UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ELETRÔNICA E SISTEMAS

MESTRADO EM ENGENHARIA ELÉTRICA

DESENVOLVIMENTO DE APLICAÇÕES ETL COMO UMA PROPOSTA PARA REDUÇÃO DE CUSTOS EM PROJETOS DE DATA WAREHOUSE

CARLA ORAN FONSECA DE SOUZA

DISSERTAÇÃO DE MESTRADO

RECIFE - PE
29 de setembro de 2003

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ELETRÔNICA E SISTEMAS

CARLA ORAN FONSECA DE SOUZA

DESENVOLVIMENTO DE APLICAÇÕES ETL COMO UMA PROPOSTA PARA REDUÇÃO DE CUSTOS EM PROJETOS DE DATA WAREHOUSE

Trabalho apresentado ao Centro de Tecnologia e Geociências da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Engenharia Elétrica com Ênfase em Redes de Computadores.

Orientador: Prof. Rafael Dueire Lins, PhD.
Co-orientador: Prof. Fernando da Fonseca de Souza, PhD.

RECIFE - PE
29 de setembro de 2003

CARLA ORAN FONSECA DE SOUZA

DESENVOLVIMENTO DE APLICAÇÕES ETL COMO UMA PROPOSTA PARA REDUÇÃO DE CUSTOS EM PROJETOS DE DATA WAREHOUSE

Trabalho apresentado ao Centro de Tecnologia e Geociências da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Engenharia Elétrica com Ênfase em Redes de Computadores.

Aprovado em 29 de Setembro de 2003.

BANCA EXAMINADORA

Rafael Dueire Lins, PhD
Orientador

Profa. Fernanda Maria Ribeiro de Alencar, PhD

Prof. José Laurindo Campos dos Santos, PhD

À minha avó Neca, minha mãe e meus irmãos

Agradecimentos

Primeiramente, agradeço à minha mãe pelo amor, carinho e todos os sacrifícios vividos para que eu alcançasse todas as minhas conquistas. À minha querida irmã Carol, que sempre me foi motivo de orgulho e ao meu irmãozinho Dudu, alegria da minha vida. Não poderia deixar de agradecer à minha tia Lenir por tanta dedicação e à minha prima Karina por ser essa pessoa doce e ter me dado um priminho tão lindo. Agradeço também à minha avozinha Neca por nunca medir esforços na sua admirável dedicação à nossa família.

À Áurea, Nis e Nívia, minhas irmãs do coração, agradeço pela amizade, apóio e risos em todos esses anos de convivência. Agradeço, com muito carinho, ao mais que amigo Casé, por sempre acreditar em mim e fazer parte da minha vida.

Não poderia esquecer a turma do viradão, em particular à Andréa, Vívian e Luciana, pelo espírito de equipe durante as madrugadas acordadas no período dos créditos.

Agradeço, em especial, ao Prof. Fernando da Fonseca de Souza, por seus ensinamentos e dedicação para que meu sonho se tornasse realidade.

Agradeço também aos amigos de faculdade Mário, Maykel e Werley por esses oito anos de amizade. E a todos que de forma direta ou indireta contribuíram para a realização desse trabalho, em particular à Julie, Jonathas, Marco e André.

Finalmente, agradeço a Deus pelas oportunidades que sempre me deu para superar minhas limitações e por ter colocado no meu caminho pessoas tão maravilhosas.

‘Viver sem errar ou fracassar em nada
durante toda a vida, é o mesmo que
a incerteza do sucesso pleno ou a
indignação de nunca termos
se quer tentado fazer algo.
—‘

Resumo

O acesso privilegiado a informações estratégicas cada vez mais vem se tornando uma arma poderosa para manutenção das empresas no mercado. De um modo geral, as organizações possuem um grande volume de dados, mas não dispõem de mecanismos capazes de tratá-los e convertê-los em informações relevantes para o processo decisório. Dentre os mecanismos empregados para prover essas necessidades, destaca-se a tecnologia de Data Warehousing, a qual começou a ser difundida no início da década de 80, com o conceito de bancos de dados coorporativos. Essa tecnologia certamente apresenta inúmeras vantagens à organização, porém exige altos investimentos para sua implantação. Durante o desenvolvimento de um Data Warehouse (DW), uma das fases mais dispendiosas inclui as etapas de Extração, Transformação e Carga dos dados (ETL), sendo um dos fatores de elevação dos custos, a aquisição de ferramentas para automatizar esse processo. Esse trabalho tem como principal objetivo apresentar uma alternativa de redução de custos para projetos de DW, por meio da implementação de uma aplicação para extração, transformação e carga de dados em ambientes com bases de dados homogêneas. A técnica adotada nessa investigação foi um estudo de caso, tendo como cenário o Departamento de Tributação da Prefeitura Municipal de Manaus. Nesse ambiente, desenvolveu-se uma aplicação para automatizar o processo de ETL, de forma a demonstrar a viabilidade dessa fase de movimentação dos dados, sem a necessidade da organização despender recursos para a aquisição de uma ferramenta ETL comercial.

Palavras-chave: Data Warehouse; Data Warehousing; ETL; Extração, Transformação e Carga; Ferramentas ETL.

Abstract

The privileged access to strategic information is becoming an even powerful tool in order to keep enterprises in the market. In general, the organizations keep a great amount of data but they usually do not have the necessary tools able to deal with and convert them into relevant information for the decision process. Among the tools used to del with these factors it is stressed the Data Warehousing technology, which was first widely known in the very beginning of the 80's as a concept of corporative database. This technology certainly offers large advantages for the organizations but it demands a high level of investments for its deployment. One of the most expensive phase of the development of Data Warehousing (DW) is the Extraction, Transformation and Load of data (ETL), and the cost increasing factor is the acquisition of tools to improve this process. The main objective of this work is to offer an alternative for cost reduction of DW projects through the implementation of a tool capable of extraction, transformation and load data in an environment with homogeneous database. The technology used during this investigation was based in a study made at the City of Manaus Department of Taxation. It was developed in the environment an application in order to automatize the ETL process and to show up that this phase of data processing could be improved without the expensive acquisition of commercial ETL tools.

Key words: Data Warehouse; Data Warehousing; ETL; Extraction, Transformation and Load; ETL tools.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Durante anos, as empresas de grande, médio e, até mesmo, pequeno porte investiram recursos no desenvolvimento de ferramentas computacionais que auxiliassem no controle e agilizassem tarefas operacionais, tais como controle de compras, contas a pagar e a receber, estoque, dentre outros.

Com o passar do tempo, novas necessidades foram surgindo, em razão, principalmente, do crescimento da competição entre as organizações, transformando a informação em uma arma poderosa para garantir o sucesso e, até mesmo, a permanência de uma empresa no mercado.

Nesse contexto, os sistemas tradicionais tornaram-se pouco eficientes, uma vez que acumulam muitos dados a respeito da organização, mas pouca informação que possam auxiliar nas tomadas de decisões.

Em muitas empresas, quando um executivo faz uma pergunta sobre o perfil dos seus negócios que fujam ao padrão conhecido por seus analistas, o tempo e o esforço necessários para respondê-la são muito grandes, uma vez que vários sistemas devem ser consultados e os dados resultantes devem ser cruzados [Edu02]. Como as decisões precisam acompanhar o ritmo do mercado, os executivos acabam por usar sua própria intuição ao invés de basearem-se em dados concretos.

Como resposta à necessidade crescente de informações rápidas a respeito da organização, no intuito de posicioná-la estrategicamente para ser mais competitiva, maximizando os lucros e reduzindo os índices de erros, introduziu-se um novo conceito no mercado, o Data Warehouse (DW), cujo objetivo é dar suporte aos processos de tomada de decisão.

A crescente preocupação em acertar com precisão, em tempos reduzidos, fez com que Universidades e instituições comerciais passassem a investir recursos em pesquisas e, conseqüentemente, no emprego desse poderoso mecanismo de tratamento de grandes volumes de dados.

O desenvolvimento de um projeto de DW, sem dúvida alguma, traz consigo enormes benefícios a uma empresa, tais como: consolidação de dados inconsistentes oriundos dos sistemas transacionais da organização, descoberta de informações estratégicas antes ocultas, dentre outros. Entretanto, para usufruir das vantagens dessa tecnologia, uma

soma considerável de recursos, tanto financeiros quanto de tempo e pessoal, deve ser despendida.

Atualmente, a preocupação com a redução de custos sem prejudicar as metas estabelecidas, vem levando as organizações a optarem por tecnologias mais baratas e até mesmo gratuitas. Um bom exemplo desse novo rumo que a Tecnologia da Informação vem seguindo é o Projeto Software Livre Paraná, idealizado em setembro de 2002, cuja idéia fundamental é a disseminação do uso do Linux (um software livre) em todos os computadores da administração pública estadual no Paraná [dE03].

Com isso, os gestores deparam-se com uma questão realmente relevante: empregar altos investimentos e utilizar os benefícios que um DW pode proporcionar ou, por aspectos de redução de custos, intimidar-se com os valores e simplesmente ignorar a existência dessa ferramenta.

Considerando, diante do cenário apresentado, que a busca por alternativas para encontrar um ponto de equilíbrio entre a utilização de ferramentas de suporte à decisão e os gastos decorrentes das mesmas, deve ser uma constante, será apresentada uma alternativa para a redução de custos de um projeto de DW, focando especialmente, a extração, transformação e carga de dados.

1.1 Motivação

A transferência de dados do ambiente operacional para o ambiente DW, consiste em um das fases mais críticas e dispendiosas do desenvolvimento desse tipo de sistema, abrangendo três atividades correlatas conhecidas como: extração, transformação e carga dos dados ou, simplesmente, ETL.

Um dos aspectos que contribui para a complexidade da ETL é o fato da mesma ser intimamente dependente dos sistemas transacionais que dão origem ao DW. Estes sistemas, muitas vezes concluídos por equipes e em épocas distintas, podem apresentar diferentes paradigmas de desenvolvimento, como por exemplo o relacional, sistemas de arquivo e o objeto-relacional [PER00].

Os investimentos em pesquisas na tecnologia de Data Warehousing, particularmente, em ETL, contribuíram para o desenvolvimento de produtos que automatizassem tais atividades. Contudo, essas ferramentas requerem o emprego considerável de recursos financeiros para sua aquisição e aprendizagem.

Segundo [CIE02], algumas ferramentas desse gênero possuem custo zero, pois vêm embutidas em um Sistema Gerenciador de Banco de Dados (SGBD). Porém, são bastante limitadas e exigem um esforço maior de codificação dos processos ETL.

É válido considerar que esse custo nulo descrito por Cielo só é possível se o SGBD for gratuito. Além disso, o mesmo deverá ser adotado pela organização para que os seus recursos de ETL possam ser utilizados.

Assim, uma boa alternativa para a substituição de uma ferramenta ETL comercial com alto custo de aquisição, é a implementação de uma aplicação com esse intuito, por parte dos próprios desenvolvedores do projeto, utilizando softwares de desenvolvimento já existentes e disseminados dentro da organização. Essa alternativa e a possibilidade de redução de gastos advinda da mesma constituem a grande motivação dessa investigação.

1.2 Descrição do Problema

A qualidade dos produtos ou serviços oferecidos pelas organizações indubitavelmente é hoje um dos maiores requisitos na conquista e manutenção de clientes. De acordo com [PET03], uma visão distorcida, porém muito difundida nas organizações, é que qualidade e custo são proporcionais, em outras palavras, melhor qualidade implica diretamente em maior custo de produção.

Esse custo é formado por diversos componentes, dentre os quais pode-se citar: matéria-prima, energia, mão-de-obra, recursos tecnológicos utilizados, etc.

Quanto aos recursos tecnológicos, empregados como suporte ao processo de produção, encontra-se a área de Tecnologia da Informação (TI), para a qual as organizações destinam boa parte de seus orçamentos. Somente o Governo Federal despende por ano um montante de aproximadamente R$ 1 bilhão apenas para renovar as licenças de softwares usados pelos órgãos públicos [dE03]. Isso representa uma soma considerável deixando de ser direcionada para outros setores certamente mais prioritários.

De um outro ponto de vista, considerando o dinamismo que rege o mundo moderno, os recursos tecnológicos são extremamente importantes para manter uma empresa em posição estratégica. Porém, os altos custos necessários para investir nos mesmos tendem a inibir muitas iniciativas de utilizar os benefícios que podem oferecer.

Uma importante ferramenta da TI que, se bem empregada, pode agregar bons valores às organizações, em particular ao processo decisório, é o DW. Desenvolver uma ferramenta desse porte certamente não é tarefa fácil, principalmente em função das redes de informação nas organizações, que servem de fonte de dados para o DW, basearem-se muitas vezes em múltiplas bases operacionais.

A movimentação de tais dados para o ambiente DW pode demandar bastante tempo de trabalho, chegando a consumir mais de 50% do tempo de desenvolvimento do pro-

jeto [C+01]. Essas características tornam o custo desse processo, conhecido como ETL, bastante significativo, impactando diretamente no projeto como um todo.

Um dos aspectos que encaressem sobremaneira esse processo é a aquisição de ferramentas para auxiliar os desenvolvedores durante a execução dos procedimentos envolvidos nessa etapa. Muitas dessas ferramentas permitem trabalhar com dados de bases heterogêneas, reduzindo a necessidade de codificação para integração de diferentes ambientes.

Entretanto, em muitos casos, os dados que constituirão o DW advêm de bases implementadas sobre plataformas homogêneas. Nesses ambientes, considerados menos complexos, sob a ótica da tecnologia Data Warehousing, a utilização de ferramentas ETL pode ser substituída pela implementação de aplicações por parte dos integrantes da equipe de desenvolvimento.

De acordo com Craig, Vivona e Bercovitch (apud [PER00]), essa é uma boa solução quando aplicada em ambientes simples, por exemplo, com apenas uma fonte de dados ou com diversas fontes de dados construídas sobre o mesmo paradigma de desenvolvimento (ex.: SGBD relacionais).

Diante desse cenário, esta pesquisa apresenta a seguinte **hipótese**:

- É possível reduzir os custos em projetos de DW com a implementação de aplicações para extração, transformação e carga de dados por parte da própria equipe de desenvolvimento.

1.3 Objetivos

Este trabalho tem como objetivo geral apresentar uma alternativa para redução de custos para projetos de DW, por meio da implementação de uma aplicação para extração, transformação e carga de dados, em um ambiente com fontes implementadas com o mesmo paradigma de desenvolvimento.

Para a implementação dessa aplicação foram empregados apenas ferramentas de desenvolvimento existentes na organização utilizada no estudo de caso, com o intuito de mostrar a viabilidade da realização desses procedimentos sem a necessidade de investir altos custos na aquisição de um pacote de software ETL comercial.

Os objetivos específicos do trabalho são:

- analisar e integrar os esquemas das fontes de dados envolvidas no estudo de caso;
- projetar e implementar um DW no cenário escolhido para estudo de caso;

- realizar a extração, transformação e carga dos dados para o ambiente DW e
- comparar a aplicação desenvolvida e três ferramentas comerciais (Oracle8i Warehouse Builder, Warehouse Workbench 4.96 e Data Transformation Service) em termos de custos de aquisição, treinamento e flexibilidade.

1.4 Metodologia

O desenvolvimento dessa investigação iniciou com o levantamento bibliográfico sobre o tema, com o intuito de compor a base teórica para a pesquisa. A técnica empregada para comprovação da hipótese apresentada foi estudo de caso, o qual teve como cenário o Departamento de Tributação da Prefeitura Municipal Manaus, por possuir um ambiente favorável aos propósitos da dissertação, envolvendo por exemplo: um conjunto de bases de dados implementadas sobre o mesmo paradgma (relacional) e licenças de ferramentas de desenvolvimento.

Nesse ambiente, foram feitas entrevistas aos usuários para definição do negócio da organização a ser mapeado para o Data Warehouse. Após essa especificação, as fontes de dados necessárias para o processo foram analisadas, com o objetivo de estudar as estruturas das mesmas.

Seguindo as etapas de construção de um DW, os esquemas das bases de dados passaram por um processo de integração, a qual baseou-se na metodologia proposta por [BLN86]

Uma vez integradas as bases de dados, a Data Staging Area e o DW puderam ser modelados e terem seus esquemas criados. Para essa implementação, utilizou-se como SGBD, o Oracle 8i.

Para a extração, transformação e carga dos dados desenvolveu-se uma aplicação para essa finalidade. Esse desenvolvimento iniciou com a modelagem do software, utilizando o padrão Unified Modeling Language ou UML [B$^+$00]. Na implementação da aplicação, foram empregados os seguintes softwares já existentes na Prefeitura: Delphi 5.0 [CAN00], SQL Navigator 3.0 d2 [Sof03] e Oracle 8i. [C$^+$01]

Após essa implementação, realizou-se uma comparação entre a aplicação desenvolvida e três ferramentas ETL comerciais, em termos de custos de aquisição, treinamento e flexibilidade. As ferramentas utilizadas nessa análise foram: Oracle8i Warehouse Builder [C$^+$01], Warehouse Workbench 4.96 [SOL03] e Data Transformation Service [COF00] [PER00].

1.5 Organização da Dissertação

Esta dissertação, além do capítulo introdutório, apresenta também outros quatro capítulos descritos a seguir.

O Capítulo 2 constitui uma base teórica, a partir dos pressupostos sobre o assunto, cuja finalidade é apresentar os conceitos relevantes para o entendimento do trabalho. No seu decorrer, são considerados aspectos a respeito da evolução dos Sistemas de Apoio à Decisão, da tecnologia Data Warehousing, seu ciclo de vida e modelagem dimensional. Por fim, são apresentados conceitos sobre o processo e ferramentas de extração, transformação e carga de dados, bem como uma metodologia para integração de esquemas de bases de dados.

No Capítulo 3, são descritas as consideração técnicas sobre a implementação da aplicação ETL e o ambiente utilizado como estudo de caso.

O Capítulo 4 apresenta três ferramentas ETL comerciais e uma análise comparativa entre tais ferramentas e a aplicação desenvolvida nesse estudo.

Finalmente, no Capítulo 5, são descritas as considerações finais da dissertação, as dificuldades encontradas durante seu desenvolvimento e sugestões de futuros trabalhos.

Capítulo 2

Fundamentação Teórica

Este capítulo inicia-se com uma abordagem sobre a evolução dos Sistemas de Apoio à Decisão, enfatizando conceitos sobre DW, suas características, arquitetura, ciclo de vida e modelagem.

Posteriormente, são traçadas considerações acerca dos processos de extração, transformação e carga de dados, empregados durante o desenvolvimento de um DW, bem como a respeito de ferramentas ETL.

Para que os dados possam ser transferidos para o ambiente DW, é essencial que os esquemas das bases envolvidas nesse procedimento sejam integrados. Portanto, para finalizar este capítulo, é apresentada uma metodologia descrita em [BLN86] para realização de tal necessidade.

2.1 Evolução dos Sistemas de Apoio à Decisão

Durante o início da década de 60, a computação consistia em aplicações escritas especialmente em COBOL, não integradas e executadas sobre arquivos mestres. Tais arquivos eram armazenados em fitas magnéticas, cujas principais características eram a alta capacidade de armazenamento a um custo reduzido. O uso continuado desses recursos trouxe como conseqüência, por volta de 1965, o surgimento de enormes quantidades de dados redundantes e a complexidade na manutenção dos programas existentes e desenvolvimento de novos [INM97]

Na década seguinte, as fitas magnéticas deram lugar ao armazenamento em disco, também conhecido como Direct Access Storage Device ou Dispositivo de Armazenamento de Acesso Direto (DASD). Esse advento era substancialmente diferente da tecnologia anterior, uma vez que os dados podiam ser acessados diretamente, ao contrário do que ocorria com as fitas magnéticas, nas quais o acesso era seqüencial.

