

UNIVERSIDADE FEEVALE

LUIZ HENRIQUE FELTES

DESENVOLVIMENTO DE UMA SOLUÇÃO DE BI PARA O ERP
SIGER

Novo Hamburgo
2010

LUIZ HENRIQUE FELTES

DESENVOLVIMENTO DE UMA SOLUÇÃO DE BI PARA O ERP
SIGER

Trabalho de Conclusão de Curso
apresentado como requisito parcial
à obtenção do grau de Bacharel em
Ciências da Computação pela
Universidade Feevale

Orientador: Edvar Bergmann Araujo

Novo Hamburgo
2010

AGRADECIMENTOS

Gostaria de agradecer a todos os que, de alguma maneira, contribuíram para a realização desse trabalho de conclusão, em especial:

A minha família pelo apoio nos momentos mais difíceis do trabalho e principalmente pela educação que me foi dada. A minha namorada Ynajara, pelo carinho, apoio e compreensão durante esse período. Ao meu orientador Edvar Bergmann Araujo que acrescentou seu conhecimento para o enriquecimento desse trabalho.

RESUMO

Com o aumento da competitividade empresarial, cada vez mais as empresas buscam soluções para realizar análises e planejamentos dos seus negócios, a fim de oferecer melhores produtos e serviço aos clientes. Dentre as soluções, estão os Sistemas ERP (*Enterprise Resource Planning*) e os Sistemas de Apoio a Decisão (SAD), também conhecidos por *Business Intelligence* (BI). Os ERPs fornecem informações operacionais que servem como base para os Sistemas de Apoio a Decisão. Desta forma, este trabalho tem como objetivo desenvolver uma solução de BI que possa ser integrada ao ERP SIGER. A solução proposta permitirá a criação de repositórios de dados na forma de um *Data Mart*, bem como a respectiva transformação e carga. Objetiva-se com isto criar uma solução independente da área de atuação da empresa e que facilmente possa ser adaptada a diferentes realidades, permitindo a criação dos repositórios pelos consultores, sem a necessidade de intervenção do desenvolvedor. Para completar a solução de BI, será contemplada a análise dos dados utilizando uma ferramenta OLAP.

Palavras-chave: *Business Intelligence*. ERP. *Data Warehouse*. Sistemas de Apoio a Decisão. OLAP.

ABSTRACT

With increasing business competitiveness, more and more companies are seeking solutions to perform analysis and planning of their businesses in order to offer better products and service to customers. Among the solutions are the ERP (Enterprise Resource Planning) and the Decision Support Systems (DSS), also known as Business Intelligence (BI). The ERPs provide operational information that serve as the basis for the Decision Support Systems. Thus, this study aims to develop a BI solution that can be integrated with ERP SIGER. The proposed solution will allow the creation of data repositories in the form of a Data Mart, and their transformation and load. It aims to create a solution to this area regardless of where the company operates and that can be easily adapted to different realities, allowing the creation of repositories by consultants without the need for intervention from the developer. To complete BI solution will be included in data analysis using an OLAP tool.

Keywords: Business Intelligence. ERP. Data Warehouse. Decision Support Systems. OLAP.

LISTA DE FIGURAS

Figura 1.1 – Estrutura do ambiente de BI	14
Figura 1.2 – <i>Data Marts</i> específicos	15
Figura 1.3 – <i>Tabela fato e tabelas dimensões</i>	20
Figura 1.4 – <i>Modelo Star</i>	21
Figura 1.4 – <i>Modelo Snowflake</i>	22
Figura 1.5 – <i>Transformação dos dados</i>	24
Figura 1.6 – <i>Opções de armazenamento/implementação OLAP</i>	26
Figura 2.1 – <i>Distribuição da carteira de clientes por atividade</i>	29
Figura 2.2 – <i>Estrutura do SIGER</i>	31
Figura 3.1 – Componente da suíte Pentaho BI	34
Figura 3.2 – A <i>GUI do Pentaho</i>	35
Figura 3.3 – A criação do repositório do SADIG Free	37
Figura 3.4 – A <i>GUI do Kettle</i>	38
Figura 3.5 – Junção dos <i>jobs</i> com <i>transformations</i> em uma única tarefa	39
Figura 3.6 – O SSIS integrado ao Microsoft Visual Studio e seus componentes	40
Figura 3.7 – Linguagem multidimensional MDX utilizada pelo Mondrian	41
Figura 3.8 – <i>Schema</i> do Mondrian	42
Figura 3.9 – Ferramenta Schema Workbench para criação e testes do <i>schema</i>	42
Figura 3.10 – Barra de ferramentas do Jpivot	43
Figura 3.11 – Consulta utilizando o JPivot	45
Figura 4.1 – Visão geral da solução de BI construída	46
Figura 4.2 – Fluxograma da solução de BI	47
Figura 4.3 – Tela Inicial da ferramenta BIIM	48
Figura 4.4 – Criação de um novo Data Mart	48
Figura 4.5 – Configurações do DM e os dicionários de dados	49
Figura 4.6 – Criação de uma tabela fato ou dimensão	50
Figura 4.7 – Criação de campos	52
Figura 4.8 – Criação de campos, aba “Avançado”	53
Figura 4.9 – Esquema DDL para criação do Data Mart	54
Figura 4.10 – Parte do arquivo XML do Job gerado para ser utilizado com o Kettle	55
Figura 4.11 – Parte do arquivo XML da transformação gerada	56

Figura 4.12 – Kettle com arquivo de transformação criado _____	57
Figura 4.13 – Tela para geração dos cubos de dados _____	58
Figura 4.14 – Tela para adicionar tabela dimensão e campos ao cubo de dados _____	59
Figura 4.15 – <i>Schema</i> XML do Mondrian gerado automaticamente pela ferramenta _____	60
Figura 4.16 – Visualização de uma consulta utilizando o JPivot _____	61
Figura 4.17 – JRubik acessando o mesmo <i>schema</i> construído para o Mondrian _____	62
Figura 4.18 – Modelo ER das tabelas utilizadas para o BIIM _____	63
Figura 5.1 – Modelo ER do DM construído para o estudo de caso _____	65
Figura 5.2 – Criação de um novo DM _____	66
Figura 5.3 – Construção da tabela dimensão “Produto” _____	67
Figura 5.4 – Construção do campo virtual “ano” da tabela dimensão “Tempo” _____	68
Figura 5.5 – Construção da tabela dimensão “Cliente” _____	70
Figura 5.6 – Construção do campo “codTempo” da tabela fato “Vendas” _____	71
Figura 5.7 – Construção do campo virtual “valorTotal” da tabela fato “Vendas” _____	72
Figura 5.8 – Processamento de carga dos dados da tabela fato “Vendas” _____	73
Figura 5.9 – Criação do cubo Vendas x Produto x Estado _____	74
Figura 5.10 – Criação da dimensão “Tempo” do cubo Vendas x Produto x Estado _____	75
Figura 5.11 – Criação da dimensão “Produto” do cubo Vendas x Produto x Estado _____	75
Figura 5.12 – Criação da dimensão “Cliente” do cubo Vendas x Produto x Estado _____	76
Figura 5.13 – Criação da dimensão “Produto” do cubo Vendas x Grupo Produto x Cliente _____	77
Figura 5.14 – Análise realizada no cubo Vendas x Produto x Estado _____	78
Figura 5.15 – Operação <i>drill-down</i> realizada no estado de São Paulo _____	78
Figura 5.16 – Análise realizada no cubo Vendas x Grupo Produto x Cliente _____	79

LISTA DE QUADROS

Quadro 1.1 – Comparação entre os dados de natureza operacional e informacional _____ 19

LISTA DE ABREVIATURAS E SIGLAS

BI	<i>Business Intelligence</i>
CRM	<i>Customer Relationship Management</i>
DDL	<i>Data Definition Language</i>
DM	<i>Data Mart</i>
DOLAP	<i>Desktop On-Line Analytical Processing</i>
DW	<i>Data Warehouse</i>
ER	Entidade Relacionamento
ERP	<i>Enterprise Resource Planning</i>
ETL	<i>Extract Transform Load</i>
GUI	<i>Graphical User Interface</i>
HOLAP	<i>Hybrid On-Line Analytical Processing</i>
HTML	<i>HyperText Markup Language</i>
JDBC	<i>Java Database Connectivity</i>
MDX	<i>Multidimensional Expressions</i>
MOLAP	<i>Multidimensional On-Line Analytical Processing</i>
ODBC	<i>Open Data Base Connectivity</i>
OLAP	<i>On-Line Analytical Processing</i>
PDF	<i>Portable Document Format</i>
PDI	<i>Pentaho Data Integration</i>
ROLAP	<i>Relational On-Line Analytical Processing</i>
RTF	<i>Rich Text Format</i>
SAD	Sistemas de Apoio a Decisão
SGBD	Sistema Gerenciador de Banco de Dados
SIGER	Sistema Integrado Gestão Empresarial Rech
SQL	<i>Structured Query Language</i>
SSIS	<i>SQL Server Integration Services</i>
TI	Tecnologia da Informação
WIZARD	Janela gráfica para realizar tarefas complexas passo-a-passo
XML	<i>Extensible Markup Language</i>

SUMÁRIO

INTRODUÇÃO	11
1 BUSINESS INTELLIGENCE	14
1.1 DATA WAREHOUSE (DW)	14
1.1.1 Data Marts	15
1.1.2 Granularidade	16
1.1.3 Arquitetura do DW / DM	16
1.1.4 Tipos de implementação	17
1.1.5 Diferenças entre ambiente operacional e o ambiente DW	18
1.2 MODELAGEM DIMENSIONAL DOS DADOS	19
1.2.1 Tabela fato	20
1.2.2 Tabela dimensão	20
1.2.3 Modelo Star	21
1.2.4 Modelo Snowflake	22
1.2.5 Metadados	22
1.3 EXTRAÇÃO, TRANSFORMAÇÃO E CARGA (ETL)	23
1.4 ON-LINE ANALYTIC PROCESSING (OLAP)	25
1.4.1 Operações OLAP	25
1.4.2 Arquiteturas OLAP	26
2 ERP SIGER	28
2.1 RECH INFORMÁTICA	28
2.2 SIGER	30
3 ANÁLISES DE SOLUÇÕES DE BI	33
3.1 SOLUÇÕES DE BI	33
3.1.1 Pentaho	33
3.1.2 SADIG BI	36
3.2 FERRAMENTAS ETL	37
3.2.1 Kettle	38
3.2.2 SQL Server Integration Services (SSIS)	40
3.3 OLAP	41
3.3.1 Mondrian/JPivot	41
3.4 CONSIDERAÇÕES FINAIS	45
4 FERRAMENTA DESENVOLVIDA	46
4.1 VISÃO GERAL	46
4.2 CRIAÇÃO DE DATA MARTS	48
4.3 CONFIGURAÇÕES	49
4.4 CONSTRUÇÃO DA TABELA FATO OU DIMENSÃO	49
4.4.1 Construção dos campos	50
4.5 GERAÇÃO DO DATA MART	54
4.5.1 Integração com Kettle (ETL)	54
4.6 CUBO DE DADOS	57
4.6.1 Integração com Mondrian/JPivot	59
4.6.2 Visualização dos cubos de dados	60
4.7 MODELAGEM DA FERRAMENTA	62
5 ESTUDO DE CASO	65
5.1 CONSTRUÇÃO DO DM	65

5.1.1 Criação do DM Faturamento	66
5.1.2 Construção da tabela dimensão Produto	66
5.1.3 Construção da tabela dimensão Tempo	68
5.1.4 Construção da tabela dimensão Cliente	69
5.1.5 Construção da tabela fato Vendas	70
5.2 CARGA DOS DADOS	73
5.3 CRIAÇÃO DOS CUBOS DE DADOS	73
5.3.1 Vendas x Produto x Estado	74
5.3.2 Vendas x Grupo Produto x Cliente	76
5.4 ANÁLISE DOS CUBOS DE DADOS	77
5.4.1 Análise do cubo Vendas x Produto x Estado	77
5.4.2 Análise do cubo Vendas x Grupo Produto x Cliente	79
CONCLUSÃO	80
REFERÊNCIAS	81

INTRODUÇÃO

Com o aumento da competitividade empresarial, cada vez mais as empresas buscam soluções para realizar análises e planejamentos dos seus negócios, a fim de oferecer melhores produtos e serviços aos clientes, buscando a satisfação dos mesmos e visando afastar a constante ameaça da concorrência.

Uma dessas soluções são os Sistemas ERP (*Enterprise Resource Planning*) que, conforme Davenport (2002), são pacotes de aplicativos que dão suporte as necessidades de informações de uma empresa, constituindo um único sistema de informações. Haberkorn (2004) complementa que o ERP visa à automação dos procedimentos, abrangendo desde o planejamento, execução e controle sob o ponto de vista econômico e financeiro.

Dentre os diversos sistemas ERP disponíveis no mercado está o SIGER (Sistema Integrado de Gestão Empresarial Rech), desenvolvido pela Rech Informática, situada na cidade de Novo Hamburgo, Rio Grande do Sul. Atualmente conta com mais de 600 clientes atuando na indústria, comércio, escritórios contábeis e prestadoras de serviços. O software é modularizado e customizado para atender as diversas necessidades do mercado.

Além do uso de sistemas ERP, as empresas buscam outras soluções como os Sistemas de Apoio a Decisão (SAD) para apoiar no momento da tomada de decisão. Conforme Binder (1994), a tomada de decisão é a atividade máxima de um líder e consiste na escolha de uma ação entre diversas alternativas existentes, seguindo determinados passos previamente estabelecidos para resolução de um problema de modo correto ou não.

Os SADs são sistemas complexos que permitem acesso a base de dados corporativa, modelagem de problemas, simulações com uma interface amigável. Caracterizam-se por auxiliar os líderes em todas as fases da tomada de decisão, fornecendo subsídios para a escolha da melhor alternativa. (BINDER, 1994)

Atualmente, em substituição ao SAD, tem-se utilizado muito o termo BI (*Business Intelligence*) que, segundo Barbieri (2001), representa a habilidade de explorar e estruturar as informações, com o objetivo de desenvolver entendimentos e conhecimentos para auxiliar na tomada de decisões. O autor também completa que o conceito BI, de uma forma mais ampla, deve ser entendido como a utilização que várias fontes de informações, sejam elas de sistemas legados ou não, para se definir estratégias de competitividade nos negócios da empresa.

O processo decisório deve ser confiável. As decisões precisam ser embasadas em informações consolidadas, mesmo que estas informações estejam espalhadas em diferentes sistemas dentro de uma empresa. Desta forma, essas diversas fontes de informação precisam ser armazenadas em uma única base de dados chamada de *Data Warehouse* ou *Data Mart*.

Segundo Inmon (1997) o DW é um conjunto de dados baseado em assuntos, integrado, não volátil, e variável em relação ao tempo, de apoio as decisões gerenciais. Já Barbieri (2001) define DW como um banco de dados destinado a sistemas de apoio a decisão, organizados em estruturas lógicas dimensionais para possibilitar o processamento analítico por ferramentas especiais como OLAP (*Online Analytical Processing*).

Em síntese, o DW tem o objetivo de armazenar os dados sumarizados para facilitar e agilizar os processos de tomada de decisão. O termo *Data Mart*, conforme Barbieri (2001), refere-se ao repositório de dados que atende a áreas específicas do negócio, sendo que DW e o DM podem ser definidos como espécies do mesmo tipo, sendo a sua diferença centrada no escopo do projeto e limites de sua abrangência.

Conforme Machado (2000), OLAP são as ferramentas que possibilitam as consultas para análise dos dados e descoberta das informações, permitindo a visualização dos dados sob diversas perspectivas, possibilitando navegar no nível de detalhe da informação.

Várias soluções de BI tendem a ficarem restritas ao repositório previamente criado no DW/DM, assim como o processo de ETL (*Extract, Transform and Load*) construído. Dessa forma não é possível de uma maneira simples e rápida a manutenção dos dados no repositório, como por exemplo, a criação de novos dados, bem como sua transformação e carga da informação. Em algumas situações pode ser necessário envolver o desenvolvedor da ferramenta acarretando um custo com desenvolvimento específico.

Por outro lado, algumas soluções de BI disponíveis no mercado buscam ser flexíveis por permitirem a integração com Sistemas Legados já existentes. Dentre elas, pode-se destacar o Pentaho e o SADIG. O Pentaho é uma ferramenta *open source*, que possibilita consultas, relatórios, análises interativas, painéis de controle, integração e mineração dos dados o que a torna uma ferramenta popular quando assunto é BI (Pentaho, 2010). O SADIG é uma ferramenta comercial, de código fechado, que tem como produtos o SADIG Análises, SADIG Performance, SADIG Painéis e SADIG ETL que formam toda a suíte de BI. (SADIG, 2010)

A proposta desse trabalho é criar uma solução de BI para o ERP SIGER que possa ser adaptada para atender diversas áreas e ramos de atividade. Através de uma interface gráfica, será possível a criação do *Data Mart* pelos consultores da empresa. Estes, devidamente treinados, poderão criar a solução adequada a cada cliente sem a necessidade de intervenção dos desenvolvedores. Objetiva-se, desta forma, total flexibilidade na implantação da ferramenta de acordo com a área de negócio e os resultados esperados pela empresa contratante. A solução proposta também permitirá todo o controle de extração, transformação e carga dos dados vindos diretamente do ERP SIGER e de outros sistemas legados, deixando a solução de BI aberta para ser utilizada com outros sistemas. Para completar a solução de BI, será contemplada a análise dos dados utilizando uma ferramenta OLAP.

Essa solução trará benefícios tanto para os usuários na tomada de decisões como também para o SIGER, trazendo um diferencial competitivo para o ERP, visando à busca por novos clientes e satisfação dos atuais.

O trabalho está dividido em cinco capítulos. O primeiro capítulo aborda o conteúdo teórico sobre os assuntos envolvidos neste trabalho. O segundo capítulo apresenta o ERP SIGER e sua estrutura. O terceiro capítulo trata sobre as análises feitas em algumas ferramentas de BI, ETL e OLAP existentes a fim de verificar quais poderiam ser utilizadas na solução proposta. O quarto capítulo detalha a ferramenta construída e todas as suas funcionalidades. O quinto e último capítulo inclui um estudo de caso que visa demonstrar todo o processo de criação de uma solução de BI utilizando a ferramenta proposta e, desta forma, apresenta os resultados obtidos.

1 BUSINESS INTELLIGENCE

Segundo Barbieri (2001), o BI representa a habilidade de estruturar, acessar e explorar as informações com o objetivo de desenvolver percepções, entendimentos e conhecimentos para auxiliar o processo da tomada de decisão.

As empresas contam com grandes volumes de dados, espalhados em diversas fontes de dados, sejam elas Sistemas Legados, ERPs, CRMs ou planilhas eletrônicas, dificultando o acesso a essas informações. O maior objetivo das técnicas de BI é a definição de regras para a formatação e padronização adequada destas fontes de dados, visando transformá-los em depósitos estruturados de informações, independente da sua origem. (BARBIERI, 2001)

A figura 1.1 apresenta a estrutura de um ambiente de BI, destacando os sistemas de origem, passando pelo processo de extração e transformação, onde após esse processo, os dados são carregados no repositório, para ser feita a exploração dos dados, através de ferramentas OLAP e/ou *Data Mining*.

