordem com que as transformações são executadas, automatizando todo o processo do início ao fim. Uma característica interessante do Pentaho Data Integration é o facto de permitir a calendarização da execução de um job. Por exemplo, é possível implementar um job que carregue um DW com dados de uma fonte de dados todos os dias às 8 da manhã, de modo a estar sempre atualizado.

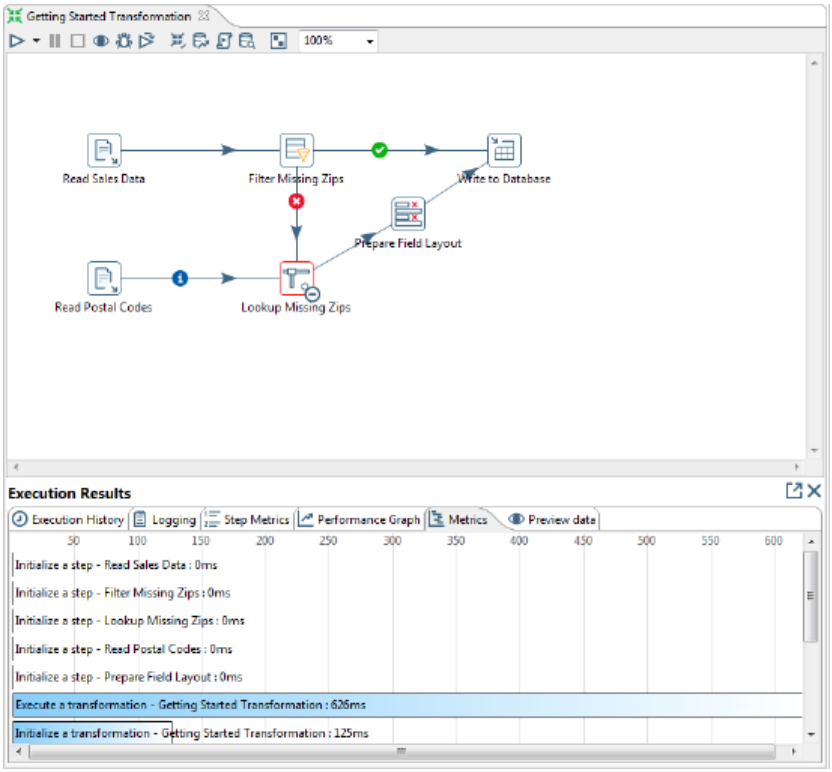


Figura 2.4: Exemplo de uma transformação implementada no PDI [2]. Neste caso, a transformação extrai informação relativa a vendas e códigos postais e prepara-a, completando alguns dados que não estão presentes e depois carrega-os para uma outra base de dados

2.5 Ferramentas de Visualização

Após a conclusão do processo de ETL, é importante visualizar os dados. Para este efeito existem uma séria de ferramentas de visualização que traduzem a informação presente nos Data Warehouses em gráficos, tabelas ou outras formas de visualização, que podem até ser interativos, permitindo navegar, filtrar, ordenar consoante a necessidade do utilizador. No contexto de qualquer organização, são estes relatórios que permitirão uma análise e tomada de decisão fundamentada.

2.5.1 Power BI

Um dos sistemas mais conhecidos e utilizados é o Power BI. Desenvolvido pela Microsoft, é constituído por um conjunto de serviços - Power BI Service, Power BI Desktop e Power BI Mobile [7]. O Power BI permite fazer uma conexão direta com uma ou várias fontes de dados (um DW, por exemplo), extraindo os seus dados para criar gráficos e dashboards interativos [11]. Para além da sua interface de utilizador simples e user-friendly, esta ferramenta disponibiliza uma grande variedade de métodos de visualização diferentes, como, por exemplo, gráficos de barras, gráficos de dispersão e bolhas e até mapas.

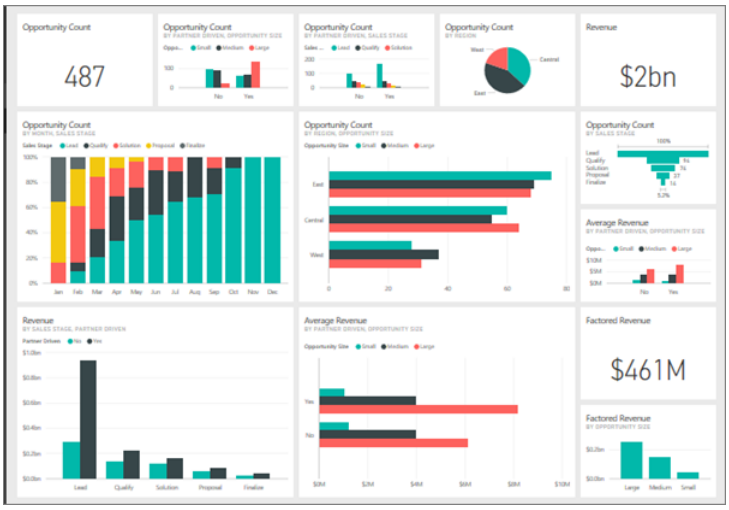


Figura 2.5: Exemplo de um dashboard desenvolvido no Power BI

2.5.2 Pentaho Report Designer

Outro exemplo é o Pentaho Report Designer, desenvolvido pela mesma empresa que foi referida na secção anterior. Por definição, tem uma integração direta com o PDI e com qualquer outro produto pertencente à Pentaho Suite[1], facilitando a extração de dados de qualquer passo de uma transformação. Gera reports em XML que podem incluir subreports, gráficos, imagens, etc.

No entanto, a sua interface de utilizador não é tão intuitiva e moderna como o Power BI e não apresenta tantas opções de visualização como outras ferramentas do tipo [12].

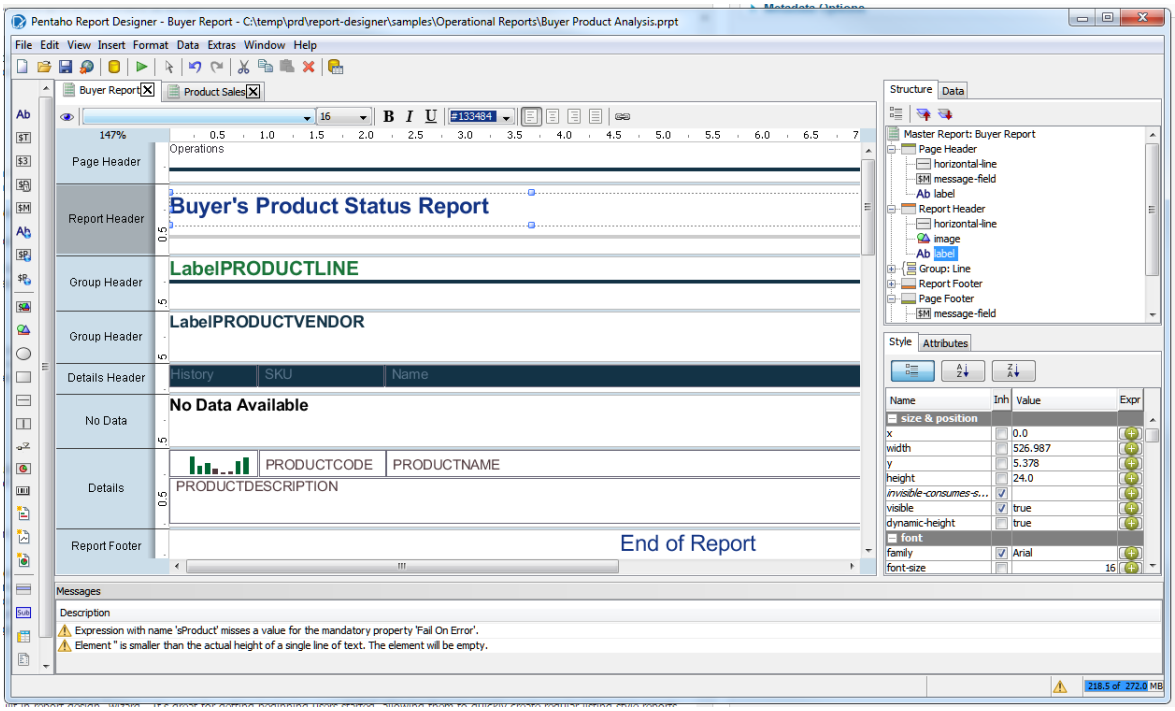


Figura 2.6: Interface de criação de um report em PRD

[1]https://xpansa.com/services/pentaho/

2.6 Link Consulting e edocLink

Este projeto insere-se no contexto da Link Consulting, mais especificamente da edoclink Enterprise. O seu produto, edoclink, é uma ferramenta de gestão documental e workflow que permite a desmaterialização de tarefas administrativas e de processos de decision-making. É um produto já utilizado em vários setores, tanto público como privado, como por exemplo Saúde, Educação, Seguros e Administração Pública [6]. Esta solução suporta as mais recentes plataformas da Microsoft, inclusivé uma série de funcionalidades capazes de satisfazer as necessidades das organizações (Análise de requisitos, consultadoria organizacional, suporte pós-produção, etc).

> O edoc é uma aplicação centrada na disponibilização de todas as funções do documento. O centro é o documento e a sua vida e não apenas o seu mero registo e pesquisa [5].

2.6.1 Conceitos Base

De forma a facilitar a compreensão da solução implementada, é importante primeiro esclarecer alguns conceitos que fazem parte do ambiente do edoclink:

- **Distribuição -** Atividade que constitui um processo de tomada de decisão, no qual intervém um dado conjunto de utilizadores. Percurso de tramitação de um ou vários documentos. Associada a um interveniente. Execução de um processo de negócio.
- **Tipo de Distribuição -** Definição do Processo de Negócio. Por exemplo, um processo de negócio de aquisições tem o tipo AQUISIÇÕES.
- **Etapa -** Atividade realizada por determinada pessoa (executante) no decorrer do processo de Negócio.
- **Formulário -** Conjunto de campos a preencher numa determinada etapa.
- **Percurso -** Conjunto de etapas que acontecem sempre independentemente do contexto de execução (excetuando a primeira etapa).
- **Interveniente -** Grupo/Pessoa associada a uma distribuição/etapa.
- **Executante -** Pessoa que fica responsável por executar uma etapa. Determinada pelo interveniente caso este seja um group. Se for uma pessoa, essa pessoa torna-se automaticamente o executante.
- **Data Recebida (in_date) -** Data em que a etapa é atribuída ao interveniente.
- **Data Aceite (read_date) -** Data em que a etapa é aceite pelo executante.
- **Data Enviada (out_date) -** Data em que o executante entrega a etapa. A etapa é dada por concluída.

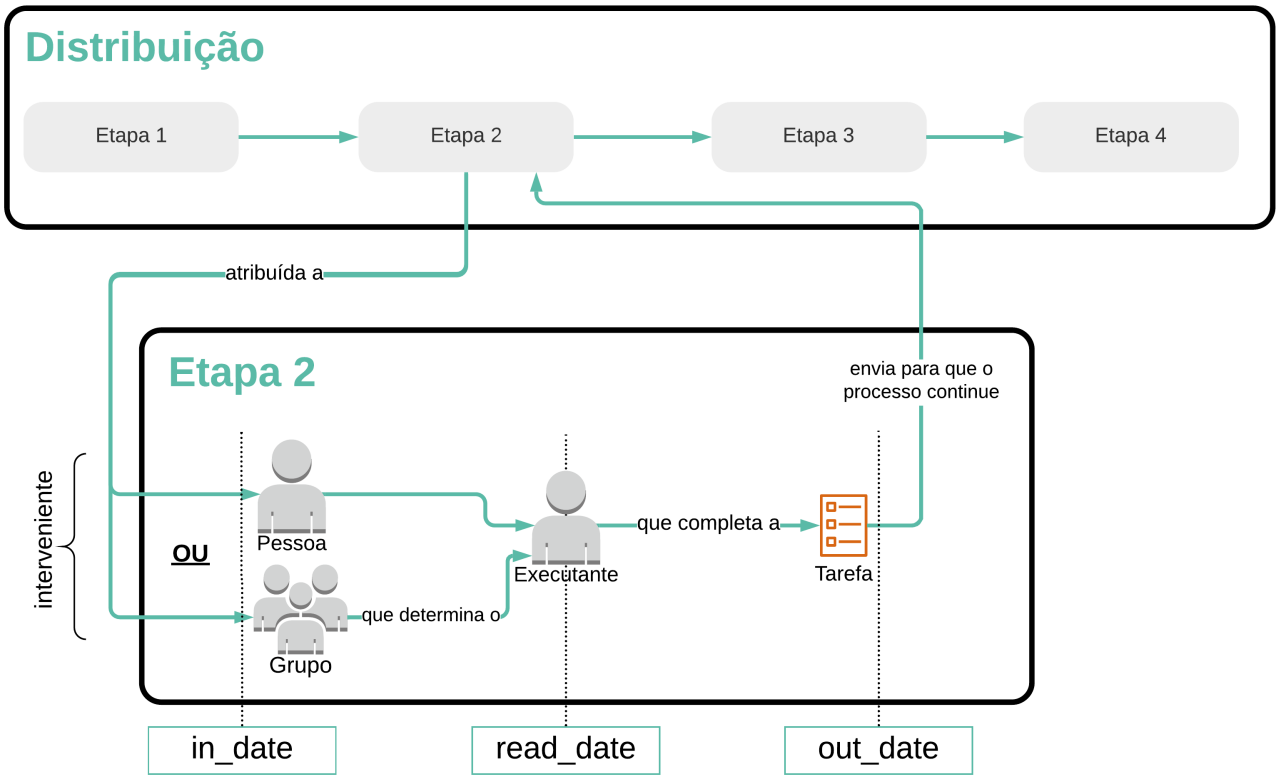


Figura 2.7: Funcionamento de um processo no edoclink

Capítulo 3

Solução proposta

3.1 Solução

A solução implementada tem como objetivo desenvolver um sistema capaz de responder a um conjunto de indicadores que ajudem a decidir a quem deverá ser alocada uma determinada etapa no edoc. Por exemplo, uma etapa A, associada ao interveniente X, deverá ser alocada ao executante 1 (que tem menos tarefas pendentes) ou ao executante 2 (que executou mais tarefas nos últimos 30 dias)? Os indicadores servirão como base para depois se tomar este tipo de decisões.

Nesta secção, serão detalhados todos os componentes da solução, incluindo desafios que foram sendo encontrados e a forma como se os ultrapassou.

3.1.1 Arquitetura e Tecnologias utilizadas

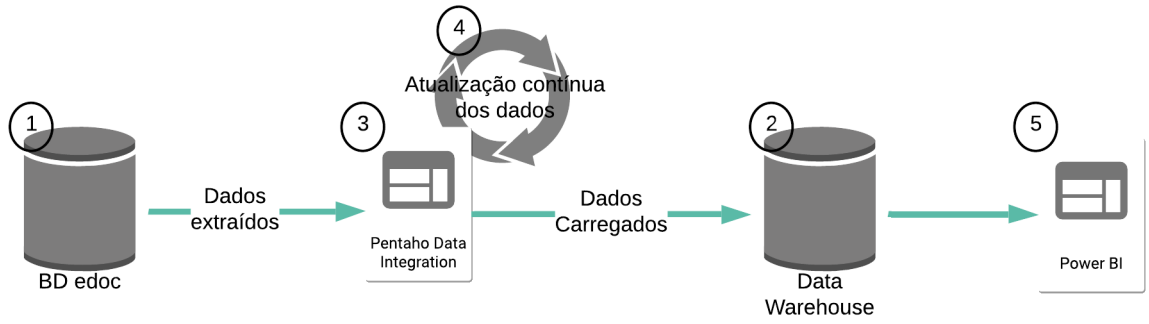


Figura 3.1: Arquitetura da solução implementada. A ordenação é relativa à ordem com que os diferentes componentes serão detalhados ao longo do documento

Conforme a arquitetura e ordenação apresentada anteriormente, segue a descrição de cada componente da arquitetura.

1. Base de dados do edoc

Infelizmente, por uma questão de proteção de dados, não foi possível ter acesso a uma base de dados com dados gerados através dos processos de negócio de um cliente. A BD utilizada foi então uma cópia de uma BD de testes disponibilizada pela

edoclink, por isso é expectável que os resultados obtidos não sejam representativos da realidade.

É uma BD em SQLServer. Tem centenas de tabelas e, como tal, foi primeiro preciso analisar as tabelas e perceber quais as que seriam importantes para o projeto a desenvolver. Foi decidido que o evento a registar na tabela de factos seria cada etapa de uma distribuição, portanto a tabela principal foi a DISTRIBUTION_STAGES que contém a maior parte da informação relativa a cada etapa. No entanto, foram usadas mais algumas tabelas de modo a conseguir a obter toda a informação necessária para responder aos indicadores definidos.

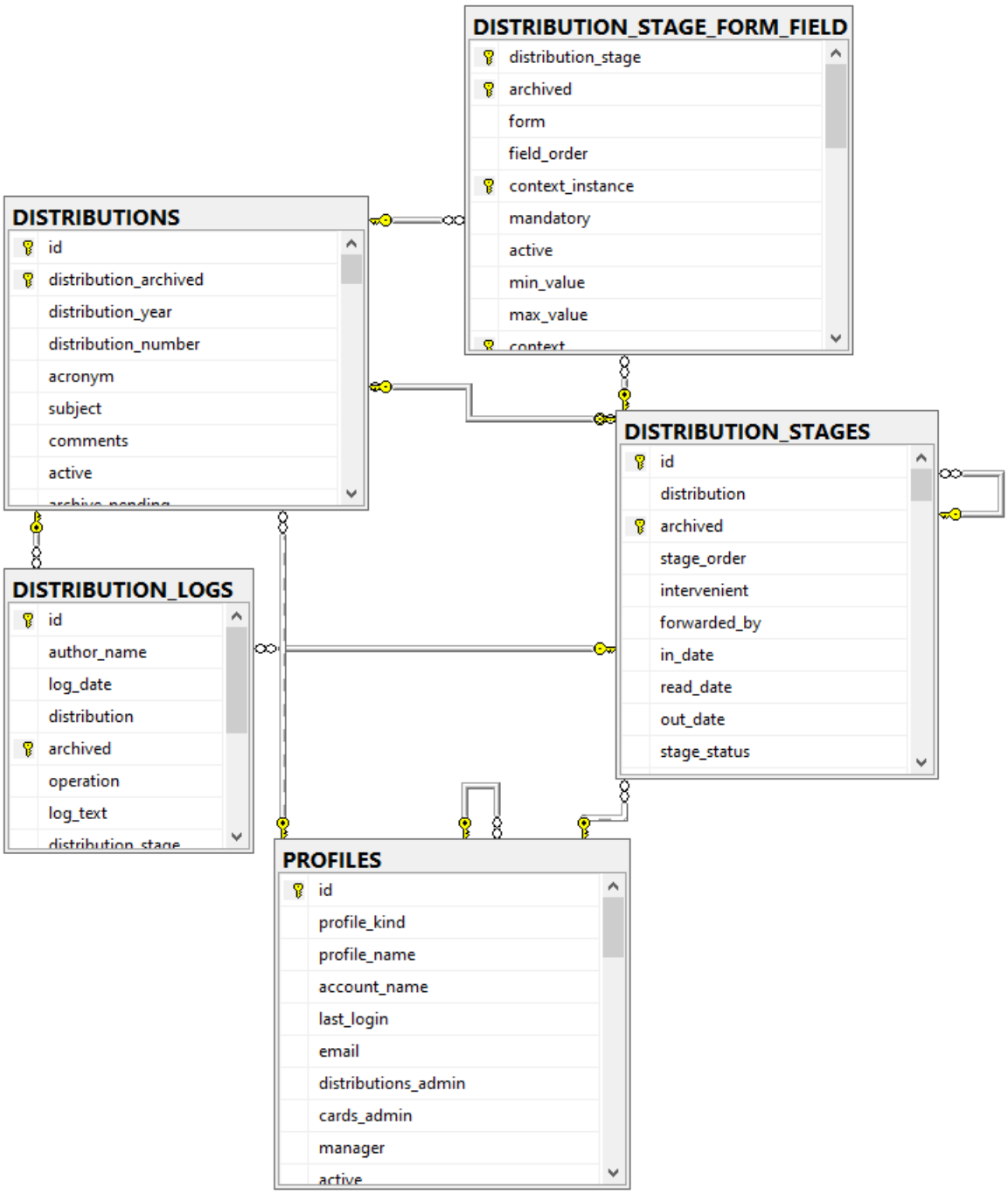


Figura 3.2: Diagrama das tabelas da BD usadas

2. Data Warehouse

O Data Warehouse criado segue uma estrutura de esquema em estrela. Apresenta 6 dimensões e uma tabela de factos com as medidas *tempoAceitacao*, *tempoExecucao* e *tempoEtapa* que ajudarão a responder aos objetivos definidos inicialmente.