Paralelo ao surgimento do DASD, veio um novo tipo de software chamado de Sistema Gerenciador de Bancos de Dados (SGBD). De acordo com Elmasri e Navathe (2000), um SGBD é uma coleção de programas para facilitar a definição, construção e manipulação de bases de dados para várias aplicações.

Ainda durante a década de 70, o processamento de transações on-line passou a ser realizado sobre os bancos de dados, proporcionando um acesso mais rápido e abrindo perspectivas totalmente novas para o uso do computador [INM97].

Posteriormente, com o aparecimento dos computadores pessoais (PC) veio também a percepção de que os dados, anteriormente utilizados exclusivamente para fins operacionais, também poderiam viabilizar decisões gerenciais. Com essa nova visão, surgiu o conceito de Sistemas de Apoio à Decisão (SAD).

Os SAD representam uma classe de sistemas fundamentados nos conceitos de inteligência artificial, que são capazes de apoiar a decisão em domínios específicos [FNC98]. Segundo Turban e Aronson *apud* [SJF02], SAD são sistemas que pretendem ser interativos, flexíveis e adaptáveis, sendo capaz de encontrar a melhor alternativa possível para problemas de tomada de decisão. Dentre os SAD encontram-se: os Sistemas de Informações Geográficas [C$^+$96], DW Geográficos [MEL03] e DW [INM97]

Os DW, detalhados na próxima seção, são SAD que tem por finalidade permitir a análise dos dados da organização e facilitar, com isso, a descoberta de informações antes difíceis de serem obtidas.

2.2 Data Warehousing e Data Warehouse

De acordo com Machado (2000), a tecnologia Data Warehousing é considerada por diversos autores da área como uma evolução natural do Ambiente de Apoio à Decisão, de forma que sua crescente utilização pelas empresas está intimamente relacionada à necessidade de se dominar informações estratégicas a fim de garantir respostas e ações mais velozes. Para a implementação dessa tecnologia, são necessários repositórios de dados denominados Data Warehouse (DW).

Segundo [INM97], um DW é uma coleção de dados orientada por assuntos, integrada, variante em relação ao tempo e não volátil, cujo objetivo é dar suporte aos processos de tomada de decisão.

A seguir são detalhadas as quatro características de um DW, propostas por Inmon.

2.2.1 Orientação por Assunto

Enquanto o desenvolvimento de sistemas convencionais tem seu foco voltado no projeto do banco de dados e implementação de processos de controle das funções operacionais, o foco de um DW é o negócio da organização. Nesse caso, os dados de interesse são agrupados por assunto.

De acordo com [FRE02], assunto é um conjunto de informações relativas à determinada área estratégica de uma empresa. Por exemplo: em um hospital, as áreas estratégicas poderiam ser os pacientes, o atendimento, epidemias, dentre outras.

2.2.2 Integração

Nos sistemas transacionais ou operacionais distribuídos pela empresa, muitas vezes, os dados neles armazenados encontram-se codificados em vários padrões. Por exemplo, em um sistema, o atributo data foi codificado como DD/MM/AAAA (Ex: 01/05/2001) e em outro sistema convencionou-se que o formato da data seria MM/DD/AA (Ex: 12/26/01).

Como tais sistemas povoarão a base dos dados do DW, é necessário integrar e padronizar seus dados antes de armazená-los. No exemplo acima, poder-se-ia adotar como padrão o formato DD/MM/AAAA. A Figura 2.1 ilustra o exemplo descrito.

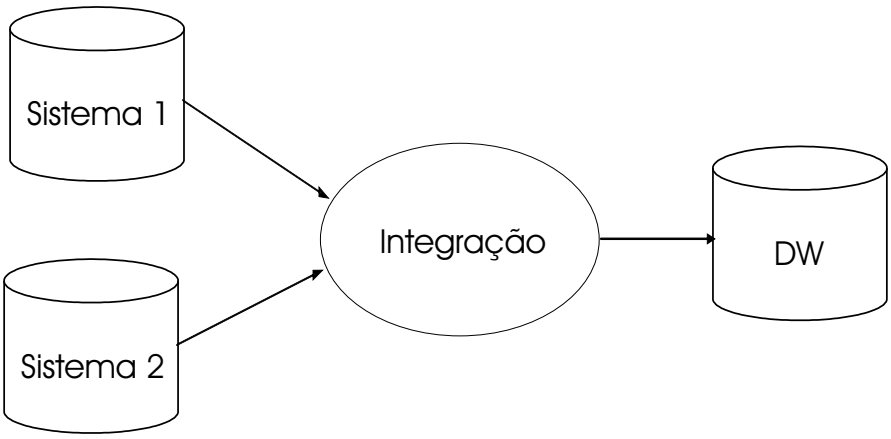

Figura 2.1. Integração de dados. Fonte adaptada [INM97]

De acordo com [INM97], a integração é marcante para o projeto de DW, sendo considerada a mais importante dentre todas as suas características, uma vez que o grande objetivo desse tipo de SAD é proporcionar informações gerenciais, as quais se tornariam inviáveis caso os dados fossem inconsistentes.

2.2.3 Variação no Tempo

Todo DW deve ser variante no tempo. Em outras palavras, este deve manter um histórico dos fatos que ocorrem na empresa por um período muito superior ao dos sistemas convencionais, para os quais um horizonte de tempo de 60 a 90 dias é satisfatório [INM97].

Essa característica é facilmente explicada, uma vez que os DW têm por objetivo municiar os tomadores de decisão, os quais devem ter, o máximo possível, a visão do

todo, não se limitando a fatos isolados que ocorram num período de tempo muito curto.

2.2.4 Não Volátil

Segundo [FRE02], um DW deve permitir uma carga inicial dos dados e, posterior a isso, disponibilizar tais dados para eventuais consultas dos usuários. Esse ambiente é conhecido como load-and-access (carga-e-acesso).

A carga dos dados no DW dá-se sob a forma de blocos de informações e não registro a registro como ocorre nos sistema transacionais, os quais precisam realizar diversas operações como *rollbacks*, *commits*, restrições, dentre outras, para validar cada entrada no banco.

2.2.5 Arquitetura

Para ser utilizável, um DW deve ser capaz de fornecer respostas rápidas aos questionamentos dos usuários, sem relegar a segundo plano detalhes relevantes que podem auxiliar no processo de tomada de decisão. Assim, para que isso seja possível, ele deve possuir uma arquitetura que lhe permita recuperar, manipular e apresentar de forma eficiente os dados. A Figura 2.2 apresenta os elementos básicos de um DW, inseridos em uma estrutura conceitual.

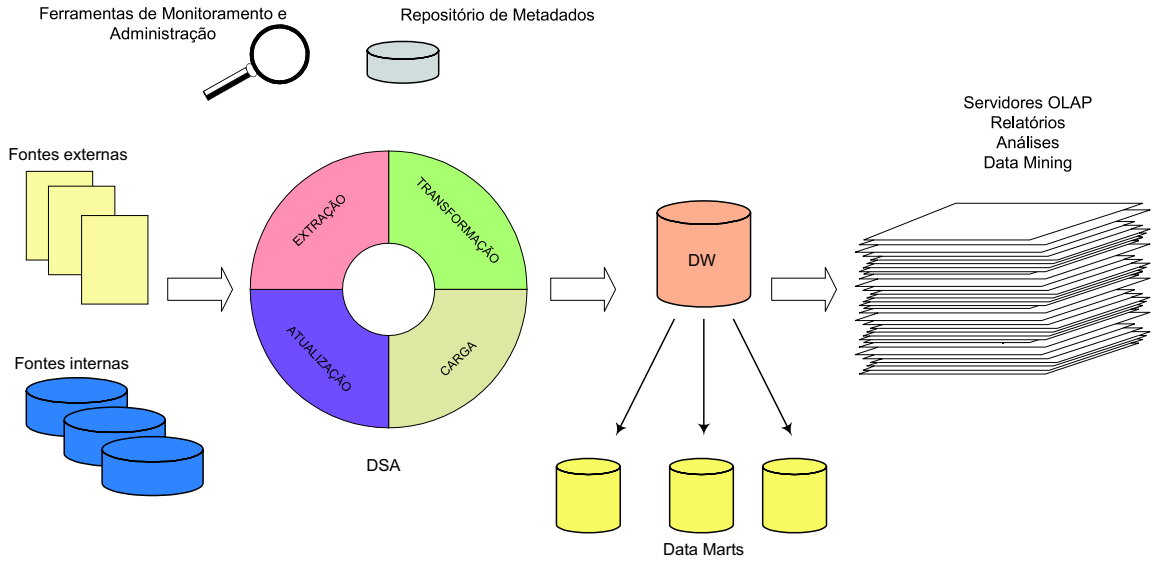

Figura 2.2. Arquitetura conceitual de um DW. Fonte adaptada [FLO99]

Os dados que constituirão o DW provêem, tipicamente, das bases de dados dos sistemas transacionais ou operacionais da organização, conhecidos como fontes internas, mas

também podem ter sua origem em fontes externas, tais como: arquivos texto, planilhas, dados provenientes da Internet, etc.

A *Data Staging Area* (DSA) é tanto uma área de armazenamento intermediária dos dados antes de serem efetivamente migrados para o DW, quanto um conjunto de processos, normalmente, denominados extração, transformação e carga [KIM02].

Como a criação de um DW requer considerável recurso de tempo, dinheiro e muito esforço por parte da equipe envolvida, muitas organizações iniciam esse tipo de projeto, centrando o foco nas necessidades de um determinado grupo de pessoas. Assim, podem existir bases de dados menores, mas também com propósitos gerenciais, chamadas Data Marts [Bra02].

Uma outra característica de um DW é a visão multidimensional dos dados. Na sua forma tradicional, a apresentação dos dados dá-se em duas dimensões (linhas e colunas). Entretanto em um projeto dessa natureza, consideram-se outras dimensões. Por exemplo: em uma dimensão estaria o mês da venda, em outra em que cidade ocorreu e na última estaria o vendedor responsável.

Como complemento às ferramentas de consultas e visualização dos dados estão as ferramentas de mineração. Esse processo de buscar informações que a princípio estão ocultas nos dados armazenados é chamado de Data Mining.

Quanto aos metadados, que são dados sobre dados, são armazenados e gerenciados pelos chamados repositórios de metadados.

> *[...]Como a maioria dos desenvolvedores possui uma aversão natural ao desenvolvimento e ao arquivamento ordenado de documentação, os metadados costumam ser retirados do plano de projeto, embora todos saibam o quanto eles são importantes [KIM02].*

Finalmente, para o gerenciamento do sistema como um todo, existem as ferramentas de monitoramento e administração.

2.2.6 Ciclo de Vida e Modelo Dimensional

Um aspecto importante de um DW, que o difere substancialmente de um sistema transacional refere-se ao seu ciclo de vida de desenvolvimento.

Enquanto em um sistema tradicional, o ciclo de vida inicia-se tipicamente pelo levantamento dos requisitos, passa pelo processo de análise e projeto para, finalmente, se implementar as necessidades dos usuários, em um DW, esse ciclo começa pelos dados, os quais uma vez disponibilizados, são transformados, integrados, testados e, somente então, é feita a codificação.

Segundo Inmom *apud* [MIR03], a compreensão dos requisitos vem com a análise dos programas que foram implementados, já que se o desenvolvedor for aguardar que todos os requisitos sejam levantados para dar início ao projeto, o DW nunca será construído.

Durante o desenvolvimento de um DW, a representação dos dados e dos relacionamentos entre eles é feita através do que se chama de Modelagem Dimensional ou MDM.

A MDM permite que o negócio da organização seja mapeado como um conjunto de valores descritos sob várias perspectivas, em outras palavras, a representação dos dados se dá em várias dimensões, onde cada uma delas é um tema ou um assunto [5498].

Ao contrário do que ocorre na modelagem de sistemas tradicionais, os modelos dimensionais são bastante assimétricos e não buscam a normalização dos dados.

Conforme Harrison *apud* [MIR03], existem cinco tipos de modelos que podem ser utilizados nesses casos. São eles: em estrela parcial, tabela de fatos particionada, tabela particional, floco de neve e estrela, sendo os dois últimos os mais empregados e conhecidos.

O modelo floco de neve procura encontrar um equilíbrio entre a normalização da base dados para evitar um alto índice de redundância e a desnormalização para obter aumento no desempenho do sistema [MIR03]. A Figura 2.3 ilustra um exemplo do emprego desse modelo para uma rede de supermercado.

No modelo estrela, existe uma tabela no centro do diagrama, chamada de tabela de fatos, e outras tabelas conectadas a esta, conhecidas como tabelas de dimensão, sendo que uma das dimensões mais importantes é a do tempo, em função da característica do DW descrita na sessão 2.2.3. A Figura 2.4 apresenta o diagrama da rede de supermercados, agora empregando o modelo em estrela.

Diferentemente do floco de neve, o modelo em estrela busca a otimização da performance do sistema, não se preocupando com a normalização da base de dados [MIR03]. Com essa característica o esquema resultante tende a ter menos relacionamentos, conseqüentemente, menos necessidade de operações de junções, uma vez que, para a realização de uma junção, faz-se necessária a criação lógica do produto cartesiano de todas as linhas das tabelas envolvidas no processo [VIE89] [DAT90]. Como esse número de

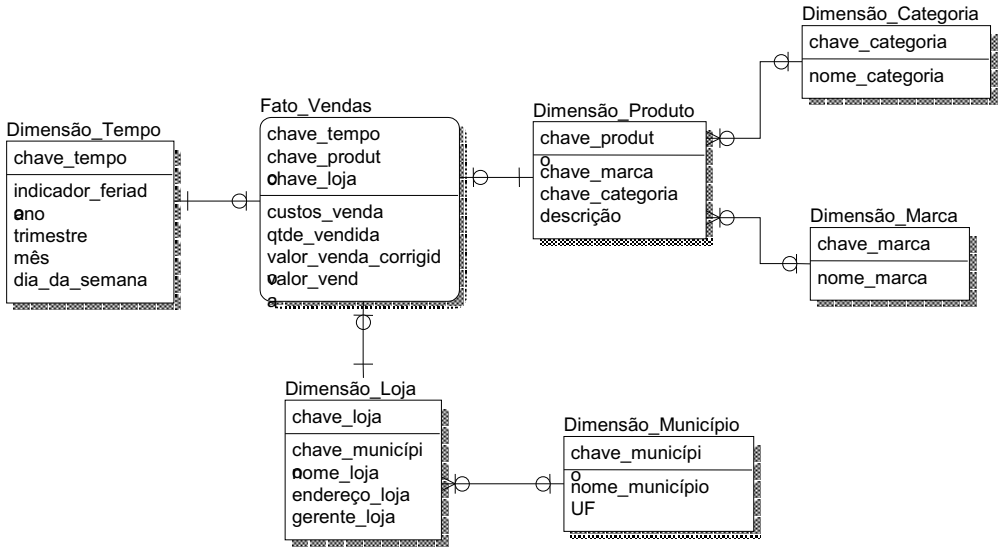

Figura 2.3. Exemplo do emprego do modelo Floco de Neve.

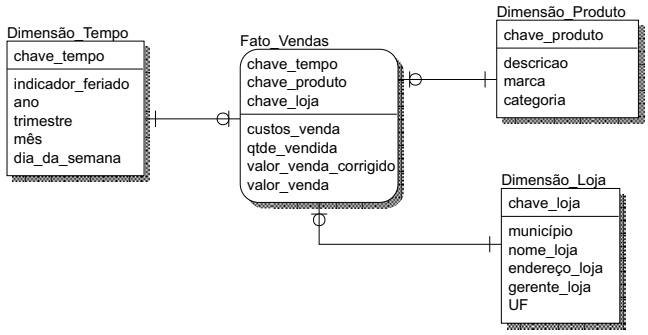

Figura 2.4. Exemplo do emprego do modelo em Estrela.

linhas manipuladas é resultante do produto das tuplas de cada tabela ou visão da junção, essa torna-se uma das operações que demandam um dos mais altos custos em termos de processamento de consultas.

2.2.7 Modelagem Relacional x Modelagem Multidimensional

Desde de que foi descrita por Codd *apud*[BLN86] em 1970, a modelagem relacional firmou-se de maneira extremamente sólida, tornando-se bastante apropriada para o desenvolvimento dos chamados sistemas transacionais, cujo foco são as operações e atividades

da empresa que podem ser automatizadas.

Entretanto, esse tipo de modelagem não atende a todos os requisitos que as novas tecnologias e necessidades de consultas e análise de dados vêm requerendo. Para este propósito, Kimball *apud* [FRE02], sugere a utilização da modelagem dimensional.

Uma das grandes vantagens da análise dimensional é sua simplicidade do ponto de vista do usuário. O desenvolvimento de um sistema tradicional (OLTP - On-Line Transaction Processing) pode envolver dezenas e, até mesmo, centenas de tabelas, ligadas por meio de relacionamentos complexos em razão da normalização da base de dados.

Esse processo de normalização tem por objetivo evitar redundâncias, mas, por outro lado, traz como conseqüência a complexidade da base de dados, a qual somente é acessada pelo usuário final por meio de uma interface amigável, conhecida como front-end. A Figura 2.5 ilustra a interação do usuário final com as bases de dados de sistemas OLTP.

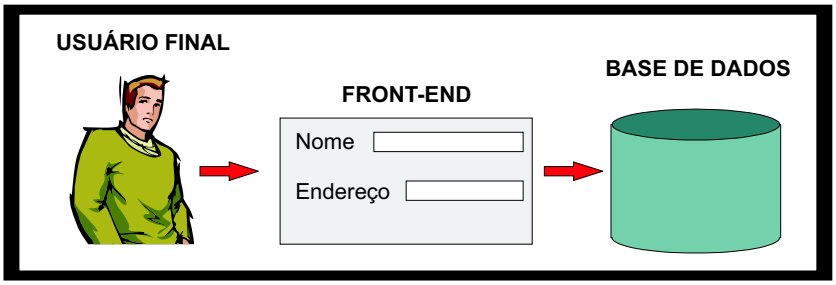

Figura 2.5. Interação dos usuários com as bases de dados dos sistemas OLTP.

Ao contrário do que ocorre em aplicações OLTP, em um DW, cujo modo de processamento é conhecido como OLAP (*On-Line Analytical Processing*), a base de dados não deve ser algo inacessível pelos usuários. Em função dessa necessidade, as tabelas e seus relacionamentos devem ser facilmente entendidos pelos mesmos.

Além disso, a normalização não é uma exigência para esse tipo de ambiente, uma vez que a preocupação é a otimização das consultas e não a manutenção da consistência interna da base, pois a carga e atualização dos dados não é feita pelos métodos tradicionais de *insert*, *update* e *delete*.

A Tabela 2.1 apresenta algumas comparações entre aplicações OLTP, cuja modelagem é feita tipicamente pelo modelo relacional, e aplicações OLAP, para as quais o emprego da modelagem dimensional é mais apropriado.