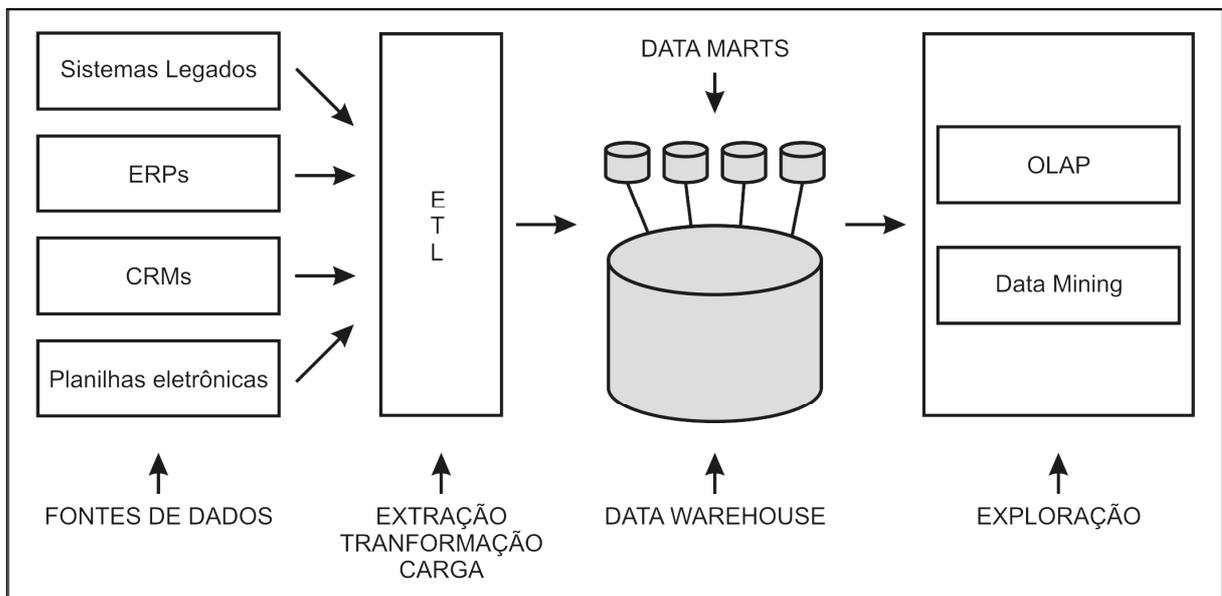


Figura 1.1 – Estrutura do ambiente de BI
Fonte: Adaptado pelo autor segundo Barbieri (2001)

1.1 DATA WAREHOUSE (DW)

O DW representa uma grande base de dados capaz de integrar as informações de interesse para a empresa, de forma concisa e confiável, que se encontram originalmente espalhadas em diversas fontes de dados, para a posterior utilização com ferramentas de análises. A finalidade principal do DW é apresentar as informações, através de dados

históricos, permitindo identificar indicadores, evoluções de valores ao longo de um período, padrões de comportamento, etc. (MACHADO, 2000)

Conforme Inmon (1997, p.33), “o *Data Warehouse* é um conjunto de dados baseado em assuntos, integrado, não-volátil, e variável em relação ao tempo”. Explicando melhor cada um dos itens:

- a) **Baseados em assunto:** a orientação por assunto significa que as informações armazenadas são agrupadas por assuntos de interesse da empresa que são mais importantes;
- b) **Integrado:** os dados devem ser codificados em um formato consistente, seguindo a mesma regra para padrões de nome, valores, tipo de dados, unidades de medidas, etc;
- c) **Não-volátil:** os dados não se alteram após serem carregados no DW. Novos dados podem ser carregados de forma incremental;
- d) **Variável em relação ao tempo:** representam resultados em um determinado momento no tempo.

1.1.1 Data Marts

Conforme Singh (2001), *Data Mart* é um subconjunto do DW. Normalmente desempenha o papel de um DW departamental da empresa como, por exemplo, contabilidade, finanças, vendas e estoque, como demonstra a figura 1.2.

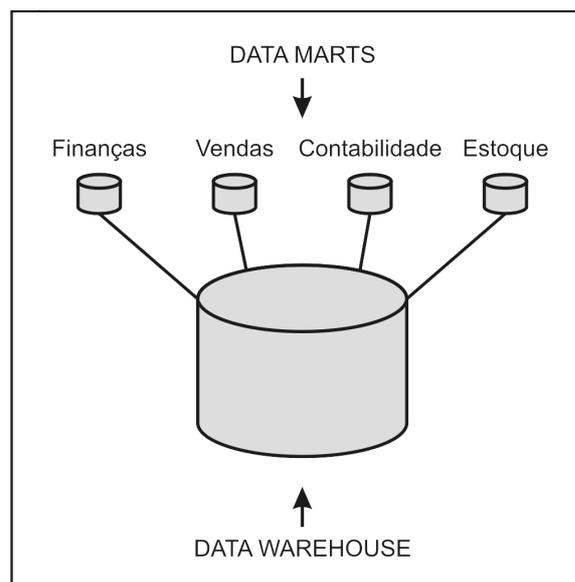


Figura 1.2 – *Data Marts* específicos

Fonte: Adaptado pelo autor segundo Inmon, Welch, Glassey (1999)

As estruturas de dados encontrado no DM, não obrigatoriamente, são menos granulares do que as encontradas no DW. A granularidade pode variar de acordo com a necessidade para a exploração dos dados. Os DM são moldados de acordo com os requisitos de cada departamento, como consequência, o projeto de cada DM é único. (INMON, WELCH, GLASSEY, 1999)

1.1.2 Granularidade

Um dos aspectos mais importantes da modelagem física de dados é a definição da granularidade. A granularidade dos dados no DW consiste no nível de sumarização dos dados e no nível de detalhe. Quanto mais detalhes, menor a granularidade, conseqüentemente, quanto menos nível de detalhe maior é a granularidade dos dados. (MACHADO, 2000)

Segundo Machado (2000), como consequência, quanto maior o volume de dados, menor a *performance* de consultas e relatórios. A correta seleção da granularidade terá um efeito significativo no volume de dados. Uma vez definida a granularidade, deve ser aplicada para todo o DW em que existir referência a esse assunto. As informações devem ser analisadas sempre pela mesma granularidade. A granularidade não se limita somente ao tempo, mas a todos os fatores de classificação da informação que estiverem sendo utilizados.

De acordo com Inmon (1997), a melhor opção para granularidade é a utilização de níveis duais de granularidade, que permite processamento de grandes quantidades de solicitações e atende a qualquer questão que precise ser respondida consultado um nível maior de detalhes. Para isso são criadas duas camadas, sendo uma camada para os dados levemente resumidos e outra para os dados históricos.

1.1.3 Arquitetura do DW / DM

A escolha da arquitetura é uma decisão importante no projeto que deve sempre que possível estar baseada nos fatores relativos à infra-estrutura disponível e ao porte da empresa. A abordagem da implementação escolhida pode causar impactos no sucesso do projeto de DW. Deve levar em consideração para a escolha o tempo para execução do projeto, o retorno do investimento, a satisfação do usuário, os recursos necessários e a velocidade da utilização das informações. (MACHADO, 2000)

Existem três tipos de arquiteturas: global, independente e integrada. A seguir a definição de cada uma das arquiteturas:

- a) **Arquitetura global:** conforme Machado (2000), a arquitetura global é considerada a que suporta toda ou a maior parte das necessidades de um *Data Warehouse* integrado com um grande grau de acesso e utilização dessas informações por todos os departamentos da empresa. O termo global reflete o escopo do acesso das informações, significando por toda a empresa. Essa arquitetura pode ser fisicamente centralizada ou fisicamente distribuída por diversos locais;
- b) **Arquitetura independente:** de acordo com Machado (2000) a arquitetura independente implica em DMs que atendem apenas as necessidades específicas de um determinado departamento, sem nenhum foco corporativo. Seu tempo de construção é menor, tendo como desvantagem seu isolamento perante os demais DMs;
- c) **Arquitetura integrada:** na arquitetura integrada os DMs são implementados separadamente por departamentos, mas possuem integração entre eles, proporcionando uma visão corporativa das informações. (MACHADO, 2000)

1.1.4 Tipos de implementação

Segundo Machado (2000), a escolha por um tipo de implementação é influenciada por diversos fatores, como a infra-estrutura de TI, arquitetura escolhida, recursos disponíveis, velocidade de implementação, investimento desejado e pela necessidade ou não do acesso corporativo dos dados. Os principais tipos de implementação do DW são:

- a) **Top Down:** nesse tipo de implementação, deve-se primeiro tomar decisões sobre as fontes de dados, segurança, estrutura, qualidade e padrões dos dados significando um trabalho inicial oneroso pelo planejamento e definições conceituais para a construção do DW. Após construído o DW os dados e metadados são extraídos para os DM. Esse tipo de implementação tem como vantagem a visão geral dos dados na empresa, facilidade de manutenção, repositório de dados centralizados. Porém, como desvantagem, requer um longo período de implementação;

- b) **Bottom-Up**: nessa abordagem primeiramente são construídos os DMs departamentais, que vão sendo gradativamente incrementados ao DW. Esse processo tem como vantagem uma rápida implementação e conseqüentemente um rápido retorno e a opção de escolher os principais departamentos da empresa para a criação dos DM. Tem como desvantagem a falta de padronização dos DM que pode vir a gerar redundância e inconsistência dos dados;
- c) **Implementação combinada**: essa forma de implementação tem o propósito de integrar as duas arquiteturas, realizando a modelagem dos dados do DW com uma visão macro e gerar os DMs a partir do macro modelo de dados do DW. Essa abordagem tem como vantagem a garantia da consistência dos dados, pelo fato de se utilizar um modelo de dados únicos para os DMs.

1.1.5 Diferenças entre ambiente operacional e o ambiente DW

De acordo com Singh (2001), há dois tipos básicos de dados em qualquer empresa. O primeiro chama-se dado operacional e o segundo de dado informativo. Os dados operacionais são aqueles que suportam as funções do negócio. Já os dados informativos suportam o processo de tomada de decisão.

Segundo Kimball (1998), o modelo relacional divide os dados em várias entidades distintas, sendo criada uma tabela no banco de dados para cada entidade. Isso gera um imenso número de conexões possíveis entre as tabelas através de junções internas entre os elementos de dados.

Conforme Barbieri (2001), a abordagem tradicional de dados desenvolvida ao longo dos anos sempre primou pela representação de estruturas que melhor se ajustassem as características transacionais. Porém, essa abordagem se demonstra inadequada para os processamentos demandados pela ótica dimensional.

O quadro 1.1 demonstra uma comparação de algumas características e as diferenças entre os dados operacionais e os dados informacionais.

Característica	Dado Operacional	Dado Informacional
Conteúdo	Valores correntes	Valores sumarizados, calculados, integrados de várias fontes
Organização dos dados	Por aplicação/sistema de informação	Por assuntos/negócios
Natureza dos dados	Dinâmica	Estática até atualização dos dados
Formato dos dados	Relacional	Dimensional
Atualização dos dados	Atualização campo a campo	Acesso, sem <i>update</i>
Uso	Altamente estruturado	Desestruturado
Tempo de resposta	Otimizado para segundos	Análises mais complexas, com tempo de respostas maiores

Quadro 1.1 – Comparação entre os dados de natureza operacional e informacional

Fonte: Barbieri (2001).

1.2 MODELAGEM DIMENSIONAL DOS DADOS

Conforme Barbieri (2001), a modelagem dimensional permite que o usuário visualize os dados de uma forma próxima do seu entendimento, permitindo várias perspectivas possíveis, como por exemplo, tempo e o espaço. As técnicas de modelagem dimensional foram criadas para modificar alguns conceitos tradicionais de Banco de Dados, principalmente do modelo relacional.

Segundo Machado (2000), a modelagem dimensional demonstra de uma forma simples para que os usuários finais facilmente entendam e naveguem pela solução de estrutura de dados construída. Desta forma, exigem um nível de abstração superior ao que é comum nos modelos relacionais.

A estrutura dimensional modifica a distribuição dos campos por entre as tabelas, o que permite uma formatação estrutural voltada para os vários pontos de entradas, também chamados de dimensões, e menos para os dados granulares, chamados de fatos. Com isso a estrutura dimensional dos dados se parece com uma forma estrelar, onde várias tabelas de entradas estão se relacionando com poucas tabelas de informações. Esse modelo oferece claramente os elementos necessários para buscar as informações sobre fatos via dimensões de referências. A figura 1.3 demonstra a tabela fato vendas e as tabelas dimensões loja e produto. (BARBIERI, 2001)

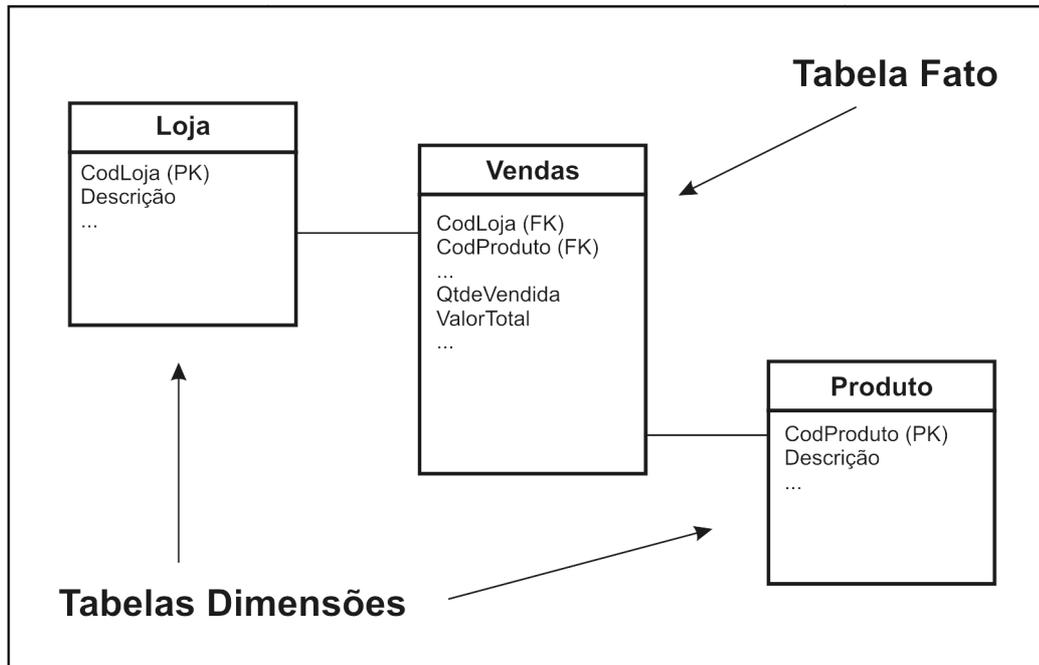


Figura 1.3 – Tabela fato e tabelas dimensões
 Fonte: Adaptado pelo autor segundo Barbieri (2001)

1.2.1 Tabela fato

De acordo com Barbieri (2001), as tabelas fato armazenam medidas numéricas associadas a eventos do negócio. A tabela fato contém vários fatos e cada um corresponde a uma linha, sendo que cada fato pode armazenar uma ou mais medidas numéricas. Possui como chave primária uma junção de várias chaves que são as chaves primárias das dimensões que com ela se relacionam. Normalmente armazenam muitos dados e merecem um cuidado especial em função do seu alto volume.

Segundo Machado (2000), fato é uma coleção de dados, composto de dados com medidas. Cada fato representa um evento de negócio, sendo utilizado para analisar o processo de negócio de uma empresa. Pode ser tudo aquilo que reflete a evolução do negócio.

1.2.2 Tabela dimensão

As tabelas dimensões representam entidades de negócios para armazenar informações como tempo, cliente, produto, geografia, etc. As tabelas dimensões tem relação um para muitos (1:N) com a tabela fato e possuem um número menor de linhas se comparada com tabelas fato. Possuem múltiplas colunas de informações, das quais algumas representam a sua própria hierarquia. (BARBIERI, 2001)

De acordo com Kimball (1998), as tabelas dimensionais armazenam as descrições das dimensões envolvidas no negócio. Para ele, os melhores atributos que devem ser selecionados para compor as tabelas dimensões são os textuais como, por exemplo, uma descrição sucinta ou longa para facilitar o entendimento do usuário.

1.2.3 Modelo Star

De acordo com Singh (2001), a principal característica do Modelo *Star*, também chamado de Modelo Estrela, são suas dimensões desnormalizadas. As vantagens na utilização desse modelo são a redução de *joins* físicos entre as tabelas dimensões no momento da consulta, melhorando muito o desempenho e a simplicidade da visualização do modelo de dados, permitindo a expansão e evolução do DW com pouca manutenção.

Para Machado (2000), o Modelo *Star* é a estrutura básica de um modelo de dados dimensional. Sua composição possui uma grande entidade central chamada de fato e um conjunto de entidades menores chamadas de dimensões, organizadas ao redor da tabela fato formando uma estrela, como se pode verificar na figura 1.4. O relacionamento entre a entidade fato e as dimensões é uma ligação entre as duas entidades com um relacionamento de um para muitos no sentido da dimensão para o fato.

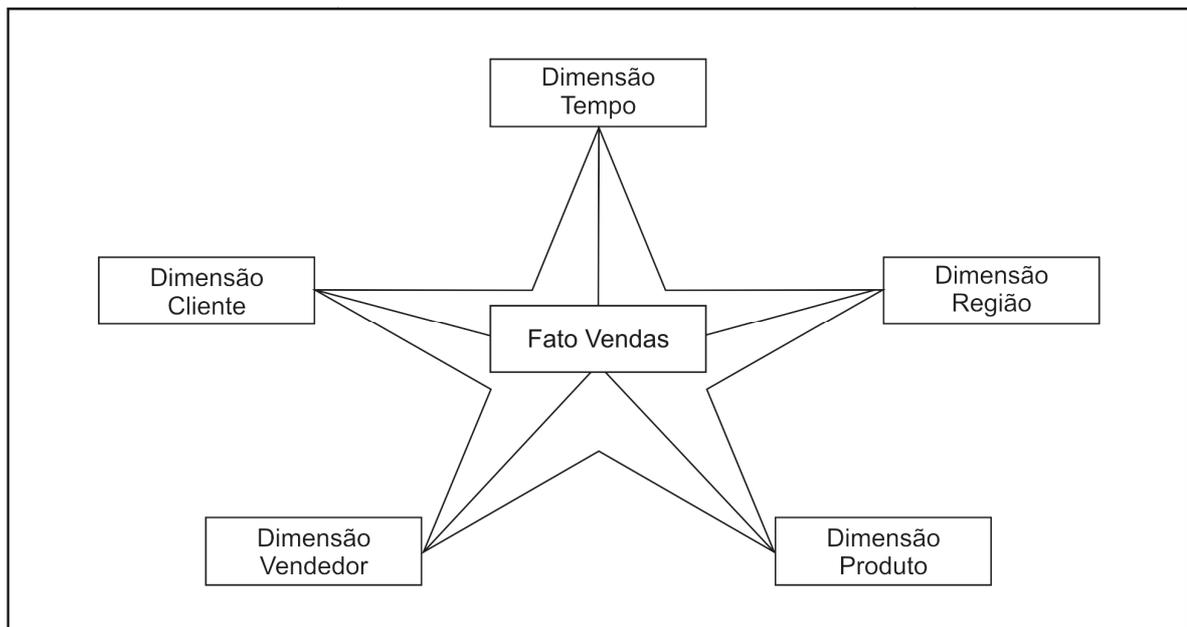


Figura 1.4 – *Modelo Star*

Fonte: Adaptado pelo autor segundo Machado (2000)

1.2.4 Modelo Snowflake

O Modelo *Snowflake*, também chamado de Modelo Floco de Neve, é o resultado da terceira forma normal sobre as tabelas dimensões do Modelo *Star*. Cada tabela dimensional armazena uma chave para cada elemento da dimensão, formando uma hierarquia. Tem como vantagem a normalização, evitando a redundância de valores textuais na tabela dimensão. (MACHADO, 2000)

Esse modelo tem como desvantagem sua complexidade da estrutura de dados normalizada, dificultando os usuários na navegação das consultas e os programas de carga e manutenção podem ficar mais difíceis de administrar de acordo com o aumento da complexidade. (SINGH, 2001)

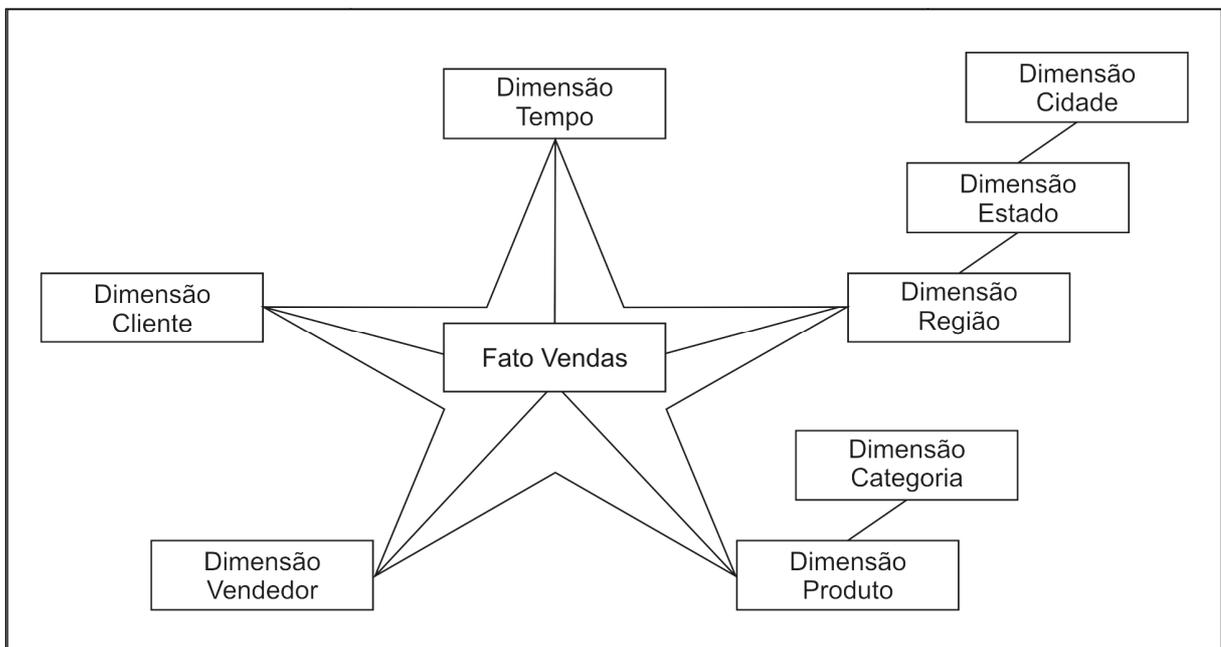


Figura 1.4 – *Modelo Snowflake*
 Fonte: Adaptado pelo autor segundo Machado (2000)

1.2.5 Metadados

Conforme Singh (2001), o metadado é um componente importante para o DW, pois contém informações sobre os dados armazenados. O metadado é utilizado para identificar quando ele foi criado, qual sua origem, a quem ele pertence e quem o administra, ou seja, são todas as informações relacionadas ao conteúdo dos dados permitindo entender os dados.