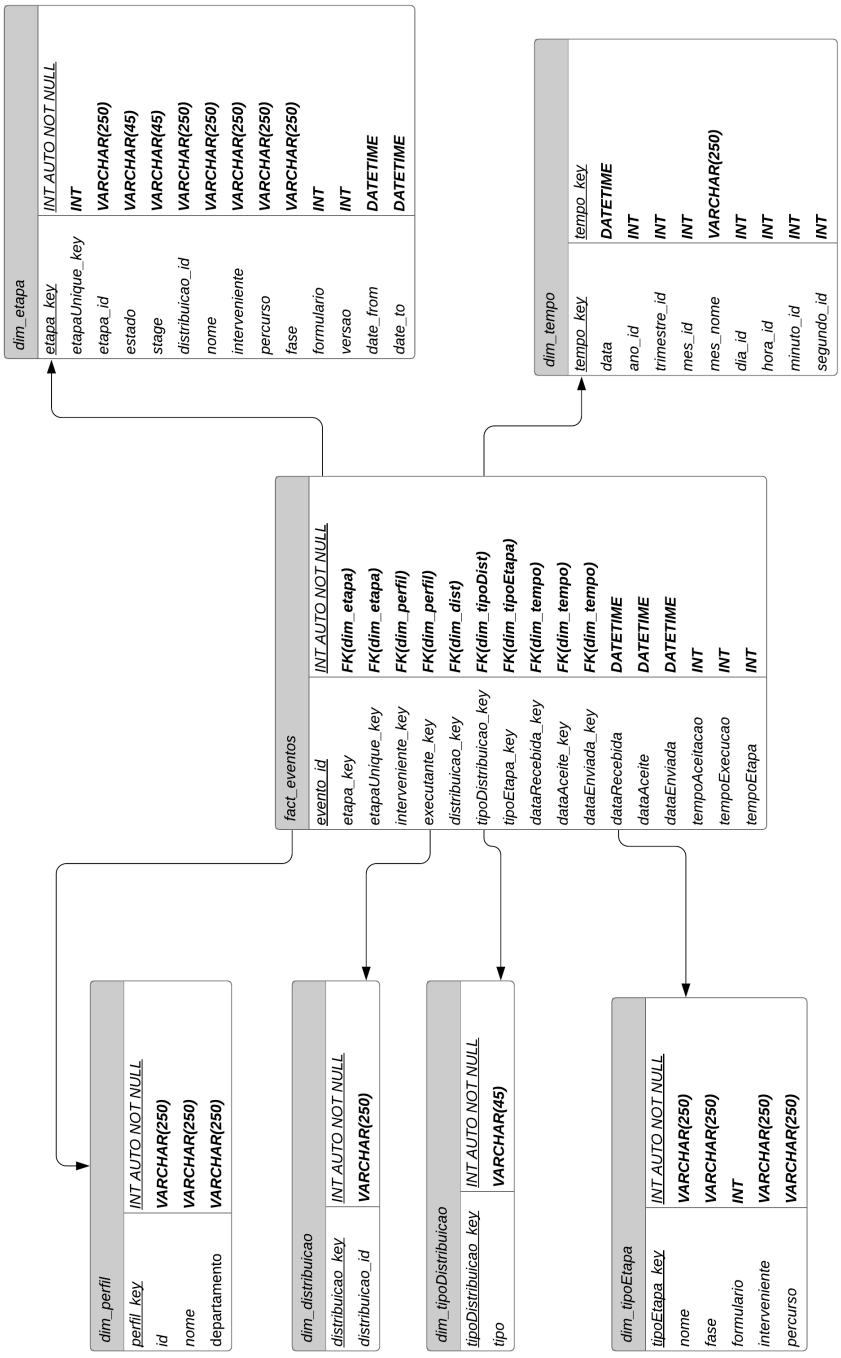


Figura 3.3: Diagrama do Data Warehouse desenvolvido

Acima está apresentada a versão final do DW. No entanto, o DW foi sofrendo algumas alterações ao longo do desenvolvimento. Algumas das mudanças foram:

- Alterou-se de duas dimensões (executante e interveniente) para uma que continha ambos (perfil), uma vez que os atributos eram os mesmos e se podia fazer a distinção na tabela de factos com uma chave estrangeira para cada.

- Durante o desenvolvimento, percebeu-se que as etapas podiam ser rejeitadas e executadas novamente. Como tal, mudou-se a dimensão das etapas para uma *Slowly Changing Dimension (SCD)*[1].

- Adicionaram-se os atributos **Versao**, **date_from** e **date_to** à dim_etapa como consequência da alteração para SCD.

- Adicionou-se o atributo **etapaUnique_key** também à dim_etapa para que fossem reconhecidas etapas executadas novamente. Este reconhecimento seria feito através da comparação entre vários atributos (**interveniente**, **distribuicao**, **percurso**, **fase** e **formulario**)

O seguinte código em MySQL foi desenvolvido para criar o DW localmente.

[1]Uma *Slowly Changing Dimension* (SCD) é uma dimensão que guarda e gere tanto o estado corrente de uma entrada no DW como os estados que foi tendo ao longo do tempo

```sql
DROP DATABASE IF EXISTS teste1_dw;
CREATE DATABASE teste1_dw;

USE teste1_dw;

CREATE TABLE dim_perfil (
    perfil_key INT NOT NULL AUTO_INCREMENT,
    id VARCHAR(250),
    nome VARCHAR(2500),
    departamento VARCHAR(250),
    PRIMARY KEY (perfil_key)

);
CREATE TABLE dim_distribuicao (
    distribuicao_key INT NOT NULL AUTO_INCREMENT,
    distribuicao_id VARCHAR(250),
    PRIMARY KEY (distribuicao_key)
);
CREATE TABLE dim_tipoDistribuicao (
    tipoDistribuicao_key INT NOT NULL AUTO_INCREMENT,
    tipo VARCHAR(45),
    PRIMARY KEY (tipoDistribuicao_key)
);
CREATE TABLE dim_etapa (
    etapa_key INT NOT NULL AUTO_INCREMENT,
    etapaUnique_key INT,
    etapa_id VARCHAR(250),
    estado VARCHAR(45),
    stage VARCHAR(45),
    distribuicao_id VARCHAR(250),
    nome VARCHAR(250),
    interveniente VARCHAR(250),
    percurso VARCHAR(250),
    fase VARCHAR(250),
    formulario INT,
    versao INT,
    date_from DATETIME,
    date_to DATETIME,
    PRIMARY KEY (etapa_key)
);
CREATE TABLE dim_tipoEtapa (
    tipoEtapa_key INT NOT NULL AUTO_INCREMENT,
    nome VARCHAR(250),
    fase VARCHAR(250),
    formulario INT,
    interveniente VARCHAR(250),
    percurso VARCHAR(250),
    PRIMARY KEY (tipoEtapa_key)
);
CREATE TABLE dim_tempo (
    tempo_key INT NOT NULL AUTO_INCREMENT,
    data DATETIME,
    ano_id INT,
    trimestre_id INT,
    mes_id INT,
    mes_nome VARCHAR(255),
    dia_id INT,
    hora_id INT,
    minuto_id INT,
    segundo_id INT,
    PRIMARY KEY (tempo_key)
);
CREATE TABLE fact_eventos (
    evento_id INT NOT NULL AUTO_INCREMENT,
    etapa_key INT,
    etapaUnique_key INT,

    interveniente_key INT,
    executante_key INT,

    distribuicao_key INT,
    tipoDistribuicao_key INT,
    tipoEtapa_key INT,

    dataRecebida_key INT,
    dataRecebida DATETIME,
    dataAceite_key INT,
    dataAceite DATETIME,
    dataEnviada_key INT,
    dataEnviada DATETIME,

    tempoAceitacao INT,
    tempoExecucao INT,
    tempoEtapa INT,

    PRIMARY KEY (evento_id),
    FOREIGN KEY (etapa_key)
        REFERENCES dim_etapa (etapa_key),

    FOREIGN KEY (interveniente_key)
        REFERENCES dim_perfil (perfil_key),
    FOREIGN KEY (executante_key)
        REFERENCES dim_perfil (perfil_key),

    FOREIGN KEY (distribuicao_key)
        REFERENCES dim_distribuicao (distribuicao_key),
    FOREIGN KEY (tipoDistribuicao_key)
        REFERENCES dim_tipoDistribuicao (tipoDistribuicao_key),
    FOREIGN KEY (tipoEtapa_key)
        REFERENCES dim_tipoEtapa (tipoEtapa_key),

    FOREIGN KEY (dataRecebida_key)
        REFERENCES dim_tempo (tempo_key),
    FOREIGN KEY (dataAceite_key)
        REFERENCES dim_tempo (tempo_key),
    FOREIGN KEY (dataEnviada_key)
        REFERENCES dim_tempo (tempo_key)
);
```

Figura 3.4: Script em SQL que cria o DW

3. Pentaho Data Integration

O PDI foi a ferramenta de ETL escolhida para o projeto. Não só apresentava todas as funcionalidades necessárias, como também já tinha sido utilizada previamente na cadeira de Análise e Integração de Dados do Mestrado em Engenharia Informática e de Computadores. Este componente é o core do projeto. É aqui que se executa a extração dos dados da BD do edoc e se transformam os dados para serem depois carregados para o Data Warehouse desenvolvido. Foi desenvolvida uma transformação para cada dimensão e um job para executar diariamente todas as transformações sequencialmente.

Transformações

Dim_distribuicao

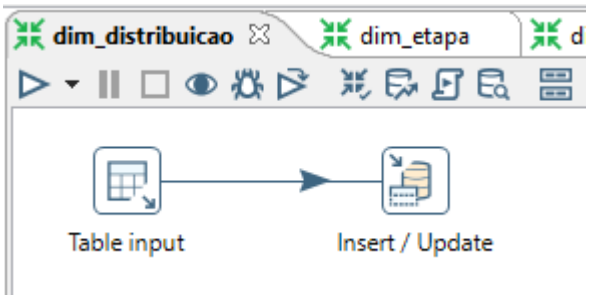


Figura 3.5: Transformação da dim_distribuicao

A dimensão das distribuições tem apenas 1 atributo que não precisa de qualquer modificação para ser carregado para o DW. Como tal, a transformação para esta dimensão é simples, apresentando apenas 2 passos. Um table input para extrair os dados diretamente da fonte origem e um Insert/Update para carregar os dados para o DW (verifica se o **distribuicao_id** obtido no passo anterior já existe no DW. Se sim, atualiza. Se não, insere no DW).

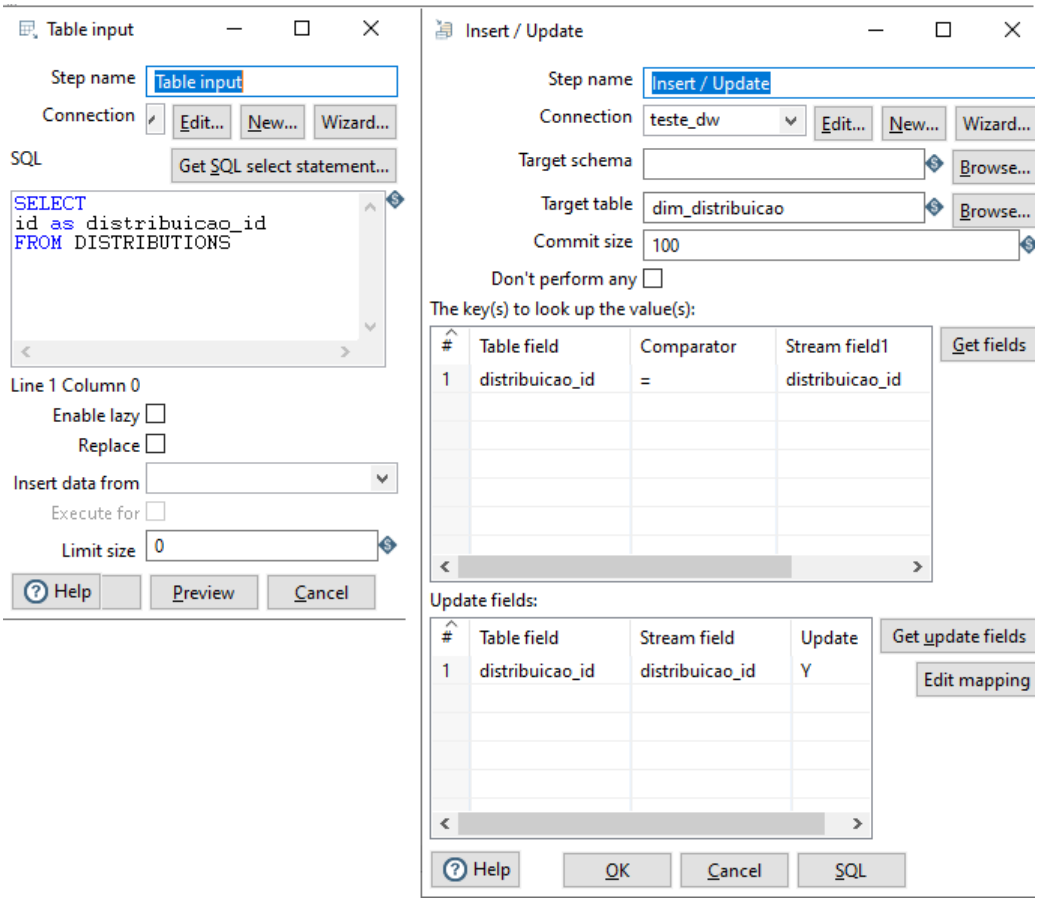


Figura 3.6: Configuração dos passos da Transformação da dim_distribuicao

Dim_tipoDistribuicao

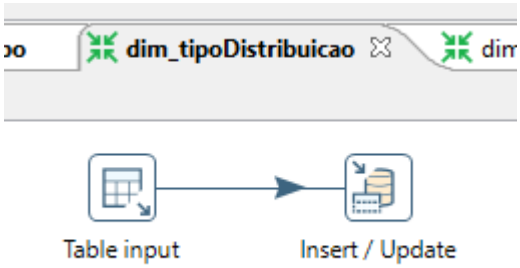


Figura 3.7: Transformação da dim_tipoDistribuicao

À semelhança da transformação anterior, a dimensão do tipo das distribuições tem apenas 1 atributo que pode ser carregado diretamente para o Data Warehouse. Foi então implementado um table input, seguido de um Insert/Update para carregar ou atualizar os dados do DW.

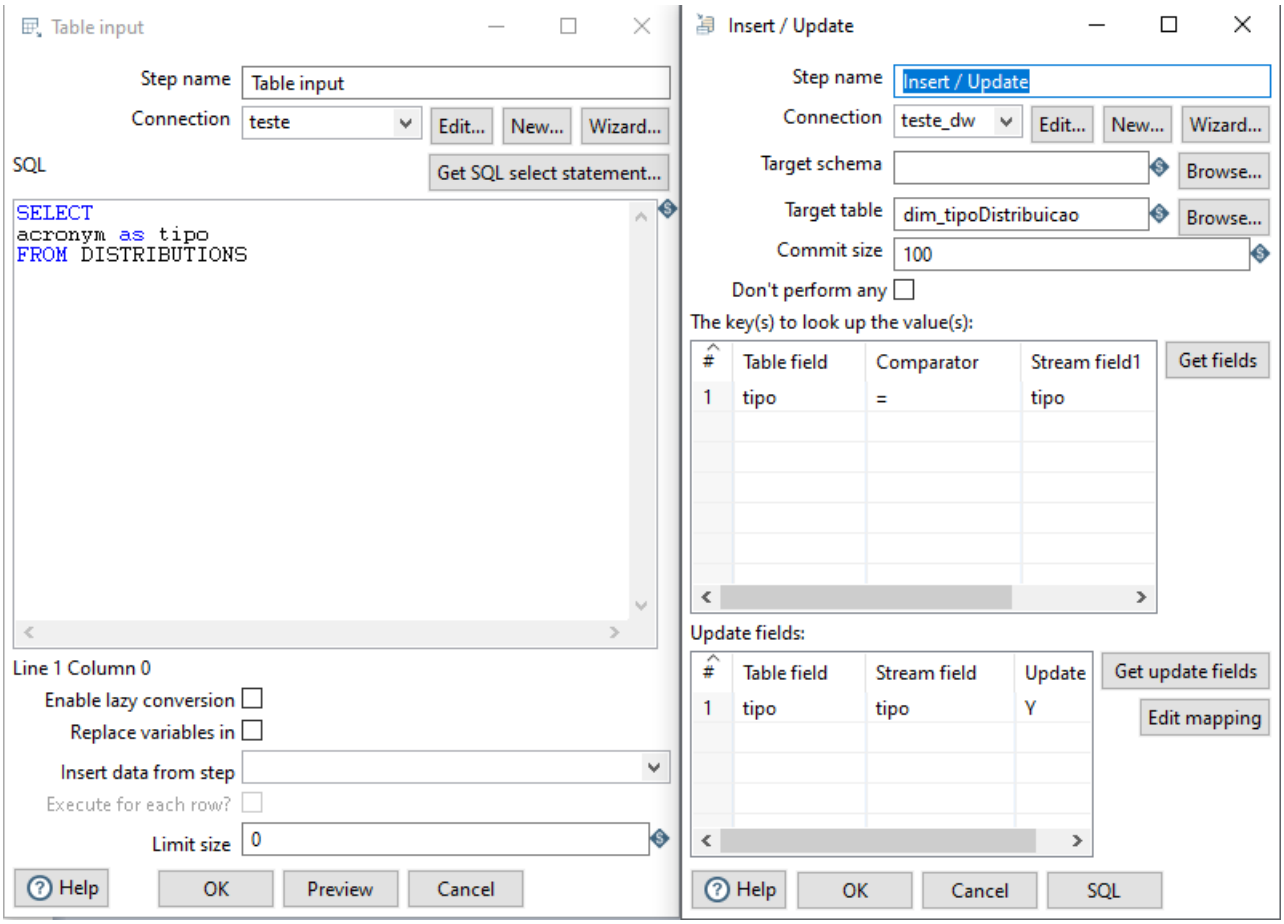


Figura 3.8: Configuração dos passos da Transformação da dim_tipoDistribuicao

Dim_perfil

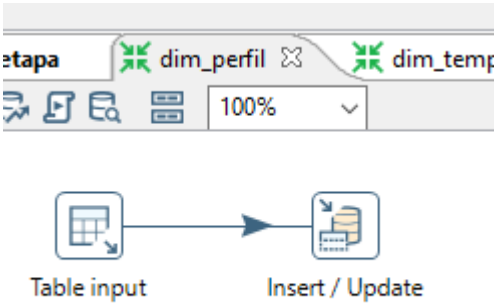


Figura 3.9: Transformação da dim_perfil

Como referido na secção do DW, a dimensão dos perfis surgiu numa fase mais avançada, após inicialmente se ter pensado ter uma dimensão para os intervenientes e outra para os executantes. No entanto, decidiu-se alterar a diferenciação entre esses 2 tipos de perfis para a transformação da tabela de factos e ter apenas uma dimensão onde estariam todos os perfis presentes na base de dados do edoc. Para tal, foram precisos novamente apenas 2 passos - 1 table input para extrair a informação necessária relativamente a cada perfil e um Insert/Update para carregar o DW.