Tabela 2.1. *Comparação entre aplicações OLTP e OLAP. Fonte adaptada [FLO99]*

Característica	OLTP	OLAP
Objetivo	Automatização das atividades operacionais	Análise do negócio da empresa para subsidiar os tomadores de decisão
Consistência entre dados de diferentes sistemas	Um sistema OLTP é consistente dentro do seu escopo. Entretanto, quando se está observando vários sistemas, estes, geralmente, não são consistentes entre si.	Os dados, provenientes, de diversos sistemas, devem ser submetidos a processos para garantir a consistência dos mesmos
Tipo de usuário	Operadores	Tomadores de decisão
Estrutura da base de dados dados	Mudanças não são desejáveis	Ocorrem mudanças constantes para acompanhar o dinamismo dos negócios da organização
Redundância de dados	Não deve ocorrer. É evitada por meio de processos de normalização	Ocorre. É aceitável para deixar a base de dados mais simples e otimizar as consultas.
Interação com os usuários	É feita por meio de consultas e operações pré-definidas.	Operações pré-definidas e *ad-hoc*
Granularidade dos dados	Os dados são armazenados de forma detalhada	Armazenamento detalhado e resumido

2.3 Processos de Extração, Transformação e Carga de Dados

Embora se tenha atualmente no mercado diversas ferramentas que auxiliam no processo de ETL, essa ainda é uma das fases mais desafiadoras e complexas do desenvolvi-

mento de um DW, uma vez que é a responsável por selecionar, lapidar e disponibilizar os dados que serão consultados pelos usuários finais.

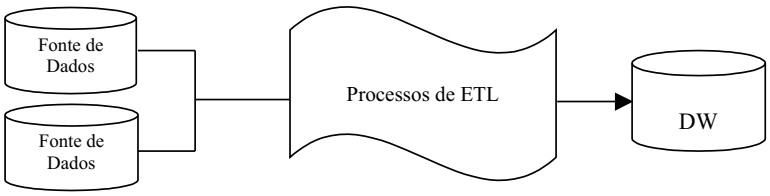

Figura 2.6. Processos de ETL dentro do ciclo de vida dos dados de um DW. Fonte adaptada [C+01]

Essa etapa se dá basicamente em três passos: extração dos dados (**E**xtraction), limpeza ou transformação (**T**ransformation) e, finalmente, a carga (**L**oad). A Figura 2.6 ilustra onde se localizam os processos de ETL dentro do ciclo de vida dos dados de um DW.

2.3.1 Extração

A tarefa de extração de dados tem por objetivos principais: definir o escopo do projeto de DW, identificar e analisar as fontes de dados e especificar os programas que farão a extração [PER00].

No escopo do projeto, são identificados os usuários que estarão envolvidos com o DW a ser implantado e levantados os requisitos a serem suportados.

Embora sem o conhecimento a respeito dos requisitos não há como saber quais dados deverão ser extraídos, é importante frisar que aguardar o levantamento de todas as necessidade para iniciar o desenvolvimento do DW pode tornar o projeto inviável e frustrar as expectativas dos usuários [7300].

Os dados contidos em um DW são proveniente de diversas origens. Assim, após a definição do escopo do projeto, a segunda etapa é a identificação das fontes de dados, as quais podem ser classificadas em dois grandes grupos: internas e externas.

As fontes internas são os sistemas chamados transacionais, os quais têm por objetivo o controle das rotinas operacionais da organização, tais como: almoxarifado, faturamento, vendas, etc. Esta, sem dúvida alguma, é a principal fonte para um DW, pois trata-se dos dados a respeito da própria empresa [J+98].

O outro tipo de fonte de dados, a externa, refere-se a dados que não fazem parte dos sistemas da organização e que muitas vezes são comprados de outras empresas que mantém bases de dados comerciais. Em alguns casos, os mesmos estão disponíveis na Internet e em outros, encontram-se de forma não computadorizada, como em jornais, boletins informativos, dentre outros.

Como se pode observar, as origens de dados de um DW são as mais diversas possíveis, podendo, portanto, ter características variadas. É dessa heterogeneidade que surgem as grandes dificuldades dessa fase. A seguir são citados alguns dos elementos dificultadores da extração de dados [CIE02]:

- algumas vezes é necessário selecionar vários campos do ambiente operacional para compor um único atributo no DW;
- dificilmente existem modelos de dados e documentação dos sistemas transacionais. Isso significa que os desenvolvedores terão que realizar trabalhos de engenharia reversa para compreender as fontes dos dados;
- os dados não se encontram padronizados. Por exemplo: em um sistema transacional o campo sexo é identificado pelas letras M (masculino) e F (feminino). Em outro, esse mesmo campo utiliza as letras H (masculino) e M (feminino);
- as bases de dados transacionais são desenvolvidas em plataformas e tecnologias distintas, tais como Oracle, SQL Server, DB2, sistemas de arquivos, etc.

Uma vez identificados, os dados devem ser coletados de suas fontes e armazenados em uma área intermediária, por meio de rotinas de extração. Existem diversas técnicas para realização desse processo de captura dos dados das fontes, porém todas elas são classificadas em estática ou dinâmica, conforme descrito a seguir [PER00].

Técnicas de Captura Estática de Dados

Esta técnica baseia-se na captura de todos os dados de interesse do DW, independentemente de terem sido modificados ou não durante o período compreendido entre uma extração e outra.

Assim, não existe a necessidade de rotinas que fiquem monitorando os sistemas transacionais em busca de dados que foram alterados, incluídos ou excluídos. No lugar disso, existe um processo automatizado para extração dos mesmos, sendo a periodicidade com que essa captura é realizada dependente do nível de granularidade do DW.

Uma característica marcante da técnica estática é a aquisição apenas da versão corrente dos dados acessados. Em outras palavras, se um registro sofreu várias alterações desde sua última captura e os sistemas transacionais não tiverem sido projetados para armazenar um histórico de tais modificações, na próxima execução das rotinas de extração, somente a última versão do registro em questão será migrada para o ambiente DW.

A grande vantagem dessa técnica é o fato de poder ser executada fora dos horários mais críticos de requisição aos bancos de dados, não interferindo na performance dos sistemas transacionais.

Técnicas de Captura Dinâmica de Dados

A captura dinâmica, também conhecida como incremental, implementa uma replicação dos dados modificados entre uma extração e outra para posterior migração para o DW.

Essa técnica utiliza recursos disponíveis nos SGBD, tais como *triggers* e arquivos de *log*, para realização da aquisição das modificações. Com isso, diferentemente do que ocorre na captura estática, as versões intermediárias dos dados podem ser adquiridas.

A desvantagem dessa abordagem é o sobrecarga dos sistemas transacionais causados pela execução das rotinas de captura das modificações dos dados, mesmo durante os horários mais críticos de trabalho.

2.3.2 Transformação e Limpeza

Segundo [FRE02], a fase de transformação e limpeza dos dados é útil para corrigir algumas imperfeições originárias ainda nas bases de dados transacionais, com o intuito de oferecer aos usuários finais do DW informações coerentes e com qualidade. Alguns problemas encontrados facilmente nessas bases, que dificultam essa fase são:

- diferenças de unidades: em um sistema transacional a unidade de um material pode ser expressa em quilos e em outro pode estar representada em gramas. Assim, é necessário converter campos iguais para as mesmas unidades de medida;
- diferenças de precisão: alguns valores podem ser representados com duas casas decimais em uma base de dados e com quatro casas em outra. É função do administrador do DW padronizar a precisão dos campos;
- diferenças de códigos ou expressões: em alguns casos, pode-se encontrar os mesmos valores representando informações distintas. Ex: o valor "SG"corresponde à cidade São Gabriel em uma base e em outra trata-se da cidade Silva e Gomes;

- diferenças de granularidade: é o caso de um campo que totalize a quantidade de vendas da empresa em uma semana e outro armazene o total de vendas em um mês. Esses campos devem ser convertidos para uma granularidade única para que possam ser comparados;

- diferenças de abstração: o campo endereço pode armazenar o logradouro, número da residência e bairro em uma tabela, mas em outra pode existir um campo para cada um desses dados.

Após as transformações aplicadas sobre os dados para assegurar a qualidade e integração dos mesmos, estes podem finalmente ser migrados para a base de dados do DW.

2.3.3 Carga

A última etapa da ETL é a carga dos dados para o novo ambiente. Vale ressaltar que, após uma carga inicial, são necessárias rotinas de atualizações, para propagar para o DW as modificações ocorridas nos sistemas fontes.

Essa manutenção dos dados também possui uma enorme complexidade, devendo-se levar em consideração os seguintes aspectos [CIE02]:

- no momento da atualização, é necessário verificar a integridade dos dados, ou seja, se as chaves estrangeiras possuem valores correspondes nas tabelas onde estão as respectivas chaves primárias e

- se a atualização será incremental (somente são propagados os dados modificados nos sistemas fontes) ou por cima dos dados (todos os dados são novamente propagados). Geralmente, a carga incremental é feita nas tabelas de fatos e a carga por cima dos dados é feita nas tabelas de dimensões.

2.3.4 Ferramentas ETL

De acordo com [PER00], ferramentas ETL podem ser definidas como um conjunto de ferramentas com o propósito de extração, transformação e carga dos dados para o ambiente DW.

Atualmente, as ferramentas ETL possuem um front-end amigável ao usuário, fato que concede às mesmas uma curva de aprendizado razoável. Além disso, apresentam funcionalidades cliente-servidor e acesso à WEB [PER00].

Em seu artigo, [CIE02] apresenta o Data Stage da Ardent (adquirido atualmente pela Informix), o ETI da IBM, Sagent da Sagent, Informática Power Conect da Informática e o Data Transformation Service da Microsoft como as principais ferramentas ETL existentes

no mercado. Além dessas, pode-se ainda citar como exemplos de softwares comerciais para esse propósito o Oracle8i Warehouse Builder [C+01] e o Warehouse Workbench [SOL03].

Embora tais ferramentas tragam benefícios bastante recompensadores, como o aumento da produtividade, essas exigem investimentos altíssimos, tanto no que diz respeito à capacitação profissional dos desenvolvedores envolvidos no projeto, quanto na sua própria aquisição. Além disso, é importante frisar que um pacote de um software com a finalidade de extração, transformação e carga dos dados não é uma solução completa. Em muitos casos, é necessária a implementação de rotinas para atender situações específicas.

2.4 Integração de Bases de Dados

Assim como a ETL, outro grande desafio do projeto de um DW é a integração dos esquemas que servirão de fontes de dados para esse ambiente. De acordo com Batini, Lenzerine e Navathe (1986), a maioria dos pesquisadores sugere que essa integração seja feita como parte da modelagem conceitual da aplicação, a qual tem por objetivo produzir uma visão abstrata dos dados.

Os principais problemas encontrados durante esse processo provêem, basicamente, das diversidades semântica e estrutural entre os esquemas envolvidos. As causas para essas diversidades, classificadas em [BLN86], são:

- perspectivas diferentes: na maioria das situações, esquemas distintos são concebidos por seus próprios grupos de projetistas e usuários, os quais adotam seus pontos de vista para modelar o mesmo objeto do mundo real;
- equivalência entre construtores do modelo: como existem várias combinações para representar o mesmo domínio de problema, uma mesma situação pode ser modelada com uma variedade de possibilidades. Por exemplo, em um esquema, um objeto do mundo real é modelado como um atributo e, em outro esquema, esse mesmo objeto é modelado como uma entidade e
- incompatibilidade de especificações de projeto: escolhas incorretas a respeito de tipos de dados, cardinalidade, restrições, etc., durante a criação dos esquemas, podem refletir de forma negativa no processo de integração. Por exemplo, supondo que o projetista de um esquema qualquer modelasse erroneamente a cardinalidade entre duas entidades como sendo um-para-um ao invés de um-para-muitos, isso poderia passar despercebido e o esquema integrado herdar o mesmo problema.

Embora existam inúmeras metodologias de integração de esquemas, tais como as descritas por Batini e Lenzerini, ElMasri *et al.* e Navathe e Gadgil *apud* [BLN86], pode-se considerar quatro fases básicas envolvidas nesse processo, são elas: pré-integração,

comparação de esquemas, adequação e união/reestruturação. A seguir essas fases são descritas conforme proposto em [BLN86].

A pré-integração tem como propósito o conhecimento inicial dos esquemas envolvidos na integração, bem como o estabelecimento de relacionamentos e correspondências entre os mesmos. O passo inicial dessa fase é a escolha da estratégia de processamento da integração, a qual pode ser classificada em dois grupos: binária e n-ária.

Na estratégia binária, os esquemas são integrados aos pares. Quando a integração ocorre com apenas dois esquemas de cada vez, chama-se de escada. Ao contrário, quando esse processo se dá com mais de um par de esquemas ao mesmo tempo, chama-se de estratégia balanceada. A Figura 2.7 ilustra esses dois tipos de estratégia binária.

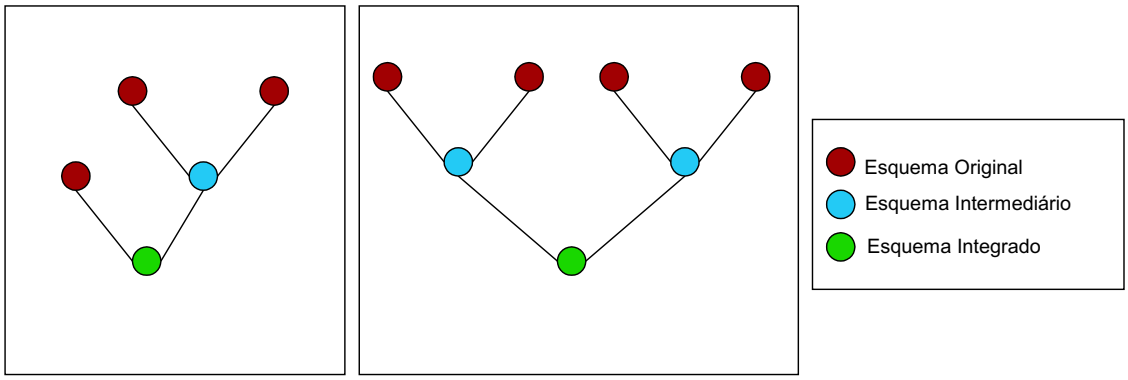

Figura 2.7. Estratégias escada e balanceada, respectivamente, de integração binária de esquemas. Fonte adaptada [BLN86]

A estratégia n-ária estabelece que n esquemas são integrados ao mesmo tempo. Quando todos os esquemas são integrados de uma única vez, chama-se *one shot*. Caso contrário, a estratégia n-ária é conhecida como iterativa. A Figura 2.8 ilustra a diferença entre esses tipos de integração.

A fase posterior à pré-integração é a comparação, a qual tem por finalidade principal levantar os conflitos existentes entre os esquemas. Embora possam existir diversos tipos de conflitos causados por inconsistência semântica entre tais esquemas, como por exemplo incompatibilidade de hardware e software, conflitos de domínio e atributos não representados [VAR01], os mesmos podem ser classificados em dois tipos, são eles: conflitos de nomes e conflitos estruturais.

O primeiro tipo de conflito refere-se a problemas relacionados aos nomes dados aos objetos modelados e podem ser classificados em dois grupos, conforme abaixo:

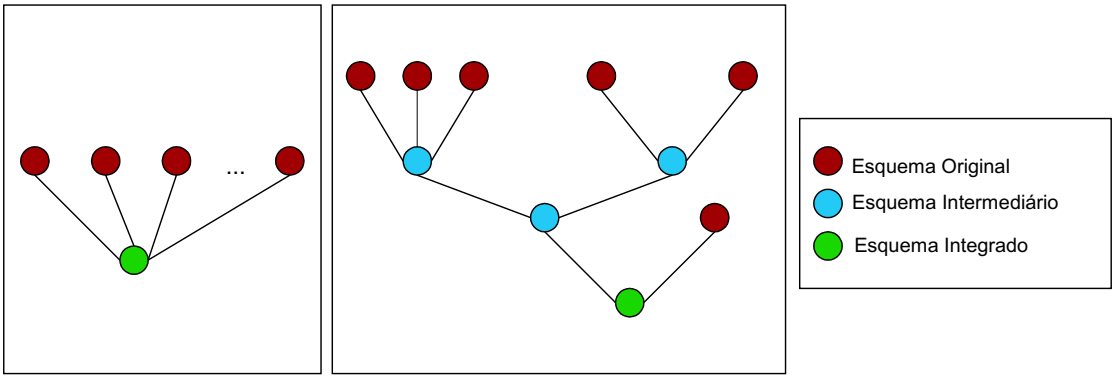

Figura 2.8. Estratégias *one shot* e iterativa, respectivamente, de integração n-ária de esquemas. Fonte adaptada [BLN86]

- homônimos: esse problema ocorre quando conceitos distintos no mundo real recebem nomes iguais em esquemas diferentes, gerando, dessa forma, inconsistência;
- sinônimos: ocorre quando conceitos iguais são modelados com nomes distintos em esquemas diferentes.

Como já foi descrito na seção anterior, em sua maioria, esquemas distintos são projetados por grupos também distintos de pessoas, com seus próprios pontos de vista a respeito do domínio do problema. Como conseqüência disso, surgem os conflitos estruturais, uma vez que a mesma situação pode ser modelada de variadas formas.

A terceira fase, adequação, vem para tornar os esquemas compatíveis para o processo de integração. Para isso, é necessário solucionar os conflitos levantados na fase anterior, tendo, em muitos casos, que transformar os objetos modelados, bem como seus relacionamentos, para tornar isso possível.

Finalmente, na fase de união e reestruturação o esquema integrado intermediário passa por otimizações relacionadas a questões de integridade, minimização (eliminação de redundâncias) e compreensão (torna o esquema integrado o mais compreensível possível).

Com o intuito de facilitar a visualização dos principais problemas encontrados durante a integração de esquemas de bancos de dados, é apresentado a seguir um exemplo dessa atividade, iniciando com a ilustração, na Figura 2.9, dos esquemas a serem integrados.

As entidades Palavra-Chave e Assunto dizem respeito ao mesmo conceito no mundo real. Dessa forma, as mesmas deverão ser unificadas no esquema integrado e receber um nome único. Nesse caso, será adotado o termo Assunto.