É pelo metadado que usuário final entende e localiza os dados do DW. O metadado também fornece informações sobre a estrutura dos dados, algoritmos utilizados para sumarização, histórico da extração e transformação do dado, estatísticas de uso,

identificadores de coluna. Ele é responsável por integrar a aplicação ao DW, permitindo que mudanças feitas no DW sejam automaticamente visualizadas na aplicação do usuário final. (SINGH, 2001)

Kimball (1998) considera o metadado como um termo vago para qualquer forma de dado auxiliar mantido. Conforme Inmon (1997), o metadado também é importante para um acompanhamento das alterações das estruturas dos dados ao longo do tempo, pois a estrutura do DW tende a sofrer alterações com o decorrer do tempo.

Inmon, Welch, Glassey (1999) consideram o metadado como um componente que compõe o centro nervoso do DW, pois sem ele os dados do DW são componentes avulsos funcionando independentemente e com objetivos separados. Os metadados são importantes para compartilhar o conhecimento criado pelo projetista, desenvolvedor e o usuário final, para que eles possam alcançar a continuidade do pensamento. Os metadados também precisam armazenar informações sobre o contexto da informação para que seja possível compreender e interpretar a informação.

1.3 EXTRAÇÃO, TRANSFORMAÇÃO E CARGA (ETL)

De acordo com Barbieri (2001), o ETL é responsável pela coleta, limpeza, preparação e carga dos dados operacionais para o DW. Esse processo tem como objetivo filtrar os dados para eliminar dados indesejáveis, integrar os dados de diferentes fontes distintas em uma única fonte e reduzir o volume de dados visando obter informações resumidas e sumarizadas, bem como a derivação de dados através de fórmulas para produzir dados virtuais a partir de dados já existentes.

Nessa camada de integração dos dados vindos do ambiente operacional para o DW, eles passam por um processo de transformação, onde são submetidos a uma conversão, que podem ser padronizações de unidades de medida, padronização de domínios e tipos de dados, como se pode verificar na figura 1.5. (INMON, TERDEMAN, IMHOFF, 2001)

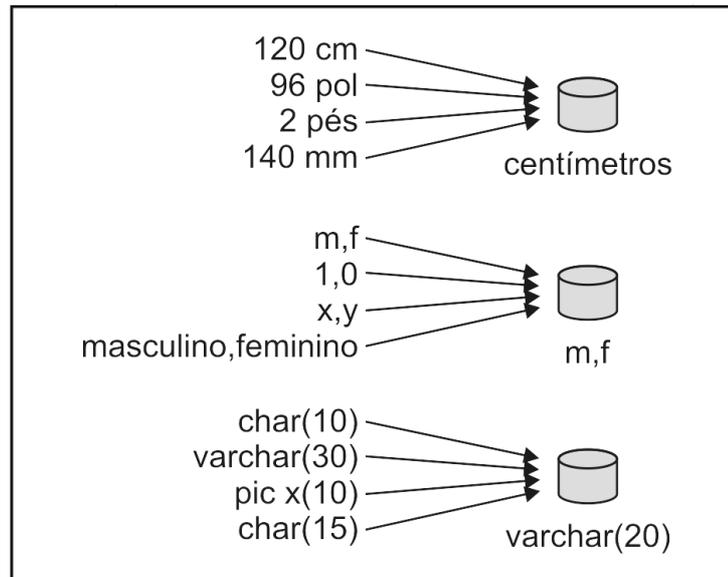


Figura 1.5 – *Transformação dos dados*

Fonte: Adaptado pelo autor segundo Inmon, Terdeman, Imhoff (2001)

Conforme Inmon (1997) existem três tipos de carga que podem ser realizadas do ambiente operacional para o DW: o carregamento de dados históricos, o carregamento de dados de valor corrente no ambiente operacional e o carregamento de alteração no DW a partir de atualizações que tenham ocorrido desde a última atualização do DW.

Inmon (1997) também indica algumas das funcionalidades necessárias para que ocorra a transformação e integração dos dados como:

- **Limpeza dos dados:** os dados devem passar por um algoritmo simples nos dados de entrada e em algumas situações podem ser aplicados algoritmos complexos com rotinas de inteligência artificial para limpar os dados de entrada e tornar uma saída aceitável;
- **Várias fontes de dados:** deve ser capaz de interpretar qualquer fonte de dados, sejam eles diferentes bancos de dados ou formatos de arquivos;
- **Valores *default*:** em algumas condições pode não existir o dado na entrada, sendo necessária a definição de um valor padrão para essas condições;
- **Resumir os dados:** vários registros de dados vindos do ambiente operacional são combinados em um único registro no DW.

1.4 ON-LINE ANALYTIC PROCESSING (OLAP)

De acordo com Barbieri (2001) o termo OLAP (*On-Line Analytical Processing*) representa a característica de trabalhar com dados e com operadores dimensionais possibilitando uma forma múltipla e combinada de análises, que permite ao usuário analisar o porquê dos resultados obtidos.

Para Singh (2001) as ferramentas OLAP fornecem subsídios para a análise complexa do negócio, possibilitando aos usuários analisar e navegar pelos dados para identificar tendências, exceções e obter alguns detalhes para entender melhor os resultados do seu negócio.

1.4.1 Operações OLAP

Conforme Barbieri (2001), Machado (2000), Singh(2001), Inmon,Welch e Glassey (1999) as ferramentas OLAP tem operações para analisar os dados, sendo que as principais são:

- ***Drill-down***: permite a movimentação da visão dos dados ao longo dos níveis hierárquicos de uma dimensão, permitindo navegação do nível mais alto até o dado detalhado;
- ***Drill-up***: tem o mesmo princípio da operação *drill-down*, mas permite a navegação inversa dos dados, ou seja, navega do nível de dados mais detalhado até o nível mais alto;
- ***Drill-across***: permite a navegação da tabela dimensão passando de um nível para outro sem passar pelos níveis intermediários. Por exemplo, tabela dimensão tempo, que possui os campos ano, semestre, trimestre, mês, dia, passar da informação ano diretamente para dia;
- ***Drill-through***: é quando há necessidade de uma informação em um nível de detalhe menor do que aquele armazenado na tabela fato, ou seja, é a operação que busca a informação além do nível de granularidade existente na estrutura dimensional;
- ***Slice-dice***: é a operação utilizada para acessar os dados do DW utilizando qualquer uma das dimensões de forma equivalente. Pode modificar a posição de

uma informação, alterar linhas por colunas, modificar a dimensão, para facilitar a compreensão dos usuários;

- **Pivoting:** é a mudança das linhas e colunas da consulta, rotacionando as mesmas a fim de se obter uma nova visão das informações.

1.4.2 Arquiteturas OLAP

De acordo com Barbieri (2001) a arquitetura OLAP está relacionada com a estratégia de armazenamento e implementação do DW/DM, conforme a figura 1.6.

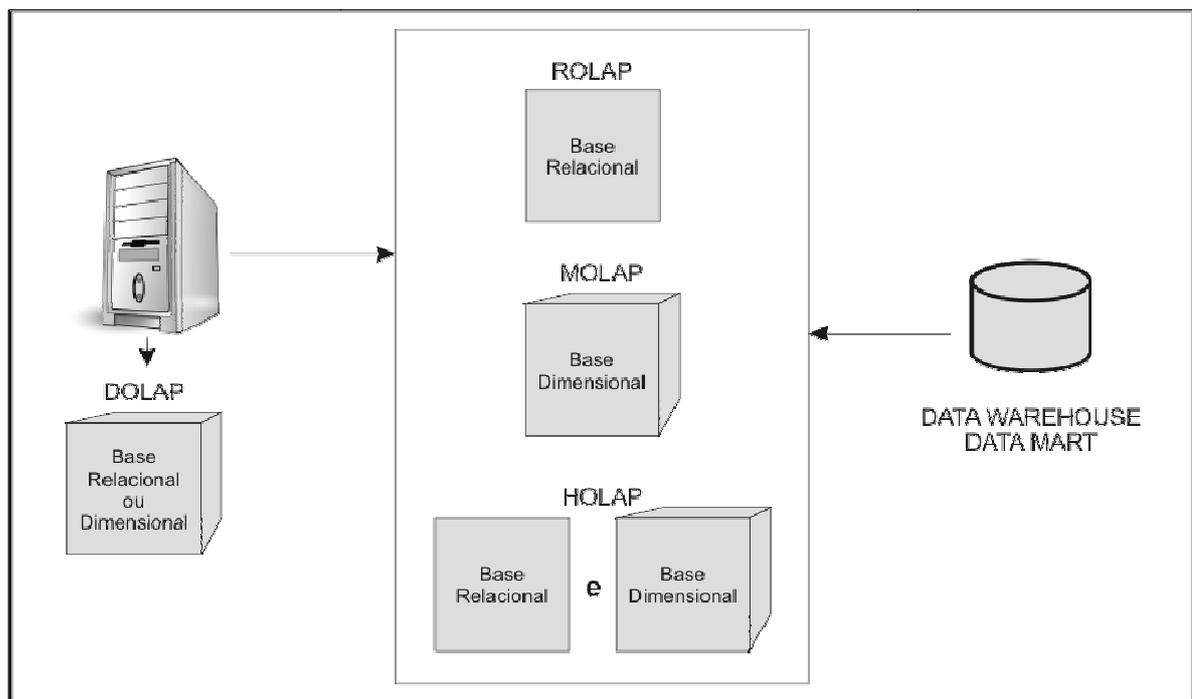


Figura 1.6 – Opções de armazenamento/implementação OLAP

Fonte: Adaptado pelo autor segundo Barbieri (2001)

A implementação ROLAP utiliza os próprios SGBDR (Sistemas de Gerência de Banco de Dados Relacionais) com as tabelas construídas em estruturas relacionais clássicas. Oferece toda vantagem de um SGBDR sendo necessário um cuidado com excesso de tabelas normalizadas, que poderá comprometer a *performance* das análises.

Na MOLAP são usados os gerenciadores de banco de dados proprietários, que possuem características de armazenamento e ferramentas para tratamento dimensional dos dados.

A HOLAP é uma opção híbrida do uso das estratégias de ROLAP e MOLAP, onde as estruturas relacionais são normalmente utilizadas para dados com maior granularidade e as estruturas dimensionais são dedicadas ao armazenamento de menor granularidade.

Já a implementação DOLAP, utiliza uma abordagem onde as estruturas, tanto dimensionais quanto relacionais, são transferidas para as estações clientes, com objetivo de facilitar a *performance* de algumas análises, diminuindo o tráfego de informações entre o servidor/cliente.

Os estudos realizados permitiram identificar a estrutura e funcionamento de uma solução de BI, desde os sistemas que irão originar os dados, passando pelas estruturas de armazenamento utilizadas, o processo de ETL até chegar à utilização de uma ferramenta OLAP que irá permitir a visualização e análise dos dados. Estes conceitos são fundamentais para o projeto da solução proposta neste trabalho, de criar uma solução de BI para o ERP SIGER.

2 ERP SIGER

O SIGER (Sistema Integrado de Gestão Empresarial Rech), desenvolvido pela Rech Informática Ltda, é um sistema ERP completo, com módulos totalmente integrados, que atende às necessidades de diversos setores, eficaz na otimização dos processos e sincronização das informações.

2.1 RECH INFORMÁTICA

A Rech Informática, empresa sediada em Novo Hamburgo, Rio Grande do Sul, vem atuando desde 1990 no mercado de software de gestão empresarial. Foi constituída a partir de um alicerce sólido de valores humanos, conjugando experiência, seriedade e competência nas áreas de informática e gestão empresarial. Fundada em 15 de maio de 1990, pelos irmãos Carlos Vanderlei Rech e Rovani Marcelo Rech, que desde o início, estiveram comprometidos em oferecer ao mercado produtos e serviços de qualidade.

A Rech possui uma sede própria com aproximadamente 1.200 m² de área construída, em um terreno de mais de 3.200 m², em pleno contato com a natureza. Conta com amplo estacionamento, podendo abrigar com total segurança, auditório para palestras, conferências e seminários, centro de treinamento e salão de festas. Todo ambiente operacional é de alta qualidade com tecnologia Dell.

O principal objetivo da Rech Informática é a conquista diária da satisfação de seus clientes. Por isto, a empresa vem efetuando investimentos em novas tecnologias e no treinamento de pessoal, aprimorando constantemente o SIGER e a qualidade dos serviços prestados.

Com o total comprometimento em atingir este objetivo, a Rech Informática reinveste todo o seu lucro na própria empresa. Estes investimentos são tanto em estrutura física, quanto em recursos de pessoal, sempre buscando melhorar cada vez mais o relacionamento da empresa com seus clientes, fornecedores e colaboradores.

A empresa possui clientes nos estados do Rio Grande do Sul, Santa Catarina, Paraná, São Paulo, Minas Gerais, Rio de Janeiro, Espírito Santo, Bahia, Pernambuco, Paraíba, Ceará, Mato Grosso, Distrito Federal e Goiás, atuando mais especificamente no Vale dos Sinos e região metropolitana de Porto Alegre. Atualmente conta com cerca de 600 clientes em diversas atividades de negócios, totalizando mais de 5.000 usuários que utilizam o SIGER no seu dia-a-dia. A figura 2.1 demonstra a distribuição da carteira de clientes por atividade.

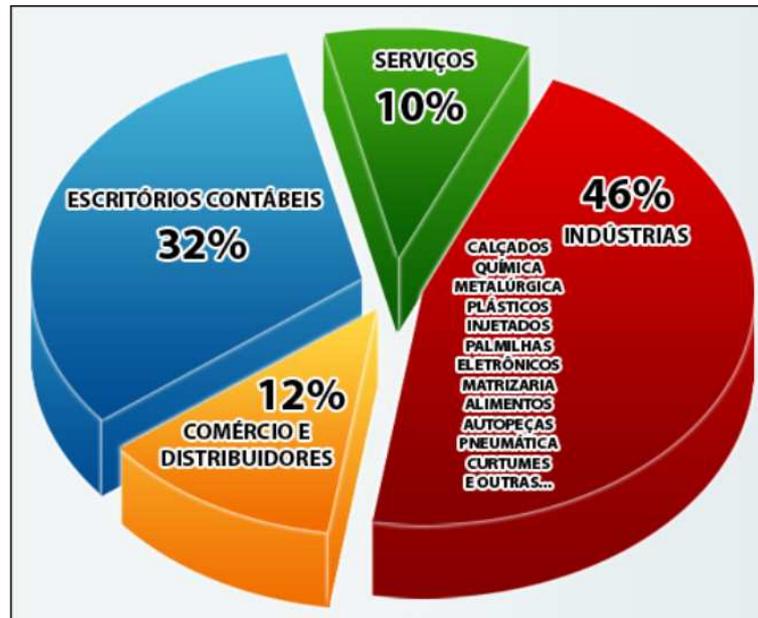


Figura 2.1 – *Distribuição da carteira de clientes por atividade*
 Fonte: Rech Informática (2010)

A empresa tem como missão fornecer produtos e serviços de alta qualidade contribuindo para que clientes e colaboradores atinjam seus objetivos estratégicos, explorando ao máximo seus potenciais e otimizando seus processos através da integração de tecnologia e sistema de informações. Sua visão é ser referência no estado do Rio Grande do Sul até o ano de 2015 em soluções de sistema de informações para gestão empresarial. Tem como princípios:

- **Satisfação:** superar as expectativas dos clientes com soluções de qualidade;
- **Tecnologia:** utilizar tecnologias confiáveis, viáveis e produtivas;
- **Recursos humanos:** valorização, espírito de equipe e aprimoramento contínuo;
- **Valores:** honestidade, humildade, seriedade, transparência, crescimento sustentado, solidez, habilidade e competência;
- **Qualidade:** foco na qualidade total como receita para o sucesso;
- **Ética:** ser ético e justo em todos os relacionamentos internos e externos;
- **Dinamismo:** Flexibilidade e agilidade no atendimento e soluções.

Atualmente, a estrutura organizacional da empresa está dividida em cinco grandes áreas (Direção, Controladoria, Suporte, Consultoria e Desenvolvimento) sendo a direção responsável pela estratégia, visão e administração da Rech Informática e a controladoria

responsável pelo monitoramento funcional e operacional da empresa e seu relacionamento com o mercado.

O suporte é responsável pelo canal direto e aberto permanentemente nos relacionamentos com os clientes e mercado e a manutenção do uso do sistema. A consultoria é responsável por reconhecer o processo de gestão das empresas e implantar o ERP. O desenvolvimento é a divisão responsável pelo planejamento e execução de projetos de desenvolvimento do sistema, relacionando demanda e novas tecnologias.

2.2 SIGER

O SIGER é composto por módulos, os quais podem ser adquiridos separadamente ou em conjunto. Isto permite que cada cliente monte a solução juntamente com a equipe de consultoria da Rech, de forma que atenda completamente sua gestão sem ter que adquirir partes do software que não lhe interessam.

O software é desenvolvido pela Rech Informática, na sede da empresa. O ERP SIGER não é vendido como um produto onde normalmente existe um pesado investimento inicial. O sistema é contratado com uma mensalidade para utilização do sistema, incluindo suporte técnico e atualizações de versões. Todo esse investimento é feito pela Rech, a qual aposta em uma parceira de longo prazo com seus clientes.

A ferramenta ERP é desenvolvida utilizando Net Express, desenvolvido pela empresa Micro Focus Corporation e possui alguns recursos extras desenvolvidos em Delphi 7, da Borland. A atual base de dados utilizada pelo sistema é a própria base nativa do Net Express, com arquivos indexados.

Está em desenvolvimento uma nova versão do ERP, que em breve estará disponível no mercado. Nesta há todas as funcionalidades existentes na atual versão, porém com novos recursos. Um destaque tecnológico e importante para o desenvolvimento deste trabalho, é a migração de toda a base de dados nativa para um sistema gerenciador de banco de dados, que nesta primeira versão do ERP será o Oracle. No entanto, está previsto para futuras versões a utilização do Microsoft SQL Server e outros SGBDs.

O ERP SIGER possui estrutura modular, sendo que alguns dos módulos são: Contabilidade, Gestão Patrimonial, Folha de Pagamento, Livros Fiscais, Faturamento/Vendas, Controle Financeiro, Controle de Estoque, Controle de Compras, Gestão Industrial, PCP para

Calçados, Automação de Escritórios, Ponto de Venda, Gestão de Serviços, Telemarketing e Comércio Exterior, como demonstra a figura 2.2.



Figura 2.2 – Estrutura do SIGER

Fonte: Rech Informática (2010)

Essa estrutura modular permite que o sistema seja instalado e configurado para atender diversas situações de acordo com a necessidade do cliente. Essa mesma estrutura pode ser utilizada para a construção dos DMs para que os gestores de cada área possam realizar suas consultas e análises.

O SIGER pode ser utilizado em diversas áreas de negócios como escritórios contábeis, indústrias de calçados, químicas, metalúrgicas, de plásticos, embalagens, injetados, palmilhas, componentes eletrônicos, moldes/matrizes, alimentos, autopeças, pneumática, curtumes, comércio, distribuidores e serviços em geral. Esses diferentes ramos de negócios exigem demandas decisórias diferentes, o que valoriza ainda mais a solução proposta.

O sistema ainda tem recurso de Gerador de Relatórios que possibilita aos usuários, além de utilizar os relatórios já disponíveis nos diversos módulos do sistema, elaborar seus próprios relatórios com informações específicas. Os campos a serem impressos nos relatórios

são provenientes das tabelas existentes no sistema, podendo haver ligação de uma tabela com outra, ou seja, fazer a criação de sub-relatórios, de modo a atender às necessidades de acompanhamento e controle necessários.

O BI é um recurso inexistente no ERP SIGER, e quando estiver disponível, agregará recursos importantes para auxiliar o processo decisório dos clientes, com mais flexibilidade e interatividade.

3 ANÁLISES DE SOLUÇÕES DE BI

Conforme indicado anteriormente, o objetivo deste trabalho contempla a realização de análises de algumas ferramentas de BI como o Pentaho e o SADIG, para avaliar as suas funcionalidades e realizar testes, a fim de escolher ou não determinadas ferramentas para utilizar em conjunto com a solução proposta.