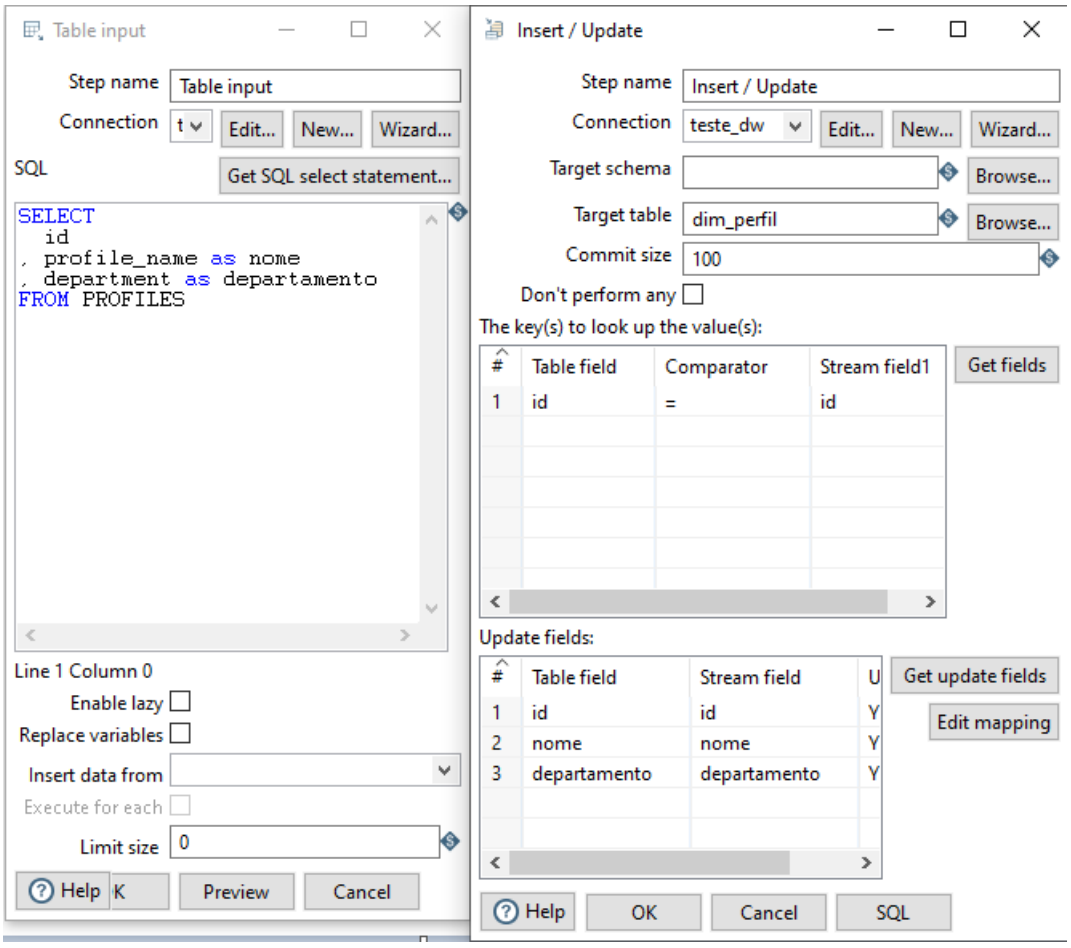


Figura 3.10: Configuração dos passos da Transformação da dim_perfil

Dim_etapa

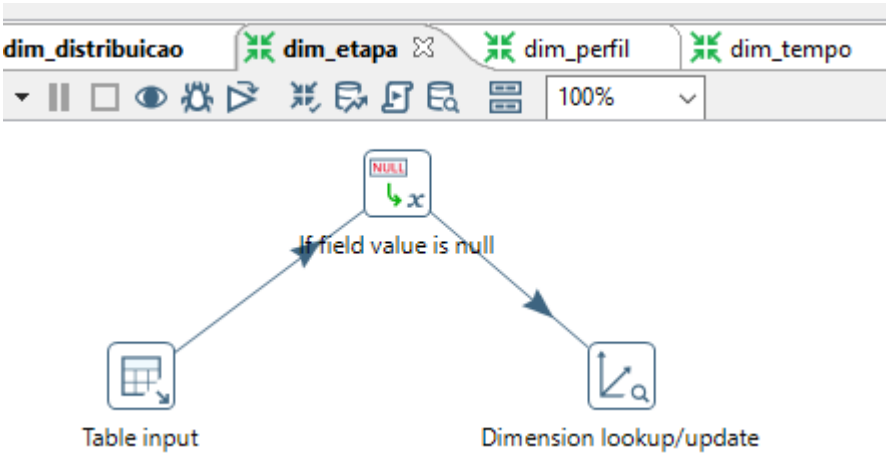


Figura 3.11: Transformação da dim_etapa

A dimensão do perfil também foi uma das que foi alterada ao longo do projeto - tornou-se uma SCD e adicionaram-se vários atributos. A mudança para SCD implicou a adição dos atributos **versao**,**date_from** e **date_to**. Outra adição importante foi o reconhecimento de etapas que estariam a ser executadas novamente. Os critérios usadaos para tal foram:

- **Distribuição**, **Nome**, **Percurso** e **Interveniente** têm de ser iguais. Se algum desses atributos for null numa das etapas, não se pode tirar nenhuma conclusão - consideram-se etapas diferentes.

- Se **Fase** ou **Formulário** não forem null, também têm de ser iguais. Caso sejam null em ambas as etapas, consideram-se etapas iguais.

Assim, a transformação precisou de 3 passos:

- Table Input onde se extraiu as informações necessárias da BD do edoc.

- If field value is null onde se substituíam os valores de **fase** e **formulario** por "1"e "2"respetivamente. Isto foi necessário uma vez que o PDI não consegue comparar null com null, então dar um valor concreto tornou possível comparar estes atributos entre 2 etapas.

- Dimension Lookup/update que é semelhante ao Insert/Update utilizado em todas as outras transformações mas permite configurar os atributos relativos a ser uma SCD.

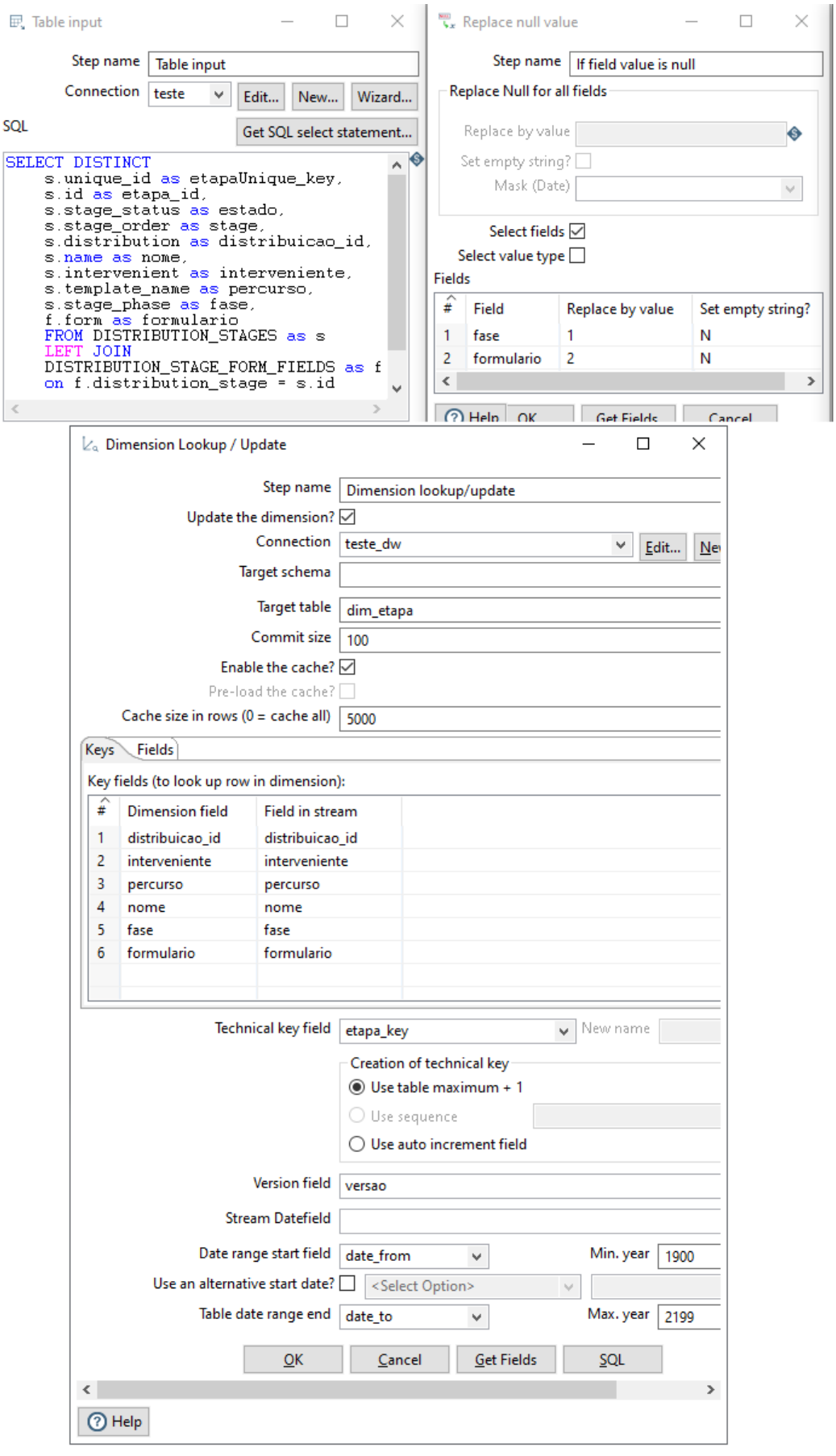


Figura 3.12: Configuração dos passos da Transformação da dim_etapa

Dim_tipoEtapa

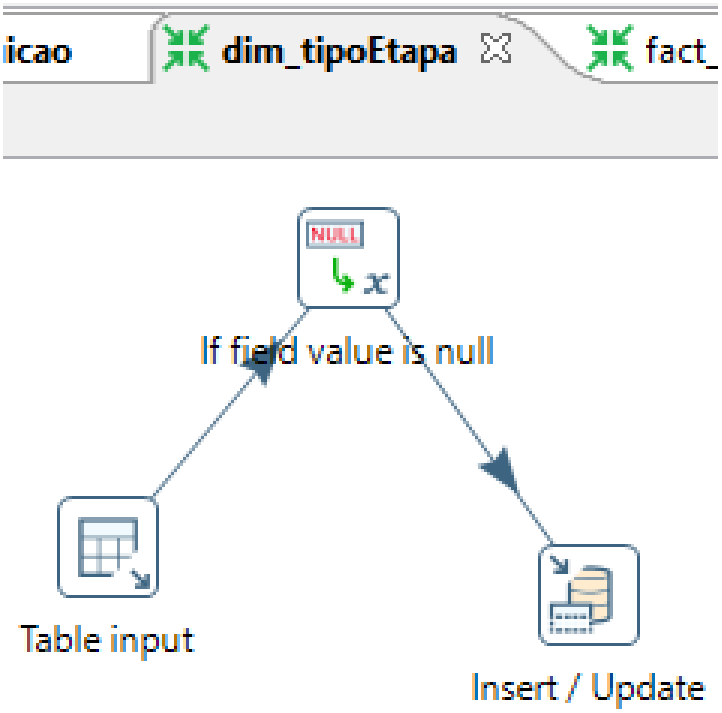


Figura 3.13: Transformação da dim_tipoEtapa

A transformação da dimensão do tipo de uma etapa é muito semelhante à transformação anterior. A única diferença é o facto de se considerarem tipos de etapa iguais, etapas que também tenham o **percurso** e/ou **nome** null (para além dos atributos **fase** e **formulario** da transformação anterior).

A transformação tem então 3 passos:

- Table Input onde se extraiu as informações necessárias da BD do edoc.
- If field value is null onde se substituíam os valores de **fase**, **formulario**, **percurso** e **nome** por "1", "2", "3"e "4"respetivamente.
- Insert/Update para carregar ou atualizar dados no DW.

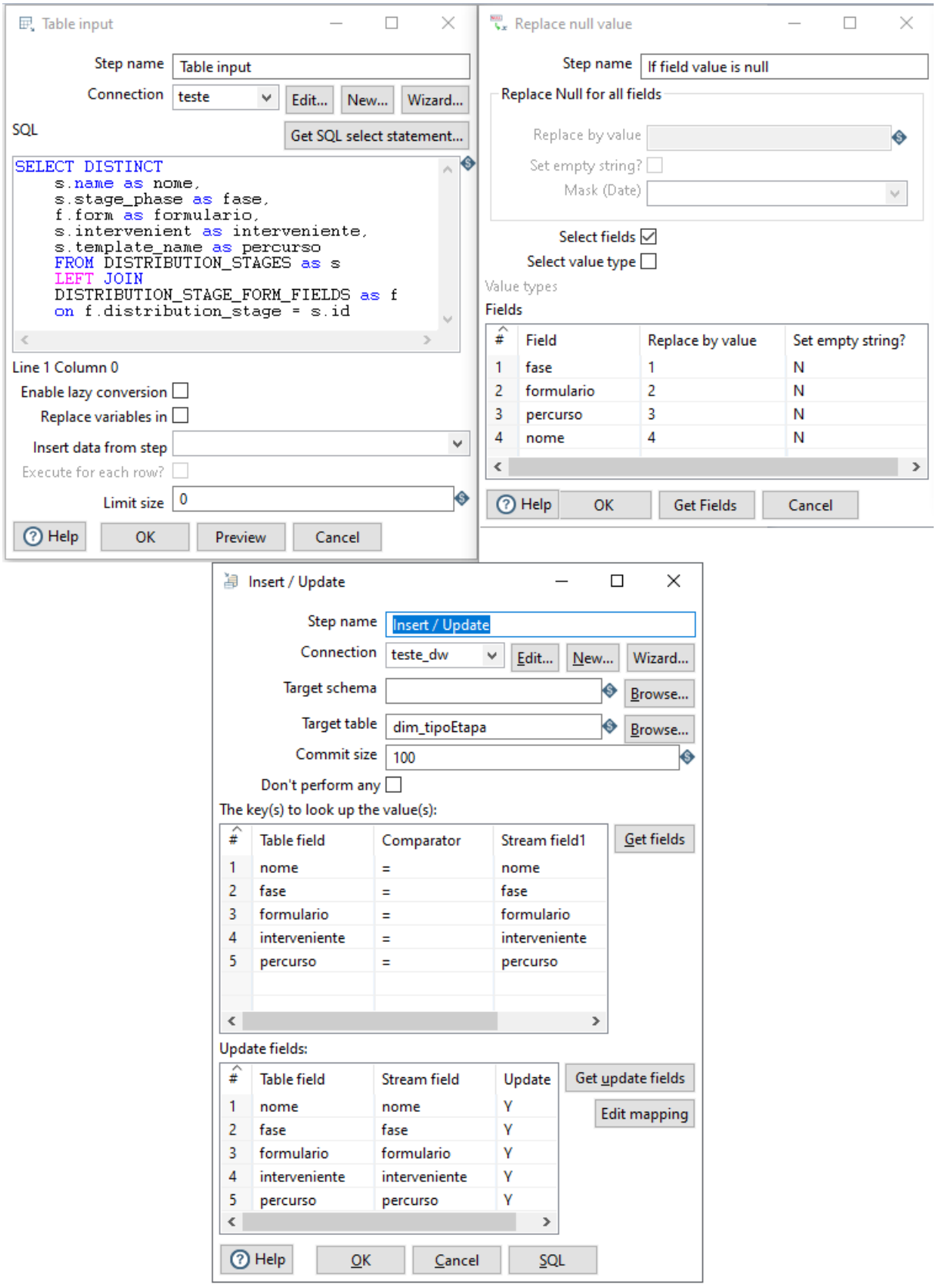


Figura 3.14: Configuração dos passos da Transformação da dim_tipoEtapa

Dim_tempo

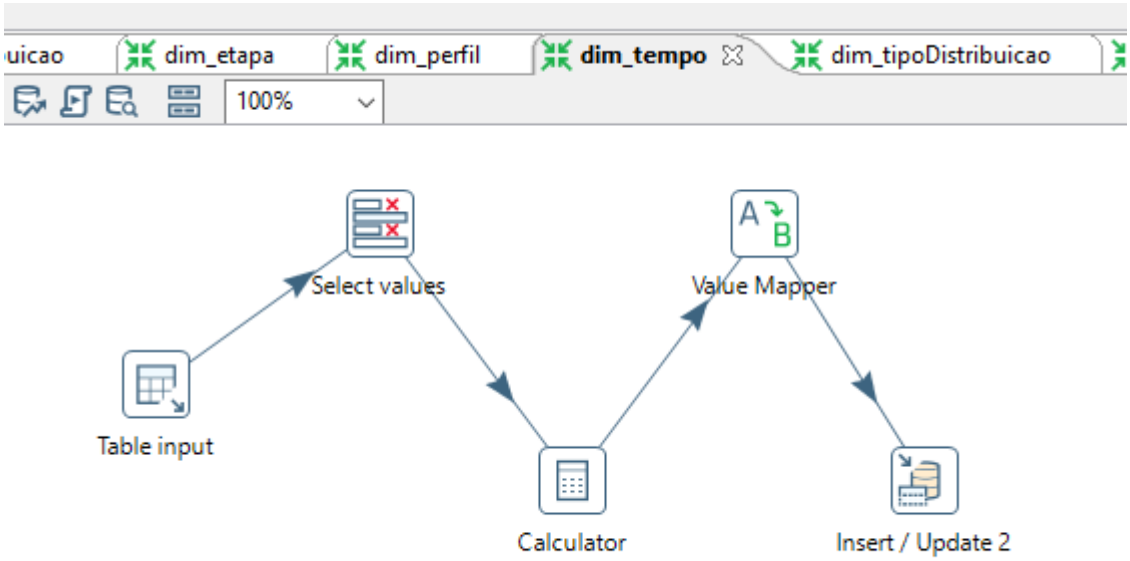


Figura 3.15: Transformação da dim_tempo

Para a transformação da dimensão do tempo, a decisão mais importante a tomar foi que datas seriam necessárias extrair da BD - à medida que se ia desenvolvendo o projeto, algumas datas iam sendo adicionadas consoante necessidade. Depois de decidido, as datas sofreram algum tipo de tratamento para depois seram carregadas para o DW. Esta transformação apresenta 5 passos:

- Table input onde foram extraídas todas as datas necessárias da BD do edoc.
- Select values que transformou o formato das datas vindas da BD no formato pretendido (yyyy-MM-dd HH:mm:ss).
- Calculator que para cada data calculou o ano, trimestre, mês, dia, hora, minuto e segundo.
- Value Mapper que fez a correspondência entre o mês obtido no passo anterior e o nome do mês.
- Insert/Update que carregou todos os dados obtidos pelos passos anterior para o Data Warehouse.