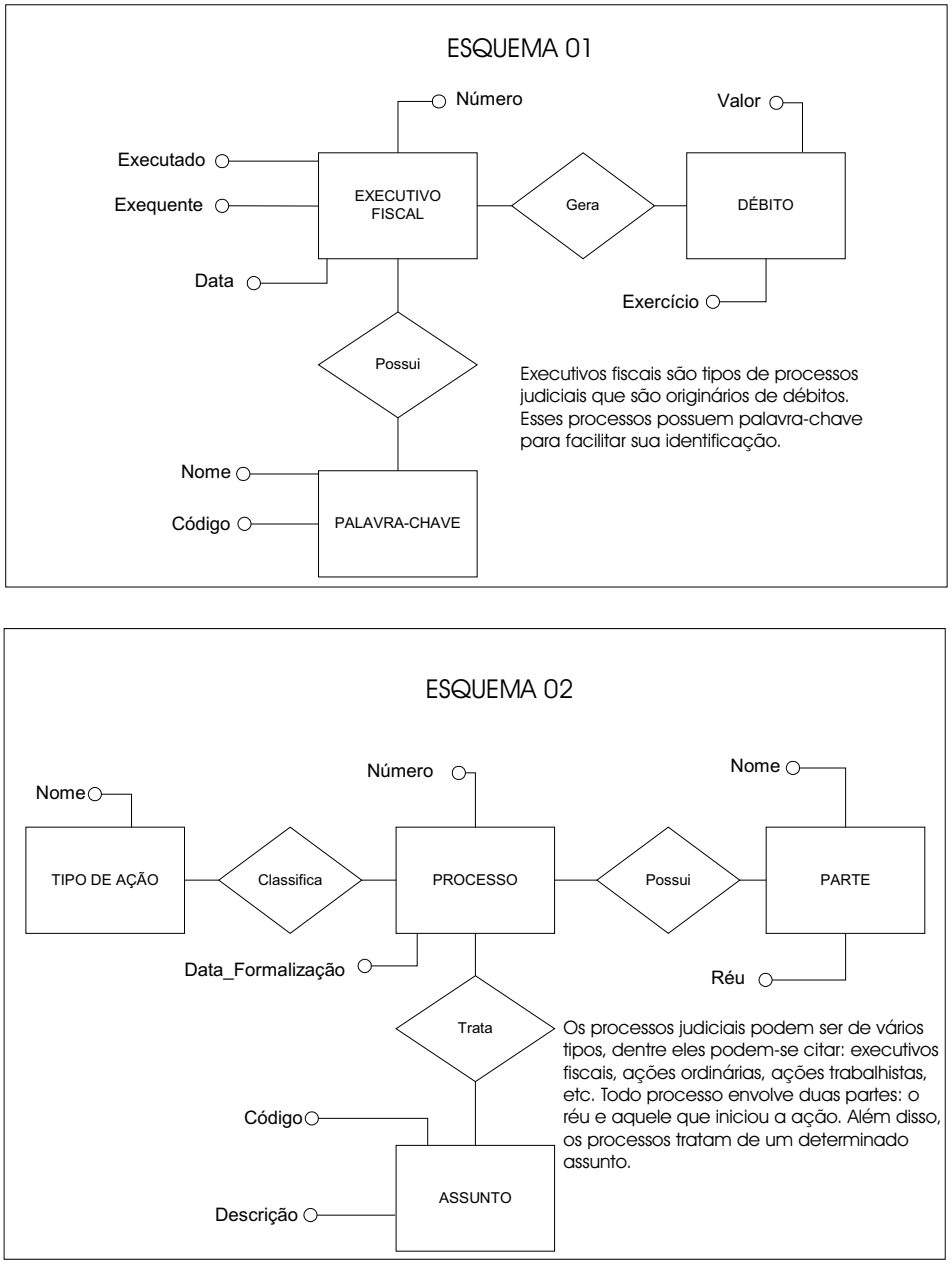

Figura 2.9. Exemplos de requerimentos e seus respectivos esquemas que serão integrados.

Uma outra diferença entre os dois esquemas originais refere-se às partes de um processo ou executivo fiscal. No esquema 01, esse conceito foi modelado como os atributos Exequente e Executado. Enquanto que no esquema 02, para o mesmo conceito foi criado a entidade Parte. No esquema integrado, os atributos acima citados serão convertidos na entidade presente no segundo esquema.

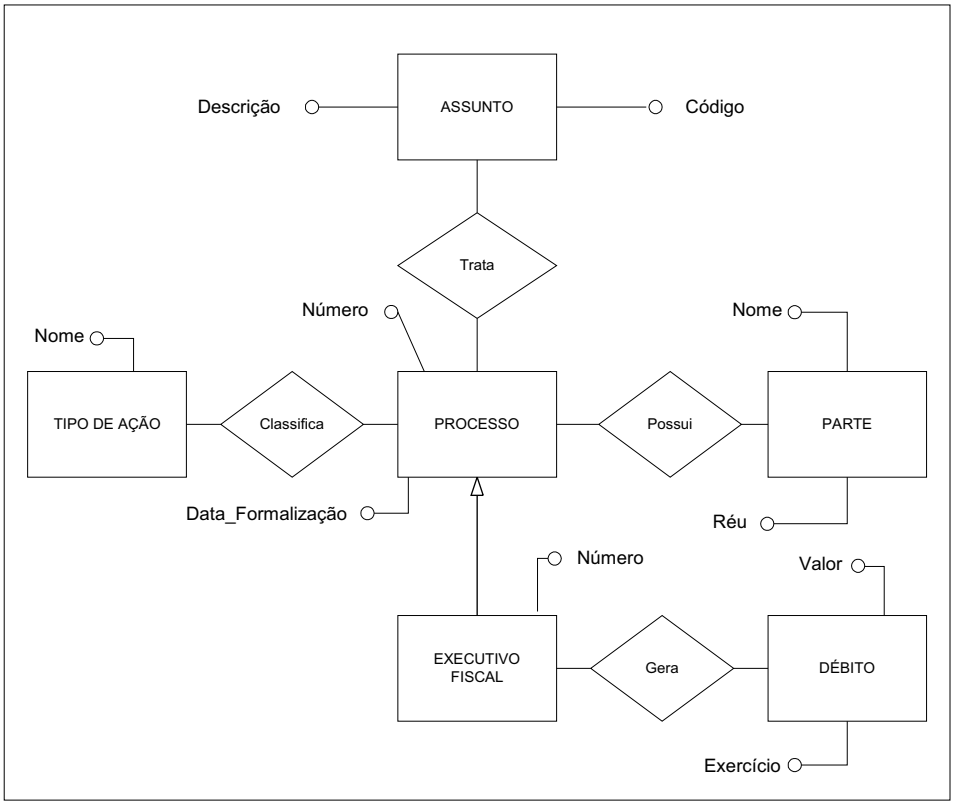

Figura 2.10. Esquema resultante do processo de integração.

Por fim, outra questão que deve ser observada é o fato da entidade Executivo Fiscal ser uma subclasse da entidade Processo. Isto posto, após essa considerações a Figura 2.10 apresenta o esquema integrado originário dos esquemas 01 e 02.

Embora o exemplo acima seja simples quando comparado com uma integração real, ele pode fornecer uma boa amostra dos problemas encontrados durante esse processo, os quais provêem, basicamente, das diversidades semântica e estrutural entre os esquemas envolvidos.

Outro aspecto que deve ser levado em consideração durante o projeto de um DW é a questão da granularidade, a qual refere-se ao nível de detalhe dos dados armazenados [ELE02].

A razão pela qual esta questão é um fator tão importante de projeto, reside no fato de que ela afeta diretamente o volume de dados a serem armazenados e os tipos de consultas solicitadas pelos usuários que poderão ser atendidas. Isso explica-se da seguinte maneira: quanto maior o detalhamento dos dados, ou seja, quanto mais baixa a granularidade, maior o volume dos mesmos e, conseqüentemente, mais espaço em disco deverá ser usado para armazená-los; por outro lado, mais opções de consultas poderão ser realizadas, uma

vez que se tem mais dados para serem trabalhados. Além disso, um grande volume de dados tende a diminuir a performance do sistema [ELE02].

A definição do nível de granularidade não é uma decisão fácil a ser tomada pelos projetistas. Esta depende dos dados disponíveis nas fontes do DW, das necessidades de consultas dos usuários, qual o volume de dados que se pretende armazenar e quais os requisitos de performance a serem observados.

Capítulo 3

Projeto e Implementação de uma Aplicação ETL: Estudo de Caso na PMM

Este capítulo apresenta o projeto e implementação de uma aplicação para extração, transformação e carga de dados, bem como a modelagem dimensional de um DW. Com isso, objetivamos mostrar a viabilidade da realização dos procedimentos de ETL por parte da equipe de desenvolvimento, utilizando para isso ferramentas de desenvolvimento de software existentes na organização.

Para comprovação da hipótese apresentada na seção 1.2 do capítulo introdutório, realizou-se um estudo de caso no Departamento de Tributação da Prefeitura Municipal de Manaus, como descrito na seção seguinte.

A Prefeitura Municipal de Manaus (PMM), em particular o Departamento de Tributação, foi adotada como cenário dessa pesquisa por possuir um ambiente com fontes de dados implementadas sob o mesmo paradigma de desenvolvimento, adequando-se, portanto, ao domínio do problema que estamos investigando.

3.1 Negócio da Organização: Arrecadação Tributária do Município de Manaus

A Constituição Federal Brasileira, em seu título VI, capítulo I, seção I, confere à União, Estados, Distrito Federal e Municípios o direito de instituir e cobrar tributos, os quais podem se apresentar na forma de impostos, taxas (em razão do exercício do poder de polícia ou pela utilização de serviços públicos) e contribuição de melhoria em função de obras públicas [FJ01].

Para que esse direito previsto na Constituição seja efetivamente exercido, existem órgãos cuja missão principal é a arrecadação de recursos provenientes de tais tributos. Em Manaus, segundo a lei municipal nº 1.073 [MAN03], o órgão que concentra os processos de arrecadação tributária é a Secretaria Municipal de Economia e Finanças (SEMEF), por meio de seu Departamento de Tributação (DETRIB).

Cabe ao DETRIB, portanto, a arrecadação dos seguintes tributos de natureza municipal (MPOG *et al apud* [S$^+$00]):

- IPTU: imposto predial e territorial urbano, o qual incide sobre a propriedade, o domínio útil ou a posse do imóvel situado na zona urbana do Município;
- ISS: imposto sobre serviços de qualquer natureza, incidindo sobre a prestação por empresa ou profissional autônomo de serviços constantes na Lista de Serviços apresentada em [FJ01];
- ITBI: imposto sobre transferência de bens imóveis, o qual incide sobre a transmissão inter vivos de bens imóveis e
- Taxas decorrentes da utilização efetiva ou potencial de serviços públicos, tal como o alvará de funcionamento.

O pagamento desses tributos é feito pelo contribuinte na rede bancária autorizada, através dos Documentos de Arrecadação Municipal, conhecidos como DAM. Os bancos, que possuem convênio com a PMM, enviam para a SEMEF os pagamentos realizados no dia anterior, com o intuito de que os débitos sejam baixados [S$^+$00].

Entretanto, o processo de arrecadação tributária não consiste simplesmente em receber os pagamentos dos contribuintes e torcer para que o valor arrecado seja suficiente para cobrir as despesas. Como em qualquer empresa privada, os órgãos públicos também necessitam de planejamento. Nesse caso, o passo inicial desse processo, conforme estabelece a Lei de Responsabilidade Fiscal (MPOG *et al* apud Silva, 2003), é a apresentação da previsão de arrecadação, constante no orçamento. Assim, com base nos valores orçados, os gestores podem avaliar se a capacidade contributiva prevista do município está sendo atingida e caso não esteja, podem traçar estratégias para que a meta seja alcançada.

De acordo com Silva (2003), as prefeituras devem possuir mecanismos para incentivar ou exigir a cobrança de tributos, uma vez que os contribuintes, sejam eles pessoas físicas ou jurídicas, nem sempre se sentem motivados a destinar parte de sua renda para essa finalidade.

Uma das formas empregadas pelas prefeituras para exigir o pagamento de tributos é a inscrição do débito em dívida ativa. Esse é um procedimento que antecede a cobrança executiva que envolve a esfera judicial.

Além da evasão fiscal gerada pelo não pagamento das obrigações legais por parte de alguns contribuintes, outro fator que afeta diretamente a arrecadação municipal são os pedidos de isenção de tributos, previstos na lei nº 1.697 de 20 de dezembro de 1983 [FJ01], os quais são formalizados como processos administrativos. Quando um processo desse gênero é deferido, significa que deixou de entrar nos cofres municipais uma certa

quantia. A Figura 3.1 ilustra o fluxo das atividades envolvidas na arrecadação tributária do município de Manaus.

Pode-se verificar, portanto, que a arrecadação municipal é um processo complexo e que exige dos administradores públicos habilidades para traçar estratégias, avaliar os resultados e por em prática planos de contingência quando o esperado não foi alcançado.

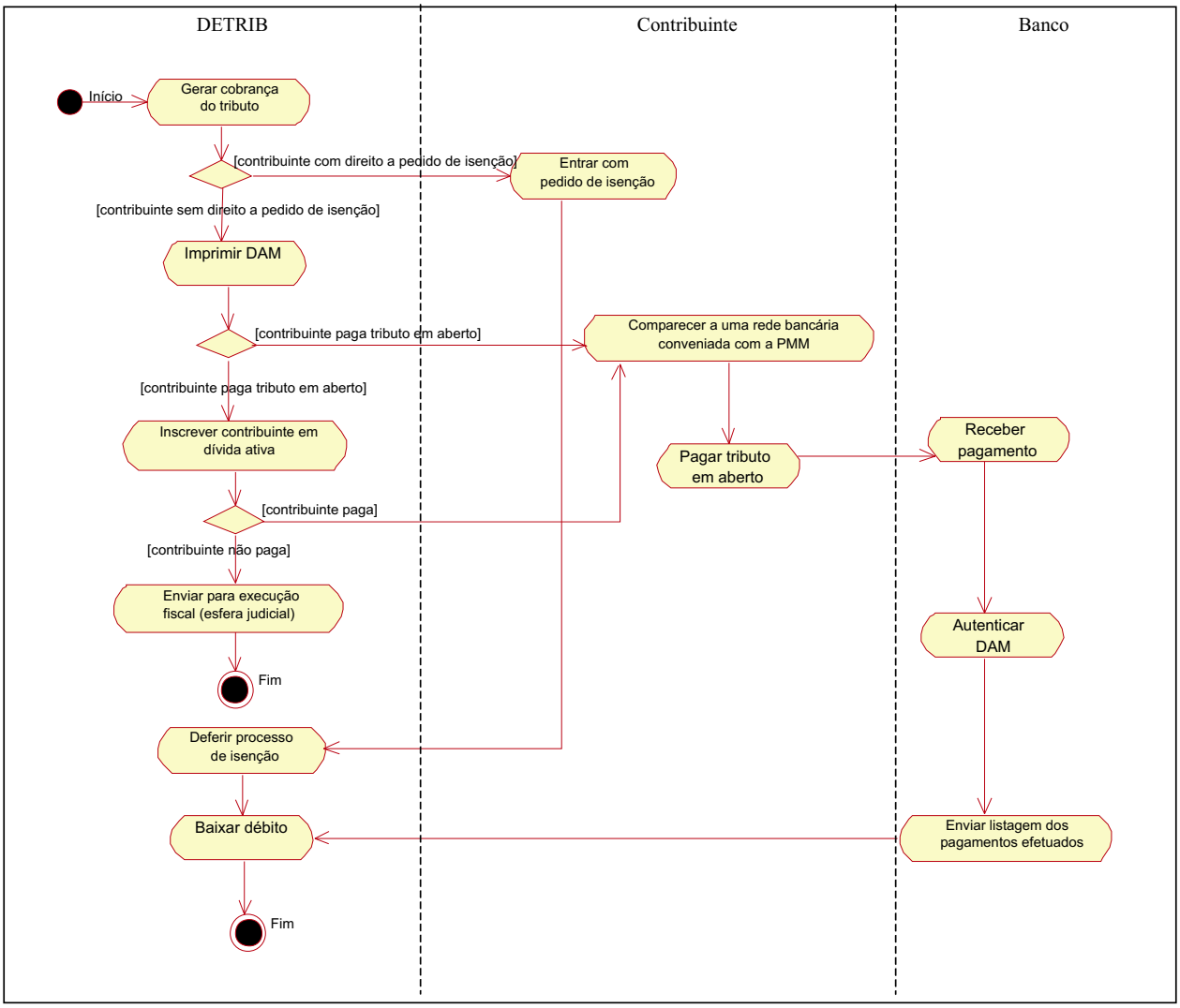

Figura 3.1. Fluxo de atividades da arrecadação tribuária do município de Manaus.

Para tanto, os gestores devem dispor de ferramentas que os auxiliem em suas atividades, principalmente, no que diz respeito ao fornecimento de informações rápidas, precisas e consolidadas. Logo, a presença de Sistemas de Apoio a Decisão, como um DW,

em órgãos públicos municipais pode ser vista como um facilitador na execução de suas atividades.

3.2 Integração das Fontes de Dados

Embora a integração das fontes de dados do DW não seja o grande foco desse trabalho, a descrição desse processo é de extrema importância para a compreensão do estudo de caso aqui empregado.

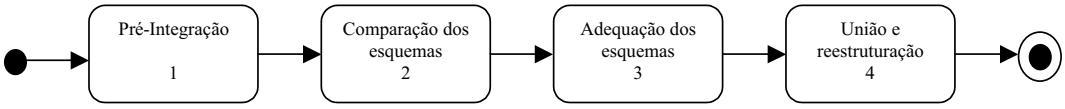

Figura 3.2. Fluxo das atividades realizadas para a integração das fontes de dados.

As atividades realizadas até a efetiva integração dos esquemas, ilustradas na Figura 3.2, foram quatro e envolveram os procedimentos descritos em (Batini et al, 1986). Embora tenha sido apresentada há quase 20 anos, essa metodologia foi escolhida por servir de referência em diversos trabalhos atuais que abordam esse assunto [DOA00] [FM02] [MP01].

Durante a primeira atividade, a pré-integração, foram definidas as bases de dados usados como fontes para o DW, as quais são apresentadas a seguir.

De acordo com Silva (2003), o Sistema Tributário Integrado (STI) e o Sistema Administrativo Integrado (SAI) são dois sistemas desenvolvidos pela SEMEF, oriundos de uma abertura de crédito firmada entre a PMM e o Banco Nacional de Desenvolvimento (BNDES), com o objetivo de promover e acompanhar o crescimento econômico da cidade.

O STI oferece suporte ao processo de arrecadação, de forma que os tributos (taxas e impostos) de responsabilidade da SEMEF são controlados e emitidos por este sistema. Enquanto que o SAI atua junto aos Departamentos Orçamentário, Contábil e Financeiro dessa mesma secretaria, auxiliando nas atividades relativas à elaboração e execução orçamentária [S$^+$00].

Para gerenciar os procedimentos referentes à inscrição de contribuintes em dívida ativa e as atividades posteriores à mesma, executadas no Setor de Dívida Ativa e na Procuradoria Geral do Município, foi desenvolvido, de forma terceirizada, o Sistema de Controle dos Débitos em Execução Fiscal (SISCODE).

Quanto ao gerenciamento dos pedidos de isenção de tributos e outros processos administrativos, foi implantado em 2001 no DETRIB o Sistema Integrado de Protocolo (SPI), atualmente em funcionamento em toda SEMEF e mais sete outras secretarias municipais.

Os sistemas apresentados acima (STI, SAI, SISCODE e SPI) servem de *front-end* para povoar cada uma das quatro bases de dados que armazenam as informações necessárias para construção do DW proposto nesse trabalho, cujo negócio é a avaliação da arrecadação tributária de responsabilidade do DETRIB.

Para tal avaliação são considerados os valores arrecadados por receita (extraídos da base de dados do STI), os valores orçados (extraídos da base de dados do SAI), inscritos em dívida ativa (extraídos da base de dados do SISCODE) e o montante que se deixou de arrecadar com pedidos de isenção (extraído da base de dados do SPI).