Ainda neste sentido, também foram realizados testes com as ferramentas ETL como o Kettle que faz parte do projeto Pentaho e o SQL Server Integration Services (SSIS) que faz parte do pacote Microsoft SQL Server, visando identificar a sua aplicabilidade no processo de ETL da solução proposta.

Todos os testes foram realizados utilizando um DM de vendas construído para testar as funcionalidades, tanto das ferramentas de análise como do processo de ETL. Somente para a ferramenta SADIG foi utilizado o próprio exemplo distribuído na versão SADIG Free, pois essa versão não permite a construção de outros DMs para análise.

3.1 SOLUÇÕES DE BI

Entre as várias soluções de BI existentes no mercado, o Pentaho e o SADIG BI foram escolhidas para a análise, a fim de identificar determinadas funcionalidades, que vão servir como base para a construção da ferramenta.

3.1.1 Pentaho

A ferramenta Pentaho é uma plataforma de BI completa, *open source*, com grande flexibilidade e que funciona em diversas plataformas. A solução oferece ferramentas de relatórios, análises, medidores de indicadores (*dashboards*), integração de dados e mineração de dados (*Data Mining*), como mostra a figura 3.1 que ilustra as ferramentas disponíveis.

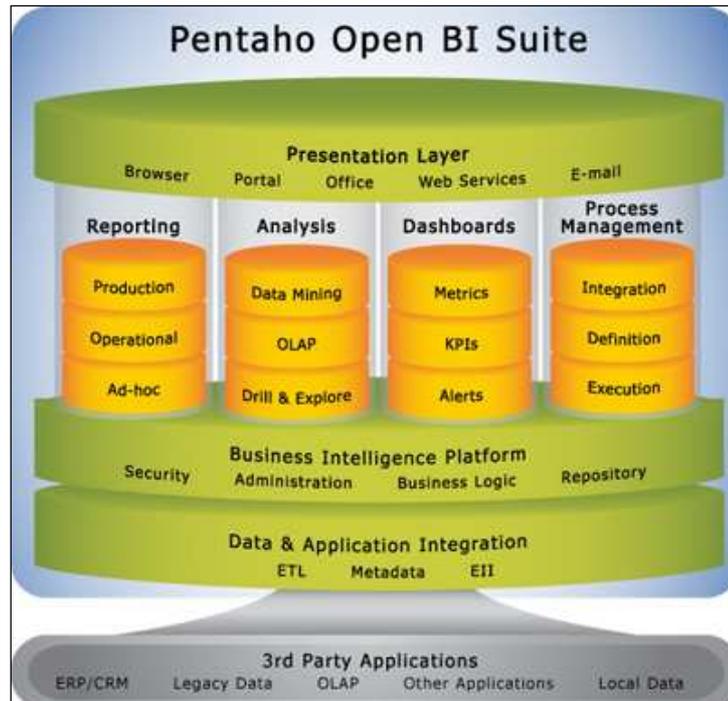


Figura 3.1 – Componente da suíte Pentaho BI
 Fonte: Pentaho (2010)

As principais ferramentas Pentaho são:

- **Pentaho Reporting:** é ferramenta para criação de relatórios e gráficos que podem ser visualizados pelo navegador e podem ser salvos em diversos formatos de saída, como por exemplo, PDF, HTML, Excel, RTF;
- **Pentaho Analysis:** é uma ferramenta analítica para exploração dos dados utilizando o navegador, que fornece todas as operações OLAP (*drill-down, drill-up, slice-dice*, etc);
- **Pentaho Dashboards:** ferramenta que permite uma visão do desempenho da empresa em forma de métricas e indicadores, com uma interface de fácil entendimento e totalmente visual;
- **Pentaho Data Integration:** é a ferramenta de ETL responsável pela extração, transformação e carga dos dados. Permite construir o fluxo das informações que serão carregadas no repositório de maneira simples utilizando *workflow*;
- **Pentaho Data Mining:** é a ferramenta responsável pela mineração dos dados, ou seja, pela exploração dos dados para localizar padrões e relacionamentos.

As ferramentas podem ser instaladas de acordo com a necessidade, atendendo empresas de médio e grande porte. Por ser uma ferramenta *open source*, está em constante evolução pela comunidade desenvolvedora, agregando novas funcionalidades e melhorias.

A instalação da ferramenta é simples, desde que atendendo alguns pré-requisitos de hardware e principalmente de software, como a versão da máquina virtual Java, necessária para execução da aplicação. A configuração da solução é mais complexa, necessitando alguns conhecimentos avançados na área de BI. Existe uma grande quantidade de documentação disponível no próprio site da ferramenta, bem como documentações produzidas por usuários avançados da ferramenta, formando uma grande base de conhecimento.

É possível citar alguns casos de sucesso de empresas que utilizam a ferramenta como a Sun Microsystems, MySQL AB, Mozilla, Swissport, Sheetz, National Health Service Islington, dentre outras. (PENTAHO, 2010)

Na figura 3.2 pode-se visualizar a GUI do Pentaho. Seu acesso pode ser feito através de qualquer navegador de Internet, facilitando sua utilização sem a necessidade de qualquer instalação nas máquinas clientes.

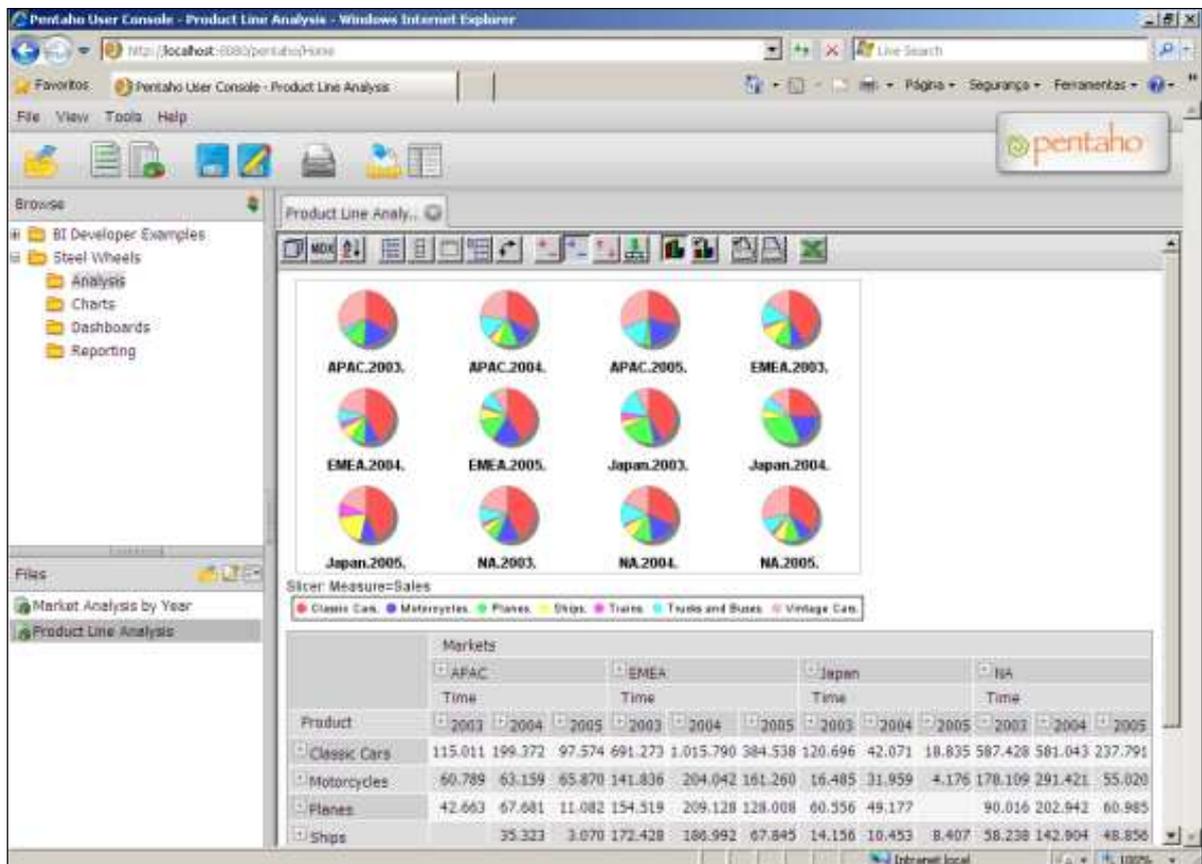


Figura 3.2 – A GUI do Pentaho

Fonte: Do próprio autor

A construção de um projeto no Pentaho é um processo complexo que necessita de conhecimentos técnicos em todas as ferramentas necessárias para a construção. Primeiro é necessário modelar o DM utilizando uma ferramenta para criação do modelo multidimensional ou fazer diretamente o modelo em linguagem SQL. Após a definição do modelo e construção no SGBD, é necessário utilizar a ferramenta Kettle para o processo de ETL desse DM. A última etapa, antes da visualização dos cubos de dados para consultas e análises, é criar o *schema* para mapear o DM para o cubo de dados. Para facilitar essa criação o Pentaho disponibiliza a ferramenta Schema Workbench. A análise e visualização é feita pelo Pentaho Analysis.

3.1.2 SADIG BI

A SADIG é uma empresa brasileira, com sede em Montenegro, Rio Grande do Sul, que atua na área de sistemas de informação desde 1986 e é reconhecida como especialista em Inteligência Estratégia e BI. Possui em torno de 600 empresas clientes e seis mil usuários distribuídos em 16 estados do Brasil e no exterior. A empresa conta com clientes como: Calçados Beira Rio, Lojas Gang, Grupo Vigor, Seven Boys, Royal Canin. (SADIG, 2010)

Possui como produtos o SADIG Análises, SADIG Painéis, SADIG Mobile, SADIG Performance, SADIG ETL e conta com uma solução *freeware* com recursos limitados, o SADIG Free. A seguir uma explicação mais detalhada de cada um desses produtos:

- **SADIG Análises:** é responsável pela integração e consolidação dos dados originados de diversos sistemas, permitindo gerar e distribuir informações, como análises de evolução temporal, *rankings*, curvas ABC, classificações e demais operações padrões OLAP;
- **SADIG Painéis:** permite que os usuários criem painéis de controle com gráficos, tabelas e sinalizadores permitindo que os gestores consigam identificar informações importantes para os processos decisórios;
- **SADIG Mobile:** permite o acesso das informações através de aparelhos celulares, e *smartphones*, em tempo real;
- **SADIG Performance:** Permite a criação de indicadores, metas, diagramas de causa e efeito e alertas para que os gestores possam avaliar o desempenho empresarial;

- **SADIG ETL:** é a ferramenta para extração, transformação e carga dos dados de forma simples e intuitiva. Essa solução foi construída em conjunto com outra empresa chamada Idealogic Software;
- **SADIG Free:** é uma versão gratuita da ferramenta com várias limitações.

Todas as ferramentas podem ser adquiridas separadamente de acordo com as necessidades da empresa. Foi utilizado o SADIG Free para a análise por se tratar de uma solução gratuita. A instalação é simples, guiada por um *wizard*. Possui um manual completo da ferramenta e principais recursos.

Com o SADIG é possível construir o repositório dentro da própria ferramenta. Entretanto, na versão SADIG Free esse recurso não está liberado. A figura 3.3 demonstra como é feita essa configuração e os campos contidos na tabela de exemplo desta versão.



Figura 3.3 – A criação do repositório do SADIG Free

Fonte: Do próprio autor

O teste realizado com o SADIG Free foi parcial por questões comerciais, pois nessa versão não é permitida a construção de novos DMs. Esta limitação acabou impedindo o êxito dos testes.

3.2 FERRAMENTAS ETL

Entre as ferramentas ETL existentes no mercado, o Kettle e o SSIS foram escolhidos para esta análise, a fim de identificar a melhor ferramenta para agregar em conjunto com a solução desenvolvida.

3.2.1 Kettle

O Kettle, também chamado de Pentaho Data Integration (PDI), é um dos componentes da suíte Pentaho sendo responsável pelo processo de ETL. A ferramenta pode ser utilizada tanto como parte do processo de BI, como para outras finalidades como migração de dados entre servidores, exportação de dados e integração de aplicações.

A instalação da ferramenta é simples, sendo necessário apenas descompactar o arquivo disponível no site da Pentaho. Para sua execução é necessário apenas iniciar script de inicialização “spoon.bat” para Windows e “spoon.sh” para Linux. A ferramenta é *open source* e funciona em plataformas Windows e Linux.

O Kettle possui uma ferramenta gráfica chamada de Spoon. Existe outro aplicativo chamado Pan, que é utilizado para executar transformações (*transformations*) criadas através de linha de comando. O outro aplicado é o Kitchen, também por linha de comando, destinado para executar os trabalhos (*jobs*) criados.

Com o Kettle é possível importar dados de diversas fontes, como por exemplo: arquivos CVS, arquivos texto, tabelas de diversos bancos de dados através de uma conexão JDBC (Oracle, SQL Server, PostgreSQL, MySQL, Sybase, DB2, Firebird, etc), XML, planilhas eletrônicas, base de dados Access, entre outros.

Sua utilização é simples e intuitiva, pois utiliza uma GUI. Nela é possível selecionar as operações de entrada, saída, transformação, entre outras, e arrastá-las para posteriormente criar a ligação entre as operações formando o fluxo em que a informação deve fluir, conforme a figura 3.4.

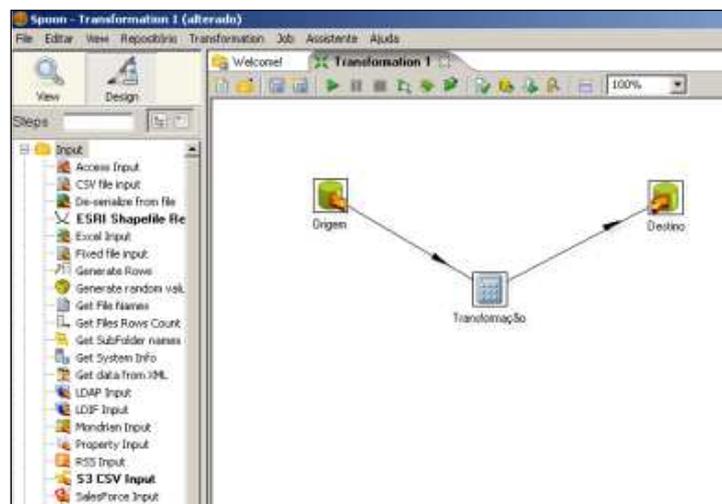


Figura 3.4 – A GUI do Kettle

Fonte: Do próprio autor

A ferramenta trabalha com dois tipos de modelagem: *jobs* e *transformations*. As *transformations* capturam o fluxo de dados de entrada e realizam uma transformação até chegar ao fluxo de dados de destino. Os *jobs* tem entradas como por exemplo, acesso a arquivo em FTP e como saída pode ser utilizado envio de e-mails. Os *jobs* também podem ser utilizados chamando outras transformações criadas e permitem realizar todas as importações de uma maneira sincronizada e de uma única vez, evitando realizar importações para cada transformação. A figura 3.5 demonstra essa junção.

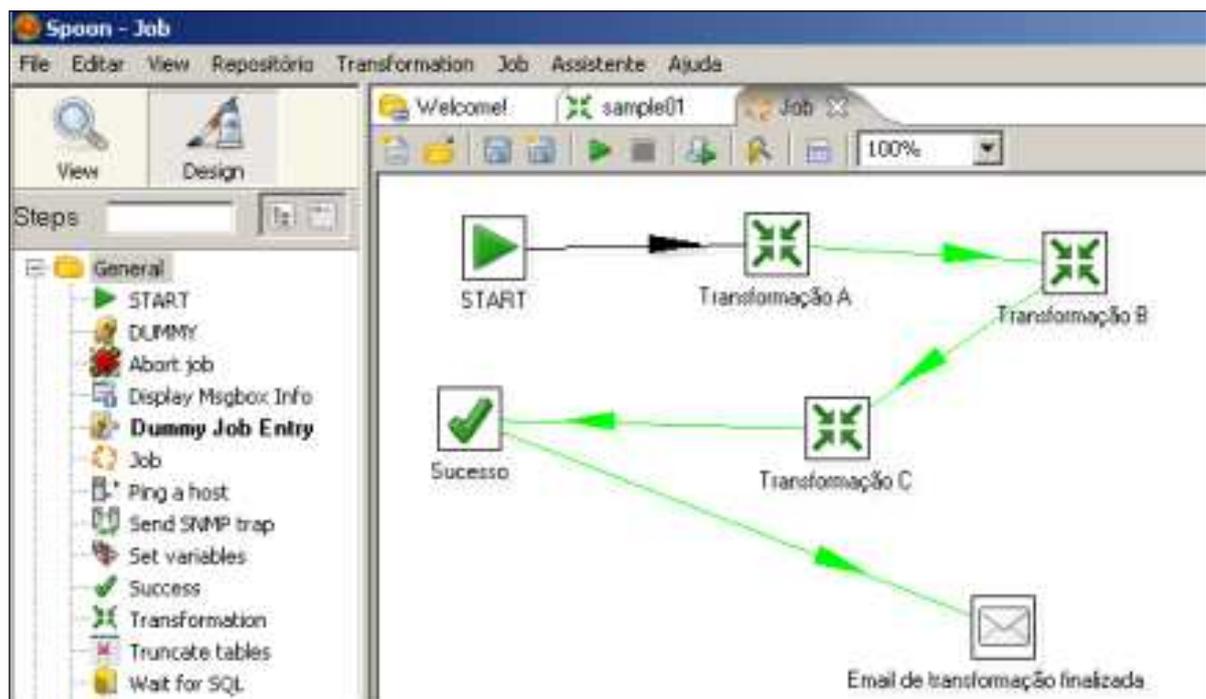


Figura 3.5 – Junção dos *jobs* com *transformations* em uma única tarefa

Fonte: Do próprio autor

Outros conceitos importantes sobre a ferramenta são *steps* e *hops*. O *step* é considerado uma unidade mínima do processo. Cada processo é um conjunto de *steps* programados para executar uma tarefa específica, seja uma leitura ou transformação de algum dado. A ligação entre esses *steps* é chamada de *hop*. Ele é representado de forma gráfica demonstrando o fluxo dos dados. Outro conceito é que um *hop* possui apenas uma origem e um destino, mas mais de um *hop* pode sair de um *step* ou entrar em um *step*, ou seja, a saída de dados pode ser copiada ou distribuída em várias ou fundida em uma única entrada. (PENTAHO, 2010)

Cada *step* de acordo com seu tipo e finalidade possui diversas configurações que podem ser realizadas, o que deixa a ferramenta bastante flexível com a customização.

3.2.2 SQL Server Integration Services (SSIS)

O SQL Server Integration Services (SSIS) é uma evolução da antiga ferramenta disponibilizada no SQL Server 2000 chamada Data Transformation Services (DTS). O DTS se apresentava na forma de menus, caixa de ferramentas de conexões e tarefas e uma única camada de *design* de pacotes. O novo SSIS ainda conta com todos esses elementos, mas a camada de *design* possui várias janelas com guias e novos recursos como Gerenciador de Soluções e Eventos de Log. (MICROSOFT, 2010)

A instalação da ferramenta é feita em conjunto com a instalação do SQL Server 2005 ou SQL Server 2008. Para os testes realizados nesse trabalho foi utilizada a versão do SQL Server 2005. A instalação é guiada por um assistente (*wizard*), sem configurações complicadas.

O SSIS é um pacote integrado da ferramenta Microsoft Visual Studio. Utiliza o recurso de *drag-and-drop* o que torna a ferramenta fácil de ser utilizada, sem a necessidade de programação via código. Porém, a ferramenta disponibiliza um ambiente de *script* para essa finalidade. A figura 3.6 demonstra a GUI do SSIS e alguns de seus componentes para trabalhar com a carga e transformação dos dados.