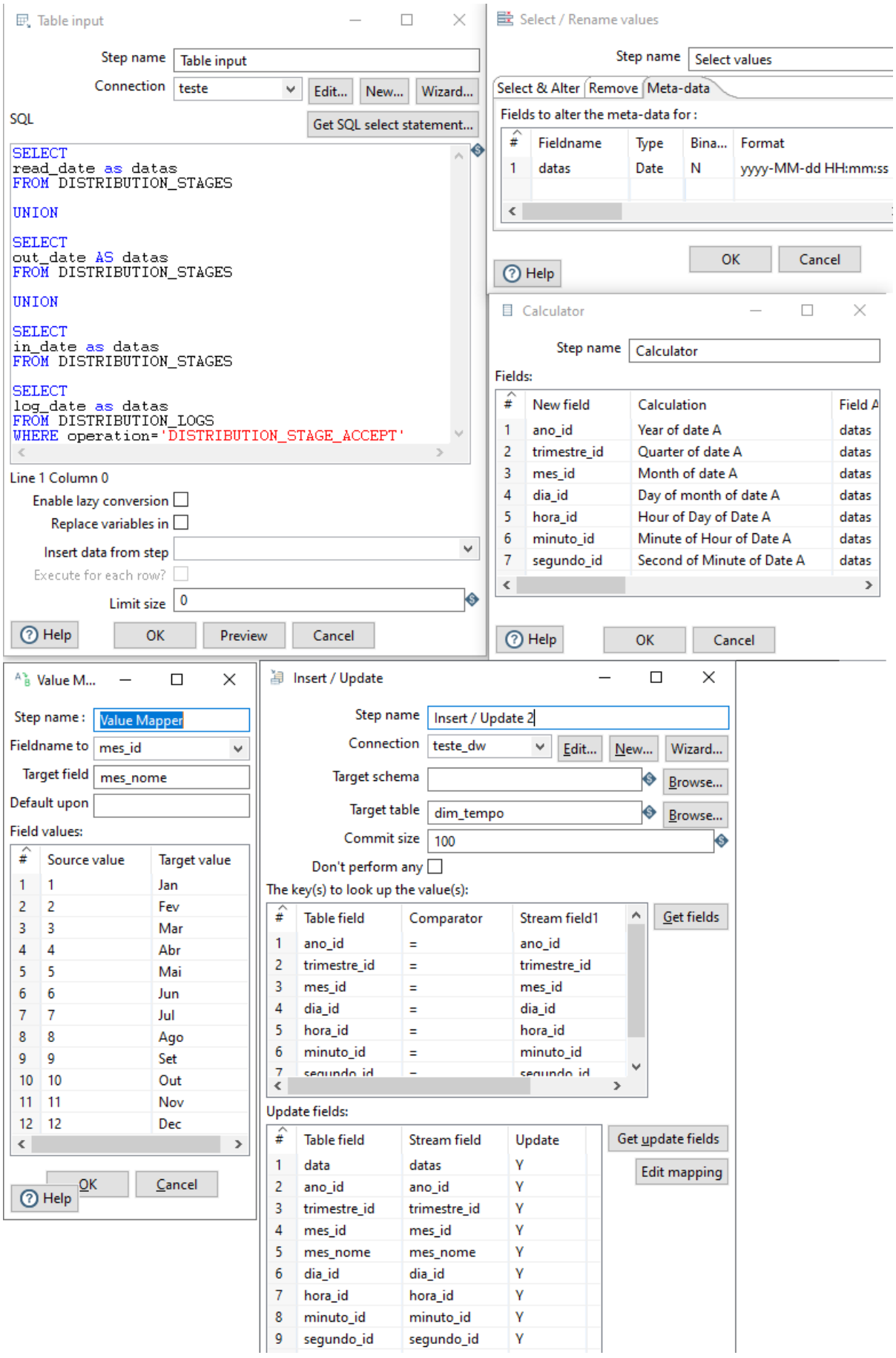


Figura 3.16: Configuração dos passos da Transformação da dim_tempo

Fact_eventos

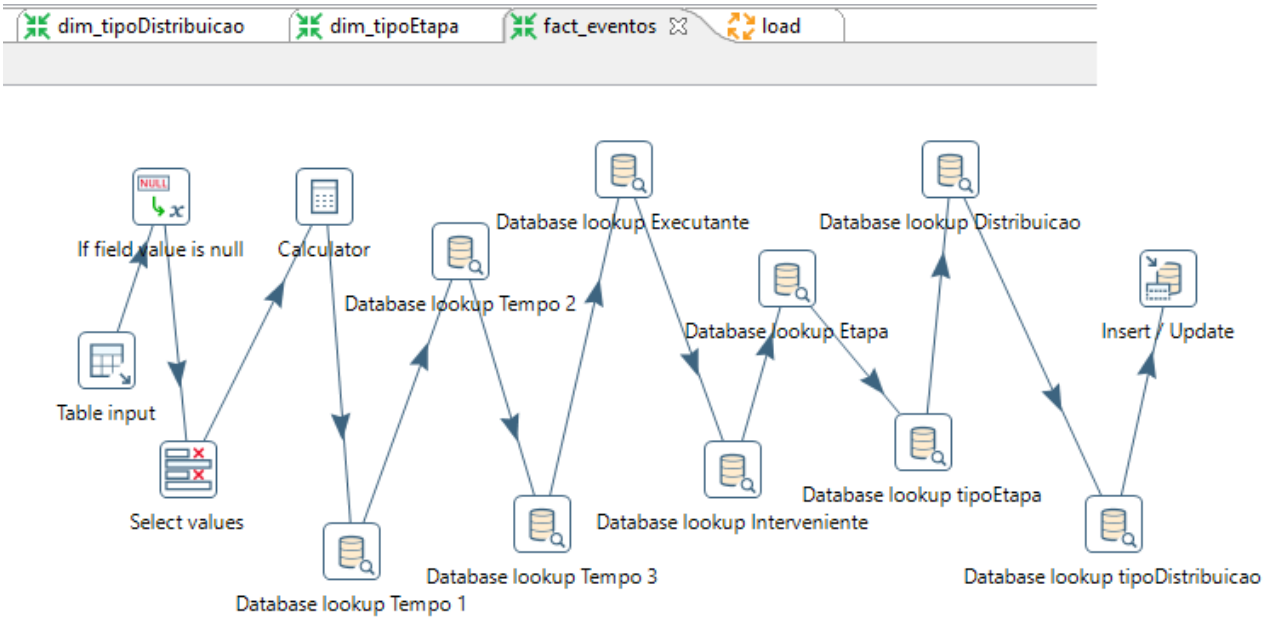


Figura 3.17: Transformação da fact_eventos

Nota: Critérios usados para definição de executante e dataAceite de uma etapa:

- **Executante -** Caso a etapa tenha sido entregue, o executante é quem a entregou. Caso não tenha sido entregue mas tenha sido aceite, o executante é quem a aceitou. Caso não tenha sido nem entregue nem aceite, se a etapa já tiver sido iniciada e o interveniente for uma pessoa e não um grupo, o executante é o inteveniente. Caso contrário, considera-se que não existe executante.
- **dataAceite -** Caso exista uma log_date com o estado "ACEITE", essa será considerada a dataAceite. Caso não, a dataAceite será a data em que a etapa foi lida pelo executante. Caso ambas não existam, considera-se que não existe dataAceite.

A transformação da tabela de factos é a que apresenta mais passos, uma vez que é aqui que é feita a relação entre todas as dimensões. Assim, foram necessários os seguintes passos:

- Table input que extraiu os dados necessários. Foi aqui que foram definidos o executantes e dataAceite de cada etapa (segundo o critério apresentada na nota acima).

- If field value is null que é equivalente ao mesmo passo na transformação da dimensão do tipo da etapa.
- Select values que também é equivalente ao mesmo passo na transformação da dimensão do tempo.
- Calculator que para além de fazer o mesmo que o Calculator da transformação da dim_tempo, calcula também as measures **tempoAceitacao**, **tempoExecucao** e **tempoTotal**.
- Database Lookup para associar a chave primária de cada entrada de cada dimensão com a tabela de factos. No caso da dimensão dos perfis e da dimensão do tempo foram precisas 2 e 3 Database Lookup , respetivamente, para fazer a distinção entre executante e interveniente e entre dataRecebida, dataAceite e dataEnviada.
- Insert/Update para finalmente inserir os dados na tabela de factos.

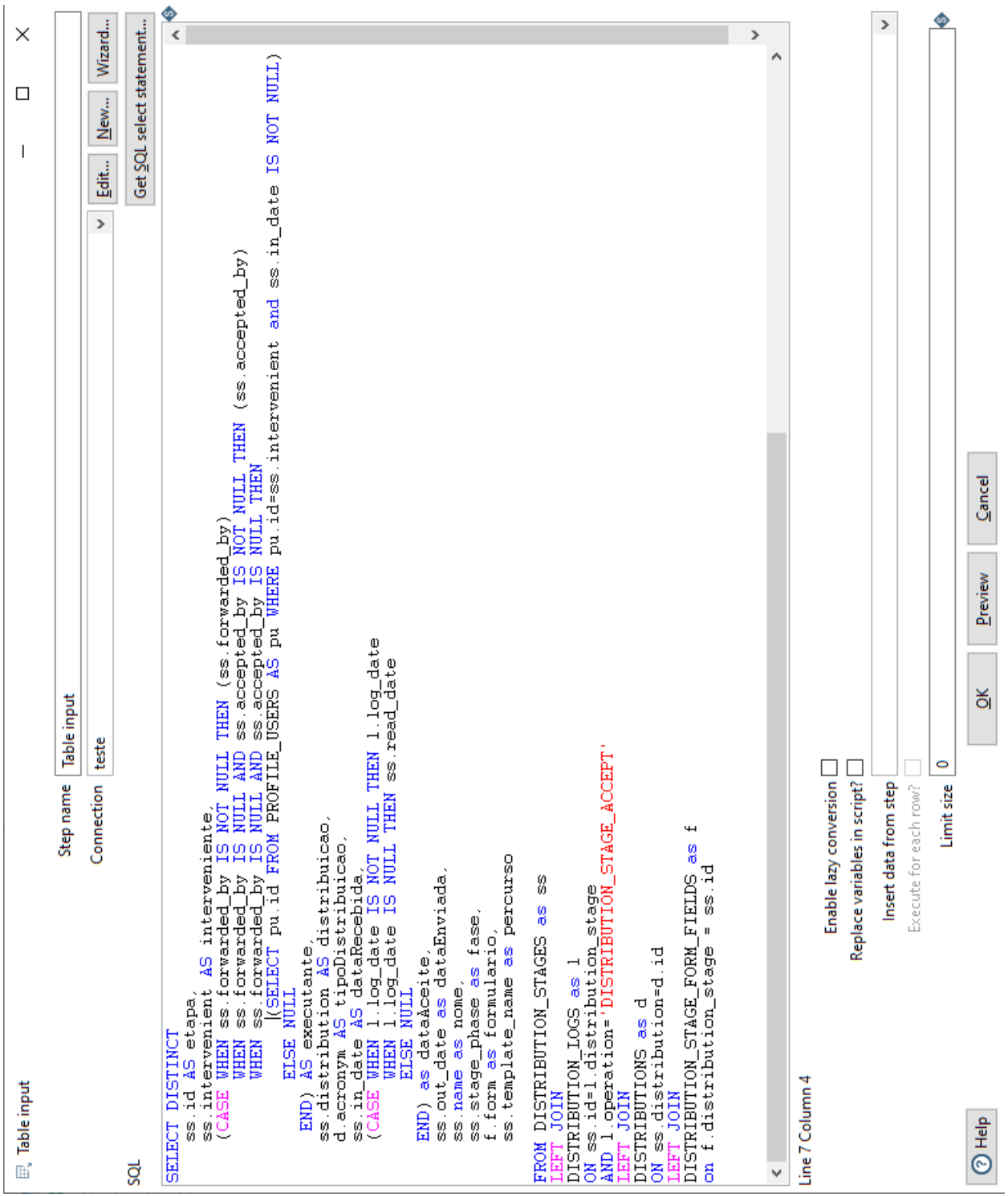


Figura 3.18: Configuração do Table Input da Transformação da fact_eventos

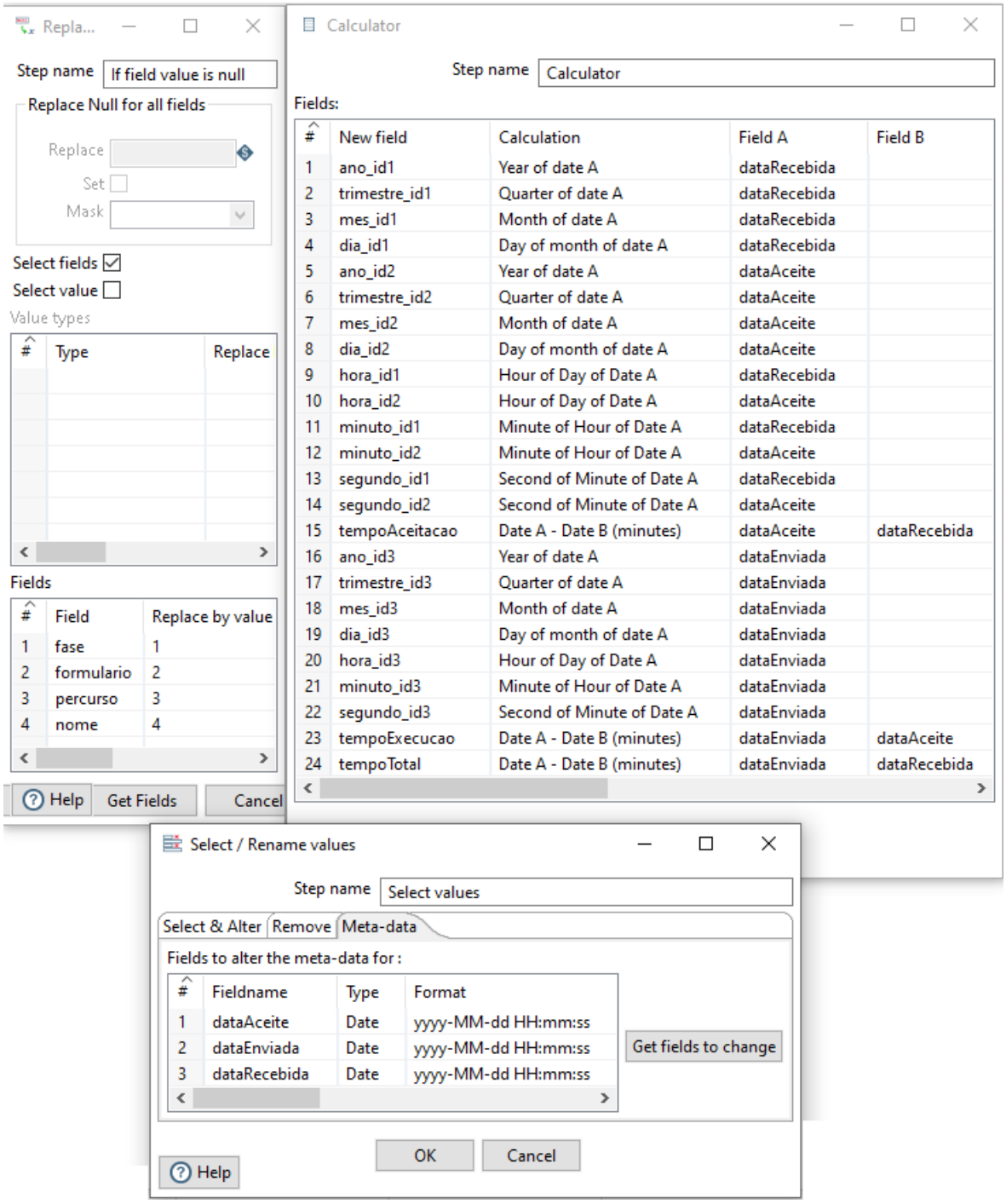


Figura 3.19: Configuração dos 2º, 3º e 4º passos da Transformação da fact_eventos

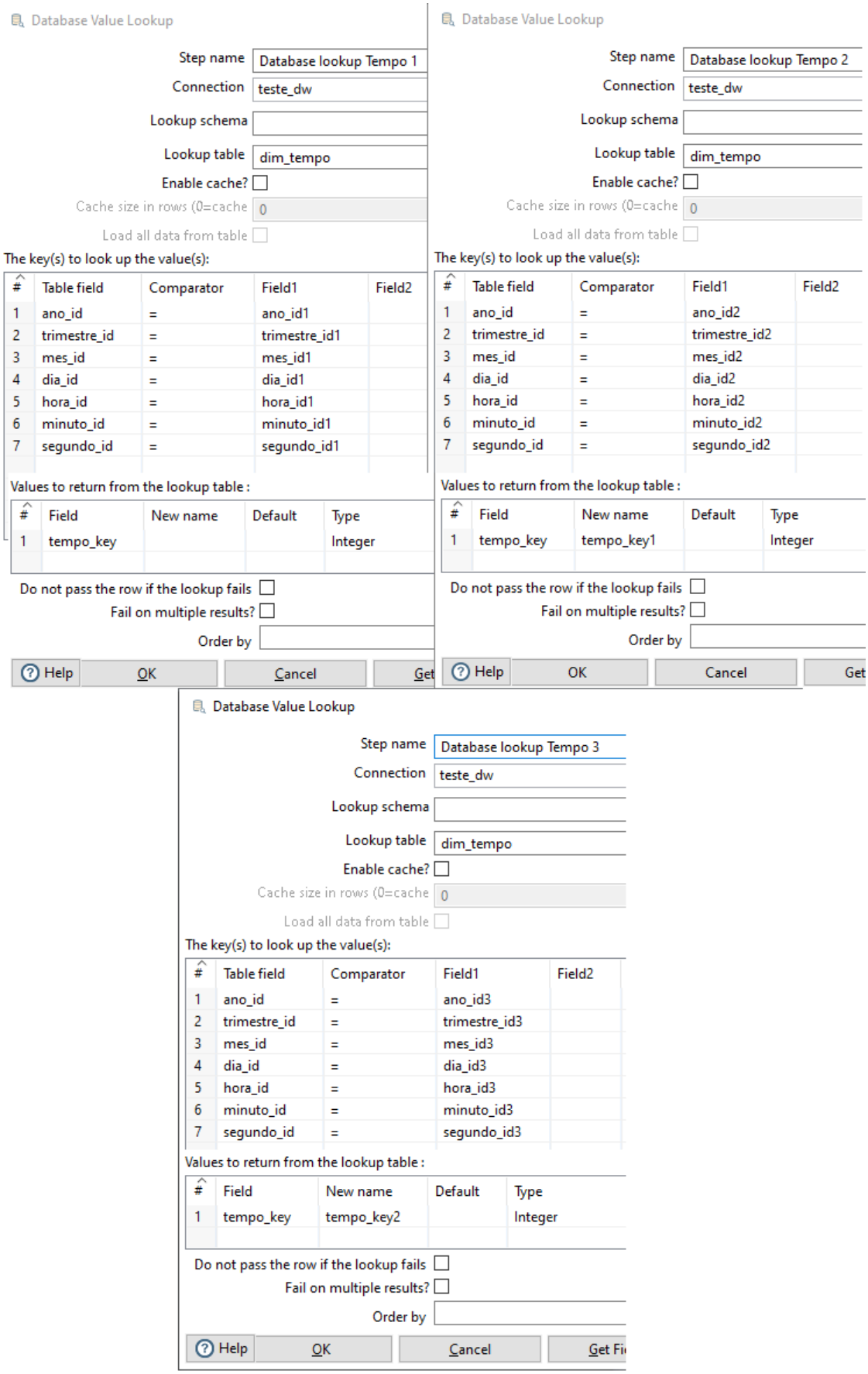


Figura 3.20: Configuração dos 5º, 6º e 7º passos da Transformação da fact_eventos