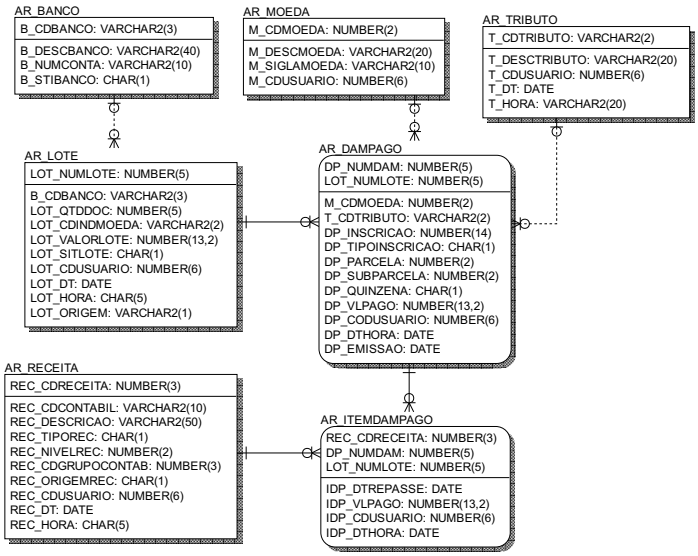

Figura 3.3. Visão parcial da base de dados do STI.

A Figura 3.3, Figura 3.4, Figura 3.5 e Figura 3.6 ilustram uma visão parcial dos esquemas das bases de dados em questão, obtidos por meio de processos de engenharia reversa, suficiente para o escopo do problema. Vale ressaltar que, por motivos de segurança, alguns atributos e nomes de tabelas foram modificados para serem ilustrados.

A finalização da primeira fase de integração deu-se com a definição da estratégia de integração utilizada na pesquisa. Por proporcionar uma visão mais abrangente do domínio

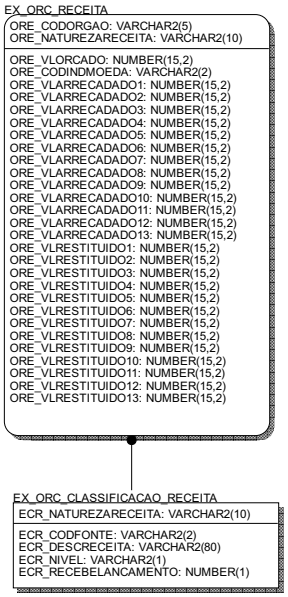

Figura 3.4. Visão parcial da base de dados do SAI.

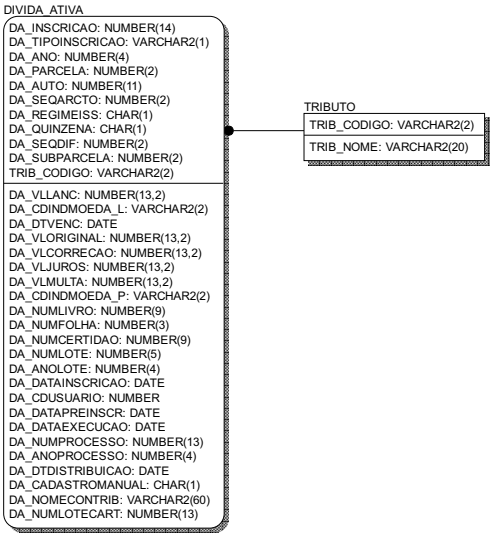

Figura 3.5. Visão parcial da base de dados do SISCODE.

do problema, uma vez que permite que todos os esquemas envolvidos no processo possam ser integrados ao mesmo tempo, foi adotada a estratégia *one shot*, descrita na seção 2.4

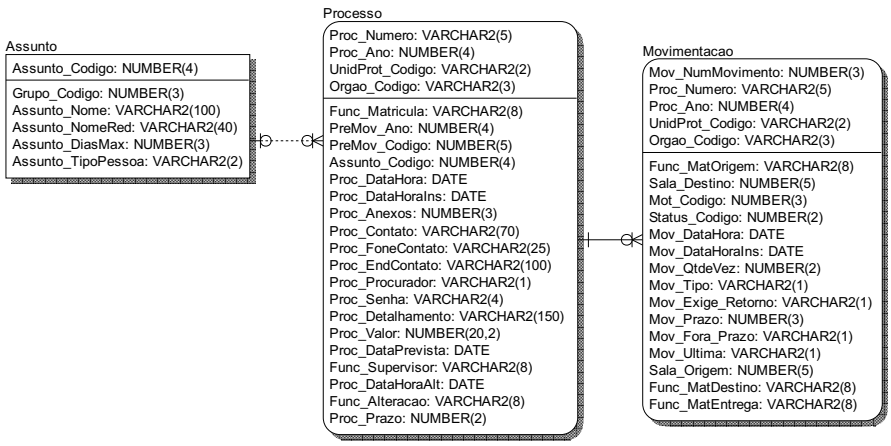

Figura 3.6. Visão parcial da base de dados do SPI.

do referencial teórico.

Durante a comparação e adequação dos esquemas, foram identificados e reparados os conflitos estruturais existentes entre as fontes de dados, os quais são descritos a seguir.

- **Conflito:** as tabelas Ar_Receita (Figura 3.3) e Ex_Orc_Classificação_Receita (Figura 3.4) representam o mesmo conceito do mundo real (classificação das receitas). **Adequação:** As tabelas deverão ser unificadas e gerada uma tabela única chamada Receita;

- **Conflito:** assim como as tabelas descritas no item anterior, Ar_Tributo (Figura 3.3) e Tributo (Figura 3.5) também são sinônimas, ou seja, representam o mesmo conceito. **Adequação:** Devem ser unificadas e criada a tabela Tributo;

- **Conflito:** no esquema do SPI (Figura 3.6), a tabela Assunto armazena os assuntos existentes na PMM para se formalizar processos. Em termos de arrecadação, os assuntos de interesse são aqueles referentes à isenção de tributos. Assim, essa tabela também deve ser integrada à Tributo. **Adequação:** relacionar os assuntos de isenção a seus tributos correspondentes. Por exemplo, "Isenção de ISS"no esquema integrado refere-se ao tributo ISS; "Isenção de IPTU"referindo-se ao tributo IPTU e assim por diante;

- **Conflito:** para o esquema integrado, somente são de interesse os processos de isenção, cujo status é arquivado deferido. Essa informação da situação dos processos encontra-se na tabela Movimentação (Figura 3.6). Daí sua utilidade para a integração. **Adequação:** no momento da integração, selecionar somente os processos de isenção de tributos, cujo último status é arquivado deferido.

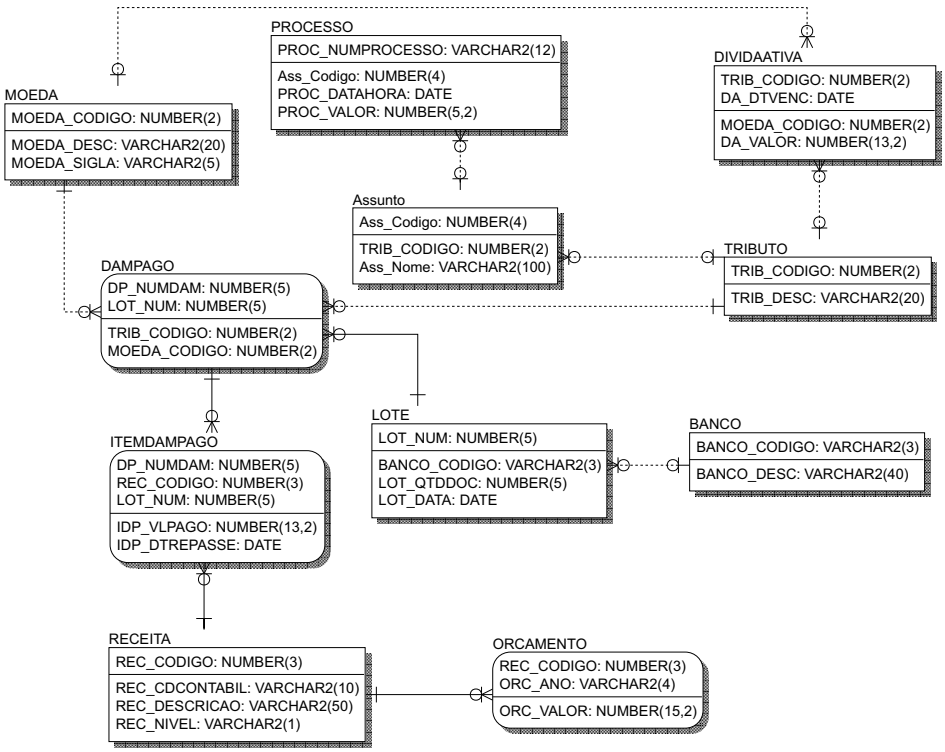

Figura 3.7. Esquema integrado.

Uma vez identificados os conflitos e propostas as adequações necessárias, os esquemas puderam, finalmente, ser integrados (união). Nessa atividade, foram eliminados atributos irrelevantes para o DW. A Figura 3.7 apresenta o esquema resultante do processo de integração.

3.3 Criação da DSA e do DW

As tabelas do esquema integrado, apresentadas na Figura 3.7, foram utilizadas para a criação da DSA. O SGBD escolhido para a implementação dessa base de dados foi o Oracle 8i, por ser o padrão adotado pela PMM [S$^+$00]

Segundo Freitas (2001), além da extração dos dados das fontes e transformação dos mesmos, o projeto e desenvolvimento da DSA também envolvem a definição e implementação dos metadados. Porém, como o foco do trabalho concentra-se na redução de custo durante o processo de ETL, os mesmos não foram considerados.

Para a modelagem do DW, foi empregado um modelo dimensional muito comum: o modelo estrela. Durante esse processo, alguns atributos da DSA passaram por agregações para se adequarem à granularidade do projeto. A Figura 3.8 a seguir mostra a estrutura concebida, a qual, assim como a DSA, também foi implementada no Oracle 8i.

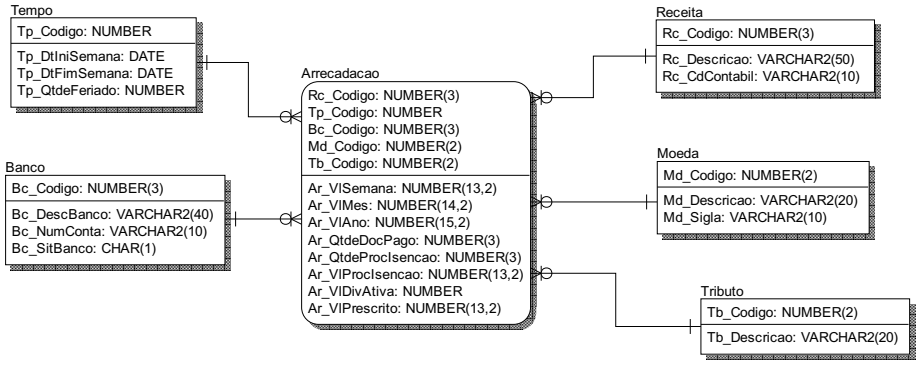

Figura 3.8. Modelo Estrela do DW

No tocante ao transporte dos dados para o DW, Freitas (2001) sugere o desenvolvimento de dois processos: um para a carga inicial dos dados e outro para as atualizações periódicas dessa base.

Para a carga inicial dos dados, foram implementadas *stored procedures*, que são procedimentos armazenados na base de dados e executados no servidor [dBdDS03]. A decisão por essa forma de implementação fundamentou-se em duas grandes vantagens das *stored procedures* sobre as consultas comuns, que são apresentadas em [CAR02]:

- redução de tráfego na rede: a maior vantagem das stored procedures é o fato de serem executadas na máquina servidora de banco de dados. Isso reduz a transferência de dados entre o servidor e o cliente pela rede e
- ganho de performance: cada vez que um comando SQL é enviado para o servidor, ele deve passar por uma análise gramatical e ser submetido ao otimizador para formulação do plano de execução. Uma *stored procedure*, por outro lado, é analisada, otimizada e armazenada em uma forma executável no momento em que é adicionada ao banco de dados. Assim, uma vez chamada, ela é executada mais rapidamente que uma consulta equivalente.

Quanto ao desenvolvimento do processo das atualizações periódicas, também conhecido como *refresh*, implementou-se uma aplicação, a qual é detalhada na próxima seção.

É interessante ressaltar que as transformações dos dados necessárias antes de serem efetivamente carregados para o novo ambiente ocorreram em duas etapas, válidas tanto

para a carga inicial quanto para as atualizações periódicas:

- primeira Etapa: os dados extraídos das fontes foram manipulados para respeitar as novas restrições impostas pelo esquema integrado. Essas transformações deram-se durante a migração dos dados para a DSA e
- segunda Etapa: os dados oriundos da DSA passaram por manipulações, com o intuito de agregá-los para que a granularidade do DW fosse respeitada.

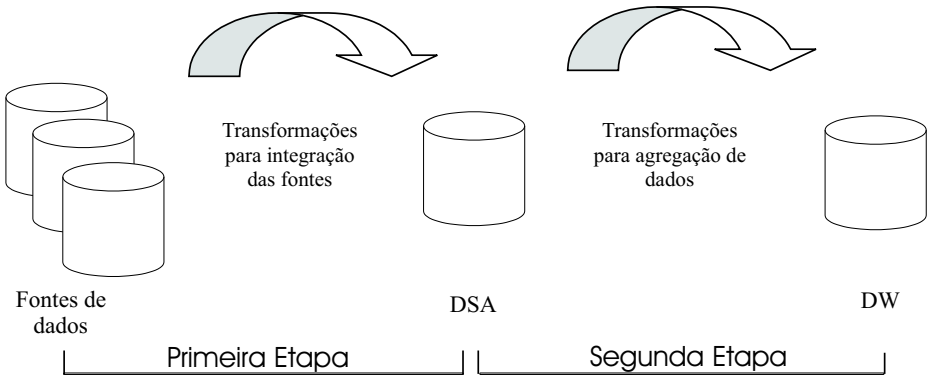

Figura 3.9. Etapas de transformação dos dados.

A Figura 3.9 ilustra essas duas etapas de transformação.

3.4 Implementação da Aplicação

Para povoar a DSA e o DW, desenvolveu-se uma aplicação em ambiente Windows. Nessa implementação, foram utilizados os seguintes produtos:

- Delphi 5.0: é um software da Borland que foi usado para o desenvolvimento do front-end da aplicação [CAN00];
- SQL Navigator 3.0 d2: este produto da Quest Software foi empregado como uma interface mais amigável para o Oracle, com o intuito de auxiliar na manipulação das tabelas da DSA e do DW, além de permitir a criação e manutenção das *stored procedures* [Sof03].
- Oracle*8i*: é um Sistema Gerenciador de Banco de Dados Objeto-Relacional e seu uso se deu para a criação das bases de dados da DSA e do DW [C$^+$01].

A escolha por estes produtos fundamentou-se em dois critérios essenciais para esse trabalho. O primeiro deles, por se tratarem de produtos já empregados na PMM, não sendo necessária aquisição de novas ferramentas. O segundo, que pode ser visto como uma conseqüência do aspecto anterior, refere-se ao fato dos mesmos serem produtos dominados pela equipe de analistas e programadores, não exigindo, portanto, investimentos em treinamentos.

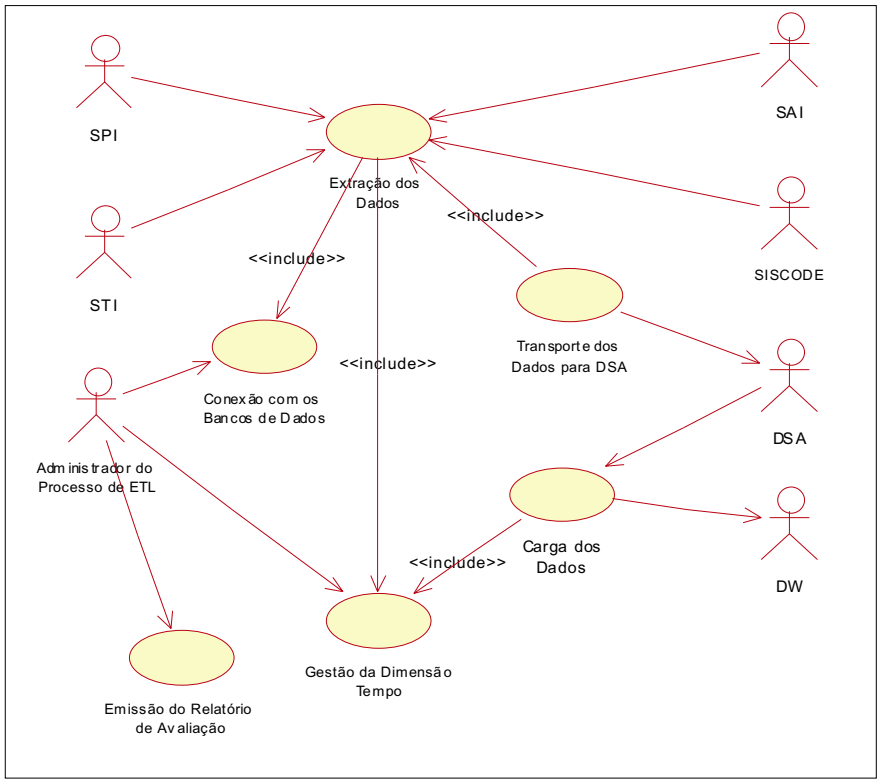

Figura 3.10. Diagrama de contexto da aplicação.

As funções da aplicação desenvolvida são ilustradas na Figura 3.10, por meio de um diagrama de contexto, o qual apresenta os principais casos de uso levantados.

Um caso de uso, representado na figura por elipses, é uma descrição de um conjunto de seqüências de ações que um sistema executa para produzir um resultado de valor observável por um ator [B$^+$00].

Um ator, por sua vez, representa um conjunto coerente de papéis que os usuários dos casos de uso desempenham quando interagem com os mesmos. Freqüentemente, um ator representa um papel que um ser humano, um dispositivo de hardware ou outro sistema desempenha com o sistema [B$^+$00]. Na figura anterior, os atores são representados por bonecos com seus respectivos nomes logo abaixo.

Os atores e os casos de uso da aplicação apresentados na Figura 3.10 são descritos a seguir:

- Administrador do Processo de ETL (ator): é um profissional responsável por realizar o processo de extração, transformação e carga dos dados para o DW, usando, para esse procedimento, a aplicação desenvolvida. Para desempenhar essa tarefa, deverá conhecer as senhas de acesso às bases que servem de fontes de dados.
- STI (ator): Base de dados do Sistema Tributário Integrado.
- SAI (ator): Base de dados do Sistema Administrativo Integrado.
- SISCODE (ator): Base de dados do Sistema de Controle dos Débitos em Execução Fiscal.
- SPI (ator): Base de dados do Sistema de Protocolo Integrado.
- DSA (ator): Base de dados da Data Staging Area.
- DW (ator): Base de dados do Data Warehouse.
- Conexão com os bancos de dados (caso de uso): Essa função permite que o administrador do processo de ETL abra uma conexão com as bases de dados do STI, SAI, SISCODE e SPI. Esse procedimento é essencial para se dar início à extração dos dados. O fluxo principal desse caso de uso é o seguinte:
  - O administrador do processo de ETL seleciona a base de dados com a qual deseja se conectar;
  - Informa o usuário para conexão;
  - Informa a senha do usuário;
  - A conexão é aberta e as tabelas, com seus respectivos campos a serem exportados, são apresentados.
- Extração dos dados (caso de uso): A função extração dos dados é utilizada para retirar das bases de dados do STI, SAI, SISCODE e SPI os dados que devem ser migrados para a DSA. O fluxo principal desse caso de uso ocorre como descrito abaixo:
  - Seleciona os dados do STI;
  - Seleciona os dados do SAI;
  - Seleciona os dados do SISCODE;
  - Seleciona os dados do SPI.
- Transporte dos dados para DSA (caso de uso): Essa funcionalidade da aplicação permite povoar a DSA com os dados selecionados do STI, SAI, SISCODE e SPI. O fluxo usado para a realização desse transporte é o seguinte:

  - Integração dos dados extraídos. Esse procedimento aplica-se sobre as tabelas que apresentam os conflitos descritos na seção 3.2 do capítulo 3;
  - Estabelecimento da conexão com a base de dados da DSA;
  - Abertura de transação;
  - Inserção dos dados na DSA;
  - Fechamento da transação.