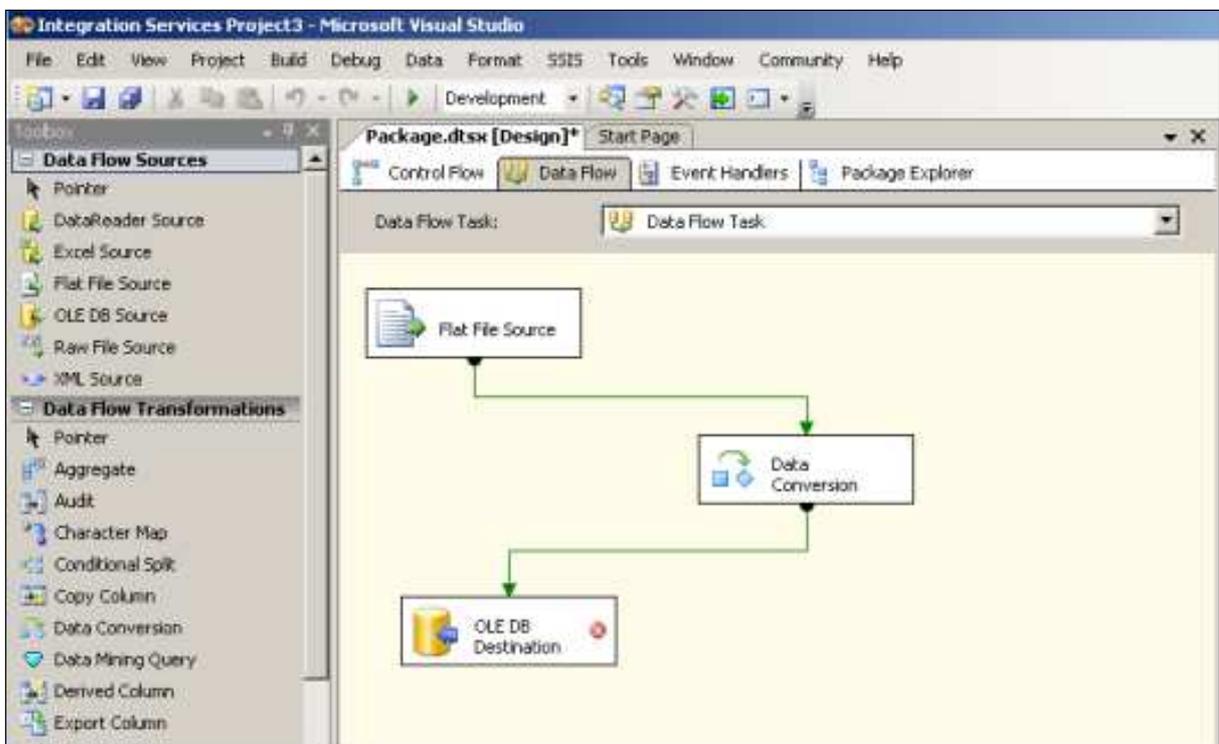


Figura 3.6 – O SSIS integrado ao Microsoft Visual Studio e seus componentes

Fonte: Do próprio autor

O SSIS não é uma ferramenta *open source*, tendo um custo alto para seu licenciamento, já que o mesmo acompanha o pacote do SQL Server 2005 e 2008. Em algumas versões do SQL Server o aplicativo não se encontra disponível, sendo necessária a aquisição das edições mais completas como é o caso da *Enterprise Edition*.

3.3 OLAP

Dentre as ferramentas existentes para as operações OLAP, foi escolhido apenas o Mondrian para ser realizada a análise, o qual será utilizado em conjunto com a solução desenvolvida.

3.3.1 Mondrian/JPivot

O Mondrian é um servidor OLAP, *open source*, escrito na linguagem Java. Ele executa as consultas utilizando uma linguagem parecida com o SQL, chamada MDX, exclusiva para consultas multidimensionais, criada originalmente pela Microsoft. A figura 3.7 demonstra um exemplo de uma consulta na linguagem MDX utilizada pelo Mondrian.

```
SELECT {[Measures].[Unit Sales]} ON COLUMNS,  
FILTER([Product].[Product Family].MEMBERS,  
  [Product].[Product Family].CurrentMember NOT IN  
  {[Product].[All Products].firstChild,  
  [Product].[All Products].lastChild}) ON ROWS  
FROM [Sales]
```

Figura 3.7 – Linguagem multidimensional MDX utilizada pelo Mondrian

Fonte: Adaptador pelo autor conforme Mondrian (2010)

A lógica do Mondrian é baseada no conceito de *schemas*, o qual define o modelo multidimensional e o mapeamento deste modelo. O modelo consiste de elementos definidos pelo *schema*. Estes elementos são: cubos, dimensões, hierarquias, níveis e membros. A construção desse esquema tem a estrutura básica de um XML, como pode ser visto na figura 3.8.

```

<Schema>
  <Cube name="Vendas">
    <Table name="Vendas_2009"/>
    <Dimension name="Clientes" foreignKey="cliente_id">
      <Hierarchy primaryKey="clientes_id">
        <Table name="Clientes"/>
      </Hierarchy>
    </Dimension>
    <Dimension name="Tempo" foreignKey="tempo_id">
      <Hierarchy primaryKey="tempo_id">
        <Table name="Hora_do_dia"/>
        <Level name="Ano" type="Numeric"/>
        <Level name="Mês" type="Numeric"/>
      </Hierarchy>
    </Dimension>
    <Measure name="Un_Vendidas" formatString="#,###"/>
    <Measure name="Preco_Un" formatString="#,###.00"/>
    <CalculatedMember name="Total_Venda" dimension="Measures"
      formula="[Measures].[Un_Vendidas] * [Measures].[Preco_Un]">
    </CalculatedMember>
  </Cube>
</Schema>

```

Figura 3.8 – *Schema* do Mondrian
 Fonte: Adaptador pelo autor conforme Mondrian (2010)

A representação destes *schemas* é feita utilizando os mesmos conceitos relacionados à análise dimensional: cubo (*cube*), representando a lógica multidimensional do sistema, bem como os fatos (*measures*) e as dimensões (*dimensions*). Uma das maneiras de criar um *schema* é editar manualmente o arquivo XML em um editor de texto. Outra maneira é utilizando a ferramenta Schema Workbench. Com ela é possível criar os esquemas de maneira visual e testar os *schemas* através das consultas MDX.

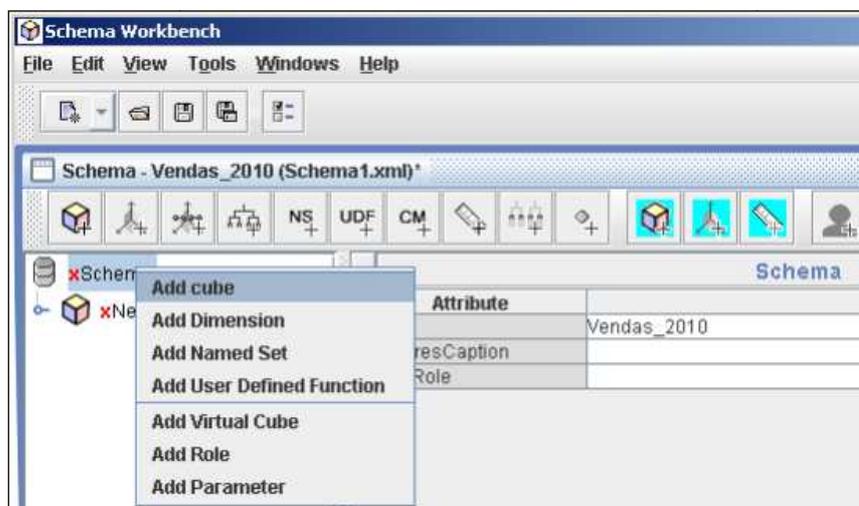


Figura 3.9 – Ferramenta Schema Workbench para criação e testes do *schema*
 Fonte: Do próprio autor

Com o uso do Mondrian o usuário pode analisar uma grande quantidade de dados armazenados em bancos de dados. O Mondrian tem a desvantagem de não possuir uma interface amigável com o usuário, sendo necessário um programa chamado Jpivot que, na verdade, atua como cliente do Mondrian. O Jpivot também é uma ferramenta *open source*.

Através do Jpivot é possível manipular, inserir e remover de uma análise, atributos e métricas. O Jpivot permite realizar operações básicas em consultas OLAP, como *slice-dice* e *drill-down*, acessando estruturas de dados multidimensionais.

O Jpivot possui uma barra de ferramentas que o transforma em uma *interface* amigável para os usuários realizarem suas consultas conforme a figura 3.10.

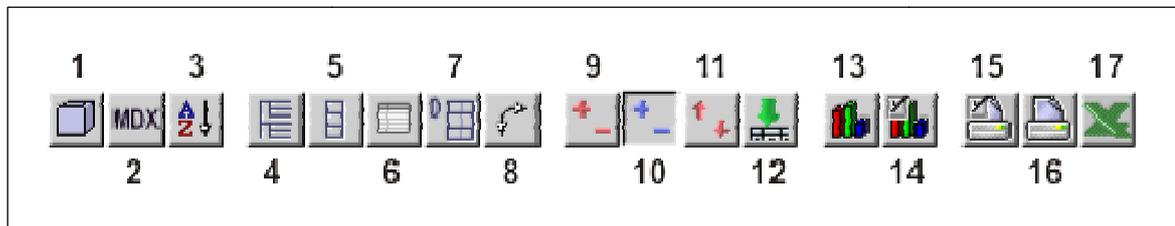


Figura 3.10 – Barra de ferramentas do Jpivot

Fonte: Do próprio autor

A seguir serão descritas as funcionalidades da barra de ferramentas, onde cada botão é indicado por um número de acordo com cada funcionalidade:

1. **Navegação OLAP:** Tem como finalidade a seleção das dimensões, medidas e filtros, para configuração do cubo de dados. Nesta opção é possível selecionar a ordem das colunas;
2. **MDX:** O objetivo deste botão é exibir e modificar a consulta de acesso ao cubo utilizando a linguagem MDX. Essa opção é útil para usuários avançados que já tem conhecimento na linguagem MDX para refinar a consulta;
3. **Ordenação:** Essa funcionalidade permite configurar o modo de ordenação da consulta e configurar o número de linhas para função de *Ranking*;
4. **Exibir membros:** Tem como objetivo definir a exibição das informações dos membros superiores, considerados como pais dentro da estrutura hierárquica, exibindo uma linha com os dados dos membros superiores;
5. **Suprimir títulos:** Permite que seja configurado para suprimir os títulos repetidos nas linhas, para facilitar a visualização dos dados;

6. **Exibir propriedades:** Mostra as propriedades das dimensões do cubo de dados, caso existam;
7. **Suprimir linhas/colunas vazias:** Suprime ou não as colunas ou linhas vazias. Essa opção permite que sejam visualizadas apenas as colunas ou linhas com informações válidas para uma determinada situação;
8. **Inversão dos eixos:** Também chamado de *pivoting*. Essa opção permite que as linhas sejam transformadas em colunas e vice-versa;
9. **Drill Member:** Abre todas as ocorrências do item na árvore;
10. **Drill Position:** Abre apenas a ocorrência selecionada do item na árvore;
11. **Drill Replace:** Substitui a raiz da dimensão atual pelo item selecionado;
12. **Drill Throungh:** Explica de onde foi obtida aquela informação. Nesta opção são exibidos os dados que geraram a informação acumulada;
13. **Gráfico:** Exibe as informações em forma de gráfico dos dados que estão sendo consultados;
14. **Configurações do gráfico:** Permite as configurações do gráfico como seu tipo (linha, barra vertical, etc), altura e largura, título, e muitas outras informações;
15. **Configurações da impressão:** As configurações da impressão permitem configurar os parâmetros de impressão como orientação, tamanho do papel, entre outros;
16. **Impressão:** Realiza a impressão da consulta em formato PDF;
17. **Exportação Excel:** Permite a exportação para formato em planilha XLS. Os dados são exportados conforme estão dispostos na tela, não permitindo que operações OLAP sejam executadas diretamente na planilha.

A figura 3.11 mostra uma consulta OLAP utilizando o JPivot com a barra de ferramentas e o resultado em forma de tabela, utilizando a base de testes padrão do Mondrian.

Test Query uses Mondrian OLAP

Promotion Media	Product	Measures		
		Unit Sales	Store Cost	Store Sales
+All Media	-All Products	266.773	225.627,23	565.238,13
	+Drink	24.597	19.477,23	48.836,21
	+Food	191.940	163.270,72	409.035,59
	+Non-Consumable	50.236	42.879,28	107.366,33

Slicer:

[back to index](#)

Figura 3.11 – Consulta utilizando o JPIVOT

Fonte: Do próprio autor

3.4 CONSIDERAÇÕES FINAIS

Na análise realizada com o SADIG destacou-se a funcionalidade na qual é possível construir um repositório de dados onde são feitas as análises e consultas. Em contrapartida, não foi possível testar este recurso em virtude de a ferramenta fazer apenas as análises e consultas no DM disponibilizado nesta versão. O SADIG não é um software *open source* e tem um custo para sua aquisição e uso. A ferramenta Pentaho é uma solução de BI completa, com licença *open source* e fornece diversos recursos para análises, com uma *interface web*, podendo ser acessada em qualquer computador com acesso a Internet ou dentro de uma Intranet.

O Mondrian/JPIVOT, que é o núcleo da ferramenta Pentaho Analysis, possui ótimos recursos OLAP, com facilidade para aplicação em outras ferramentas. Dentre as ferramentas de ETL, tanto o Kettle como o SSIS atendem todas as premissas de uma ferramenta ETL, com destaque ao Kettle, por ser *open source* e possuir uma *interface* intuitiva e de fácil utilização.

Para a utilização em conjunto com a solução proposta optou-se pelo Mondrian/JPIVOT para realizar as consultas OLAP e pelo Kettle para o processo de ETL. Os principais motivos para a escolha foram: (i) a qualidade e variedade dos recursos disponíveis nestas ferramentas; (ii) a possibilidade de serem utilizadas sem custo; (iii) pela facilidade de integração com a solução proposta, visto que o funcionamento destas ferramentas é todo baseado em arquivos XML que podem ser gerados pela solução de BI proposta neste trabalho, permitindo, desta forma, criar uma solução integrada e de fácil uso por parte dos usuários finais da solução de BI do ERP SIGER.

4 FERRAMENTA DESENVOLVIDA

A proposta deste trabalho é criar uma solução de BI para o ERP SIGER que possa ser adaptada para atender diversas áreas e ramos de atividade, de fácil utilização e que permita a construção de novos DMs para que seja possível criar de acordo com a área de negócio dos clientes. Foi desenvolvida uma solução integrada para atender a todos esses requisitos de forma simples e intuitiva sem a necessidade do usuário possuir conhecimento em várias ferramentas para a construção de uma solução de BI.

Conforme pode ser visto na figura 4.1 a solução desenvolvida empacota em uma única ferramenta a construção do DM, o processo de ETL e a criação dos cubos de dados para futuras consultas e análises.

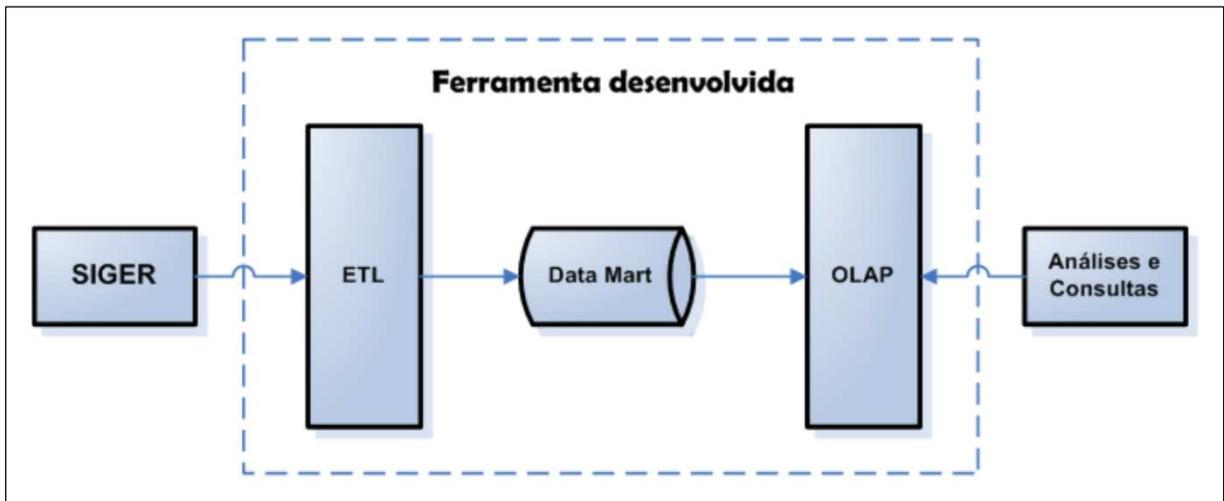


Figura 4.1 – Visão geral da solução de BI construída

Fonte: Do próprio autor

A solução utiliza ferramentas como o Kettle para o processo de ETL e o Mondrian/JPIVOT para a tarefa de OLAP. Porém, a ferramenta construída é quem interage de forma automática com essas duas ferramentas sem a necessidade de conhecimento em nenhuma das duas ferramentas.

4.1 VISÃO GERAL

A ferramenta foi desenvolvida utilizando linguagem de programação *Java* com interface gráfica *Swing*. Sua principal tarefa é fornecer uma interface gráfica simples para criação e modelagem de vários DMs e demais partes de acordo com a necessidade. A solução construída foi chamada de BIIM (*Business Intelligence Integration Module*).

A figura 4.2 demonstra o fluxograma da solução construída como um todo, onde está localizado cada processo e os dados do sistema. Como base de toda a solução está o SIGER onde usuário final está constantemente lançando os dados nos diversos módulos do sistema, que são gravados em um banco de dados. Um dos pontos principais do SIGER e o dicionário de dados que contém as informações sobre as tabelas e campos utilizados no sistema.

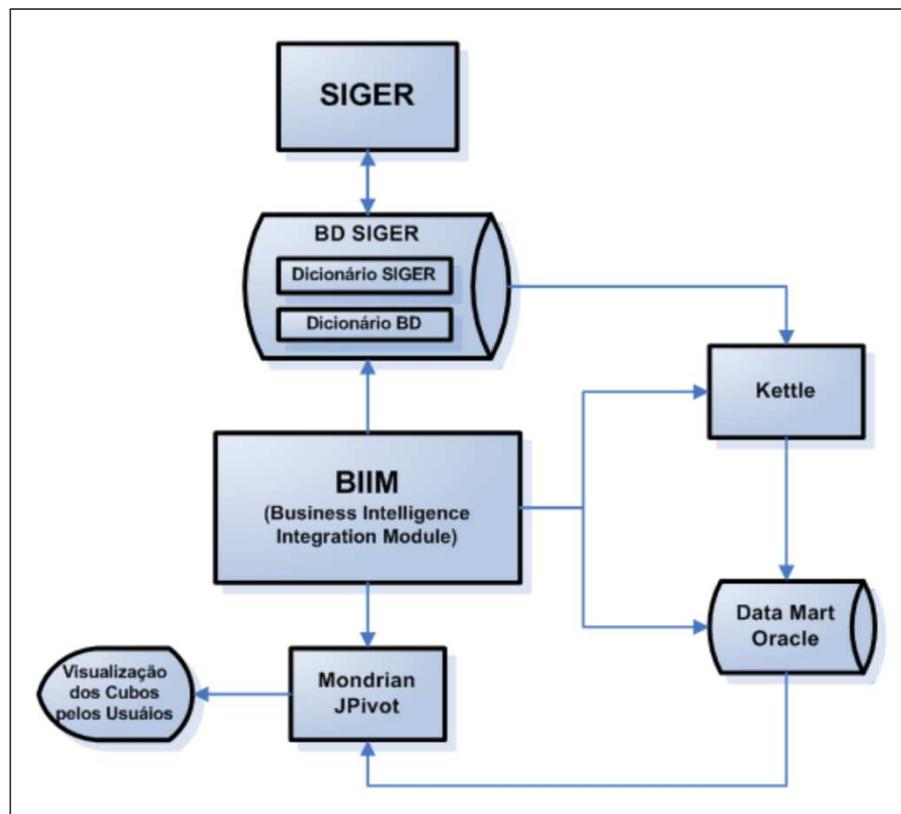


Figura 4.2 – Fluxograma da solução de BI
Fonte: Do próprio autor

O dicionário de dados do SIGER juntamente com o dicionário de dados do próprio SGBD é fundamental para ferramenta, pois é através deles que os dados do banco de dados do ERP serão vinculados ao DM.

Seguindo o fluxo, entra o BIIM para a construção do DM, que utiliza esse dicionário de dados do SIGER e o próprio dicionário de dados do SGBD para outras tabelas disponíveis no banco de dados. A ferramenta por sua vez faz a integração com o Kettle, responsável pela parte de extração, transformação e carga dos dados vindos do banco de dados do SIGER para o banco de dados do DM construído. O BIIM também é responsável pela integração com o Mondrian para a construção dos cubos de dados do DM.

Por fim está a *interface* para o usuário final visualizar os cubos de dados construídos no Mondrian. Para essa visualização é utilizado o JPivot, que permite através de um

navegador de internet realizar todas as consultas, mas pode ser utilizada qualquer outra ferramenta ou *plugin* que utilize o Mondrian como servidor OLAP. A figura 4.3 demonstra a tela inicial da ferramenta desenvolvida que permite a criação de DMs com suas tabelas fato e dimensões, geração física do DM, carga dos dados (ETL) e criação dos cubos de dados.