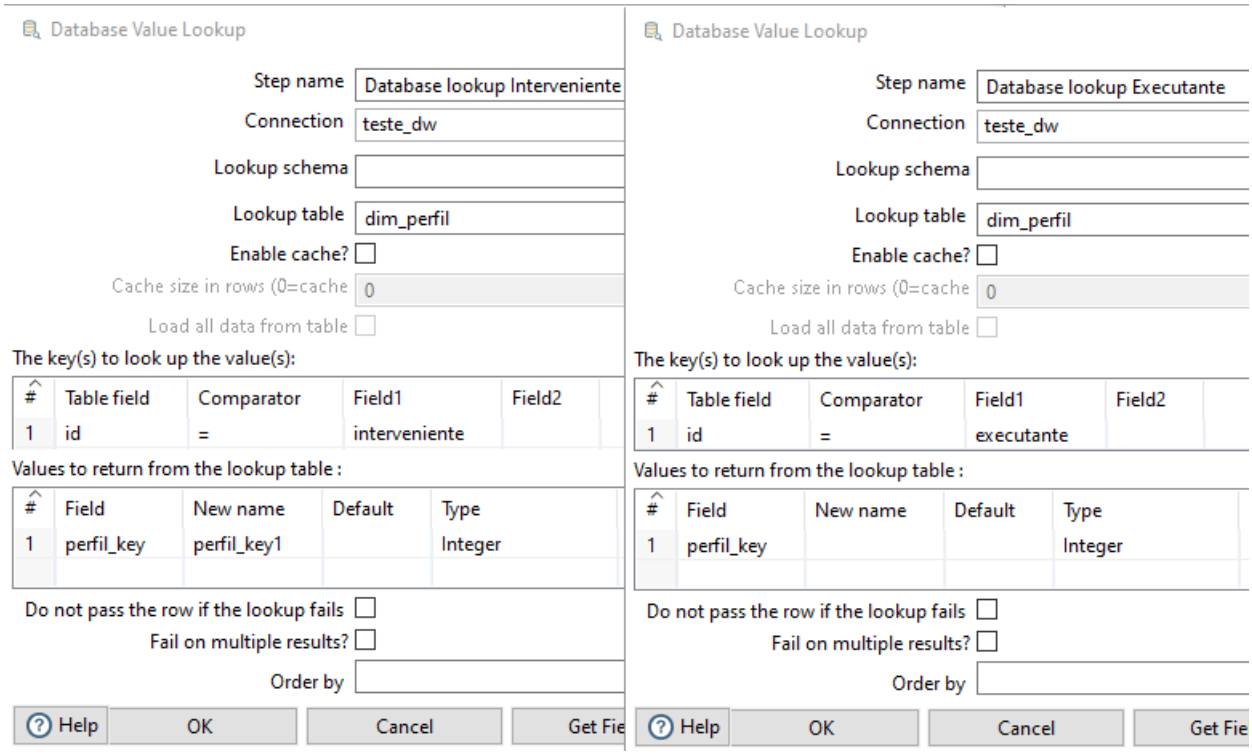


Figura 3.21: Configuração dos 8º e 9º passos da Transformação da fact_eventos

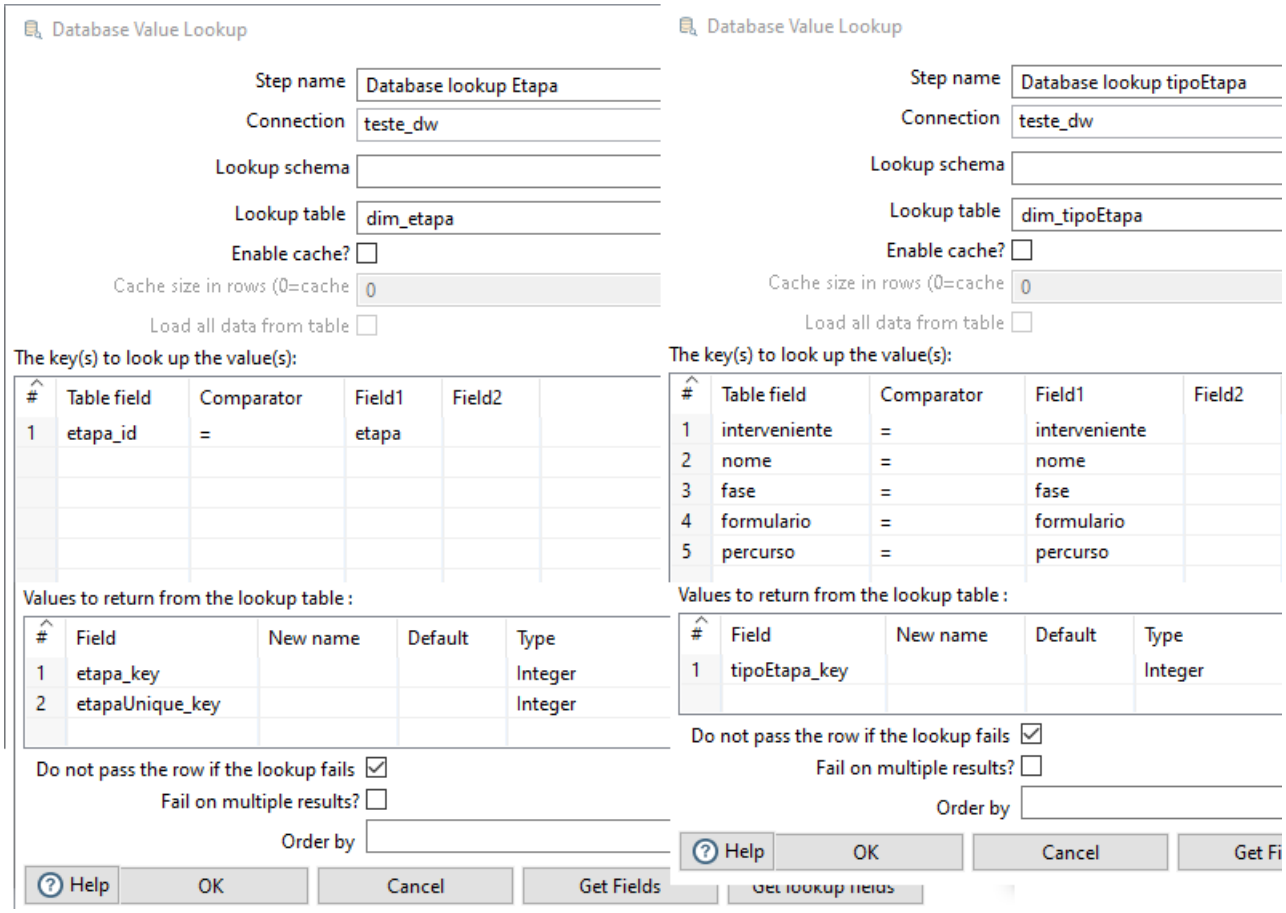


Figura 3.22: Configuração dos 10º e 11º passos da Transformação da fact_eventos

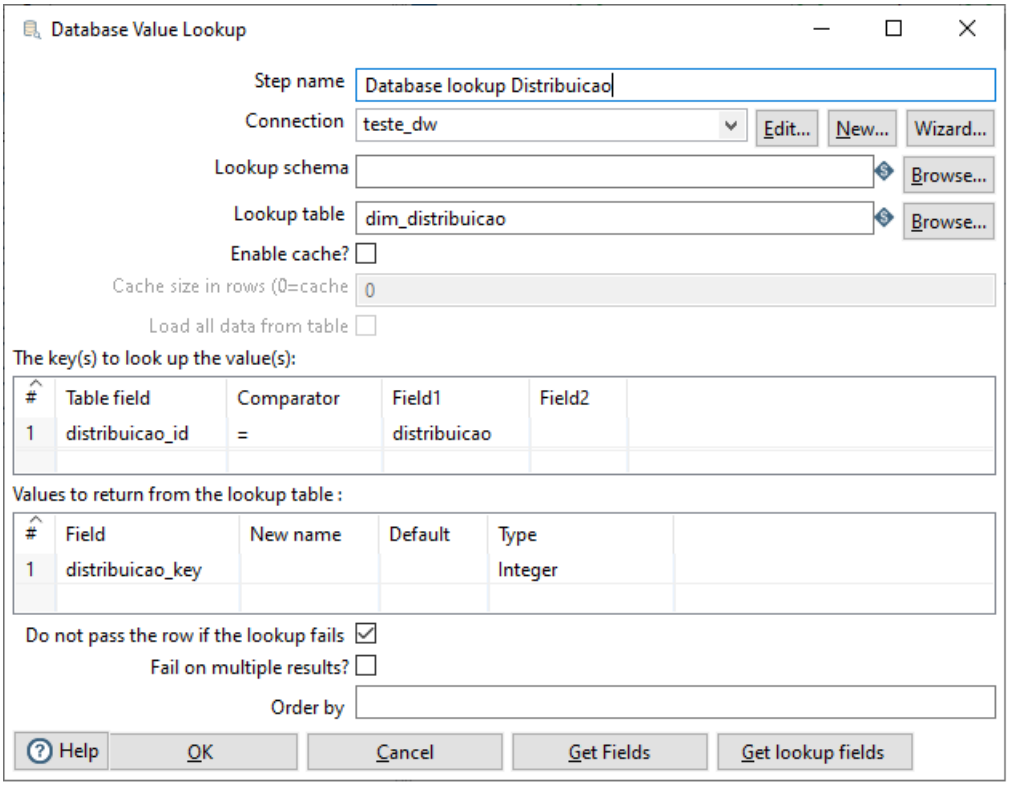


Figura 3.23: Configuração do Database Lookup da dim_distribuicao da Transformação da fact_eventos

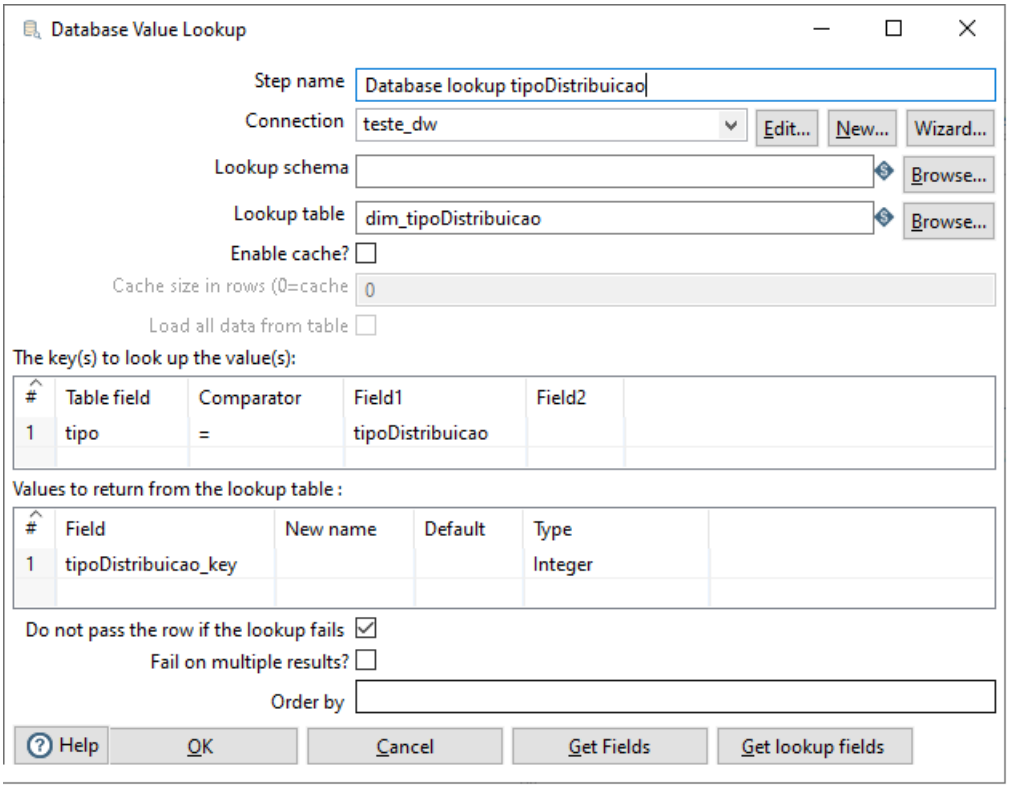


Figura 3.24: Configuração do Database Lookup da dim_tipoDistribuicao da Transformação da fact_eventos

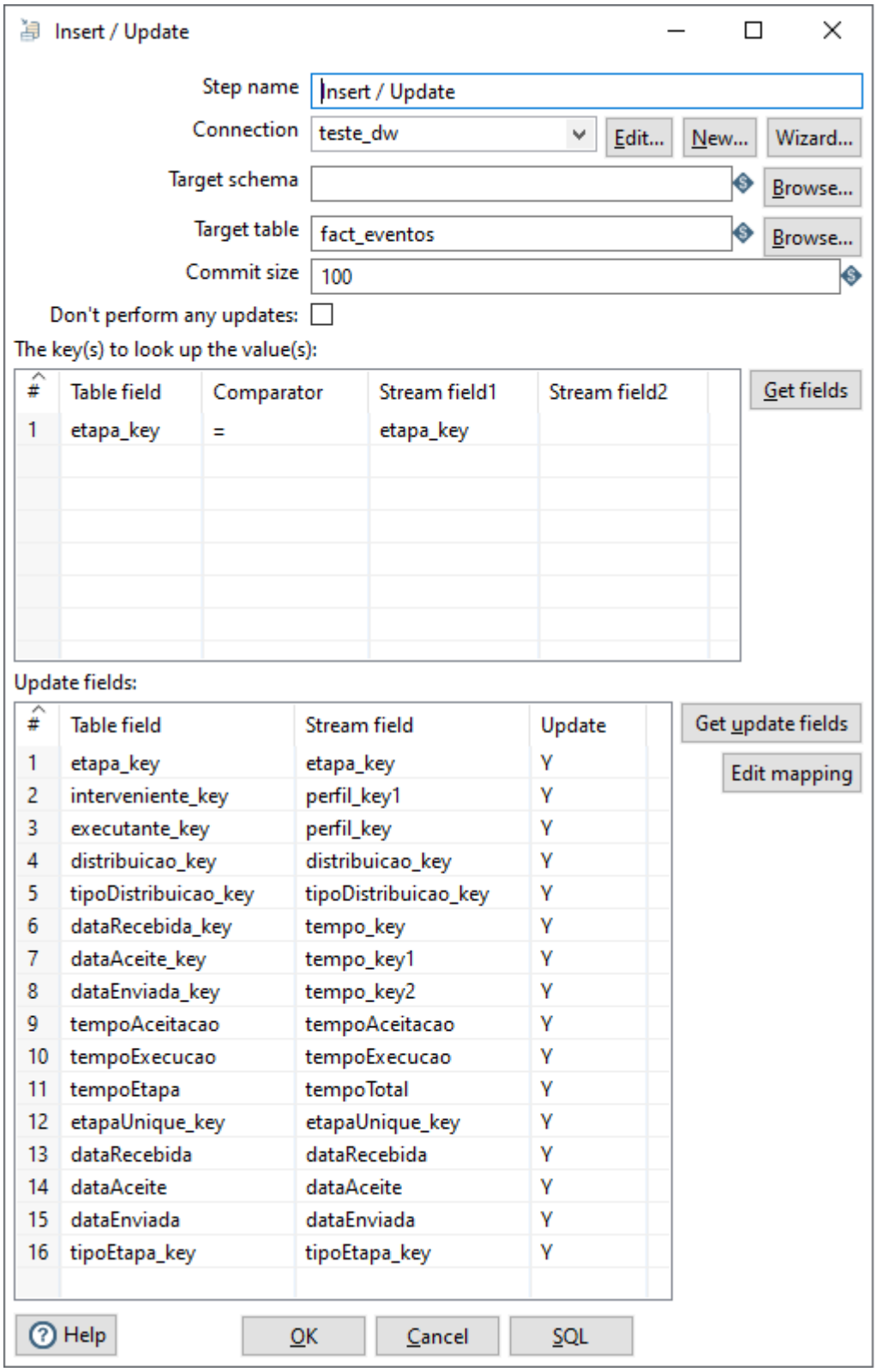


Figura 3.25: Configuração do Insert/Update da Transformação da fact_eventos

Job

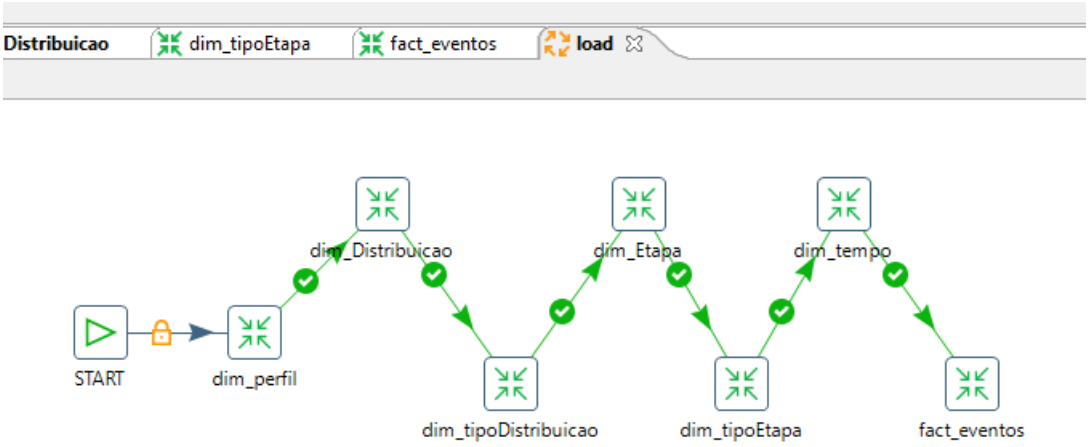


Figura 3.26: Job responsável por executar todas as transformações sequencialmente

Para finalizar a componente do PDI, foi criado um job que executa todas as transformações criadas em sequência. Caso haja algum erro na execução de qualquer transformação, o job gera um log com o erro e para imediatamente todo o processo.

A configuração do job é simples, necessitando apenas de um passo "START", seguido de um link para todas as transformações a executar. Em termos de ordem, neste caso, a única preocupação a ter é em executar a transformação da tabela de factos por último, uma vez que esta depende das outras.

Quando o job corre até ao final sem qualquer tipo de erros, significa que todos os dados terão sido corretamente extraídos da BD do edoc, transformados no formato necessário e carregados para o Data Warehouse desenvolvido, dando por concluído o processo de ETL. Em baixo, é apresentada a tabela de factos preenchida.

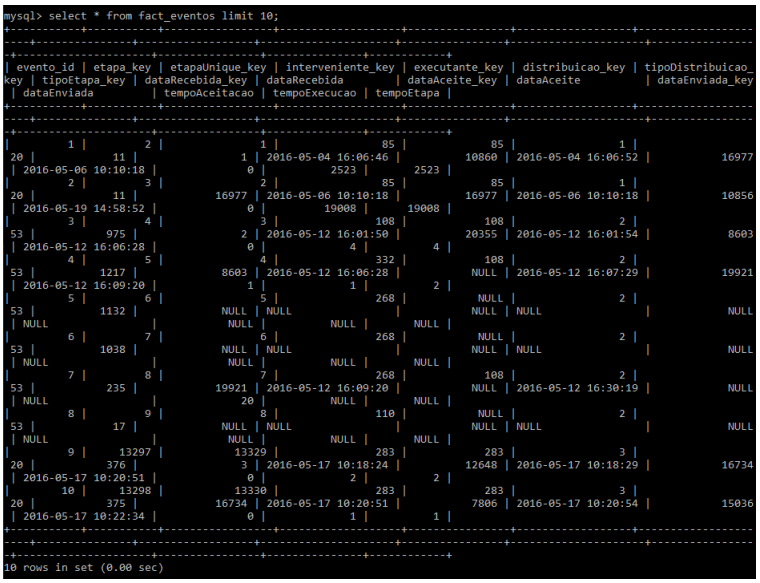


Figura 3.27: Fact_eventos após execução do job

4. Atualização contínua dos dados

A contínua atualização dos dados é muito importante porque é o que nos permite fazer uma análise precisa e ao longo do tempo. Para tal, o Pentaho Data Integration permite que seja definido o scheduling desejado diretamente no job.

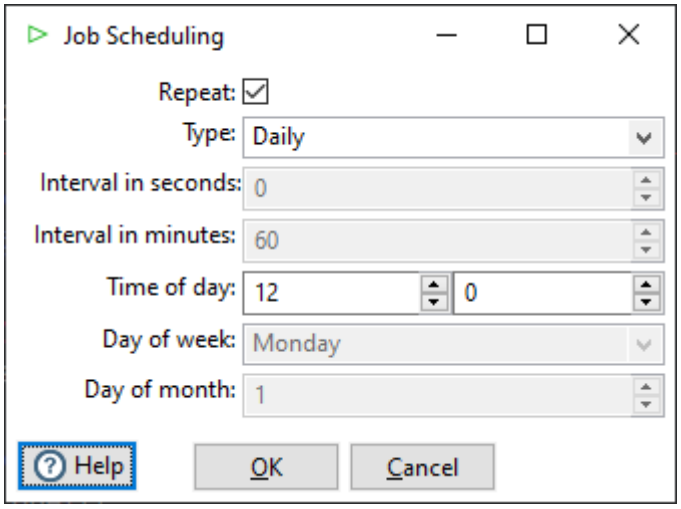


Figura 3.28: Configuração da calendarização do job

O scheduling foi definido para executar o job todos os dias às 12:00. Esta execução é feita localmente, mas poderá também ser definida para executar num servidor (do PDI ou outro, do edoclink, por exemplo). No caso deste projeto, dado que a BD é estática e não é atualizada com dados novos, a execução diária não carregará/atualizará nova informação para o DW. No entanto, a implementação foi feita para demonstrar o conceito.

5. Power BI

Tal como o PDI, já havia uma experiência prévia no Pentaho Report Designer. No entanto, decidi usar o Power BI uma vez que em realidades empresariais é mais reconhecido, tem uma interface muito mais user-friendly, apresenta uma enorme variedade de visualização dos dados e uma documentação mais detalhada. É através do Power BI que os indicadores serão desenhados, tendo portanto uma relação direta com os objetivos definidos no início do documento (um ficheiro Power BI por indicador). É importante referir que ao ter uma conexão direta com o DW, os dashboards criados no Power BI terão sempre os dados atualizados. Sempre que um dashboard é aberto, os dados serão baseados na última versão dos dados presentes no DW.

De seguida, serão detalhados os indicadores definidos na secção dos Objetivos. Para cada indicador, serão primeiro apresentados os passos executados para obter os dados necessários e depois o dashboard criado com os gráficos e filtros que dão resposta ao objetivo.

I. Tarefas Pendentes

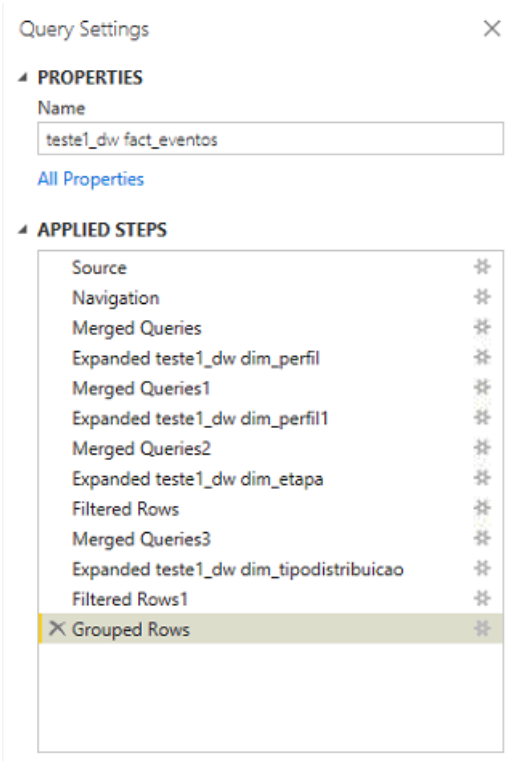


Figura 3.29: Transformação dos dados para o Indicador I

O Indicador I combina as dimensões perfil e etapa com a tabela de factos. Assim, foram precisos 4 merges para juntar os dados das etapas, dos tipos das distribuições, dos intervenientes e executantes. Nos merges da dim_perfil e dim_tipoDistribuicao, expandiram-se os atributos **nome** e **tipo** para se identificarem os intervenientes, executantes e tipos das distribuições. No merge da dim_etapa, expandiu-se o atibuto **estado** para de seguida se filtrarem apenas as etapas com o estado "P"(pendente). Como a BD original é uma BD de testes, foram filtradas também algumas etapas que não faziam sentido para tentar ter um resultado o mais parecido com a realidade possível. Por fim, fez-se um GROUP BY por COUNT ROWS para se obter o número de etapas por tipo, por interveniente e executante.

O Dashboard criado apresenta 3 filtros - por Tipo, por Interveniente e por Executante. Os gráficos e contador são atualizados automaticamente quando é feito algum tipo de filtro. O gráfico da esquerda mostra uma visão mais geral (etapas pendentes por Tipo de Distribuição), enquanto que o da direita é mais específico (etapas pendentes por interveniente e executante).