- Gestão da dimensão tempo (caso de uso): Essa função permite que o administrador do processo de ETL informe os dados necessários para a dimensão tempo existente no DW, cuja granularidade é semanal. O fluxo principal desse caso de uso segue abaixo:
  - Informação da data inicial da semana;
  - Informação da data final da semana;
  - Informação da quantidade de feriados que ocorreram durante a semana.

- Carga dos dados (caso de uso): Essa função é empregada para realizar a carga dos dados no DW. O fluxo desse caso de uso é descrito a seguir:
  - Seleção dos dados da DSA;
  - Realização dos procedimentos de agregação dos dados, a fim de respeitar a granularidade definida para o DW;
  - Estabelecimento de conexão com a base de dados do DW;
  - Abertura de transação;
  - Inserção dos dados selecionados e agregados;
  - Fechamento de transação.

3.4.1 Componentes da Aplicação

Um diagrama de componentes mostra um conjunto de componentes e seus relacionamentos, sendo seu enfoque a modelagem de aspectos físicos de sistemas orientados a objetos [B$^+$00]. A Figura 3.11 apresenta o diagrama de componentes da aplicação.

De acordo com a notação UML [B$^+$00], as bases de dados são representadas por cilindros. Na figura, pode-se observar seis desses componentes: STI, SAI, SISCODE e SPI de onde os dados são extraídos; DSA representando a base de dados da Data Staging Area e DW representando o Data Warehouse.

A folha com a ponta dobrada representa um documento contendo código-fonte ou dados. No caso específico da aplicação, é um relatório gerado pela aplicação para apresentar um resumo dos dados migrados para o DW.

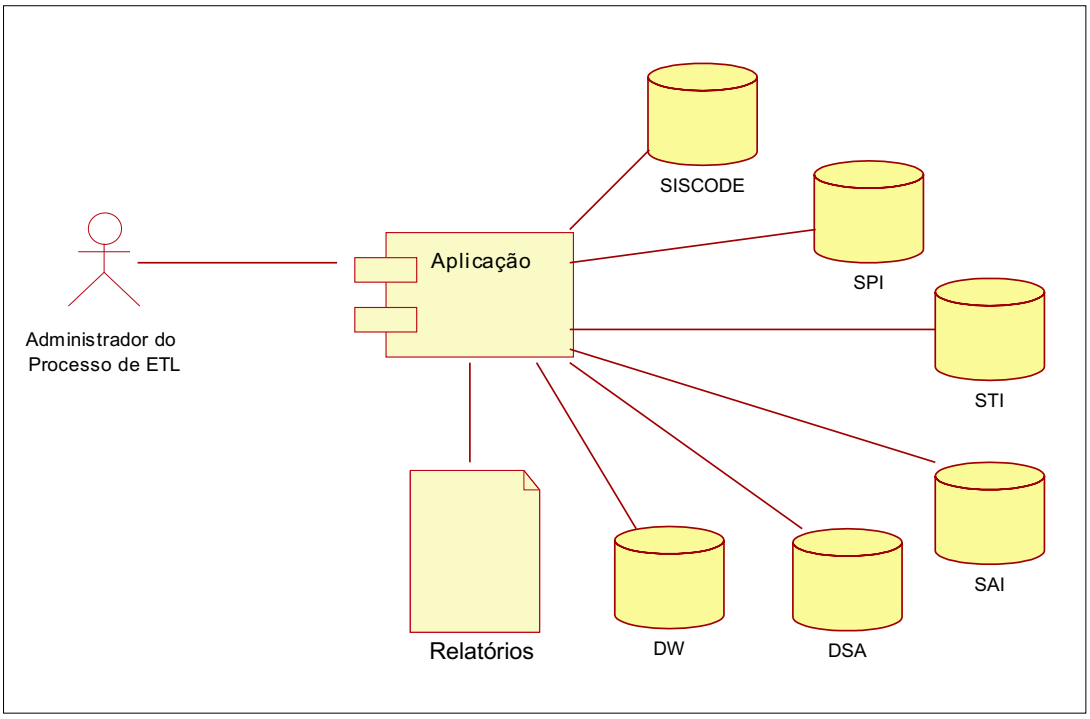

Figura 3.11. Diagrama de componentes da aplicação.

Finalmente, a caixa com dois retângulos sobrepostos representa um componente que pode ser executado. Na Figura 3.11, esse elemento padrão está ilustrando o software, cuja missão é a extração, transformação e carga dos dados a respeito da arrecadação tributária do município de Manaus. As interações entre os componentes são representadas por linhas que os conectam.

3.4.2 Diagrama de Implantação da Aplicação

Um diagrama de implantação mostra a configuração de nós de processamento e os componentes que neles existem. Assim como o diagrama de componentes, o enfoque desse também é a modelagem de aspectos físicos [B$^+$00].

A Figura 3.12 ilustra o diagrama de implantação da aplicação seguindo o padrão UML.

De acordo com a Figura 3.12, os nós de processamento são quatro: Servidor Data Warehouse, Servidor de Banco de Dados 01, Servidor de Banco de Dados 02 e Estação do Usuário.

O Servidor Data Warehouse, onde está localizada a base de dados do DW e da DSA, funciona com sistema operacional Windows 2000 Server e está localizado fisicamente no

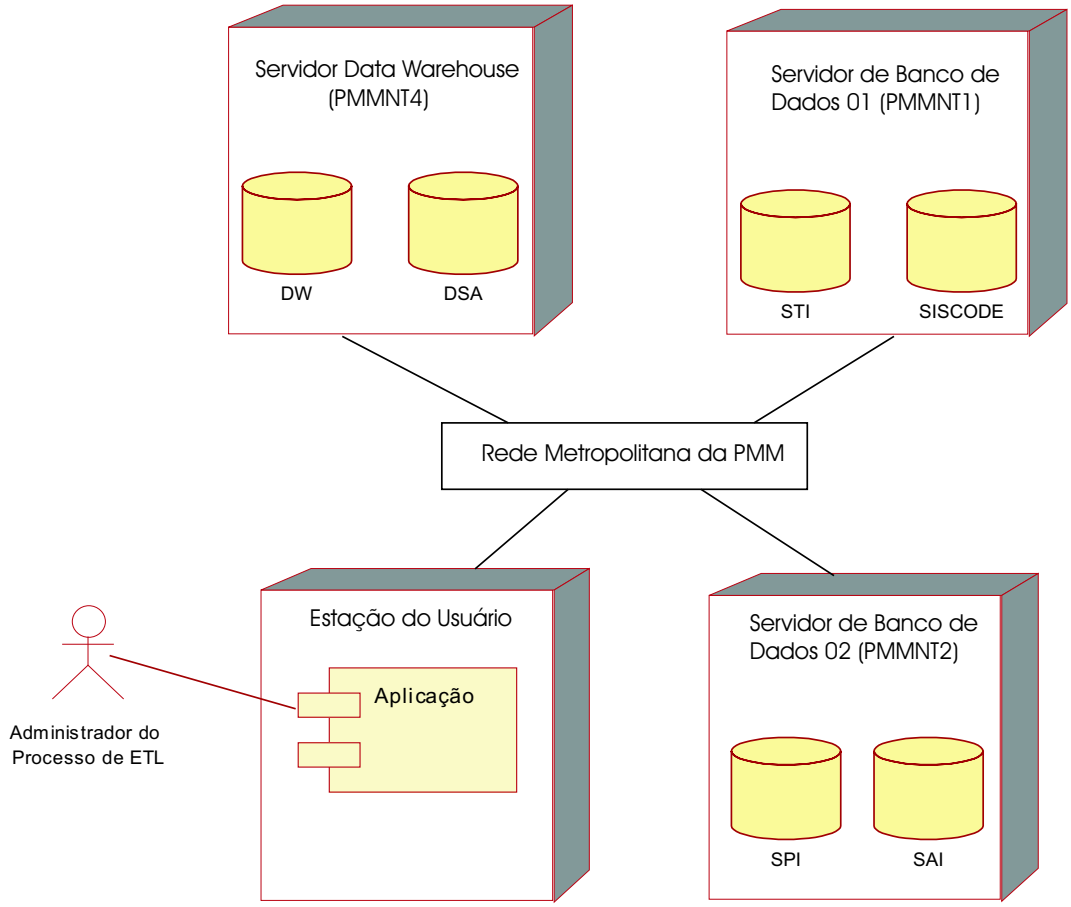

Figura 3.12. Diagrama de implantação da aplicação.

prédio da sede da PMM, assim como o Servidor de Banco de Dados 02, cujo sistema operacional é o Windows NT 4.0.

No Servidor de Banco de Dados 01 estão armazenadas as bases de dados do STI e do SISCODE. Este servidor localiza-se no prédio do DETRIB, situado no centro da cidade de Manaus.

A Estação do Usuário pode ser qualquer máquina com o sistema operacional Windows 98 ou superior, o Oracle Client e o BDE do Delphi 5.0 instalados. Por meio dessa estação, o administrador do processo de ETL inicializa a aplicação para realizar as atualizações periódicas do DW.

A comunicação lógica entre os nós se dá por meio da rede metropolitana da PMM. Fisicamente, a ligação entre o prédio do DETRIB e da sede da Prefeitura é feita por fibra ótica, sendo o protocolo de comunicação o TCP/IP [S+00].

3.4.3 Front-End da Aplicação

O *front-end* da aplicação, implementado no ambiente de programação Delphi 5.0, fundamentou-se nos casos de uso apresentados na seção 3.3 desse capítulo, para o desenvolvimento de uma interface amigável, para plataforma Windows, voltada para auxiliar o administrador do processo de ETL na realização das atualizações periódicas do DW.

A interação do usuário com essa interface dá-se através do mouse e teclado. Para sua implementação utilizou-se componentes padrões do Delphi 5.0, presentes em sua barra de componentes, tais como: maskedit, datetimepicker, stringgrid, bitbtn, edit, dentre outros [CAN00].

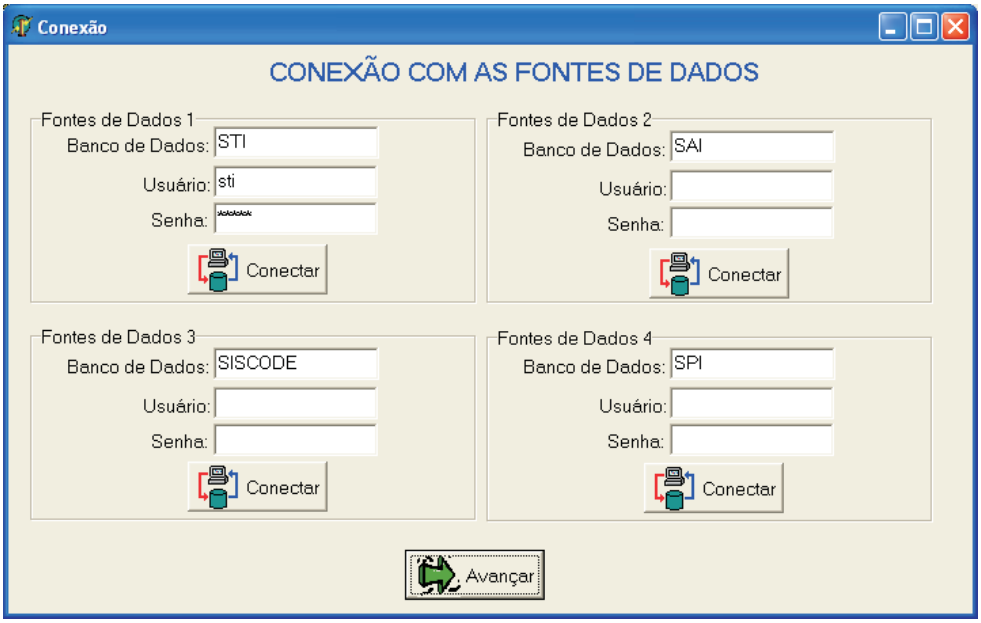

Figura 3.13. Tela inicial da aplicação

Ao ser executada a aplicação, é aberta a tela inicial, mostrada na Figura 3.13, na qual podem ser feitas as conexões às bases de dados do STI, SAI, SISCODE e SPI. Para tanto, deve-se informar o nome do usuário e a senha de acesso.

Na tela seguinte à inicial, ilustrada na Figura 3.14, o administrador do processo de ETL visualiza as tabelas de onde os dados serão extraídos (Origem), bem como aquelas onde os mesmos serão inseridos (Destino). Além disso, deve-se informar ainda nessa tela os dados necessários para a gestão da dimensão tempo do DW, os quais também são utilizados como parâmetros de entrada para as *stored procedures* chamadas durante o processo de extração. Como apresentado no trecho de código a seguir, as chamadas a tais procedimentos foram implementadas no evento *onClick* do botão *Migrar dados*.

```
with DM.sp_ext_receita do begin
   Close;
   ParamByName('@data_inicial').AsDateTime := data_inicial.date;
   ParamByName('@data_final').AsDateTime   := data_final.date;
   ExecProc;
end;
```

Figura 3.14. Tela de gestão da dimensão tempo do DW.

No caso do administrador do processo ETL desejar visualizar os comandos SQL responsáveis pela extração, transformação e carga dos dados para a DSA, basta clicar em uma das células da coluna Detalhe da Figura 3.14, correspondente à tabela de seu interesse. Ao clicar em uma dessas células, é aberta a tela apresentada na Figura 3.15. Na mesma, é possível alterar e salvar os comandos SQL, se houver necessidade.

Vale ressaltar que, durante esse processo de migração, as *stored procedures* utilizadas já implementam as transformações dos dados para que os mesmos sejam inseridos na nova base de dados limpos e consistentes.

Finalmente, o último passo das atualizações periódicas do DW é a carga dos dados propriamente dita. Esse procedimento também é implementado por meio de stored procedures, as quais são chamadas ao se clicar no botão Migrar dados que aparece na tela ilustrada na Figura 3.16.

Nessa tela, assim como naquela que é mostrada na Figura 3.14, é possível visualizar os comandos SQL, clicando nas células da coluna Detalhe.

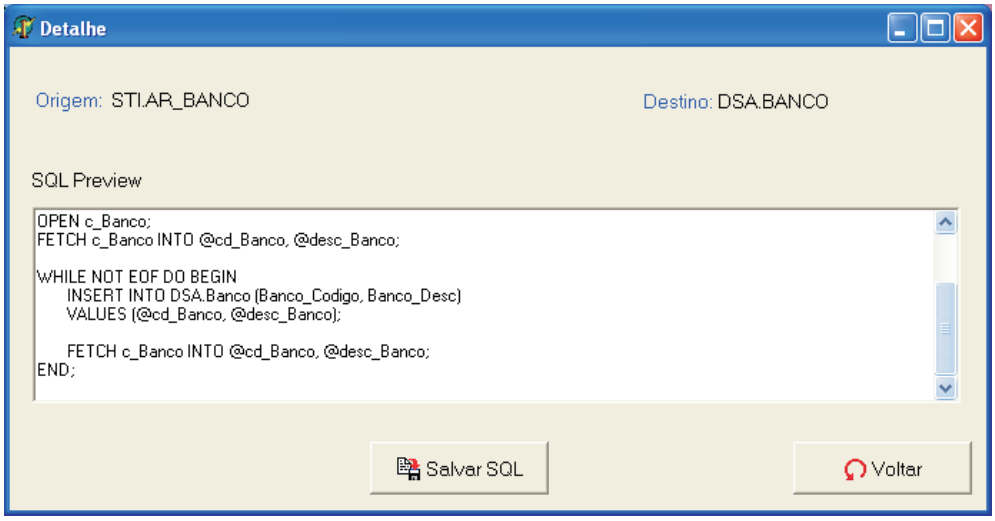

Figura 3.15. Tela da aplicação que permite a visualização e alteração dos comandos SQL responsáveis pelo processo de ETL.

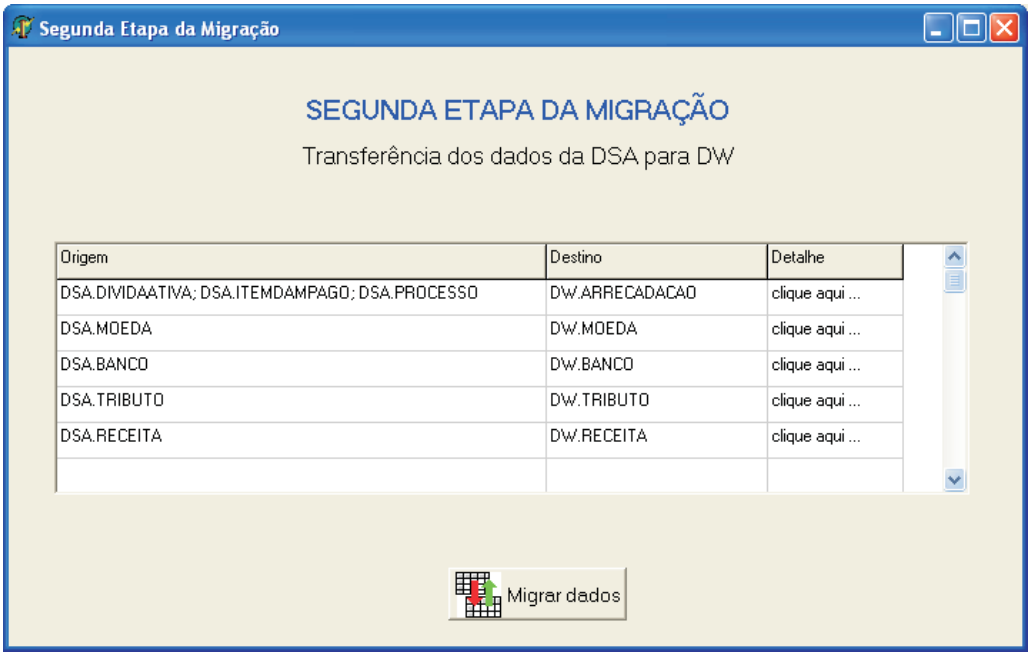

Figura 3.16. Tela da aplicação que permite a migração dos dados para o DW.

Uma vez completada a carga dos dados para o ambiente do DW, o administrador do processo tem a opção de imprimir o Relatório de Avaliação do Processo de ETL, no qual podem ser visualizados o período de extração, quais as fontes de dados utilizadas, a quantidade de registros extraídos por tabela, dentre outros dados.