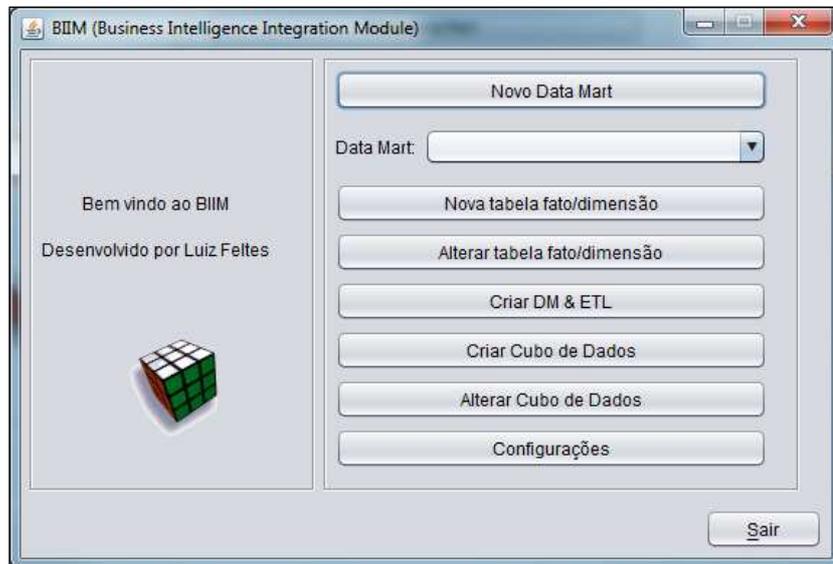


Figura 4.3 – Tela Inicial da ferramenta BIIM

Fonte: Do próprio autor

4.2 CRIAÇÃO DE DATA MARTS

Uma das premissas da ferramenta é permitir a construção de vários DMs, sendo possível criar um DM para cada área de negócio da empresa, como por exemplo, um DM para o Faturamento, outro para Financeiro, entre outros módulos do sistema. Essa funcionalidade garante que a ferramenta possa ser adaptada para atender diversas situações em diversos tipos de áreas de atuação da empresa. A construção de um novo DM pode ser utilizada a qualquer momento, possibilitando que o próprio cliente ou um consultor da Rech crie novos DMs de acordo com a necessidade. A figura 4.4 demonstra a criação de um DM.



Figura 4.4 – Criação de um novo Data Mart

Fonte: Do próprio autor

Para cada DM existente, é possível definir suas tabelas fato e dimensão, sendo possível adicionar novas tabelas fato/dimensão a qualquer momento, bem como fazer alterações nas mesmas. Considerando o DM selecionado, a ferramenta permite também criar o DM no banco de dados, fazer a carga dos dados e a construção de vários cubos de dados, que depois são disponibilizados ao usuário final para suas análises e consultas.

Pode se destacar também que o DM criado fica disponível em um banco de dados e pode ser utilizada quaisquer ferramentas de relatórios como o iReport ou Excel, utilizando ODBC ou JDBC. Este acesso aos dados armazenados no repositório configura-se mais uma flexibilidade para os usuários.

4.3 CONFIGURAÇÕES

Algumas configurações são necessárias após a criação do DM. Essas informações são utilizadas para a conexão com o dicionário de dados do ERP SIGER ou com o dicionário do próprio SGBD. Nesta janela também é necessário indicar a localização do SGBD onde o DM será criado, configurando o tipo de banco e as informações sobre a conexão, conforme a figura 4.5.



A imagem mostra uma janela de configuração intitulada "Configurações". No topo, há três abas: "Dicionário SIGER", "Dicionário BD" e "Data Mart", sendo a última selecionada. O formulário principal contém os seguintes campos:

- Banco: Oracle (menu suspenso)
- IP: localhost (campo de texto)
- Porta: 1521 (campo de texto)
- Nome: xe (campo de texto)
- Usuário: datamart (campo de texto)
- Senha: datamart (campo de texto)

Na base da janela, há dois botões: "Salvar" e "Voltar".

Figura 4.5 – Configurações do DM e os dicionários de dados
Fonte: Do próprio autor

4.4 CONSTRUÇÃO DA TABELA FATO OU DIMENSÃO

A modelagem de um DM, usando os conceitos da Modelagem Multidimensional, prevê a organização dos dados em tabelas Fatos e tabelas Dimensão, onde as tabelas Fatos

contém os dados quantitativos e as relações que a tabela Fato tem com suas tabelas Dimensões, que descrevem quem, o que, quando e onde esses fatos ocorreram. As tabelas Dimensões têm as informações descritivas sobre esses fatos, cada um sobre uma perspectiva. A solução desenvolvida utiliza o modelo estrela para a criação das tabelas Fato e Dimensão.

Para a construção de uma tabela fato ou dimensão, inicia-se a ação a partir do botão “Nova tabela fato/dimensão” da tela principal. A partir daí, é necessário informar: qual o tipo de tabela (Fato ou Dimensão) que está sendo criada através de uma *combo box*, o nome que será utilizado para a tabela e uma breve descrição, como mostra a figura 4.6.

Figura 4.6 – Criação de uma tabela fato ou dimensão
Fonte: Do próprio autor

Podem ser construídas quantas tabelas fato e dimensões forem necessárias para o DM que está sendo desenvolvido, sem nenhuma limitação. Este recurso é interessante, pois permite que em um mesmo DM se crie várias visões e associações diferentes entre as tabelas, mantendo-as em um repositório único, para que depois sejam construídos os cubos de dados (tarefa que será abordada na seção 4.6 com maiores detalhes).

4.4.1 Construção dos campos

A construção dos campos das tabelas Fato e Dimensão têm um grande facilitador que é a integração com dicionário de dados do SIGER e o próprio dicionário de dados do SGBD, permitindo que seja selecionado o relacionamento entre o campo que está sendo criado no DM com o dado de origem da informação.

Essa ligação entre o campo criado na ferramenta construída, com o campo de qualquer um dos dicionários de dados é utilizada também para o processo de ETL. Com a ligação entre os dois, o processo de ETL vai criar um *link* entre a informação de origem no

banco de dados do sistema com o campo de destino do DM. A construção dos campos da tabela fato ou dimensão pode ser feita de três formas:

- **Utilizando o dicionário de dados do SIGER** - com esse recurso é possível acessar todas as tabelas e campos do sistema SIGER de forma simples e intuitiva. A partir dessa integração do campo criado no DM com o campo do dicionário de dados do SIGER é feita a associação para a carga de dados posteriormente com o Kettle;
- **Utilizando o dicionário de dados do SGBD** – é possível utilizar outras tabelas e campos disponíveis no banco de dados utilizando o próprio dicionário de dados do SGBD;
- **Campos virtuais** – com esta funcionalidade é possível criar campos virtuais que podem ser campos calculados, utilizando operadores matemáticos como soma, subtração, divisão e multiplicação baseado nos campos do dicionário de dados do SIGER. Também é possível utilizar todas as funções do banco de dados, como por exemplo, para capturar apenas uma parte do campo, o que é muito útil, para criação de um campo “dia” a partir de um campo no formato “Date” do banco de dados com uso da função “TO_CHAR” do Oracle.

Mesmo depois de criado o campo, é possível modificar seu relacionamento com o dicionário de dados, selecionando outro campo ou tabela quando se fizer necessário.

Outras informações são necessárias para a criação de campos no DM que está sendo construído, como pode ser verificado na figura 4.7. É preciso informar um nome para o campo e uma descrição sucinta, que serão utilizadas posteriormente para criação do DDL do DM e para criação do *schema* do Mondrian.

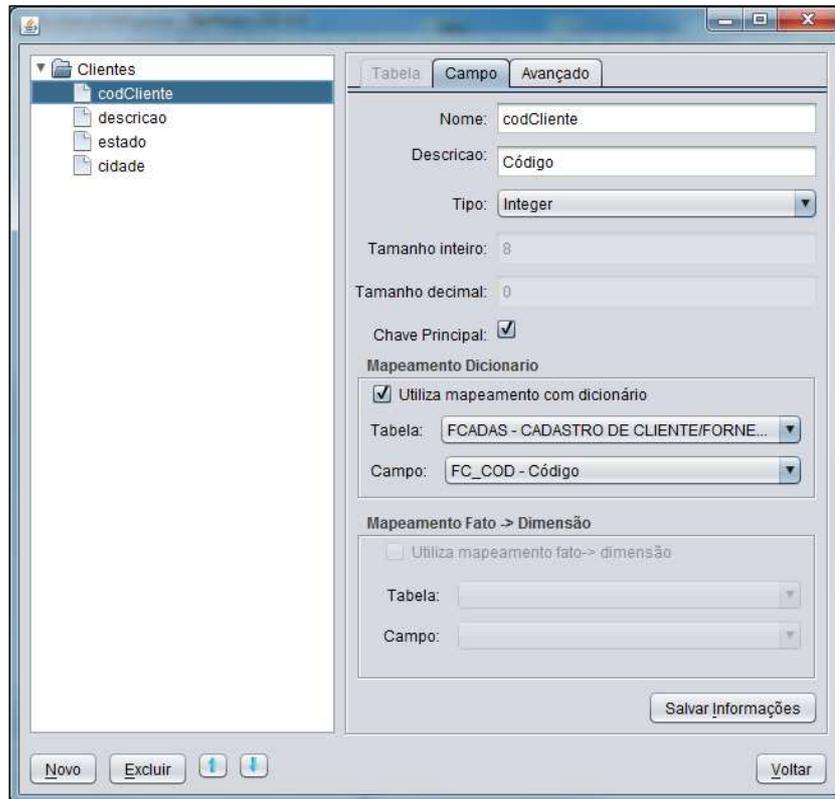


Figura 4.7 – Criação de campos

Fonte: Do próprio autor

A informação “Tipo” do campo é muito importante para a criação do DDL, pois é através dela que é criado o *data type* do SGBD. Essa informação é carregada automaticamente quando o campo utilizado for proveniente do dicionário de dados do SIGER ou do SGBD, não sendo possível modificar, pois deve ser exatamente do mesmo tipo e tamanho da informação de origem. Quando o campo criado for do tipo virtual é possível informar seu tipo e tamanho. Em alguns tipos, como por exemplo, o tipo “Decimal”, é possível também informar a precisão e a escala.

Para os campos participantes da chave principal da tabela Fato ou Dimensão é necessário marcar o indicador “Chave Principal”. O BIIM permite a criação de tabela com mais de um campo como chave principal, que ocorre principalmente nas tabelas Fatos. Essa informação é importante, pois é através dela que as demais tabelas irão se relacionar com as tabelas Fato ou Dimensão.

Outro aspecto importante para a construção do DM é o mapeamento entre a tabela Fato com as tabelas Dimensões. Esse mapeamento serve para indicar a relação entre o fato ocorrido com as respectivas Dimensões associadas. É através dele que posteriormente o cubo de dados será montado.

Esse mapeamento entre os campos chave da tabela Fato com os campos chave das tabelas Dimensões é feito através de duas *ComboBox*, uma informando a tabela Dimensão e na outra o campo. Essa informação é utilizada para realizar a criação do *schema* do Mondrian.

Algumas informações avançadas podem ser criadas como, por exemplo, a criação de filtro em um determinado campo. O filtro serve para coletar apenas as informações importantes para o contexto que está se criando. Esse filtro pode ser feito por campo, permitindo então criar diversas filtragens para coletar apenas os dados que satisfaçam os filtros selecionados. É possível utilizar operadores como Igual, Diferente, Maior, Maior ou Igual, Menor e Menor ou Igual, e um valor numérico ou textual para que ocorra o filtro no momento da carga dos dados feita pelo Kettle.

Outra informação que está disponível na aba “Avançado” é o campo fórmula. A funcionalidade da fórmula é importante para criar campos, seja juntando dois valores em campos distintos, por exemplo, juntar o nome com sobrenome ou juntar informações de endereço, como também para criar campos de porcentagem ou totalizadores. Também é possível utilizar operadores matemáticos entre os campos do dicionário ou funções do SGBD como SUM, MAX, TO_CHAR, entre outras. A figura 4.8 mostra as funcionalidades existentes na aba “Avançado” do BIIM.

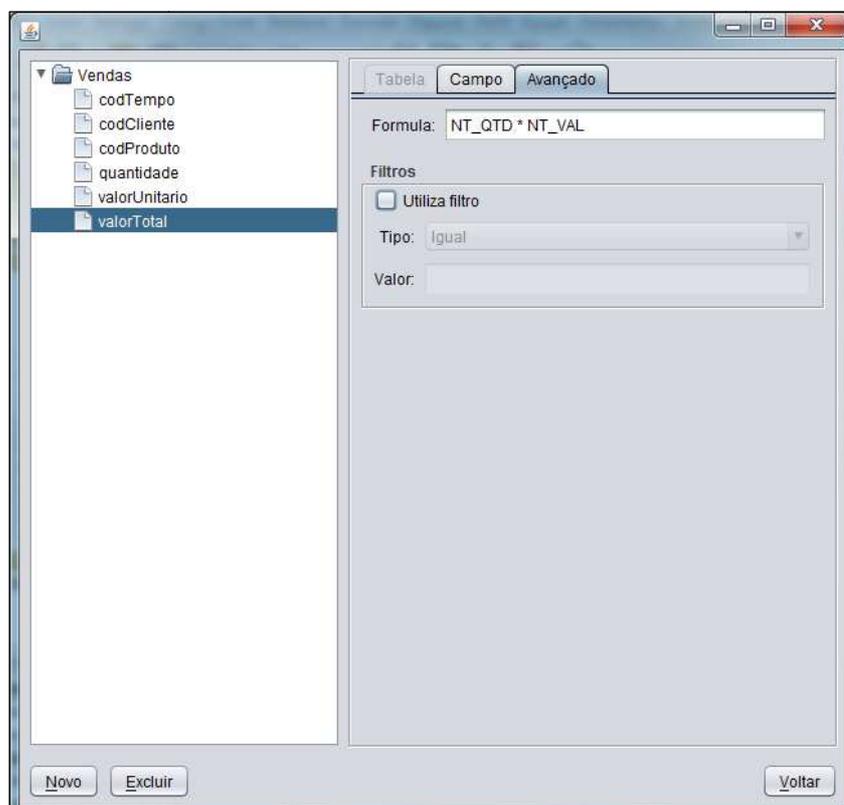


Figura 4.8 – Criação de campos, aba “Avançado”

Fonte: Do próprio autor

4.5 GERAÇÃO DO DATA MART

Após definidas as tabelas Fato e Dimensão, bem como seus campos, os mapeamentos entre elas, é possível de forma automatizada a ferramenta gerar o DDL para criação das tabelas no banco de dados. Esse processo automático para criação das tabelas do DM é um grande facilitador da ferramenta, sendo que o usuário que está construindo o DM não precisa ter conhecimentos em comandos SQL para criação física do DM.

A geração do DDL segue o esquema básico para criação de uma tabela, envolvendo o nome da tabela, seus campos com suas definições de tipo e as informações sobre a chave primária da tabela, conforme exemplificado na figura 4.9.

```
CREATE TABLE [nome_tabela] (  
    [nome_campo_1] [definição_tipo_1],  
    ...  
    [nome_campo_N] [definição_tipo_N],  
    CONSTRAINT [nome_tabela]_PK PRIMARY KEY (  
        [campo_chave_1],  
        ...  
        [campo_chave_N]  
    )  
)
```

Figura 4.9 – Esquema DDL para criação do Data Mart

Fonte: Do próprio autor

Para cada tabela gerada no DM selecionado é executado no banco de dados o comando de exclusão quando tabela já existe e após o comando de criação da tabela, que posteriormente será carregada pela integração com o Kettle.

4.5.1 Integração com Kettle (ETL)

A ferramenta ETL escolhida para integração com a ferramenta desenvolvida foi o Kettle. Um dos fatores decisivos para escolha foi a facilidade de se criar através de um arquivo XML a transformação e carga dos dados a ser realizada pelo Kettle e o mesmo permite que seja possível visualizar essa transformação criada automaticamente no Kettle e realizar outros ajustes manualmente caso seja necessário.

Os arquivos do Kettle se baseiam em arquivos com estrutura XML. Algumas informações estão disponíveis na documentação e outras foram descobertas utilizando a ferramenta, criando as transformações e posteriormente entendendo o arquivo XML gerado.

Com base nas informações do DM são criados dois arquivos XML. O primeiro arquivo XML é relacionado ao *Job* do Kettle. O *Job* é o responsável por realizar a chamada para o segundo XML responsável pela transformação em si. O arquivo *Job* é o arquivo iniciado pela ferramenta para iniciar o processo de extração, transformação e carga dos dados. A figura 4.10 demonstra uma parte do arquivo XML do *Job* gerado.

```
<?xml version="1.0" encoding="UTF-8"?>
<job>
  <name>Clientes</name>
  <entries>
    <entry>
      <name>START</name>
      <type>SPECIAL</type>
      <start>Y</start>
    </entry>
    <entry>
      <name>TransformacaoClientes</name>
      <type>TRANS</type>
      <filename>Clientes.ktr</filename>
    </entry>
  </entries>
  <hops>
    <hop>
      <from>START</from>
      <to>TransformacaoClientes</to>
    </hop>
  </hops>
</job>
```

Figura 4.10 – Parte do arquivo XML do Job gerado para ser utilizado com o Kettle
Fonte: Do próprio autor

O segundo arquivo XML criado é o responsável pelas informações de como ocorrerá a extração dos dados, transformações e filtros. Esse arquivo é responsável por ler as tabelas de origem, realizar os filtros e transformações necessárias e posteriormente realizar o mapeamento entre os campos carregados com os campos de destino no DM.

```

<?xml version="1.0" encoding="UTF-8"?>
<transformation>
  <info>
    <name>TransformacaoClientes</name>
  </info>
  <connection>
    <name>BaseDestino</name>
    <server>localhost</server>
    <type>ORACLE</type>
    ...
  </connection>
  <connection>
    <name>BaseOrigem</name>
    <server>localhost</server>
    <type>ORACLE</type>
    ...
  </connection>
  <order><hop><from>Entrada</from><to>Saida</to></hop></order>
  <step>
    <name>Entrada</name>
    <type>TableInput</type>
    <connection>BaseOrigem</connection>
    <sql>SELECT FC_COD,FC_DES,FC_EST,FC_CID FROM FCADAS</sql>
  </step>
  <step>
    <name>Saida</name>
    <type>TableOutput</type>
    <connection>BaseDestino</connection>
    <table>Clientes</table>
    <fields>
      <field>
        <column_name>codCliente</column_name>
        <stream_name>FC_COD</stream_name>
      </field>
      ...
    </fields>
  </step>
</transformation>

```

Figura 4.11 – Parte do arquivo XML da transformação gerada

Fonte: Do próprio autor

O XML da transformação contém as informações mais importantes para este processo. Nele estão os dados sobre a conexão de origem onde se encontra o dicionário de dados e a conexão de destino onde se encontra o DM criado. Também contém as informações sobre a ordem de execução dos processos e o que cada um deles deve fazer.

Após a criação do XML é possível abrir os arquivos criados com o Kettle para visualizar o processo, e se necessário realizar algum ajuste fino ou utilizar demais ferramentas disponíveis pelo Kettle. A figura 4.12 demonstra a transformação criada, que pode ser alterada posteriormente utilizando o Kettle.

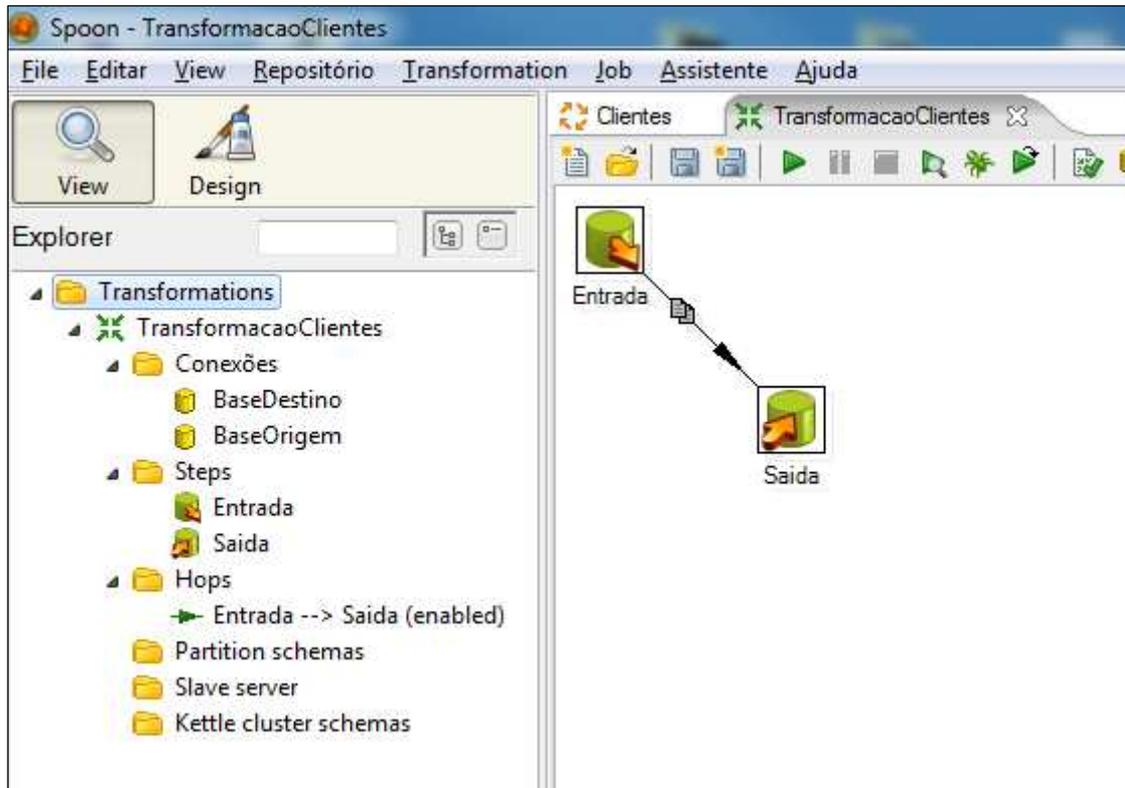


Figura 4.12 – Kettle com arquivo de transformação criado
 Fonte: Do próprio autor

Depois de gerados os arquivos XML do Kettle, automaticamente a ferramenta executa o Kettle por linha de comando chamando cada um dos *Jobs* criados, fazendo a carga no DM, deixando o mesmo populado para serem criados os cubos de dados. Essa funcionalidade permite que o usuário mesmo sem conhecimentos no Kettle e sem a necessidade de intervenção manual na ferramenta carregue todos os dados no DM construído.