Este Dashboard permite então responder a perguntas do tipo:

- Quem tem mais etapas pendentes?
- Qual o tipo de Distribuição com mais tarefas por fazer?
- No interveniente X, quem tem menos tarefas pendentes?

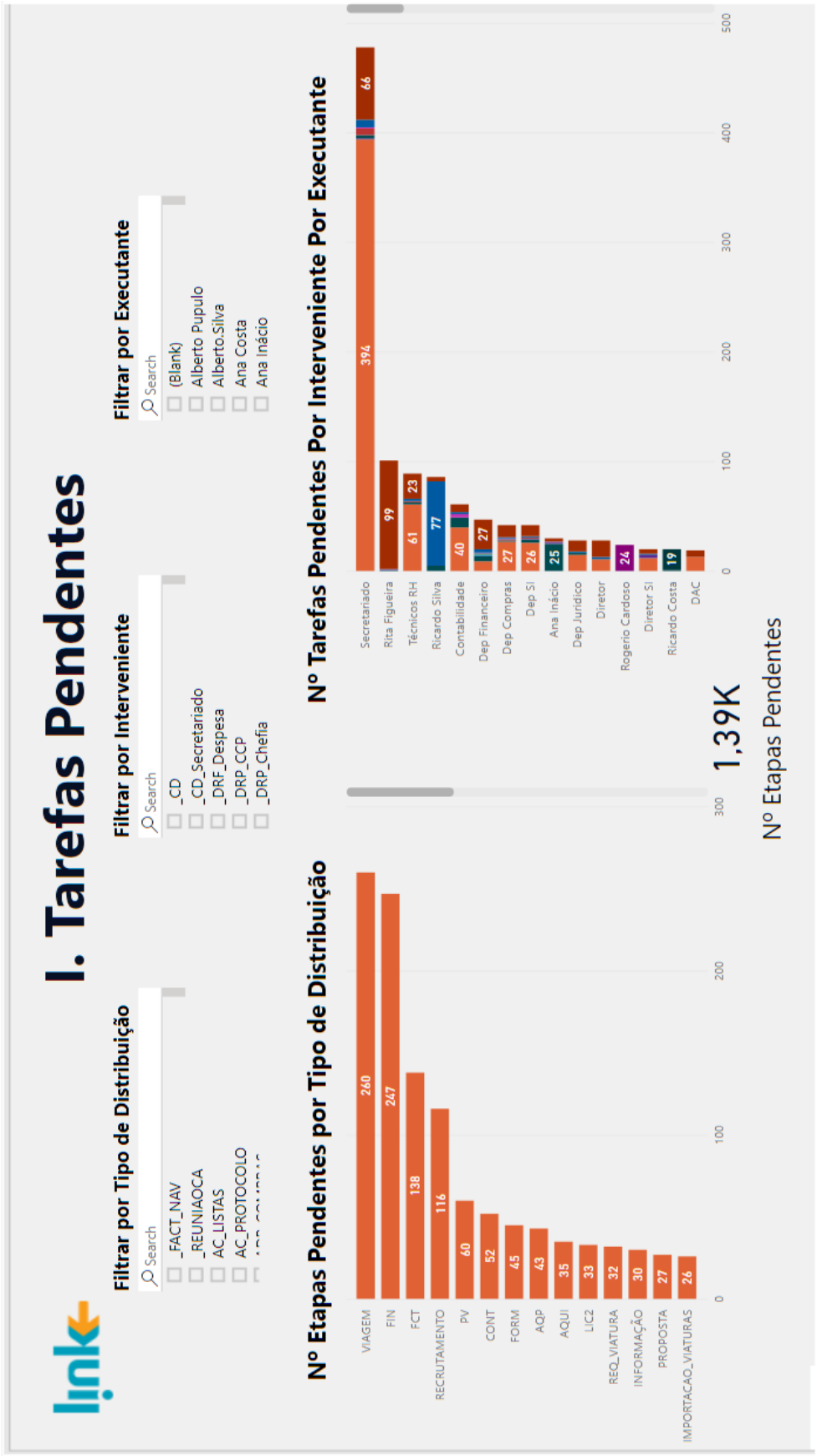


Figura 3.30: Dashboard para o Indicador I

II. Tarefas Entregues

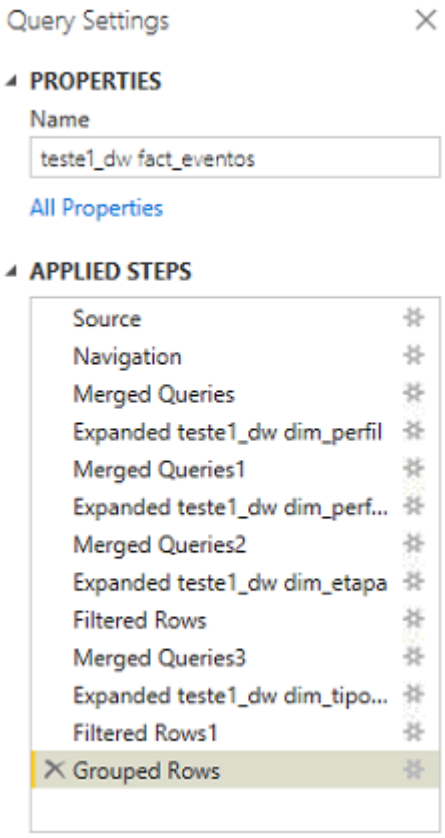


Figura 3.31: Transformação dos dados para o Indicador II

O indicador II é muito semelhante ao Indicador I, sendo apenas diferente o filtro que se faz nas etapas. Em vez de se filtrarem as etapas com **estado** "P", filtram-se as etapas com **estado** "D"(delivered).

O Dashboard criado apresenta novamente 3 filtros - por Tipo, por Interveniente e por Executante.

Este Dashboard permite responder a perguntas do tipo:

- Quem entregou mais tarefas?
- Qual o tipo de Distribuição com mais tarefas entregues?
- No interveniente X, quem tem menos tarefas entregues?

Figura 3.32: Dashboard para o Indicador II

III. Tempo Médio de Aceitação

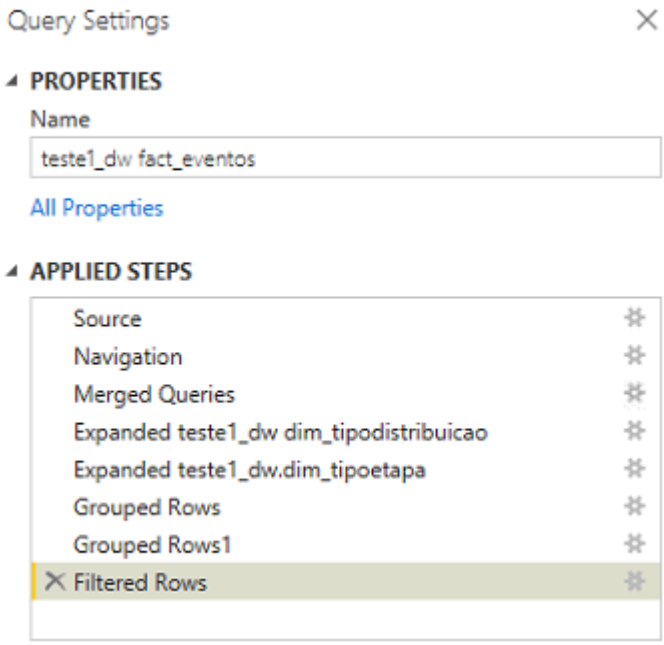


Figura 3.33: Transformação dos dados para o Indicador III

Este indicador combina as dimensões dos tipos da distribuições e etapas com a tabela de factos. Primeiro juntaram-se as tabelas correspondentes a essas dimensões e expandiram-se os atributos **tipo** e **nome** das dim_tipoDistribuicao e dim_tipoEtapa (respetivamente) para melhor se identificarem. Depois, fizeram-se dois GROUP BY - um que fez a soma do tempo total de todas as etapas com o mesmo **etapaUnique_key** (para o caso de as etapas terem sido executadas mais do que uma vez) e outro que fez a média do tempo por tipo de distribuição e por tipo de etapa.

O Dashboard criado apresenta 2 filtros - por Tipo de Distribuição e por Tipo de Etapa. Os gráficos e contador são atualizados automaticamente quando é feito algum tipo de filtro. O gráfico da esquerda mostra uma visão mais geral (tempo de Aceitação por Tipo de Distribuição), enquanto que o da direita é mais específico (tempo de Aceitação por tipo de Etapa).

Este Dashboard permite então responder a perguntas do tipo:

- Qual o tipo de Distribuição que demora mais tempo até ser aceite?

- Quais são as etapas/distribuições em que o executante é determinado mais rapidamente?

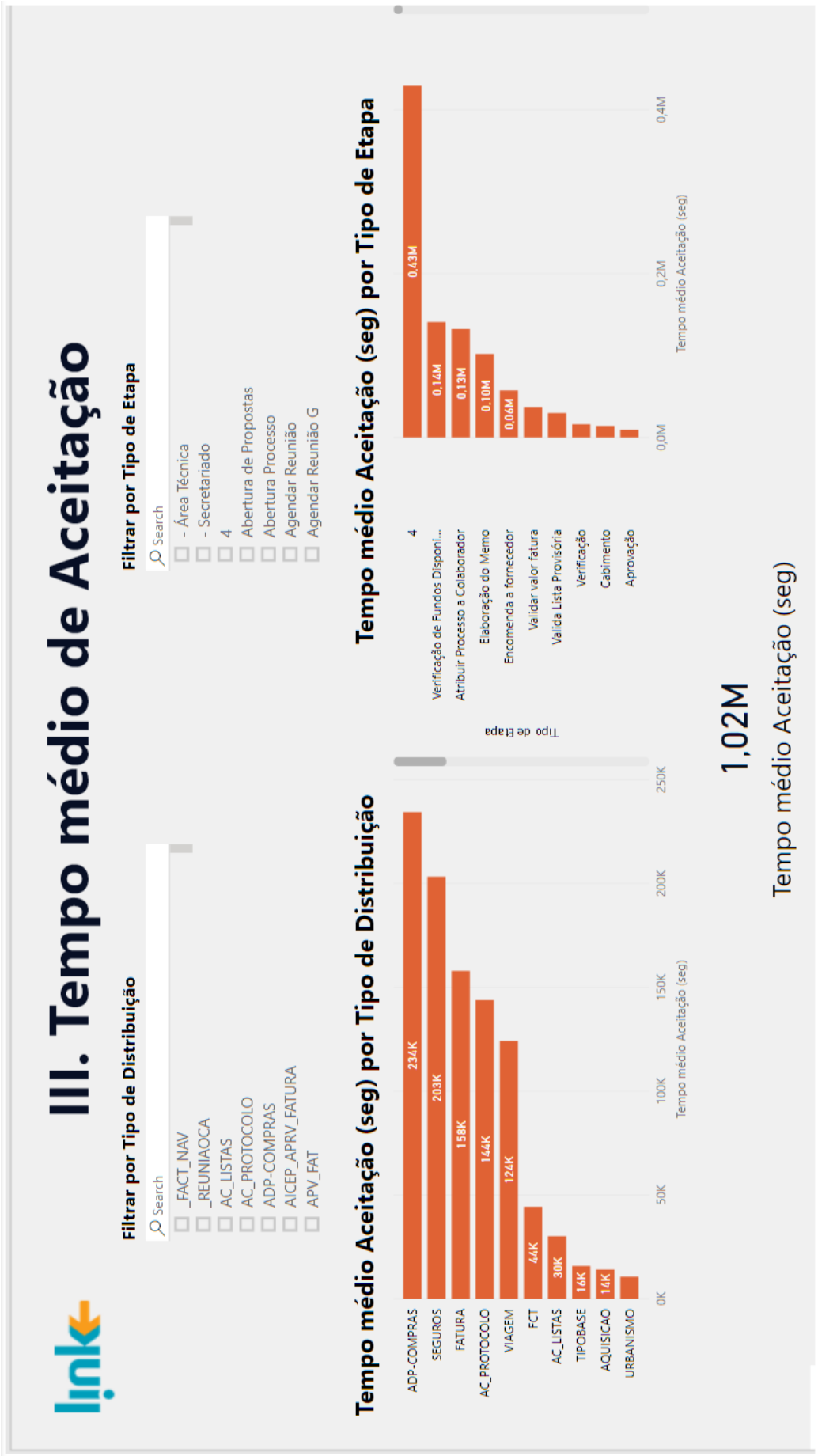


Figura 3.34: Dashboard para o Indicador III

IV. Tempo Médio Total por Tipo de Distribuição

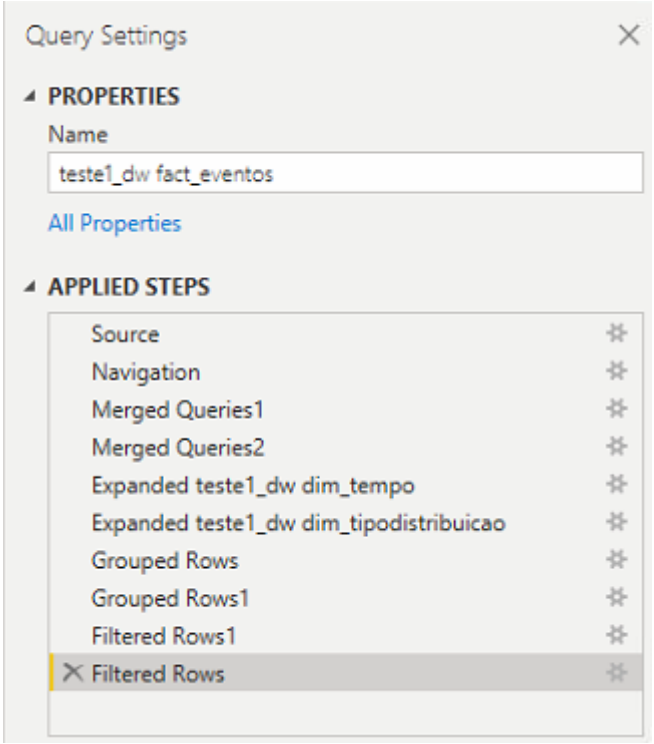


Figura 3.35: Transformação dos dados para o Indicador IV

Neste indicador, foi primeiro necessário fazer merge entre as dimensões do tempo e do tipo das distribuições e a tabela de factos. De seguida, fizeram-se 2 GROUP BY - o primeiro em que fez a soma de todas as etapas por cada distribuição e o segundo que fez a média de todas as distribuições por tipo de Distribuição e por mês. Por fim, removeram-se etapas com tipos de distribuição específicos de um ambiente de testes para ter resultados o mais semelhantes com a realidade.

O Dashboard criado apresenta 2 filtros - por Tipo de Distribuição e por mês. O gráfico permite ter uma visualização do tempo médio total das distribuições por tipo ao longo dos meses. Também foi adicionado um contador que indica o número de etapas para perceber se existe alguma relação entre o tempo total e o número de etapas.

Este Dashboard permite então responder a perguntas do tipo:

- Qual o tipo de Distribuição que é executado em menor tempo ?

- Qual foi a evolução do tipo X ao longo dos meses ?

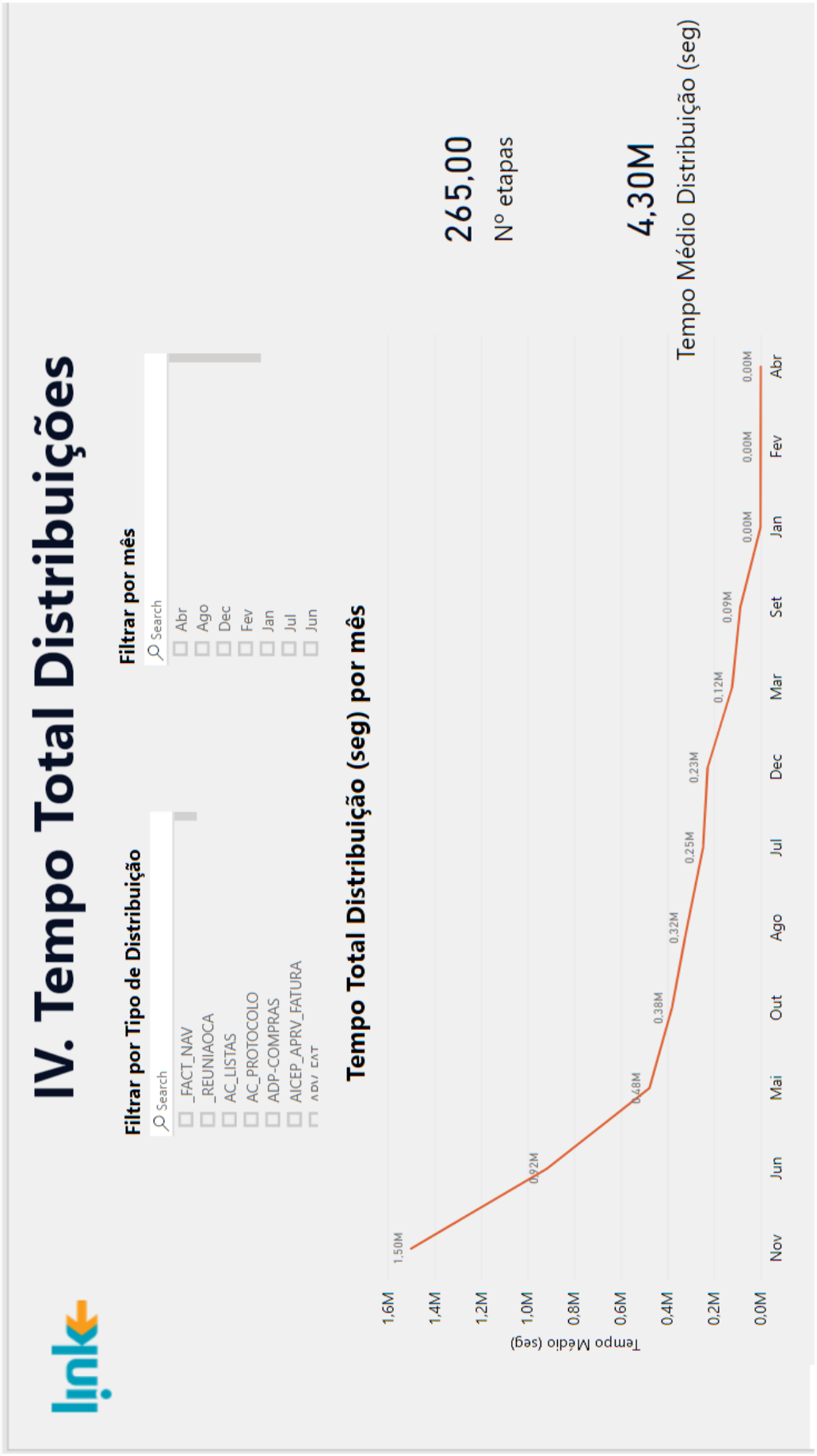


Figura 3.36: Dashboard para o Indicador IV

V. Tempo Médio Total por Etapa

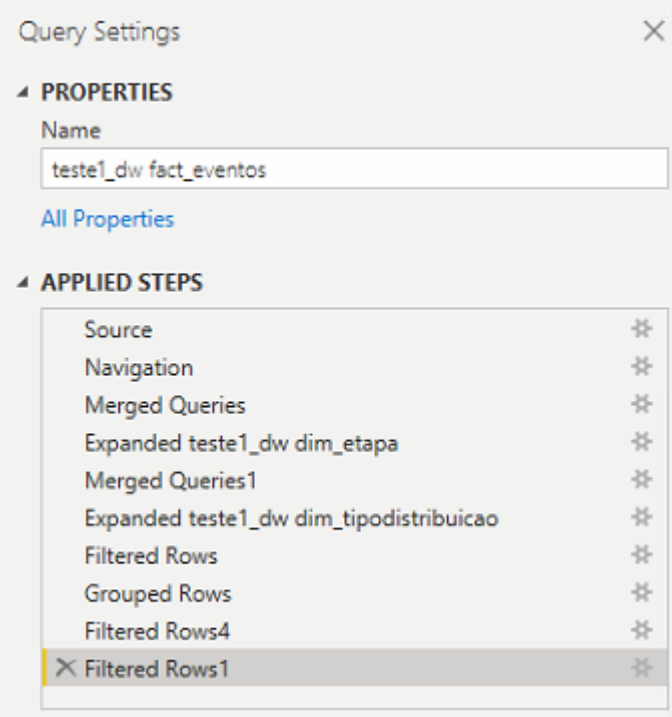


Figura 3.37: Transformação dos dados para o Indicador V

O indicador V junta a tabela de factos com as dimensões das etapas e dos tipos das distribuições. Logo, inicialmente foram precisos fazer 2 merges. Depois, fez-se um GROUP BY por **etapaUnique_key** que fez a soma do tempo total de cada etapa. Por último, filtraram-se as etapas com tempos totais que fizessem sentido (entre 120 a 30000 segundos) e removeram-se os tipos de etapas específicos de um ambiente de testes.