3.4.4 Agoritmos

Como descrito na seção 3.3, o processo de transferência dos dados das fontes para o ambiente do DW ocorreu em duas etapas: a primeira para povoar a DSA e a segunda para migrar tais dados para o DW. A seguir são apresentados alguns dos algoritmos utilizados durante tais etapas, para exemplificar como ocorreu a implementação das mesmas.

```
//ALGORITMO 1

//Extrai as receitas das bases de dados do STI e SAI

  //Seleciona as receitas existentes da base de dados do STI
     Declare c_Receita_STI
        Select Rec_CdContabil, Rec_Descricao, Rec_NivelRec
        From AR_Receita

  //Seleciona as receitas existentes da base de dados SAI que não estão na base de dados do STI
     Declare c_Receita_SAI
        Select ECR_NaturezaReceita, ECR_DescReceita, ECR_Nivel
        From Ex_Orc_Classificacao_Receita R
        Where R.ECR_NaturezaReceita not in (Select Rec_CdContabil from AR_Receita)

     Open c_Receita_STI
     Fetch Next into @CdContabil, @Descricao, @Nivel
     While not EOF do begin
          Insert into Receita (Rec_CdContabil, Rec_Descricao, Rec_Nivel)
               values (@CdContabil, @Descricao, @Nivel)
          Fetch Next into @CdContabil, @Descricao, @Nivel
     End;

     Open c_Receita_SAI
     Fetch Next into @CdContabil, @Descricao, @Nivel
     While not EOF do begin
          Insert into Receita (Rec_CdContabil, Rec_Descricao, Rec_Nivel)
               values (@CdContabil, @Descricao, @Nivel)
          Fetch Next into @CdContabil, @Descricao, @Nivel
     End;

     Close c_Receita_STI
     Close c_Receita_SAI
```

Esse primeiro algoritmo implementa, a extração dos tipos de receita das fontes de dados do DW. Para tanto, são declarados dois cursores, os quais são abertos e percorridos registro-a-registro para que seja feita a inserção nas tabela Receita da DSA.

O algoritmo 2 é o responsável pela captura dos dados de processos de isenção de tributos da base do SPI, para que sejam inseridos na tabela Processo da DSA. Esse

```
//ALGORITMO 2
//Extrai os processos de isenção de ISS da base de dados do SPI
  //Seleciona os processos cujo último status é arquivado deferido
      Declare c_Processos_ISS
        Select (proc_ano+"/"+orgao_codigo+unidprot_codigo+proc_numero) as num_proc
               proc_datahora, proc_valor
        From Processo p, Assunto a
        Where p.assunto_codigo = a.assunto_codigo
          and (a.assunto_nome like %isenção%
              or a.assunto_nome like %isencao%
              or a.assunto_nome like %isençao%
              or a.assunto_nome like %isencão%)
        and  a.assunto_nome like %ISS%
        and p.status_codigo = 4                  //status arquivado deferido

   //Seleciona o código do assunto ISS
   //Caso o assunto ainda não esteja cadastrado, o mesmo é inserido
   If not exists(Select @ass_codigo = ass_codigo from Assunto where ass_nome like %ISS%)
     insert into Assunto (ass_nome) values ("Isenção de ISS")

   Open c_Processos_ISS
   Fetch Next into @proc_numero, @proc_datahora, @proc_valor

   While not EOF do begin
        Insert into Processo(proc_numprocesso, proc_datahora , proc_valor , ass_codigo)
             values (@proc_numero, @proc_datahora, @proc_valor , @ass_tributo)
        Fetch Next into  @proc_numero, @proc_datahora, @proc_valor
   End;

   Close c_Processos_ISS
```

algoritmo apresenta um exemplo de uma das funções de transformação, a concatenação de dados para compor os números dos processos.

A implementação da carga dos dados da DSA para o DW é exemplificada com o algoritmo 3 seguinte, o qual realiza a carga da tabela de fatos Arrecadacao.

```
//ALGORITMO 3

//Seleciona os dados para compor a tabela de fatos
   Select trib_codigo, banco_codigo, rec_codigo,
          moeda_codigo, sum(idp_vlpago) as vl_semana,
          count(dp_numdam) as qtdedoc,
          (select count(proc_numprocesso) as qtdeproc.
                  Sum(proc_valor) as vlproc
           from Processo
           where proc_datahora between @data_inicial and
                                       @data_final)
           (select sum(da_valor) as vldivida
            from DividaAtiva da
            where da_dtvenc between @data_inicial and
                                    @data_final
              and da_prescrito = 'N')
           (select sum(da_valor) as vlprescrito
            from DividaAtiva da
            where da_dtvenc between @data_inicial and
                                    @data_final
              and da_prescrito = 'S')
    From ItemDamPago idp, Moeda m, Tributo t,
         Lote l, Receita r, DamPago, dp
    Where idp.dp_numdam   = dp.dp_numdam
      and m.moeda_codigo = dp.moeda_codigo
      and idp.rec_codigo      = r.rec_codigo
      and dp.trib_codigo      = t.trib_codigo
      and dp.lot_num          = l.num_lote
    Group by trib_codigo, banco_codigo,
             rec_codigo, moeda_codigo
```

Capítulo 4

Avaliação da Aplicação e Resultados Alcançados

Este capítulo apresenta uma avaliação, em termos de funcionalidade e de custos, a respeito da aplicação implementada. Essa avaliação pauta-se em uma análise comparativa entre essa aplicação e três ferramentas comerciais existentes, as quais são: Oracle*8i* Warehouse Builder [C+01], Warehouse Workbench 4.96 [SOL03] e Data Transformation Service [COF00] [PER00].

A escolha por tais ferramentas fundamentou-se na disponibilidade de fontes bibliográficas que as descrevem e na facilidade de contato com o suporte técnico das mesmas. Além disso, na escolha pelo Oracle*8i* Warehouse Builder, considerou-se o fato da plataforma Oracle ser o padrão de SGBD utilizado na PMM.

4.1 Solução Oracle - Oracle8*i* Warehouse Builder

O Oracle*8i* Warehouse Builder, também conhecido como OWB, é uma ferramenta ETL desenvolvida pela Oracle Corporate, altamente integrada ao Oracle8i. Essa característica elimina a necessidade de um servidor de transferência de dados adicional e de um SGBD de um outro fornecedor para armazenamento dos dados do DW.

A arquitetura de software do OWB apresenta um aplicativo de múltiplas camadas, conforme ilustrado na Figura 4.1.

O *repositório de metadados* do OWB utiliza o padrão Oracle Common Warehouse Model [C+01]. Este repositório promove o intercâmbio entre os metadados e os demais produtos Oracle, como Oracle Discoverer e Oracle Express, a fim de que os mesmos possam ser utilizados pelos usuários.

A *interface com o usuário*, escrita inteiramente em Java, apresenta um conjunto de editores gráficos que permitem projetar e desenvolver objetos de DW, além de oferecer módulos de transformações amigáveis.

A biblioteca de transformações do OWB, usada durante os mapeamentos de dados das fontes para o destino, inclui um conjunto de funções padrões classificadas em uma das seguintes categorias:

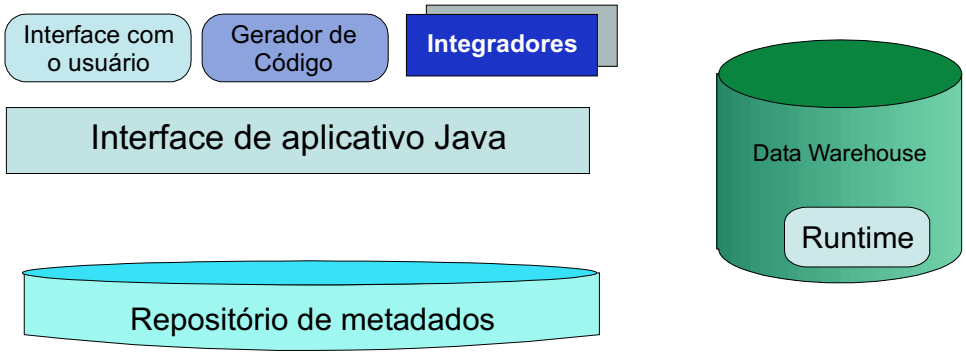

Figura 4.1. Arquitetura de software do OWB. Fonte adaptada [C$^+$01]

- básica: são transformações simples para mover dados de uma coluna de origem para uma coluna de destino;
- caracter: transformações envolvendo strings, tais como concatenação de dois valores, mudança de letras maiúsculas para minúsculas, etc.;
- conversão: esse tipo de função é usada em conversões de tipos de dados, como por exemplo, mudar um tipo inteiro para caracter;
- data: transformações usadas em um valor definido como um tipo de data, cujo retorno também é um valor de um tipo de data Ex.: identificação do último dia de um mês, a recuperação da data do sistema, dentre outras e
- numérica: são transformações que recebem uma entrada numérica e retornam valores também numérico.

Após a definição dos objetos (tabelas de fatos, dimensões, transformações, dentre outros) e dos mapeamentos necessários à criação do DW, o OWB utiliza seu gerador de código para produzir os scripts de cada um desses itens, com o intuito de implementar a base de dados. Esse procedimento ocorre em três estágios, como descrito a seguir:

- estágio um (configuração): antes que qualquer script possa ser processado para geração de objetos no DW, é necessário definir algumas características, por meio de parametrização. Como exemplo, pode-se especificar se será permitido ou não processamento paralelo durante a execução de consultas;
- estágio dois (validação): consiste na verificação das definições dos objetos com intuito de detectar possíveis erros. Alguns erros que podem ser observados durante esse estágio incluem erros de chave estrangeira e a não correspondência de tipos de dados entre origem e destino e
- estágio três (geração): uma vez definidos, configurados e validados os objetos do DW, o código pode finalmente ser gerado.

Como apresentado no segundo capítulo, os dados que constituirão o DW geralmente são provenientes de fontes distintas, representando um elemento dificultador do processo de extração. Para amenizar os problemas decorrentes dessa característica, o OWB apresenta componentes integradores que oferecem suporte a SGBD Oracle versões 7.x até Oracle8i, além de outras plataformas relacionais, tais como DB2, Informix, SQL/Server e Sybase. Essa integração é feita utilizando gateways transparentes do Oracle, detalhados na próxima seção.

Para que tenha utilidade, o DW deve ser atualizado constantemente com novos dados de acordo com intervalos de tempo predefinidos. O maior objetivo do OWB *runtime* é fazer auditorias dessas tarefas e gerar estatísticas de processamento (data em que os dados foram carregados pela última vez, duração da carga, registros carregados/rejeitados, etc.). Além disso, esse componente também permite validar os dados antes de serem efetivamente migrados para o DW, de acordo com critérios de qualidade definidos.

O último componente da arquitetura OWB é a interface de aplicativo Java, utilizada na integração dos demais componentes ao repositório de metadados.

4.1.1 Oracle Transparent Gateways

A tecnologia Oracle Transparent Gateways foi desenvolvida para permitir o acesso transparente e modificações de dados armazenados em bases não-Oracle, como ilustra a Figura 4.2. Para que essas funcionalidades possam ser usufruídas, o Gateway deve ser instalado e executado na máquina onde o banco de dados externo reside [C+01].

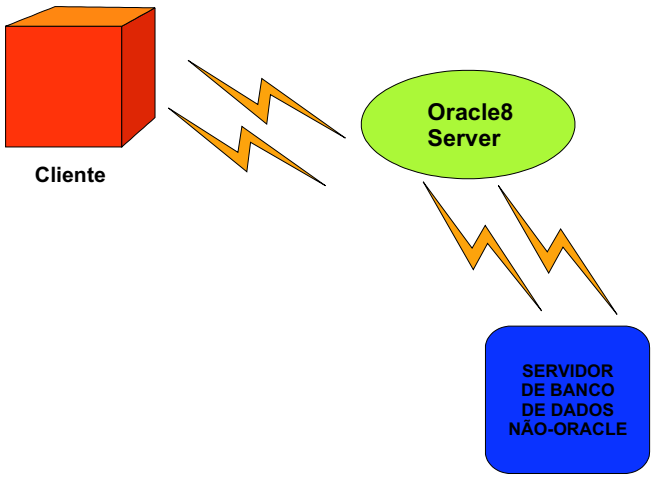

Figura 4.2. Ilustração do funcionamento da tecnologia Oracle Transparent Gateway. Fonte adaptada [C+01]

As vantagens que essa tecnologia pode oferecer são inúmeras, dentre as quais pode-se relacionar:

- integração de diversos ambientes por meio da configuração de acesso transparente aos dados;
- acesso a dados armazenados em bases não-Oracle com o uso da linguagem SQL padrão da Oracle (PL/SQL) e
- mapeamento de forma transparente dos tipos de dados na plataforma não-Oracle para os tipos de dados nativos Oracle.

Em resumo, os Gateways permitem a leitura de dados armazenados em SGBD de diversos fornecedores, dentre os quais destacam-se: APPC da IBM, DB2 da IBM, RDMS da Unisys, DRDA da IBM, SQL Server, Sybase e Informix.

4.1.2 SQL*Loader

Embora não seja empregado exclusivamente em ambiente DW, diversos projetos dessa natureza, quando optam por uma solução Oracle, utilizam o SQL*Loader para realização do processo de carga de dados.

```
          1         2         3         4         5         6         7
01234567890123456789012345678901234567890123456789012345678901234567890123456789
0-07-882390-0O Oracle8 Tuning                    Corey Abbey Dechichio Abramson
8-01-758673-4Data Warehouse                      Wilson José de Oliveira
5-56-457276-9Projeto de Data Warehouse           Felipe Nery Rodrigues Machado
```

Figura 4.3. Exemplo de entrada de dados para o SQL*Loader. Fonte adaptada [C+01]

Essa ferramenta é usada para mover dados alfanuméricos para tabelas de banco de dados Oracle. Normalmente, a entrada para o SQL*Loader é um arquivo simples de texto ASCII, como mostra a Figura 4.3.

As duas primeiras linhas da **Figura 4.3** determinam as posições das colunas no arquivo texto, enquanto que as três últimas contêm dados a respeito de livros. Uma vez que o SQL*Loader tenha conhecimento que o ISBN do livro encontra-se entre as colunas 0 e 11, o título entre as colunas 12 e 38 e o autor entre as posições 39 e 79, os dados podem ser carregados corretamente para as tabelas de destino.

Dentre os recursos que o SQL*Loader pode oferecer, estão inclusos:

- extrair dados de disco e fita magnética;
- ler registros de comprimento fixo ou variável;
- carregar dados seletivamente em uma ou mais tabelas, com base em critérios de filtragem;
- produzir relatórios de erro para auxiliar possíveis conferências e processamentos de dados inconsistentes e
- processar dados previamente, antes que os mesmos sejam efetivamente migrados para o Oracle.

4.2 Solução SOLONDE - Warehouse Workbench 4.96

O Warehouse Workbench 4.96 é um sistema de componentes para automatização de processos de extração, transformação e integração de fontes de dados, desenvolvido pela SOLONDE [SOL03].

A arquitetura dessa ferramenta é baseada em uma estrutura em formato de anel, a qual provê a integração de diversos recursos dentro de uma organização, tais como: bases de dados, sistemas CRM, DW, Internet e ERP.

Essa estrutura fundamenta-se em quatro componentes básicos, os quais são: Meta-Connectors, Global Object Store, Transformation Bus e Designer Components. Figura 4.4 ilustra a arquitetura em questão, incluindo os seus componentes.

4.2.1 Meta-Connectors

Os Meta-Connectors atuam dentro da arquitetura Warehouse Workbench 4.96 como uma ponte de conexão entre os vários sistemas de origem e destino de dados existentes na organização. A SOLONDE disponibiliza Meta-Connectors para os seguintes ambientes: Oracle, Sybase, DB2, SQL Server, Informix, ODBC, Flat Files, HTML e SAPR/3.

Esses componentes procuram ocultar a complexidade e as diferenças dos sistemas conectados à arquitetura. Desta forma, os usuários podem concentrar o foco de seu trabalho na lógica de transformação dos dados e não na compreensão das diversas plataformas.

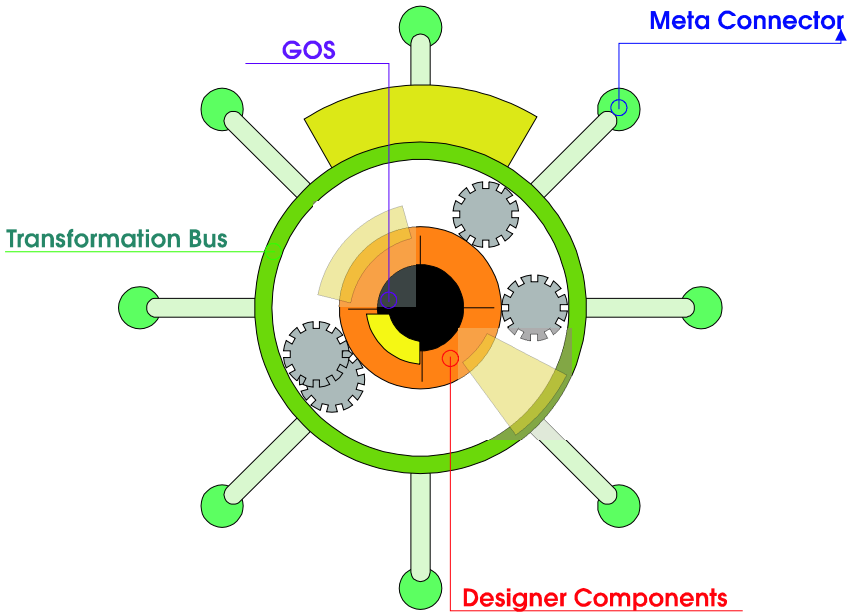

Figura 4.4. Arquitetura Warehouse Workbench. Fonte adaptada [SOL03].

4.2.2 Global Object Store

O segundo componente básico da arquitetura, Global Object Store (GOS) é um repositório de regras de transformação, o qual permite a reutilização das mesmas entre múltiplos projetos e equipes de desenvolvimento.

O GOS possibilita um gerenciamento eficiente dos processos de transformação, uma vez que separa o ambiente de projeto do ambiente de execução. Essa característica permite que as regras sejam definidas uma única vez e utilizadas sempre que necessário.

4.2.3 Transformation Bus

O terceiro componente, Transformation Bus, é uma via de comunicação bi-direcional entre as fontes de dados conectadas à arquitetura de integração de informações da SOLONDE.

Esse barramento, em formato de anel, representa uma estrutura eficiente de interconexão, uma vez que, juntamente com o GOS, permite a reutilização de regras de transformação.

4.2.4 Designer Components

Este último componente da arquitetura representa um conjunto de elementos para monitoramento do processo de integração.

Um desses elementos é o Process Designer, o qual é um console para projetar processos de transformação. Além disso, possui uma interface de simulação interativa que possibilita detectar potenciais erros e auxiliar o desenvolvedor na solução dos mesmos.

O Dependency Analyzer permite ao desenvolvedor analisar os objetos envolvidos em uma transformação. O Impact Analysis, por sua vez, mostra os objetos efetivamente envolvidos em uma transformação. O Runtime Manager, outro elemento dessa ferramenta de monitoramento, possibilita o agendamento de transformações a qualquer momento.

A combinação desses componentes oferece ao usuário um controle completo até mesmo em ambientes complexos de extração e transformação de dados.

4.3 Solução Microsoft - Data Transformation Service

O Data Transformation Service (DTS) é uma ferramenta da Microsoft, embutida em seu SGBD SQL Server 7.0, para importação, exportação e transformação de dados entre fontes heterogêneas.