4.6 CUBO DE DADOS

O processo de criação do cubo de dados é necessário para viabilizar a visualização dos dados contidos no DM sob várias perspectivas. Esse processo é importante, pois com ele são criadas as visões de dados que o usuário, o tomador de decisão, vai utilizar para realizar as análises com as informações que foram fornecidas para cada situação.

Destá forma, o BIIM permite que sejam construídos vários cubos de dados para o DM selecionado, o que possibilita que o mesmo DM forneça várias formas de enxergar os dados, selecionando apenas as tabelas e campos necessários.

Os dados necessários para geração de um cubo podem ser verificados na figura 4.13, como um nome para o cubo, uma breve descrição, qual a tabela fato que o cubo de dados vai

ser relacionado e seus campos disponíveis. Devem também ser informadas as tabelas dimensões e os seus respectivos campos para permitir a visualização do cubo de dados de diversas perspectivas.

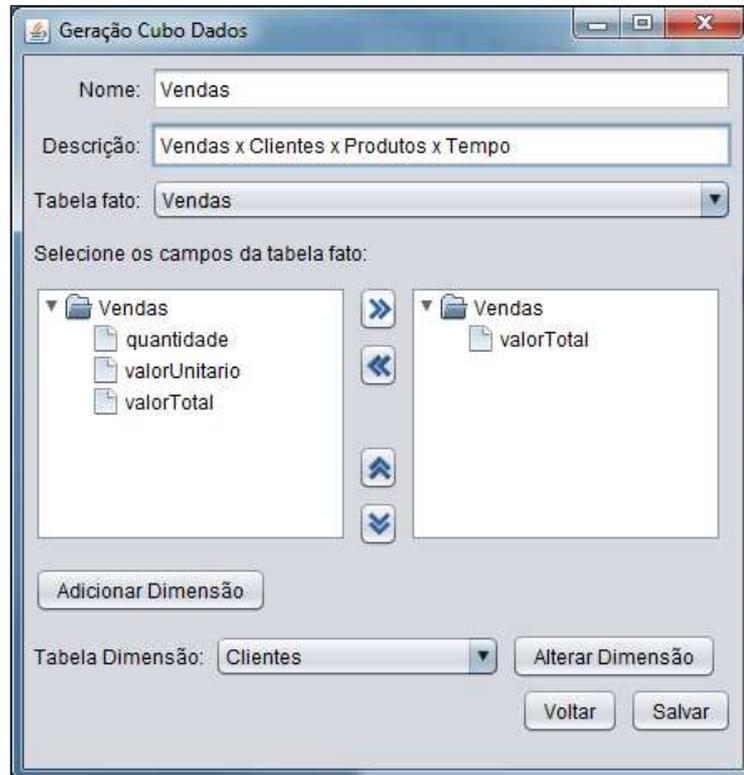


Figura 4.13 – Tela para geração dos cubos de dados

Fonte: Do próprio autor

Na definição dos cubos de dados, é possível de forma simplificada escolher apenas os campos necessários. Para tanto, basta selecionar o campo na *Treeview* da esquerda e clicar no botão com seta para direita, passando o campo selecionado para a *Treeview* da direita, que contém todos os campos do cubo de dados. Também é possível modificar a ordem dos campos, permitindo que determinado campo venha antes de outro com os botões com seta para cima ou para baixo.

A manutenção das tabelas dimensões ocorre da mesma forma que a seleção da tabela fato e seus campos. No caso das tabelas Dimensões é permitido adicionar quantas tabelas dimensões forem necessárias para o cubo de dados. A figura 4.14 demonstra a janela de manutenção do cubo onde é possível adicionar ou remover campos do DM criado e alterar sua ordem para montar o nível hierárquico na navegação na ferramenta OLAP.



Figura 4.14 – Tela para adicionar tabela dimensão e campos ao cubo de dados
 Fonte: Do próprio autor

4.6.1 Integração com Mondrian/JPivot

O Mondrian é um servidor OLAP que funciona a partir de um *schema* em formato XML. Nesse XML é definida a estrutura do cubo de dados, como sua tabela fato e seus campos, tabelas dimensões com seus respectivos campos e todos os relacionamentos entre eles. Esse *schema* é a base para o funcionamento do Mondrian e para todas as ferramentas de visualização do cubo de dados, que para este trabalho foi utilizado o JPivot.

Com base nas informações definidas para o cubo de dados é feita a integração com o Mondrian. Essa integração ocorre com a geração automática do *schema* XML do Mondrian com todos os dados anteriormente coletados pela aplicação, como relacionamentos entre as tabelas dimensões e a tabela fato e gerando todos os campos de acordo com os selecionados para cada tabela fato ou dimensão.

```

<Schema name="Vendas realizadas">
  <Cube name="Vendas">
    <Table name="VENDAS" schema="DATAMART" />
    <Dimension foreignKey="CODPRODUTO" name="Produto">
      <Hierarchy name="Produto" primaryKey="CODPRODUTO">
        <Table name="PRODUTO" schema="DATAMART" />
        <Level name="Grupo" column="GRUPO" type="Integer" />
        <Level name="Subgrupo" column="SUBGRUPO" type="Integer" />
        <Level name="Descricao" column="DESCRICAO" type="String" />
      </Hierarchy>
    </Dimension>
    <Dimension foreignKey="CODCLIENTE" name="Cliente">
      <Hierarchy name="Cliente" primaryKey="CODCLIENTE">
        <Table name="CLIENTES" schema="DATAMART" />
        <Level name="Estado" column="ESTADO" type="String" />
        <Level name="Cidade" column="CIDADE" type="String" />
        <Level name="Descricao" column="DESCRICAO" type="String" />
      </Hierarchy>
    </Dimension>
    <Dimension foreignKey="CODTEMPO" name="Tempo">
      <Hierarchy name="Tempo" primaryKey="CODTEMPO">
        <Table name="TEMPO" schema="DATAMART" />
        <Level name="Ano" column="ANO" type="Integer" />
        <Level name="Mes" column="MES" type="String" />
        <Level name="Dia" column="DIA" type="Integer" />
      </Hierarchy>
    </Dimension>
    <Measure name="Quantidade" column="QUANTIDADE" datatype="Numeric" />
    <Measure name="Valor unitario" column="VALORUNITARIO" datatype="Numeric" />
    <Measure name="Valor total" column="VALORTOTAL" datatype="Numeric" />
  </Cube>
</Schema>

```

Figura 4.15 – *Schema XML* do Mondrian gerado automaticamente pela ferramenta

Fonte: Do próprio autor

Conforme pode se verificar na figura 4.15, a criação do *schema XML* tem a definição do cubo, indicando sua tabela fato. Em seguida todas as tabelas dimensões devidamente relacionadas com a tabela fato com uma *Foreign Key*, e seus campos que podem vir a serem utilizados, por exemplo, para agrupamentos ou filtros, e outros com seu tipo de dados definido.

Por fim, na estrutura do *schema*, estão todos os campos da tabela fato e das tabelas de dimensões que serão utilizados para análise de acordo com a visualização definida.

4.6.2 Visualização dos cubos de dados

Depois de gerados os cubos de dados, é possível utilizando a ferramenta JPivot, visualizar esses cubos criados, realizando as consultas OLAP de forma visual ou até mesmo criando ou modificando as consultas através da linguagem MDX para usuários avançados.

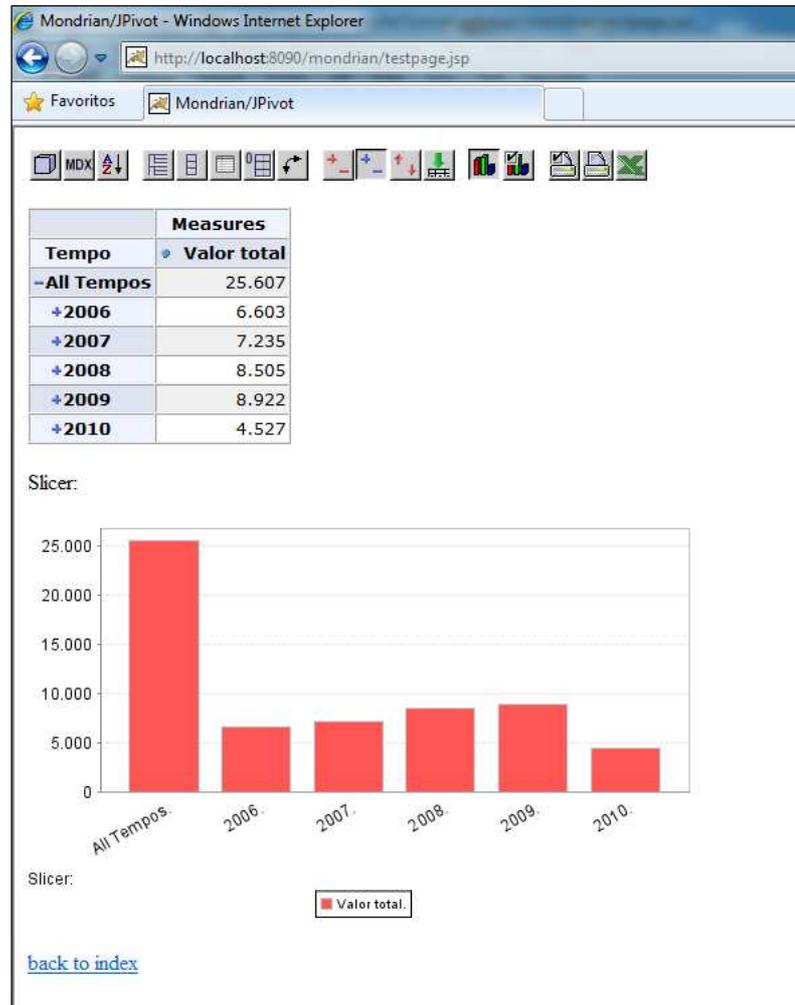


Figura 4.16 – Visualização de uma consulta utilizando o JPIVOT
 Fonte: Do próprio autor

A visualização dos cubos de dados gerados é feita utilizando um navegador de livre escolha do usuário, podendo ser acessado em uma rede local ou pela Internet, conforme a figura 4.16. O JPIVOT utiliza o Apache Tomcat e páginas JSP para a visualização do cubo de dados do Mondrian. Para cada cubo de dados criado é gerada automaticamente uma página JSP para que seja possível realizar a visualização dos dados.

Pode também ser utilizada qualquer outra ferramenta ou *plugin* que seja compatível com a *engine* do Mondrian para a visualização dos dados, como o JPalo, JRubik, entre outros. A figura 4.17 demonstra a utilização do JRubik usando o mesmo *schema* criado pela ferramenta construída para visualizar o cubo de dados.

Figura 4.17 – JRubik acessando o mesmo *schema* construído para o Mondrian

Fonte: Do próprio autor

Nesse ponto cabe ao tomador de decisão ou ao gestor explorar os dados de acordo com sua necessidade. O JPivot, assim como as outras ferramentas OLAP, permitem explorar os dados utilizando várias visões, filtros e todos os outros recursos OLAP já discutidos anteriormente.

4.7 MODELAGEM DA FERRAMENTA

O BIIM precisa armazenar informações sobre os DM construídos, tabelas fatos e dimensões, campos e os cubos de dados. A figura 4.18 demonstra o modelo ER construído para armazenar tais informações no SGBD e, desta forma, atender os requisitos da solução.

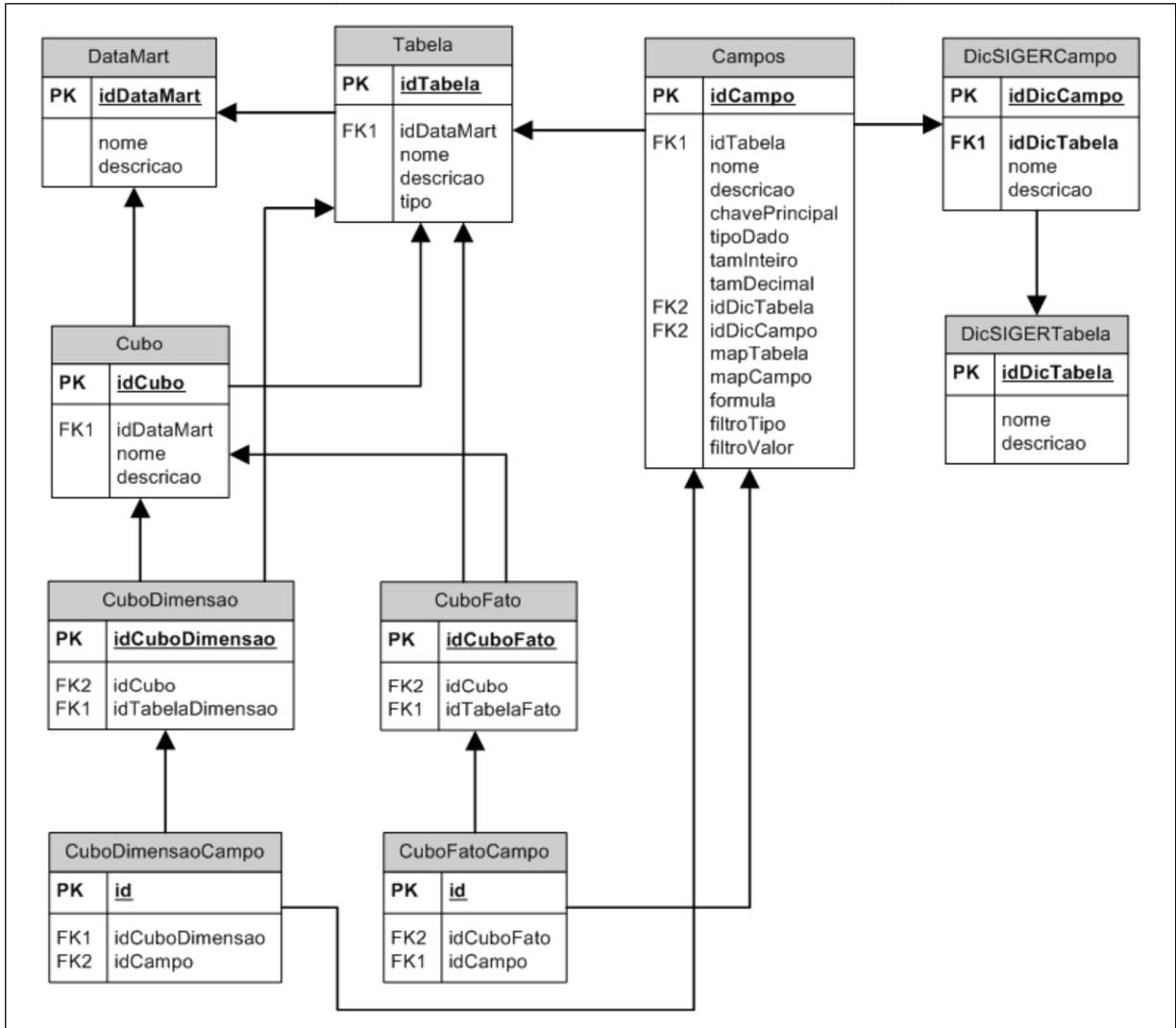


Figura 4.18 – Modelo ER das tabelas utilizadas para o BIIM

Fonte: Do próprio autor

A entidade “DataMart” é responsável pelas informações de todos os DMs criados na ferramenta. Contém como campos o nome e a descrição para cada DM criado.

A entidade “Tabela” contém as informações das tabelas Fato e Dimensões criadas para cada DM. Tem informações sobre o nome da tabela, sua descrição e informação indicando qual o tipo de tabela, Fato ou Dimensão.

A entidade “Campos” possui os campos da tabela Fato ou Dimensão e informações como nome, descrição, indicação se é uma chave principal, qual o tipo de dado (Texto, Numérico, Data) e informação sobre esse tipo como tamanho e sua precisão em caso de numérico. Essa entidade possui também informações sobre o mapeamento do campo com o dicionário de dados do SIGER, indicando qual tabela e campo do dicionário o mesmo se

relaciona. Informações sobre o tipo de filtro e o valor utilizado pelo filtro e informações caso o campo tenha alguma fórmula.

As entidades “DicSIGERCampo” e “DicSIGERTabela” contém as informações sobre as tabelas e campos utilizadas pelo ERP Siger. Como se trata de uma ferramenta comercial, várias informações foram omitidas na figura 4.18. Essas duas entidades tem as informações necessárias para identificar de onde devem vir as informações para o DM que está sendo construído.

A entidade “Cubo” em conjunto com as entidades “CuboDimensao”, “CuboDimensaoCampo”, “CuboFato” e “CuboFatoCampo” possuem as informações dos cubos de dados criados para o DM selecionado. Informações como nome, descrição, qual a tabela fato e os campos selecionados da tabela Fato e as tabelas Dimensões com seus respectivos campos selecionados.

5 ESTUDO DE CASO

A fim de validar a ferramenta construída com um caso prático, foi desenvolvido um estudo de caso com uma base de dados real. Alguns dados foram modificados ou omitidos para preservar seu conteúdo.

Esse capítulo está organizado de acordo com as etapas de todo o processo de criação de uma solução de BI usando a ferramenta proposta neste trabalho. Inicia-se com a construção do DM, criando as tabelas dimensões com seus campos necessários e a tabela fato com os campos. Posteriormente será abordada a criação e carga dos dados no DM utilizando a integração com o Kettle. Na sequência a criação de dois cubos de dados com finalidades diferentes, mas utilizando o mesmo DM construído. Por fim, a análise de alguns dados com a ferramenta JPivot.

5.1 CONSTRUÇÃO DO DM

Para o estudo de caso será criada uma tabela fato chamada “Vendas” e três tabelas dimensões, sendo uma para “Tempo”, outra para “Produto” e uma tabela “Cliente”. As próximas seções abordam a criação de cada uma das tabelas e seus respectivos campos. A figura 5.1 demonstra o modelo ER com estas tabelas e seus campos.