O Dashboard criado apresenta 2 filtros - por Tipo de Distribuição e por nome da Etapa. O gráfico da esquerda mostra uma visão mais geral (tempo médio por Tipo de Distribuição), enquanto que o da direita é mais específico (tempo médio por etapa).

Este Dashboard permite então responder a perguntas do tipo:

- Quais são as etapas que geralmente demoram mais a executar?

- Que tipo de Distribuição tem etapas que se executam mais rapidamente ?

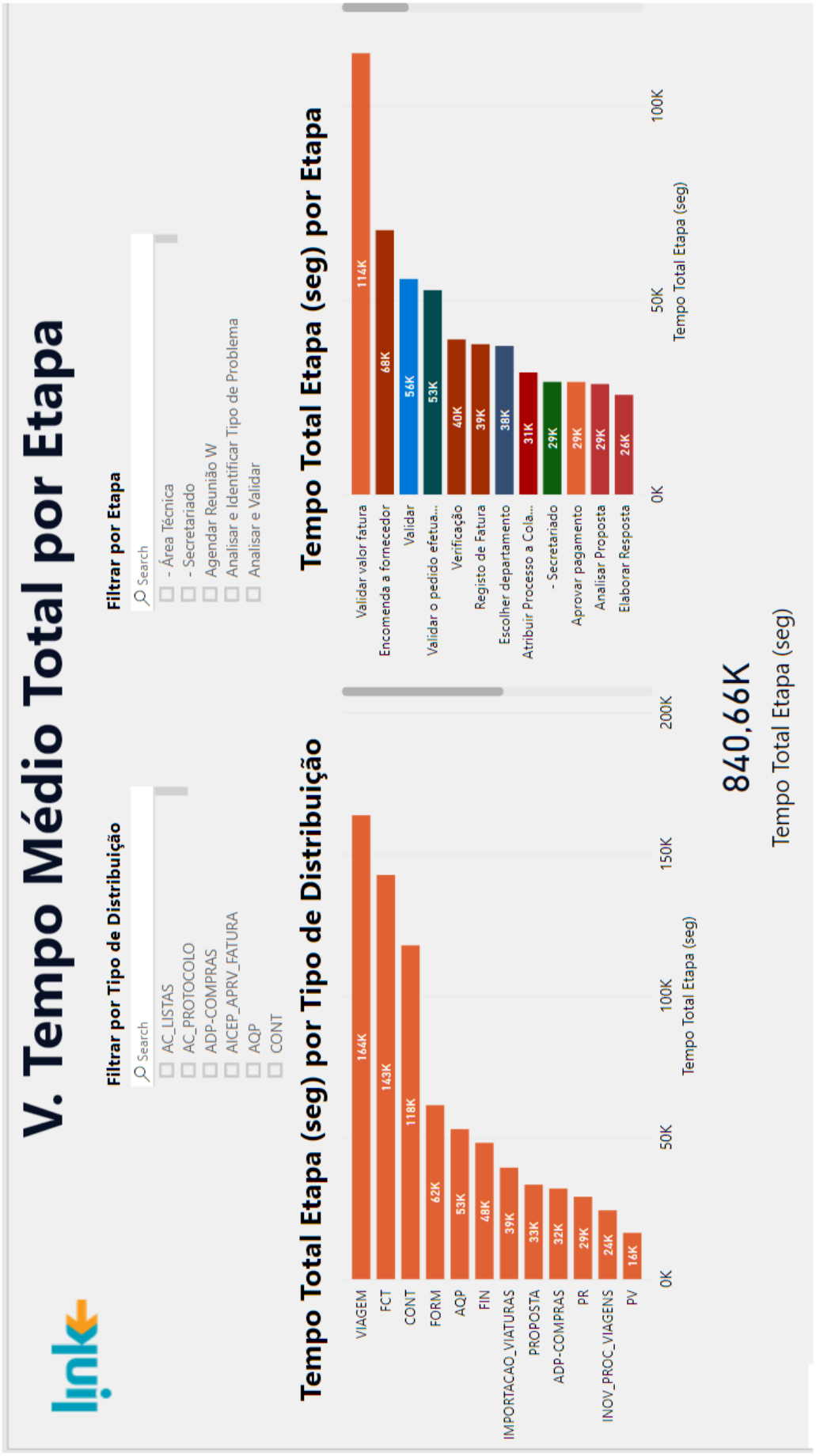


Figura 3.38: Dashboard para o Indicador V

VI. Tempo Médio Total por Fase

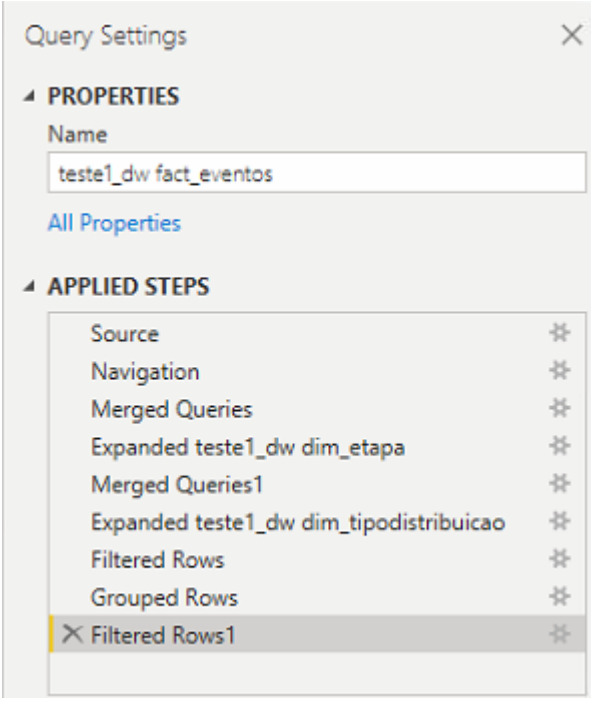


Figura 3.39: Transformação dos dados para o Indicador VI

O indicador VI é muito semelhante ao indicador V, sendo a diferença o facto de analisar por Fase e não por Etapa.

O Dashboard criado apresenta 2 filtros - por Tipo de Distribuição e por fase. O gráfico da esquerda mostra uma visão mais geral (tempo médio por Tipo de Distribuição), enquanto que o da direita é mais específico (tempo médio por fase).

Este Dashboard permite então responder a perguntas do tipo:

- Quais são as fases que geralmente demoram mais a executar?

- Que tipo de Distribuição tem fases que se executam mais rapidamente ?

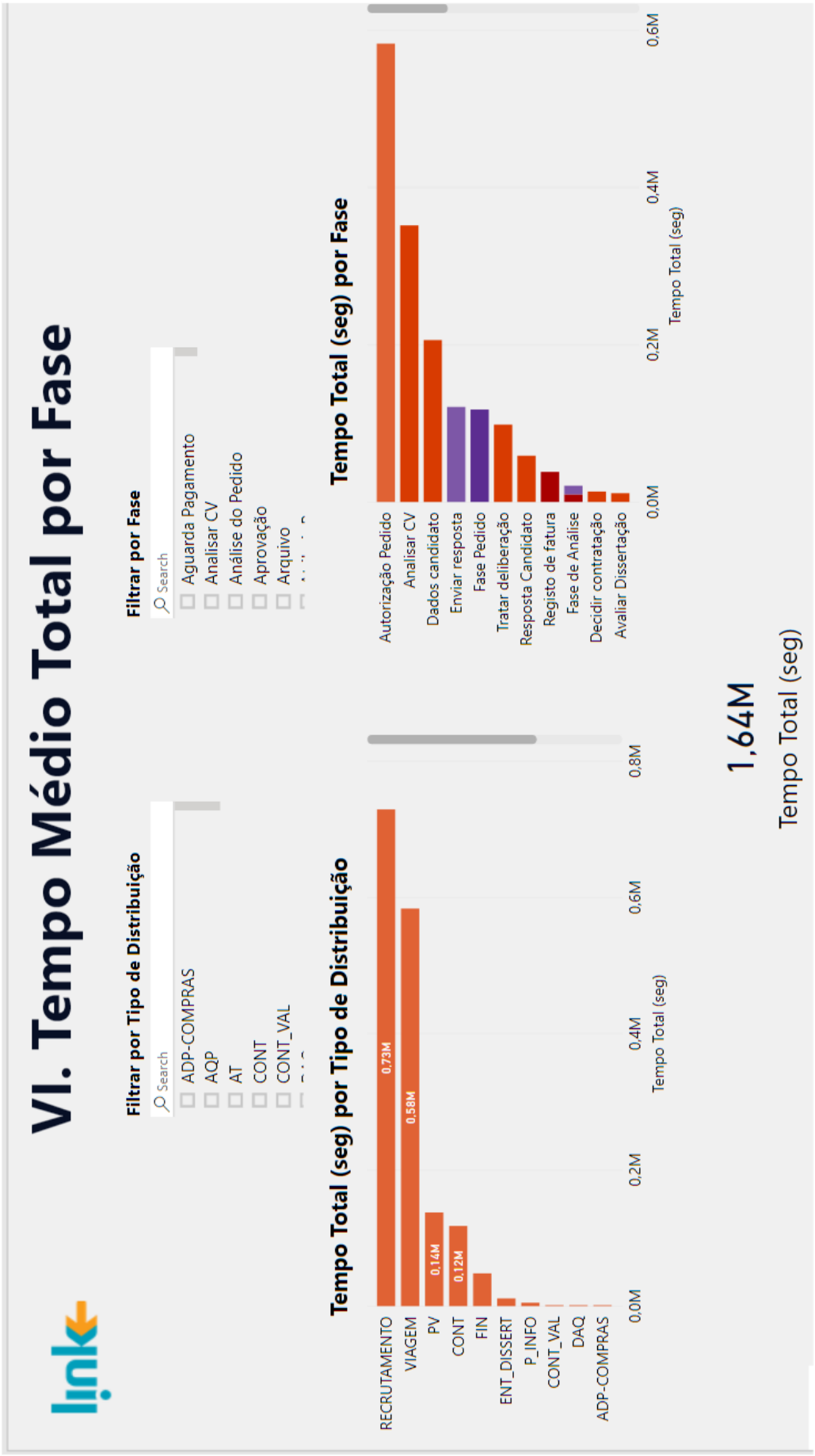


Figura 3.40: Dashboard para o Indicador VI

VII. Volume de Distribuições

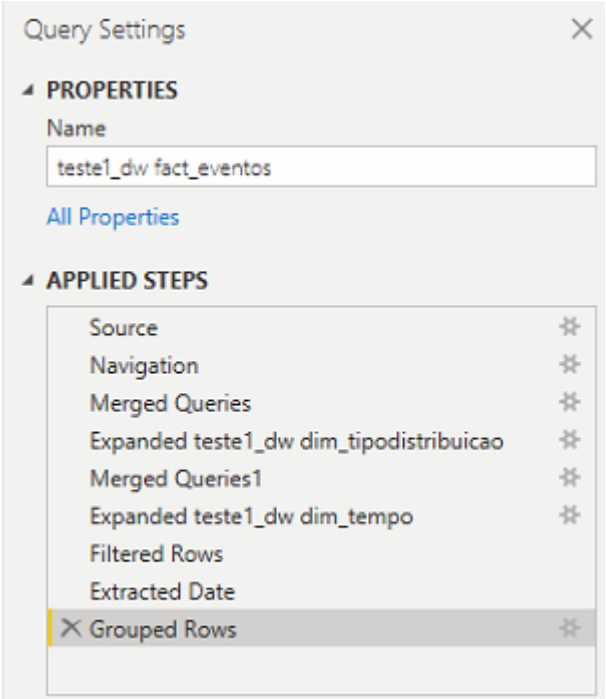


Figura 3.41: Transformação dos dados para o Indicador VII

O indicador VII mostra a evolução temporal do volume das distribuições. Junta as dimensões do tempo e do tipo das Distribuições com a tabela de factos através de 2 merges. Por fim, fez-se um GROUP BY que fez o count das distribuições por tipo, por mês e por ano.

O Dashboard criado apresenta 3 filtros - por Tipo de Distribuição, por Ano e por Mês. Os filtros temporais permitem que se faça uma análise para um ano/mês específico ou para um intervalo de tempo.

Este Dashboard permite então responder a perguntas do tipo:

- Em que intervalo de tempo houve um maior volume de distribuições?

- O volume das distribuições mantém-se ao longo dos anos?

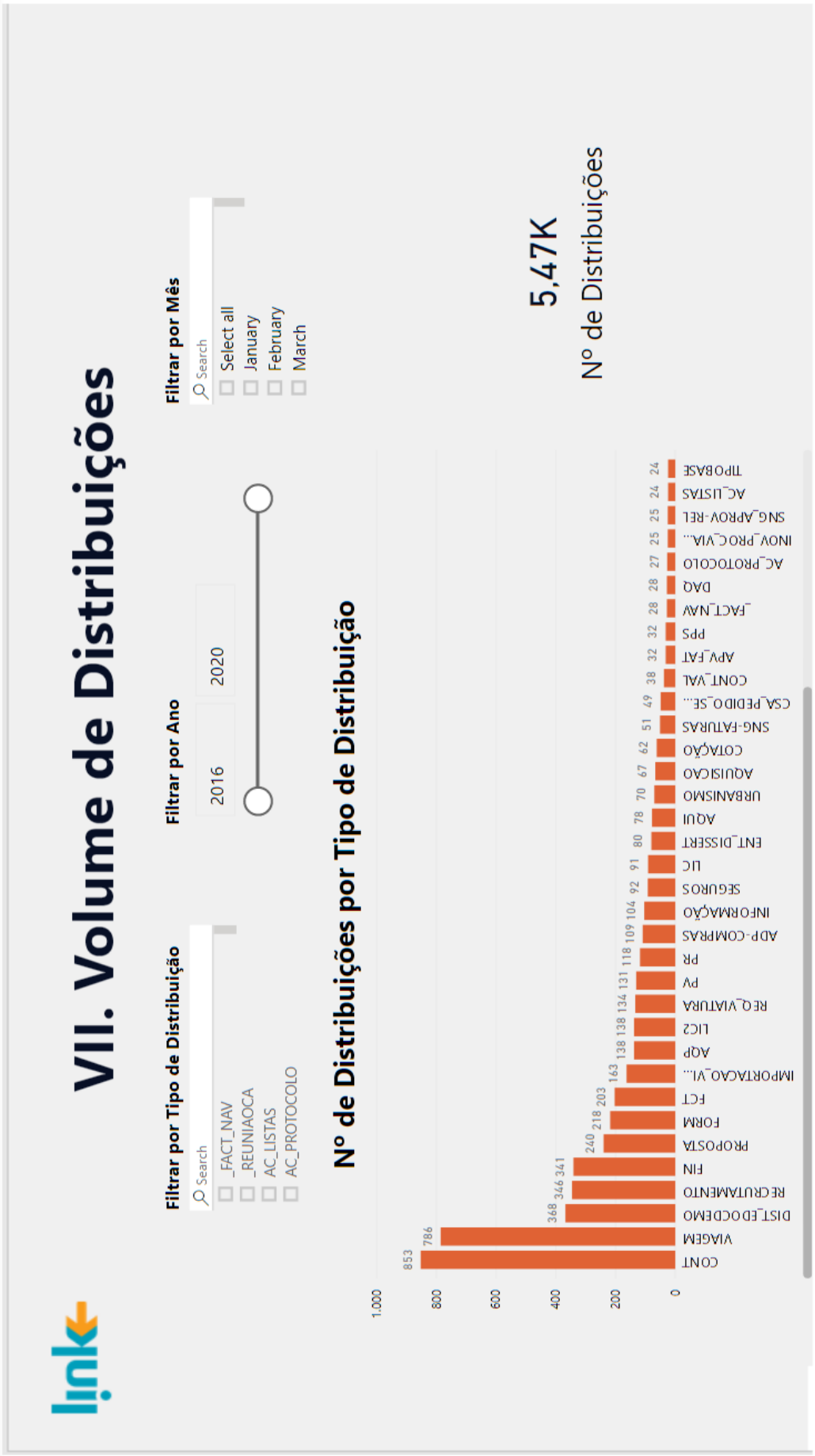


Figura 3.42: Dashboard para o Indicador VII

3.1.2 Dificuldades

Ao longo do documento, foram mencionadas algumas decisões que foram tomadas de modo a ultrapassar vários obstáculos que foram surgindo ao longo do projeto. Nesta secção será feita uma exposição mais detalhada das principais dificuldades e explicado como foram superadas.

Indicadores

O maior desafio terá sido a definir concretamente os indicadores. Apesar de sempre se ter tido uma ideia do tipo de perguntas às quais a solução teria que dar resposta, foi só mais no final que se chegou à versão final dos indicadores apresentada no documento. Esta decisão foi tomada em conjunto com colaboradores do edoclink, que deram a sua opinião sobre o que seria importante analisar. Antes da versão final, foram existindo outras versões tais como as apresentadas em baixo. Como se pode observar, as versões iniciais eram muito mais generalizadas o que dificultava o desenvolvimento da solução (especialmente dos gráficos no Power BI). As measures da tabela de factos também só puderam ser pensadas após fechar a definição dos indicadores - o facto de termos indicadores sobre o tempo de aceitação e tempo total de uma etapa/distribuição clarificou que measures tinham de ser adicionadas à tabela.

	Indicadores
Versão 1	- Quantos processos têm neste momento? - Quantos processos alguém efetuou o mês passado? - Tempo médio por processo/etapa? - Número de processos por área? - Quantas tarefas tem a pessoa X por fazer?
Versão 2	Conjugar cada uma das seguintes métricas - Número de tarefas - Tempo médio de tratamento - Tempo médio até aceitação Com - Por pessoa - Por tempo (dia, semana, mês, ano) - Por tipo de distribuição - Por tipo de etapa
Versão 3	- Etapas pendentes por pessoa - Etapas pendentes por pessoa por grupo - Tempo médio por etapa por pessoa por grupo - Tempo médio por tipo de etapa - Número etapas num período de tempo

Tabela 3.1: Versões dos Indicadores ao longo do desenvolvimento do projeto

Ambiente edoc

Outra dificuldade que surgiu logo no início do projeto terá sido em compreender o ambiente do edoc. Antes de qualquer desenvolvimento em termos de solução, foi primeiro preciso aprender todos os conceitos, funcionamento e organização da empresa/ferramenta. O edoclink disponibilizou acesso ao edoc, junto de uma formação que permitia interagir e visualizar grande parte dos conceitos, facilitando a experiência inicial. No entanto, foi a contínua interação com colaboradores da empresa e com a BD fornecida que permitiu a compreensão de todo o sistema.

Etapas refeitas

No decorrer do projeto, levantou-se a questão de como seriam tratadas etapas que estariam a ser executadas mais do que uma vez, já que na BD têm IDs diferentes e não há nenhum atributo que identifique duas etapas como iguais. Isto é importante uma vez que uma etapa que seja executada várias vezes deverá ter como tempo de execução o tempo total de todas as execuções e não apenas do primeiro, por exemplo.

Foi então estudado como poderíamos implementar esta identificação no projeto e a solução foi, como referido anteriormente na descrição da solução, mudar a dimensão das etapas para uma Slowly Changing Dimension, onde seriam guardas as diferentes execuções (através de versões) associadas ao mesmo **etapaUnique_key**, quando os critérios definidos para se identificar se duas tarefas são a mesma se verificassem. Estas versões também ficariam associadas a diferentes datas. Esta solução permitiu atingir o pretendido, uma vez que ao agrupar por **etapaUnique_key** se obtia o tempo total de todas as execuções da mesma etapa.

Power BI

O Power BI foi outro ponto que inicialmente gerou alguma dificuldade. Não havendo experiência prévia com o programa, nem tendo exemplos práticos de gráficos que podiam ser feitos, os gráficos criados numa primeira fase estavam num nível muito fraco em comparação com os apresentados na secção anterior. O utilizador não conseguia ter uma compreensão instantânea do que o gráfico representava, os filtros não eram user-friendly e os dados específicos de um ambiente de testes não eram filtrados. Foi com a ajuda direta do edoclink que se passou dos gráficos iniciais para os dashboards apresentados. Desde exemplos feitos para clientes a opiniões sobre o que se deveria adicionar/remover, a empresa foi crucial para chegar às versões finais. Em baixo, é demonstrada uma das versões iniciais do indicador I.