Para executar as funções de importação e exportação de dados, o DTS dispõe de dois aplicativos, DTS Import Wizard e DTS Export Wizard. A origem e destino dos dados podem ser uma das seguintes opções: dBase (5, III e IV), Microsoft Data Link, Microsoft Visual FoxPro Driver, Microsoft OLE DB Provider for Oracle, Microsoft ODBL Driver for Oracle, Microsoft ODBC Driver for SQL Server, Microsoft Access, Microsoft Excel (3.0, 4.0, 5.0 e 8.0), Microsoft for SQL Server OLE DB Provider, Paradox (3.x, 4.x e 5.x) e Text File [COF00]. Quanto às transformações de dados no DTS, incluem funcionalidades como: alterações em nomes de colunas, tipos de dados, nulidade, tamanho e formatos de campos.

A seqüência de passos empregada em uma movimentação e transformação de dados utilizando a ferramenta em questão, pode ser armazenada em um objeto conhecido como package [PER00]. Dessa forma, tal objeto poderá ser reutilizado e ter sua execução agendada quando houver necessidade.

De acordo com Youness *apud* [PER00], as seguintes ferramentas do SQL Server 7.0 são utilizadas para definição de packages:

- DTS Designer: é uma interface gráfica do SQL Server Enterpriser Manager usada para projetar e executar *packages*;
- DTS Import e Export Wizard: disponibilizam componentes que facilitam o processo de criação do packages para importação e exportação de dados entre fontes heterogêneas e
- Interfaces de programação DTS: são um conjunto de interfaces de automação para criação de aplicações de importação, exportação e transformação de dados.

4.4 Análise Comparativa

Essa seção tem por finalidade traçar uma comparação entre as ferramentas ETL apresentadas (OWB, Warehouse Workbench 4.96 e DTS) e a aplicação desenvolvida.

Embora, para esse trabalho, o custo de aquisição seja o principal aspecto a ser avaliado, outros dois parâmetros também foram considerados, uma vez que são bastante relevantes para a análise dos objetos da comparação. Tais parâmetros são: treinamento e flexibilidade.

É importante ressaltar que, em virtude dos altos investimentos requeridos para adquirir as ferramentas ETL descritas, a análise foi realizada com base em bibliografias existentes [C$^+$01], [SOL03], [COF00], [PER00] e em consultas ao suporte técnico das mesmas.

4.4.1 Custo de Aquisição

Esse aspecto tem por objetivo comparar as três ferramentas ETL e a aplicação desenvolvida em termos dos investimentos necessários para sua aquisição.

Dos quatro objetos da comparação, a solução Oracle, OWB, apresentou maior custo para sua aquisição, ficando seu valor em cerca de US$ 5.000,00. O custo da ferramenta Warehouse Workbench 4.96 é US$ 1.995,00.

A ferramenta DTS da Microsoft, como é um recurso embutido no SQL Server 7.0, para sua utilização, a organização deve-se adquirir esse SGBD, cujo investimento é de aproximadamente US$ 1.825,00, com direito a cinco licenças de uso.

Como já descrito no capítulo anterior, na implementação da aplicação ETL foram empregados somente softwares já existentes na organização, para os quais os desenvolvedores já possuíam completo domínio. Desta forma, como não houve a necessidade de investir em novas ferramentas nem em treinamentos, na composição dos custos dessa solução, considerou-se a quantidade de horas técnicas utilizadas pela equipe de desenvolvimento.

Considerando o valor médio dessa hora como sendo US$ 15,00 e que foram necessárias 70h para implementação, o custo de desenvolvimento dessa aplicação ficou em tornou de US$ 1.050,00.

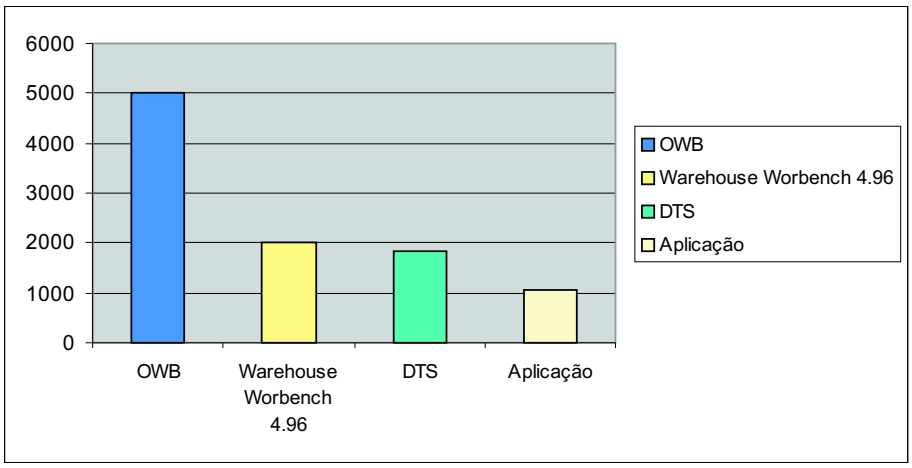

Figura 4.5. Gráfico comparativo entre as três ferramentas ETL e a nossa solução, com relação ao aspecto custo de aquisição.

Convém frisar que, a análise comparativa desse parâmetro não tem por objetivo determinar que uma ferramenta é melhor que outra, por ter um custo mais acessível. Em algumas ocasiões, as necessidades da organização podem requerer soluções mais dispendiosas, para as quais os investimentos acabam por ficar em segundo plano. A Figura 4.5 apresenta um gráfico resumo do parâmetro descrito nessa seção.

4.4.2 Treinamento

No caso da aplicação apresentada no corrente estudo, não houve necessidade de investimentos em treinamento, uma vez que, uma das principais preocupações foi utilizar em sua implementação somente softwares que faziam parte do domínio de conhecimento da equipe de desenvolvimento.

O treinamento para utilização do OWB é feito em um único módulo, com o tempo de duração de três dias, sendo o investimento necessário em torno de US$ 387,00. O fato de se fazer esse treinamento, não significa que o desenvolvedor seja certificado pela Oracle. Para obtenção de tal título, o mesmo deve ainda ser aprovado em uma avaliação, cujo valor é US$ 145,00, sendo o local de sua realização fora da cidade de Manaus. Isso implica em gastos adicionais com passagem aérea e hospedagem.

No tocante à ferramenta Warehouse Workbench 4.96 da SOLONDE, o seu treinamento ocorre em dois módulos: Basic Training e Advanced Training. O primeiro deles requer um investimento de US$ 7.995,00 e o segundo de US$ 9.995,00. Somando-se a isso, tais cursos são realizados somente nos Estados Unidos e oferecidos apenas em inglês e alemão.

Quanto ao DTS, não foi encontrado junto ao suporte técnico ou bibliografia, o custo referente ao treinamento específico dessa ferramenta.

4.4.3 Flexibilidade

Para efeitos desse estudo em particular, considera-se flexibilidade a capacidade da ferramenta ETL poder movimentar dados entre diversas plataformas.

No que tange esse aspecto, as três ferramentas apresentam-se bastante flexíveis, permitindo especificar fontes e destinos de dados que incluem SGBD relacionais, arquivos textos, HTML, planilhas, etc.

Ao contrário disso, a aplicação desenvolvida mostra-se pouco flexível, uma vez que movimenta dados de uma base de dados Oracle para outra. Essa característica deve-se ao fato da mesma ter sido proposta para funcionar em um ambiente homogêneo.

Além disso, observou-se na literatura que as ferramentas DTS e Warehouse Workbench 4.96 disponibilizam mecanismos para armazenar e reutilizar regras de transformação de dados. No OWB, tais funcionalidades não foram encontradas nas bibliografias consultadas.

Capítulo 5

Conclusão

A crescente preocupação dos gestores com o domínio da informação sobre as suas organizações, conduziu os mesmos a direcionarem recursos financeiros, de tempo e pessoal para tirarem o maior proveito possível de seus sistemas transacionais.

Essa atenção especial sobre tais sistemas deve-se ao fato dos mesmos armazenarem dados importantes a respeito das empresas, os quais, se forem bem explorados, podem gerar informações estratégicas que as coloquem em uma posição vantajosa diante de seus concorrentes.

Uma boa alternativa para transformar esses dados em informações que auxiliem o nível estratégico das organizações é o emprego de tecnologias de suporte à decisão, como o DW, o qual permite extrair os dados dos sistemas transacionais e transferi-los para um novo ambiente, possibilitando a descoberta de informações antes ocultas.

Essa transferência de dados não é tarefa trivial, pois, muitas vezes, estes estão distribuídos em plataformas de hardware e software distintas dentro da organização. Dessa forma, para facilitar a realização desse processo pelos desenvolvedores, surgiram as ferramentas ETL. Entretanto, a aquisição desse tipo de software requer o emprego considerável de investimentos financeiros por parte da organização, intimidando, em muitos casos, o desenvolvimento de um projeto de DW.

Assim, com a preocupação de reduzir custos durante a fase de ETL e, conseqüentemente, durante o projeto de DW como um todo, o presente trabalho abordou uma alternativa para este problema, com a implementação de uma aplicação.

5.1 Contribuições

As diretrizes propostas nesse estudo conduziram à implementação, por parte da própria equipe de desenvolvimento da organização, de uma aplicação ETL, no ambiente do Departamento de Tributação da PMM, com o objetivo de apresentar uma alternativa de redução de custos em projetos de DW.

Para a implementação dessa aplicação foram utilizadas as seguintes ferramentas de desenvolvimento de software: Delphi 5.0, SQL Navigator 3.0 d2 e Oracle 8i, para as quais o DETRIB já possuía as devidas licenças de uso. Além das licenças, outro fator levado

em consideração para a escolha de tais ferramentas foi o fato da equipe de desenvolvimento possuir domínio sobre as mesmas, não necessitando de investimentos em termos de treinamento.

De acordo com as análises apresentadas no capítulo 4, as quais consideraram os aspectos custo de aquisição, treinamento e flexibilidade, esta alternativa mostrou-se uma solução viável em ambientes homogêneos e apresentou-se com um custo bastante razoável quando comparada a outras ferramentas ETL comerciais, embora seja pouco flexível.

Além do custo mais baixo para sua aquisição, a aplicação desenvolvida mostrou-se uma alternativa interessante por não necessitar de investimentos em termos de treinamento, os quais, de acordo com a análise comparativa, chegaram a representar mais de 900% do valor das ferramentas.

Desta forma, conclui-se que a hipótese apresentada de que é possível reduzir custos em projetos de DW, com o desenvolvimento de aplicações ETL em ambientes com bases de dados construídas sobre plataformas homogêneas, foi comprovada.

Para a realização dessa comprovação, algumas dificuldades precisaram ser superadas. A principal delas refere-se à falta de documentação dos sistemas transacionais que serviram de fontes de dados para o DW proposto nesse trabalho (STI, SAI, SISCODE e SPI). Essa falta de documentação dificultou a compreensão dos esquemas de tais sistemas e, conseqüentemente, o processo de integração dos mesmos.

Além dessa, outra dificuldade observada durante o desenvolvimento dessa investigação, referiu-se à obtenção de informações sobre as três ferramentas comerciais utilizadas nas análises comparativas com a aplicação. Como o custo de aquisição de tais ferramentas apresentou-se bastante elevado para que elas fossem efetivamente testadas, todas as informações descritas tiveram que ser coletadas junto às referências bibliográficas e ao suporte técnico das mesmas.

5.2 Trabalhos Futuros

Para dar continuidade a este trabalho, são sugeridos os seguintes estudos:

Utilização de Software Livre

Para a implementação da aplicação proposta nesse estudo, foram empregadas ferramentas de desenvolvimento de software usadas na organização, porém não gratuitas.

A utilização de software livre, como Linux e MySQL, além de seguir a tendência atual do mercado como descrito em [C+01], reduziria ainda mais os custos de projetos de DW em ambientes que não dispõem de recursos como a PMM.

Ferramenta Flexível

Como apresentado no capítulo 4, a aplicação ETL implementada mostrou-se pouco flexível, uma vez que foi projetada para um ambiente homogêneo, no qual as bases de dados foram todas desenvolvidas em plataforma Oracle.

O desenvolvimento de uma aplicação mais flexível, que permitisse a movimentação de dados entre ambientes heterogêneos, seria bastante interessante, pois não restringiria o seu uso a um único cenário.

Gerenciador de Metadados

Neste trabalho, foram tratados assuntos referentes aos processos de extração, transformação e carga de dados, no entanto, não houve uma preocupação quanto aos metadados, importantes tanto para os usuários finais quanto para os desenvolvedores.

Assim, o desenvolvimento de um gerenciador de metadados integrado a uma ferramenta ETL flexível, implementada com software livre, apresenta-se como uma alternativa mais completa para auxiliar a equipe de desenvolvimento em projeto de DW.

Referências Bibliográficas

[5498] INFORMATIVO TÉCNICO NRO 54. Análise multidimensional. *Revista Unicamp*, 1998.

[7300] INFORMATIVO TÉCNICO NRO 73. Procurado qualquer um com experiência em... *Revista Unicamp*, 2000.

[B+00] Grady BOOCH et al. *UML Guia do Usuário*. Rio de Janeiro: Campus, 2000.

[BLN86] C. BATINI, M. LENZERINI, and S. NAVATHE. A comparative analysis of methodologies for database schema integration. *ACM Computing Surveys, v.l 18, n.4, dez*, 1986.

[Bra02] DW Brasil. Data mart. http://www.dwbrasil.com.br/html/dtmart.html. Acesso em: 22 set.2002, 2002.

[C+96] Gilberto CAMARA et al. *Anatomia de Sistemas de Informação geográfica*. Campinas: Instituto de Computação, UNICAMP, 1996.

[C+01] COREY et al. *Oracle 8i Data Warehouse*. Rio de Janeiro: Campus, 2001.

[CAN00] Marco CANTÚ. *Dominando o Delphi 5: A Bíblia*. São Paulo: Makron Books, 2000.

[CAR02] Rodrigo CARDOSO. Introdução a stored procedure e triggers no firebird (parte1). http://www.infosquad.net/colunas/firebird/index.php?ID=96. Acesso em: 20 abr. 2003, 2002.

[CIE02] Ivã Rafael CIELO. Etl: Extração, transformação e carga de dados. http://www.datawarehouse.inf.br/artigos/etl.asp. Acesso em: 17 set. 2002, 2002.

[COF00] Gayle COFFMAN. *SQL Server Completo e Total*. São Paulo: Makron Books, 2000.

[DAT90] C. J. DATE. *Introdução a Sistemas de Bancos de Dados*. Rio de Janeiro: Campus, 1990.

[dBdDS03] Grupo de Base de Dados SCE. Resumo dos comandos da linguagem sql: comandos para triggers e stored procedures. SCE-ICMSC-USP. http://gbdi.icmc.sc.usp.br/disciplinas/sce-228/storedprocedure.pdf. Acesso em: 10 abr. 2003, 2003.

[dE03] Companhia Paranaense de Energia. Copel incentiva e apóia iniciativas do projetos software livre paraná. COPEL. http://www.softwarelivre.org/index.php?menu=mais_noticias2&cod=1057439058&tab=1. Acesso em: 05 jul. 2003, 2003.

[DOA00] AnHai DOAN. Learning to map between structured representations data. http://www.citeseer.nj.nec.com/doan02learning.html. Acesso em: 15 jul. 2003, 2000.

[Edu02] FAGUNDES Eduardo. O que é um data warehouse? Disponível em: http://www.efagundes.com/disc1/00000027.htm. Acesso em: 08 out. 2002, 2002.

[ELE02] REVISTA ELETRÔNICA. Data warehouse. http://www.unifieo.br/revista/data.html. Acesso em: 15 fev. 2003, 2002.

[FJ01] Edson N. FERNANDES JR. *Consolidação da Legislação Tributária do Município de Manaus*. Manaus: Valer, 2001.

[FLO99] Feltrin Christian FLORES. Projeto e desenvolvimento de data warehouse hospitalar. Universidade Federal de Santa Maria, 1999.

[FM02] A. FUXMAN and R. J. MILLER. Towards incosistency management in data integration systems. http://citeseer.nj.nec.com/fuxman03towards.html. Acesso em: 15 jul. 2003, 2002.

[FNC98] M. A. FELDENS, F. B. NARDON, and J. M. CASTILHO. Sistemas de apoio à decisão baseados em componentes. Universidade Federal do Rio Grande do Sul, 1998.

[FRE02] Gilmar Meira FREITAS. Uma ferramenta de apoio à modelagem de dados dimensional. Belo Horizonte: Escola de Governo. Fundação João Pinheiro, 2002.

[INM97] W.H. INMON. *Como Construir o Data Warehouse*. Rio de Janeiro: Campus, 1997.

[J+98] N. Celina JORGE et al. *Processo de Extração no Data Warehouse*. Belo Horizonte, 1998.

[KIM02] Raph KIMBALL. *The Data Warehouse Toolkit*. Rio de Janeiro: Campus, 2002.

[MAN03] MANAUS. Lei municipal nº 1.073, de 16 de novembro de 1973. dispõe sobre a lei orgânica do município de manaus. http://www.interlegis.gov.br/processo_legislativo. Acesso em: 12 abr.2003, 2003.

[MEL03] Áurea H. S. MELO. *Uma Estratégia para Desenvolvimento de Data Warehouse Geográfico com Integração Híbrida Aplicada ao Monitoramento de Queimadas na Amazônia*. Recife: Departamento de Eletrônica e Sistemas, UFPE, 2003.

[MIR03] Rosemary Moreira MIRANDA. Utilização do modelo dimensional para diagnósticos de saúde no município de belo horizonte. Belo Horizonte. Prodabel PUC-IRT. http://www.pbh.gov.br/informatica/programa-formacao/especializacao/rosemary.pdf. Acesso em: 10 jan. 2003, 2003.

[MP01] Peter MCBRIEN and Alexandra POULOVASSILIS. Data integration by bi-diretional schema transformation rules. http://citeseer.nj.nec.com/mcbrien03data.html. Acesso em: 15 jul. 2003, 2001.

[PER00] Denise Maciel PEREIRA. *Uso do Padrão OIM de Metadados no Suporte às Trasformações em Ambiente Data Warehouse*. Rio de Janeiro: Universidade Federal do Rio de Janeiro, 2000. (dissertação de mestrado).

[PET03] Ademir José PETENATE. Utilização do método de taguchi na redução de custos de projetos. Campinas: Universidade Estadual de Campinas, Departamento de Estatística, 2003.

[S+00] João Guilherme SILVA et al. Rede metropolitana da prefeitura municipal de manaus. Manaus: UFPE/UTAM, 2000.

[SJF02] M. A. SILVA JÚNIOR and José Paulo A. FUSCO. Data warehouse - uma ferramenta para o sucesso competitivo, 2002.

[Sof03] QUEST Software. Sql navigator. http://www.quest.com/sql_navigator. Acesso em: 07 ago. 2003, 2003.

[SOL03] SOLONDE. Warehouse workbench: Integration information architecture. http://www.solonde.com/downloads.html. Acesso em: 18 jul. 2003, 2003.

[VAR01] Aytekin VARGUN. Semantic aspects of heterogeneous databases. http://www.cs.colorado.edu/ getrich/Classes/csci5817/Term_Papers/vargun/. Acesso em: 12 ago. 2002, 2001.

[VIE89] John VIESCAS. *SQL: a linguagem padrão de banco de dados relacionais*. Rio de Janeiro: Campus, 1989.