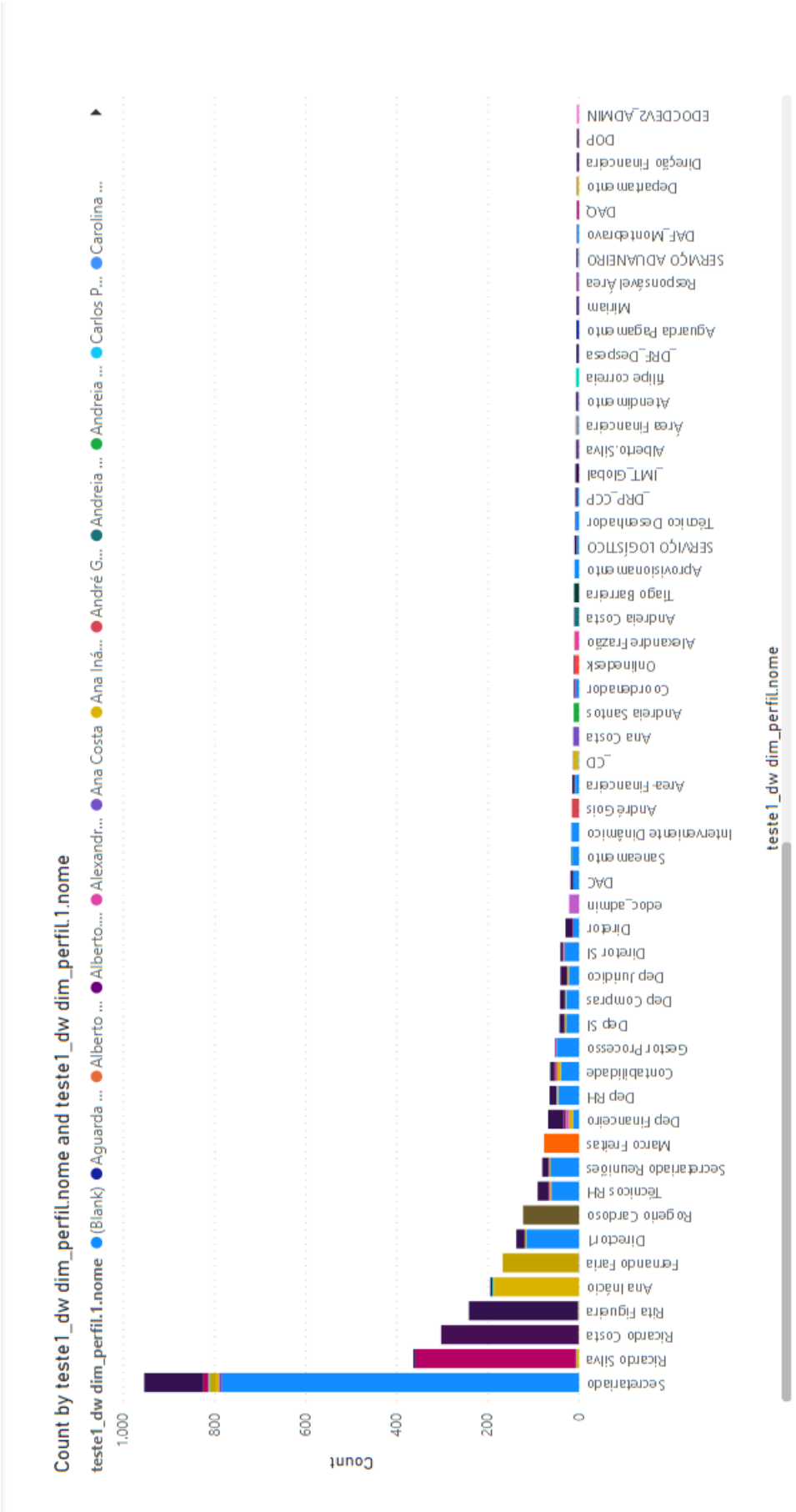


Figura 3.43: Versão Inicial do indicador I

Capítulo 4

Avaliação

4.1 Extensibilidade

A solução implementada foi desenvolvida a pensar nos indicadores definidos anteriormente. No entanto, quão fácil seria adicionar outra dimensão caso se quisesse desenvolver outro indicador? Por exemplo, caso se quisesse fazer uma análise por processos, quais seriam as alterações necessárias?

1º - Relação entre o modelo corrente e a nova dimensão

A primeira análise seria à relação entre as dimensões e tabela de factos existentes e a nova dimensão (dim_processos). Ao inspecionar a BD do edoc, existe uma tabela chamada PROCESSES que contém informação sobre os vários processos. Daí escolhe-se os atributos que se consideram importantes a ter no modelo - por exemplo, assunto, código do processo e distribuição associada.

2º - Criar dim_processos

De seguida, é preciso criar a tabela da nova dimensão e adicionar ao código MySQL já desenvolvido. Tal como as outras, esta tabela teria também um contador automático que seria a chave primária, seguida dos atributos definidos como importantes. Por fim, adicionaria-se à tabela de factos, a chave primário da dim_processos como chave estrangeira.

3º - Criar transformação no PDI e adicionar ao job

Neste momento, seria preciso criar uma nova transformação no Pentaho que iria extrair os dados necessários da tabela PROCESSES da BD do edoc, transformar e colocar na tabela criada no passo anterior. Para terminar este passo, seria preciso adicionar o passo de Database Lookup na tabela de factos e adicionar a transformação ao job.

4º - Analisar no PowerBI

Tendo o modelo já definido corretamente e os dados necessários carregados, seria apenas preciso criar um novo dashboard no Power BI que respondesse ao indicador desejado. Por exemplo, caso o indicador fosse "Volume de distribuições por código

de processo"seria preciso fazer um merge com a dim_processos, filtrar os dados específicos de um ambiente de testes, fazer um GROUP BY por código de processos que fizesse o COUNT de entradas e depois mostrar o gráfico.

Pode-se concluir que a solução desenvolvida permite a adição de novos indicadores. Desde que os passos demonstrados acima sejam cumpridos, é de facto relativamente fácil implementar novos indicadores. Apenas é preciso uma reflexão clara sobre o que é pretendido e perceber que tipo de relação apresenta com a tabela de factos.

4.2 Escalabilidade/Desempenho

Relativamente ao desempenho, foi avaliado o tempo que o processo de ETL demora, ou seja, o tempo que os dados demoram a serem extraídos da BD do edoc, transformados pelas regras definidas no PDI e carregadas para o DW. Em baixo, pode-se ver o log de uma execução do job.

Execution Results

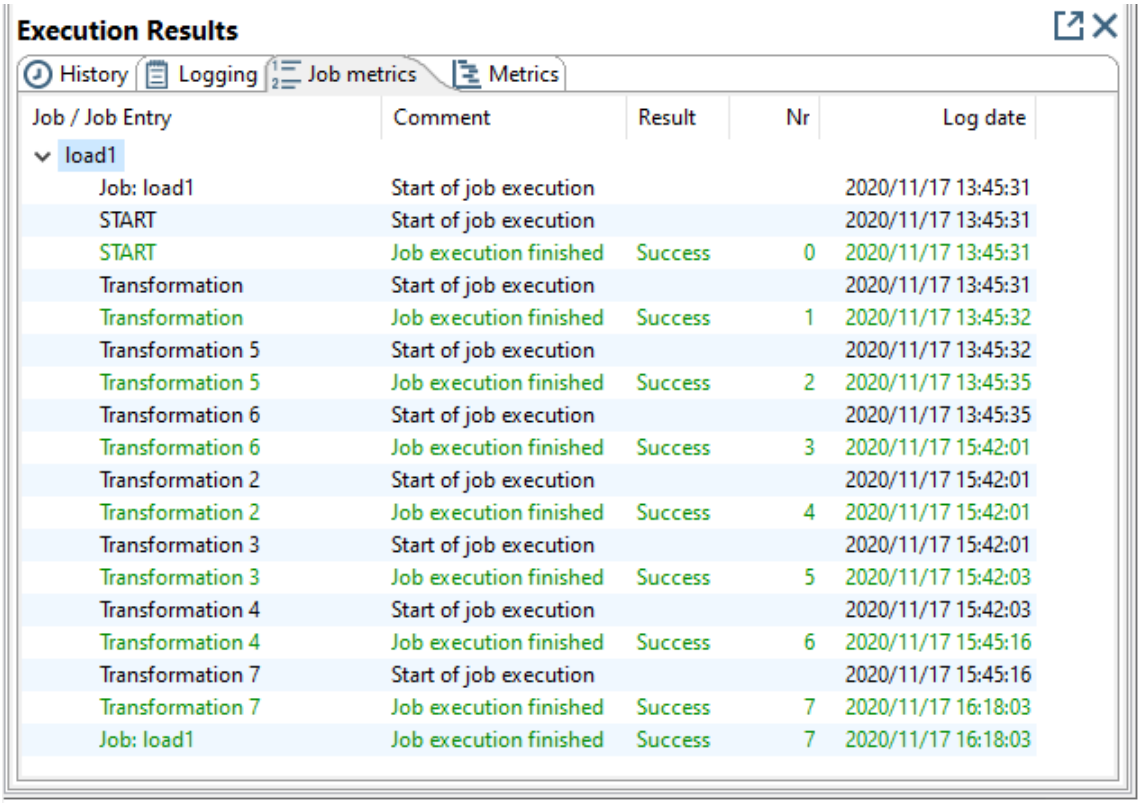
History | Logging | Job metrics | Metrics

Job / Job Entry	Comment	Result	Nr	Log date
load				
Job: load	Start of job execution			2020/11/13 17:06:09
START	Start of job execution			2020/11/13 17:06:09
START	Job execution finished	Success	0	2020/11/13 17:06:09
dim_perfil	Start of job execution			2020/11/13 17:06:09
dim_perfil	Job execution finished	Success	1	2020/11/13 17:06:10
dim_Distribuicao	Start of job execution			2020/11/13 17:06:10
dim_Distribuicao	Job execution finished	Success	2	2020/11/13 17:06:15
dim_tipoDistribuicao	Start of job execution			2020/11/13 17:06:15
dim_tipoDistribuicao	Job execution finished	Success	3	2020/11/13 17:06:16
dim_Etapa	Start of job execution			2020/11/13 17:06:16
dim_Etapa	Job execution finished	Success	4	2020/11/13 17:11:42
dim_tipoEtapa	Start of job execution			2020/11/13 17:11:42
dim_tipoEtapa	Job execution finished	Success	5	2020/11/13 17:11:43
dim_tempo	Start of job execution			2020/11/13 17:11:43
dim_tempo	Job execution finished	Success	6	2020/11/13 17:24:14
fact_eventos	Start of job execution			2020/11/13 17:24:14
fact_eventos	Job execution finished	Success	7	2020/11/13 17:27:26
Job: load	Job execution finished	Success	7	2020/11/13 17:27:26

Figura 4.1: Log da Execução do job

A duração foi cerca de 20 minutos. Apesar de ser um tempo aceitável, como referido no capítulo anterior, a BD utilizada é uma BD de testes e estática. Como tal, caso fosse usada uma BD com dados de cliente reais e que estivesse a ser constantemente atualizada, era expectável que este valor fosse aumentando sempre que fosse executada uma nova iteração do job até que deixasse de ser aceitável. Para testar esta situação, foi criada uma nova tabela de etapas diretamente na BD de testes fornecida pela edoclink com um volume de dados de 10 vezes superior. Os

dados acrescentados são apenas cópias dos dados já existentes, por isso não têm qualquer tipo de valor para análise, servem apenas para simulação da duração do processo de ETL.



Job / Job Entry	Comment	Result	Nr	Log date
load1				
Job: load1	Start of job execution			2020/11/17 13:45:31
START	Start of job execution			2020/11/17 13:45:31
START	Job execution finished	Success	0	2020/11/17 13:45:31
Transformation	Start of job execution			2020/11/17 13:45:31
Transformation	Job execution finished	Success	1	2020/11/17 13:45:32
Transformation 5	Start of job execution			2020/11/17 13:45:32
Transformation 5	Job execution finished	Success	2	2020/11/17 13:45:35
Transformation 6	Start of job execution			2020/11/17 13:45:35
Transformation 6	Job execution finished	Success	3	2020/11/17 15:42:01
Transformation 2	Start of job execution			2020/11/17 15:42:01
Transformation 2	Job execution finished	Success	4	2020/11/17 15:42:01
Transformation 3	Start of job execution			2020/11/17 15:42:01
Transformation 3	Job execution finished	Success	5	2020/11/17 15:42:03
Transformation 4	Start of job execution			2020/11/17 15:42:03
Transformation 4	Job execution finished	Success	6	2020/11/17 15:45:16
Transformation 7	Start of job execution			2020/11/17 15:45:16
Transformation 7	Job execution finished	Success	7	2020/11/17 16:18:03
Job: load1	Job execution finished	Success	7	2020/11/17 16:18:03

Figura 4.2: Log da Execução do job com um volume de dados 10x superior

Como esperado, o tempo da execução do job foi consideravelmente superior ao anterior (20 minutos vs 2 horas e 30 minutos), demonstrando que a solução não é escalável à medida que o volume de dados da BD aumente. Isto poderia ser resolvido caso o job, sempre que fosse executado diariamente, apenas inserisse/atualizasse dados que tivessem sido carregados/alterados desde a última execução. Umas das maneiras de se conseguir isto poderia ser através da criação de uma tabela de controlo que contivesse a data da última execução e o intervalo de tempo desejado (no caso de a execução ser diária, o intervalo seria 24 horas). Assim, cada transformação filtraria os dados vindos da BD do edoc com base nos dados da tabela de controlo, carregando/atualizando apenas os novos dados desde a última execução.

4.3 Produtização

De modo a analisar se a solução é passível de ser posta em produção, é preciso analisar a arquitetura e tecnologia utilizada e perceber que alterações teriam que ser feitas aos componentes da versão corrente para uma versão de cliente.

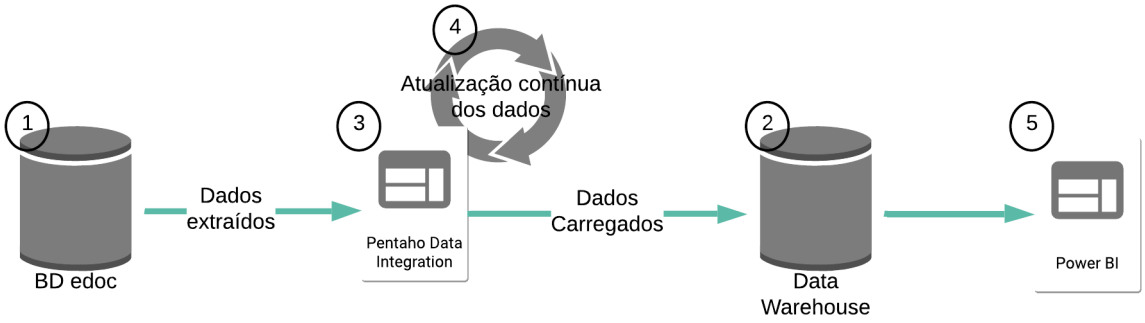


Figura 4.3: Arquitetura da solução implementada

- **BD de origem -** Em vez de se utilizar a BD de testes da versão corrente, utilizaria-se a BD do cliente com os dados gerados pelos seus processos de negócio. A única alteração necessária seria a configuração da conexão estabelecida entre o PDI e a BD.
- **Data Warehouse -** O DW da versão corrente não teria que sofrer nenhuma alteração, uma vez que mesmo mudando de BD de origem, os dados gerados, sendo na mesma através do edoc, apresentariam o mesmo formato.
- **Pentaho Data Integration -** Como referido, as únicas alterações necessárias seriam ao nível da configuração da conexão entre cada transformação do PDI e a BD do cliente. Poderia-se também fazer uma alteração das measures na transformação da tabela de factos para apresentar o tempo em minutos (em vez de segundos), uma vez que os dados seriam mais realísticos do que na BD de testes.
- **Atualização dos dados -** Na calendarização do job, também não haveria qualquer tipo de problema, podendo o cliente decidir se queria fazer a execução localmente ou num servidor seu - facilmente configurado diretamente no PDI.
- **Power BI -** Finalmente, no Power BI a única alteração necessária seria à conexão entre o programa e o DW, dependendo do host do DW. Caso fosse local, não seria preciso qualquer alteração. Os dashboards apresentariam automaticamente os dados do DW do cliente, que consequentemente continha dados vindos da BD gerada através da execução dos processos do mesmo.

Pode-se concluir que a solução é de facto produtizável, sendo apenas necessárias algumas alterações simples, maioritariamente ao nível da conexão entre a BD de cada cliente e cada compentente da solução.

4.4 Usabilidade dos dashboards

Por fim, para avaliar os dashboards com base na sua usabilidade e no que permitem visualizar, foi pedida uma opinião a um colaborador da edoclink.

Figura 4.4: Compilação dos dashboards

Os dashboards cumprem o objetivo definido inicialmente. A cada indicador corresponde um dashboard que apresenta os dados de uma forma interativa, direta e user-friendly. O mais importante é o facto de mesmo um cliente (que poderá não ter um conhecimento vasto do ambiente, conceitos e funcionamento do edoc) conseguir abrir um dashboard e compreender de imediato o que está a querer ser apresentado.

No entanto, apesar de não comprometer a análise dos dados, foi unânime que uma melhoria a fazer seria juntar todos os dashboards num mesmo ficheiro de Power BI para que o acesso aos mesmos fosse facilitado. Isto poderia ter sido implementado, caso tivesse sido criado um modelo redundante diretamente no Power BI que estabelecesse as relações entre todas as tabelas e permitisse que todos os indicadores fossem visualizados através do mesmo.

Capítulo 5

Conclusão

Ao longo do documento, foram apresentados o tema e conceitos relativos ao projeto, detalhada a solução proposta para dar resposta aos objetivos definidos e discutidos os resultados e a sua avaliação. Os objetivos do projeto foram cumpridos, tendo-se desenvolvido uma solução capaz de, através da extração e transformação de dados da BD do edoc, dar resposta aos indicadores definidos através de dashboards simples e interativos. Para além disso, concluiu-se que a solução é extensível e produtizável, o que permite a adição de novos indicadores/atributos e a produção de versões para clientes.

No entanto, e relativamente também a trabalho futuro, seria importante tornar a solução escalável, uma vez que na versão corrente, é ineficiente e o tempo de execução do job irá aumentar à medida que o volume de dados vai aumentando. Como sugerido no capítulo anterior, uma possível melhoria seria criar uma tabela de controlo com a data da última iteração do job e o intervalo de tempo entre cada execução, que filtrasse os valores a analisar em cada iteração do job. Outro aspeto que também pode ser explorada é utilizar os resultados obtidos para automatizar o processo de alocação de tarefas diretamente no edoc. Por exemplo, caso se decida que o critério é atribuir ao executante com menos tarefas pendentes, quando uma etapa fosse atribúida a um interveniente, seria automaticamente atribuída a esse executante, não necessitando de ser decidido e aceite manualmente. Isto é algo que já está a ser posto em prática por uma colega que começou a sua dissertação na Link.

Em conclusão, o projeto foi de facto interessante, cumpriram-se os objetivos e obtiveram-se resultados relevantes. Permitiu-me ter a minha primeira experiência a nível empresarial e interagir com várias ferramentas e aplicações desconhecidas para mim até ao início do projeto. Queria agradecer uma vez mais à Link Consulting e, mais especificamente, ao prof. Pedro Sousa, à Carolina Marques e ao João Guilherme por todo o apoio e ajuda que me deram ao longo de toda a duração do projeto e pela forma como me receberam.

References

[1] João Alves. «Sistema de Business Intelligence no Projeto Educativo de Guimarães». Em: 2015.

[2] Kevin Bartley. URL: https://rivery.io/etl-vs-elt-whats-the-difference/.

[3] Alessandro Berti, David Zang e Magdalena Lang. «An Open-Source Integration of Process Mining Features into the Camunda Workflow Engine: Data Extraction and Challenges». Em: set. de 2020.

[4] OCDE - Organization for Economic Co-operation e Development. «Glossário de termos-chave na avaliação». Em: 2002.

[5] edoclink. «edoclink White Paper». Em: 2016.

[6] edoclink Enterprise. URL: https://www.linkconsulting.com/what-we-do/products/edoclink1/.

[7] Michele Hart. URL: https://help.pentaho.com/Documentation/7.1/0J0/0C0/020.

[8] Muhammad Zafar Karmani et al. «A Review of Star Schema and Snowflakes Schema». Em: mai. de 2020, pp. 129–140. DOI: 10.1007/978-981-15-5232-8_12.

[9] Ralph Kimball et al. *The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing and Deploying Data Warehouses with CD Rom.* 1998.

[10] Michal PENČIKOV. «Techniques and Methods for Effective Collection and Transformation of Process Metrics Data». Em: 2014.

[11] Eugénio Santos. «Indicadores de Desempenho de Bases de Dados». Em: (2017).

[12] Innovent Solutions. URL: http://www.innoventsolutions.com/pentaho-review.html.

[13] Filipe Taveira. «Análise de risco com base em indicadores». Em: 2014.

[14] UNAIDS. *An Introduction to Indicators.*