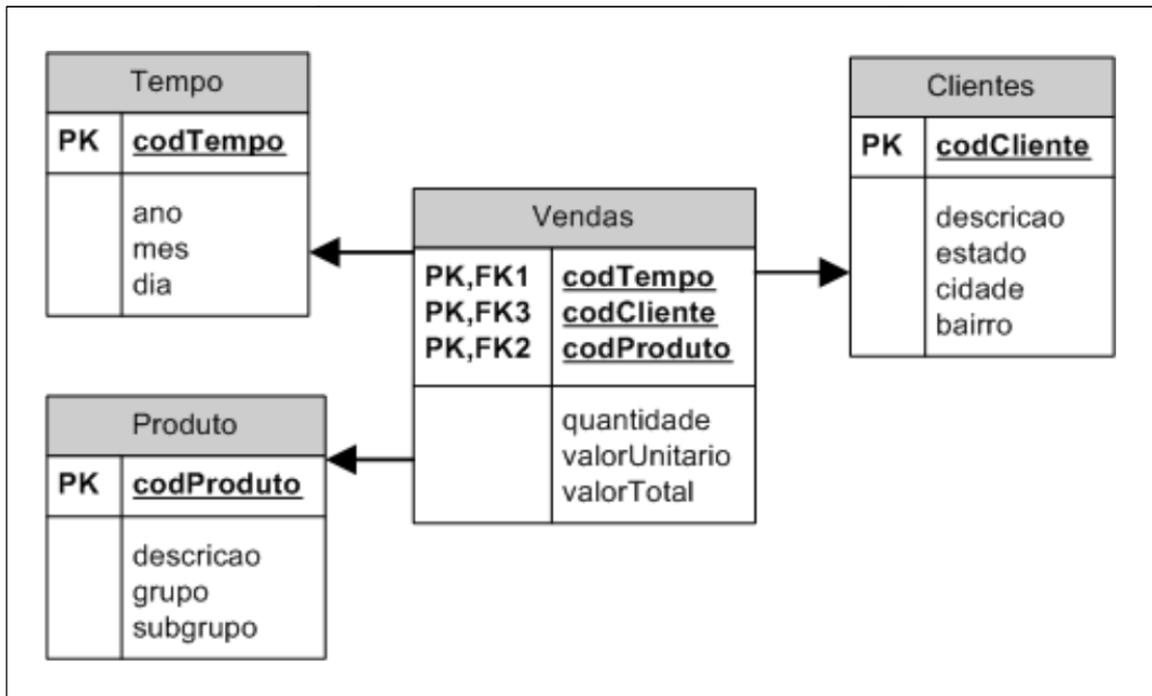


Figura 5.1 – Modelo ER do DM construído para o estudo de caso

Fonte: Do próprio autor

5.1.1 Criação do DM Faturamento

O primeiro processo para a construção do DM é a criação de um novo DM. Na tela inicial da ferramenta proposta deve-se clicar no botão “Novo Data Mart” e informar um nome e uma breve descrição para o DM que está sendo criado. A figura 5.2 mostra a tela para criação do novo DM.



Figura 5.2 – Criação de um novo DM
Fonte: Do próprio autor

Os próximos passos devem ser realizados selecionando na caixa de seleção “Data Mart”, na tela principal da ferramenta, o DM que será utilizado para construção, neste caso, o DM “Faturamento”.

5.1.2 Construção da tabela dimensão Produto

A tabela dimensão “Produto” possui as informações relacionadas aos produtos que foram vendidos e que cujo fato da venda está na tabela fato “Vendas”. A figura 5.3 demonstra a tabela criada, com detalhes de um dos campos.

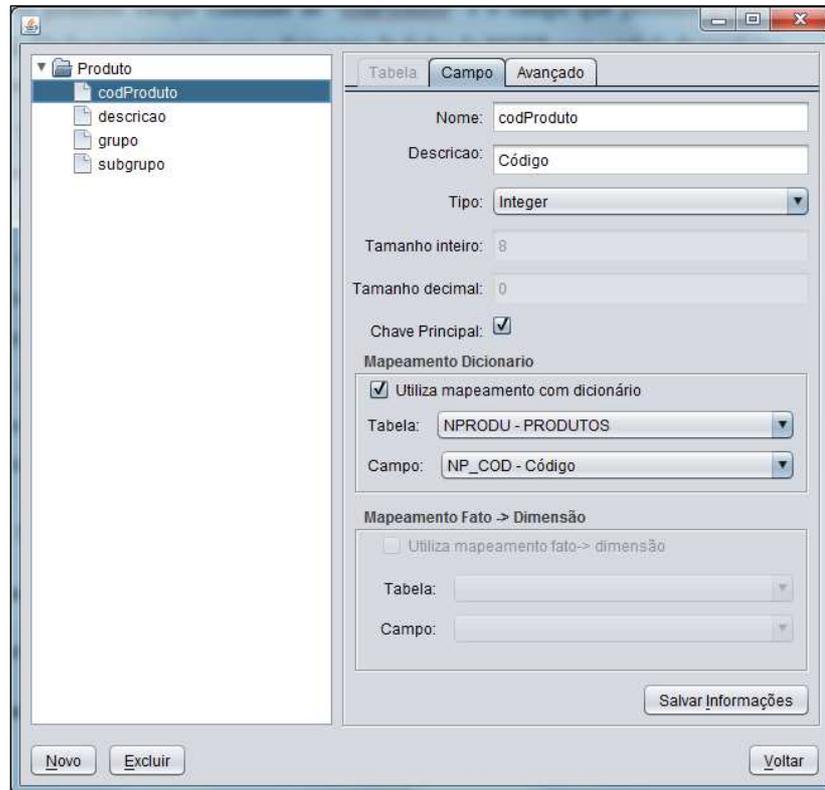


Figura 5.3 – Construção da tabela dimensão “Produto”

Fonte: Do próprio autor

O primeiro campo chamado de “codProduto” é a chave principal. Para o campo ser considerado como chave principal é necessário marcar a caixa de seleção com o nome de “Chave Principal”. Essa informação é importante para criar restrição de chave única na tabela e para o relacionamento com a tabela fato que será criada.

O campo faz mapeamento com o dicionário de dados do SIGER com a tabela de produtos e com o campo que representa o código do produto desta tabela do dicionário. O mapeamento do dicionário de dados, tanto do SIGER quanto do SGBD, é feito marcando a caixa de seleção “Utiliza mapeamento com dicionário”. Com isso, habilita as caixas de combinação (*combo-box*) da tabela e campo.

O segundo campo, chamado de “descricao”, possui a descrição do produto. Este campo tem mapeamento com o dicionário do SIGER com a tabela produto e o campo que contém a descrição.

O terceiro campo possui a informação do grupo de produto ao qual o produto faz parte. Tem o relacionamento com o dicionário do SIGER com a tabela de produtos com a informação do grupo de produto.

O quarto e último campo desta tabela dimensão é a informação sobre o subgrupo do grupo ao qual o produto pertence. Também possui relacionamento com o dicionário do SIGER na tabela produto que referencia o subgrupo.

5.1.3 Construção da tabela dimensão Tempo

A dimensão “Tempo” irá conter as informações temporais sobre o fato ocorrido. Como foi definido que a granularidade utilizada será o “dia”, é importante criar toda a estrutura desta tabela dimensão, contendo as informações de ano, mês e dia separadas para que seja possível criar os cubos utilizando essas informações para navegar entre eles.

A primeira informação da tabela dimensão “Tempo” é o “codTempo”. Esse campo vai conter a mesma informação da tabela fato “Vendas”. Por este motivo deve referenciar a tabela de pedidos e o campo data de faturamento.

O segundo campo da tabela dimensão é chamado de “ano”. Esse campo vai conter apenas o conteúdo do ano da data correspondente. Para a criação desse campo deve ser utilizado na aba “Avançado”, o campo fórmula, utilizando uma função específica do Oracle “TO_CHAR(NI_DAT, ‘yyyy’)” para separar de um campo data apenas a informação do ano, conforme a figura 5.4.

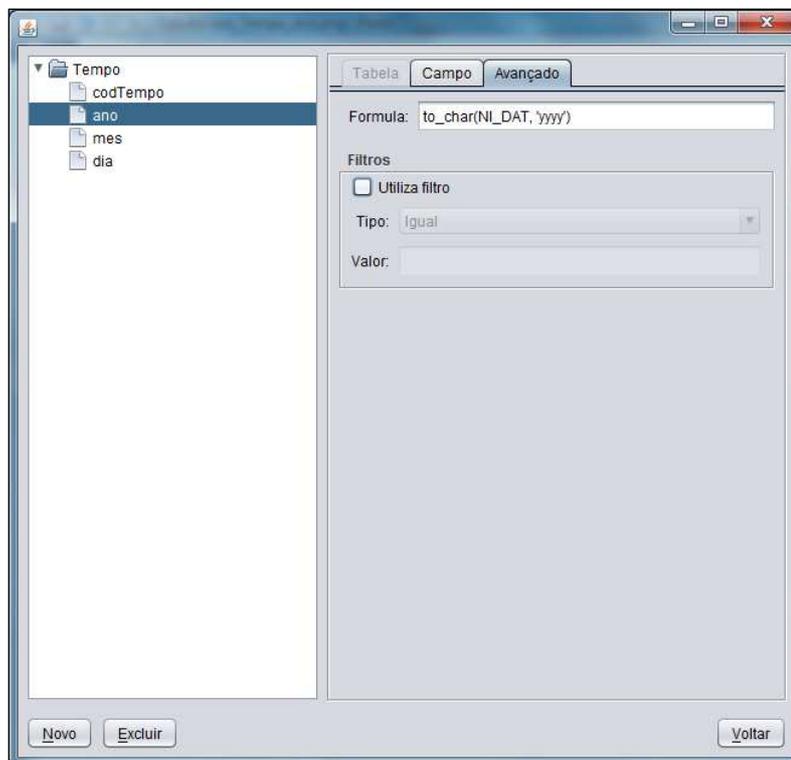


Figura 5.4 – Construção do campo virtual “ano” da tabela dimensão “Tempo”
Fonte: Do próprio autor

O terceiro campo chamado de “mês” contém a informação do mês. Nesse estudo de caso optou-se por disponibilizar o mês em formato texto, ou seja, “jan”, “fev”, etc. Para essa funcionalidade deve ser utilizada a função do Oracle como no campo anterior, porém informando que deve capturar essa informação da data. O comando utilizado foi o `”TO_CHAR(NI_DAT, 'MON')”`

O quarto e último campo desta tabela chamado de “dia” representa o dia de uma determinada data. O procedimento para criação é o mesmo utilizado para os campos anteriores, utilizando a função do Oracle `”TO_CHAR(NI_DAT, 'dd')”` para separar o dia de uma data.

5.1.4 Construção da tabela dimensão Cliente

A última tabela dimensão armazena a informações sobre o cliente que efetuou a compra, seja ela pessoa física ou jurídica. O primeiro campo possui a informação sobre o código do cliente (“codCliente”). O campo é chave principal desta tabela e faz o mapeamento com o dicionário do SIGER na tabela de clientes com o campo código do cliente.

O segundo campo da tabela dimensão diz respeito ao nome do cliente que efetuou a compra, chamado de “descricao”. Tem mapeamento com o dicionário do SIGER com a tabela de clientes e o campo descrição. Caso o cliente seja pessoa física contém o nome do cliente ou se for pessoa jurídica possui a razão social da empresa.

O terceiro campo desta tabela referencia o Estado em que o cliente se encontra. Para este campo é utilizado no formato de UF com dois dígitos, como por exemplo, “RS”, “SP”, etc. Tem relacionamento com o dicionário do SIGER na tabela cliente e o campo de Estado.

A quarta e quinta coluna tem as informações da cidade e o bairro em que o cliente se encontra respectivamente. As duas informações se relacionam com o dicionário do SIGER na tabela cliente (figura 5.5).

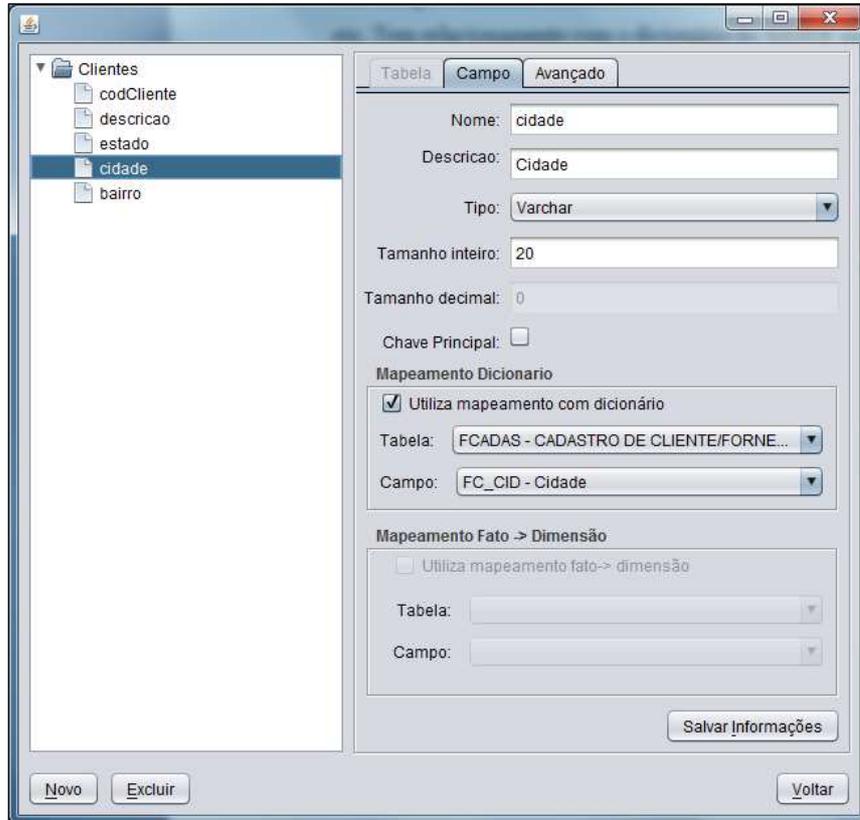


Figura 5.5 – Construção da tabela dimensão “Cliente”

Fonte: Do próprio autor

5.1.5 Construção da tabela fato Vendas

A tabela fato “Vendas” terá os dados factuais ocorridos em um determinado período de tempo, realizado por um cliente que adquiriu um determinado produto. Serão armazenadas informações como a quantidade vendida, o preço unitário e um valor total que representa a multiplicação do campo preço unitário com a quantidade vendida.

Conforme a figura 5.6, o primeiro campo a ser criado é o “codTempo” que contém a informação sobre o dia em que o fato ocorreu. Neste cenário foi definido como granularidade o dia, sendo armazenada a informação com ano, mês e dia. Esse campo vai ter relação com o dicionário de dados do SIGER, referenciado pela tabela de pedidos e o campo que tem a informação da data do faturamento. Como este campo é uma data completa, foi utilizado como tipo “Date”. Também deve ser feito o relacionamento deste campo com o “codTempo” da tabela dimensão “Tempo” e definido como um campo de chave principal.

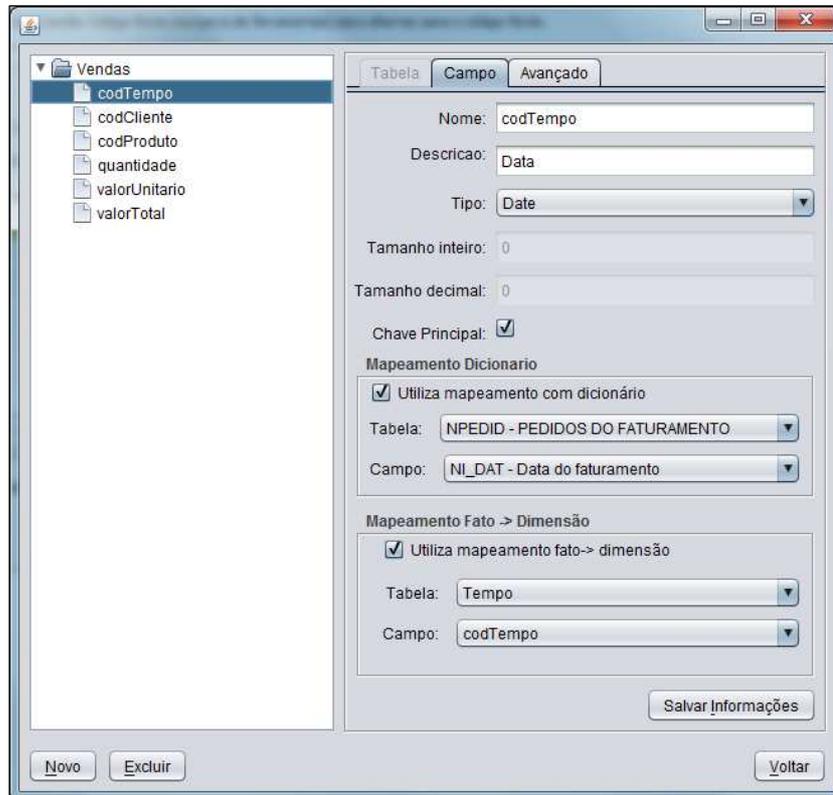


Figura 5.6 – Construção do campo “codTempo” da tabela fato “Vendas”

Fonte: Do próprio autor

O mapeamento entre a tabela fato com a tabela dimensão é realizado marcando a caixa de seleção “Utiliza mapeamento fato -> dimensão”. Com isso é habilitado a caixa de combinação para seleção das tabelas dimensões e o campo chave da tabela dimensão selecionada. Esse mapeamento é importante para interligar o campo da tabela fato com a tabela dimensão para a criação dos cubos de dados.

A segunda informação da tabela fato é o “codCliente”. Esse campo terá mapeamento com dicionário do SIGER da tabela de pedidos com o campo que possui a informação do código do cliente. Relaciona-se com a tabela dimensão “Cliente”, com campo “codCliente”. Esse campo também faz parte da chave principal da tabela fato. Como o campo é proveniente do dicionário, o tipo de dado é definido automaticamente de acordo com o tipo de dado do campo original.

O terceiro campo é o “codProduto”, o qual referencia o dicionário do SIGER, tabela de pedidos com o campo contendo o código do produto vendido. Relaciona-se com a tabela dimensão “Produto” com o campo “codProduto” e faz parte da chave principal.

O quarto campo da tabela fato contém a informação sobre o fato ocorrido em um determinado dia, com um cliente e um produto. Essa informação se refere à quantidade

vendida do produto, chamado de “quantidade”. Esse campo não faz parte da chave principal, mas tem relação com dicionário do SIGER na tabela de pedidos com o campo de quantidade vendida do produto.

A quinta informação presente na tabela armazena o valor unitário de cada produto vendido. Referencia o dicionário do SIGER na tabela pedido com a informação sobre o valor do produto vendido.

O sexto e último campo da tabela fato “Vendas” é um campo virtual, pois seu valor é calculado utilizando outros dois valores. Nesse campo está a informação sobre o valor total da venda realizada, calculando o preço unitário do produto com a quantidade vendida. Como o campo não tem relação direta com o dicionário de dados do SIGER é necessário definir seu tipo de dado, o qual foi definido como “Decimal (10,2)”. Para realizar o cálculo foi utilizado na aba “Avançado” o campo fórmula, onde foram definidos quais os campos que devem ser multiplicados, conforme a figura 5.7.

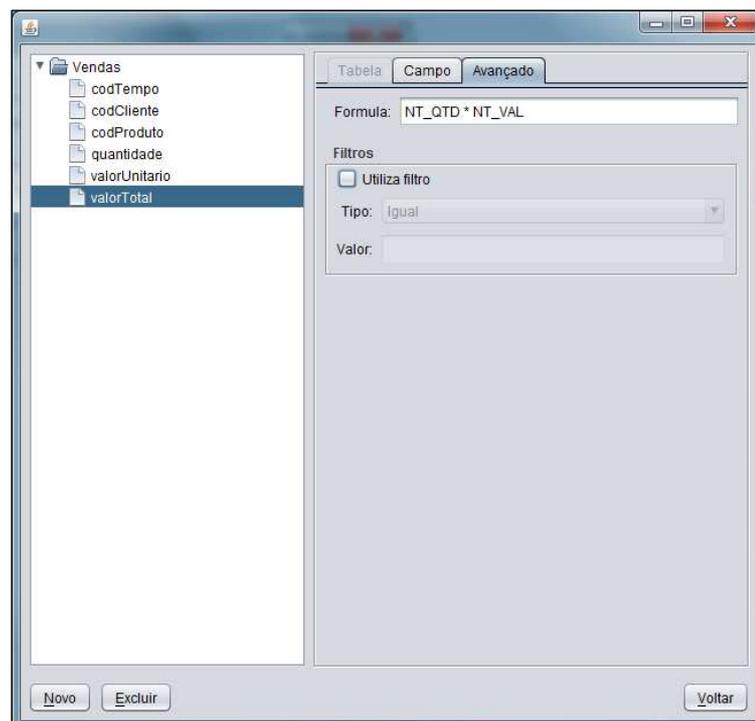


Figura 5.7 – Construção do campo virtual “valorTotal” da tabela fato “Vendas”

Fonte: Do próprio autor

Com isso, encerra-se a construção da tabela fato, contendo as informações necessárias para a posterior análise dos fatos ocorridos.

5.2 CARGA DOS DADOS

O processo de carga dos dados é automatizado pela ferramenta com base nas informações cadastradas no processo de criação do DM descrito anteriormente. Neste estudo de caso, todas as informações virão diretamente das tabelas do ERP SIGER, a partir do relacionamento feito com o dicionário de dados do ERP.

As informações são sempre carregadas de forma total, ou seja, são excluídos todos os registros das tabelas fato e dimensões e importado todos os dados novamente. Este processo é simples, automático e funciona sem interações com usuário no decorrer da execução. Pode ser executado através da ferramenta ou por um agendamento no sistema operacional utilizando uma chamada via linha de comando.

Para cada tabela criada, seja fato ou dimensão, é feito um processamento com algumas etapas como: exclusão da tabela existente, a criação da tabela, geração da transformação e carga dos dados pelo Kettle. A figura 5.8 mostra a janela de processamento de uma das tabelas.

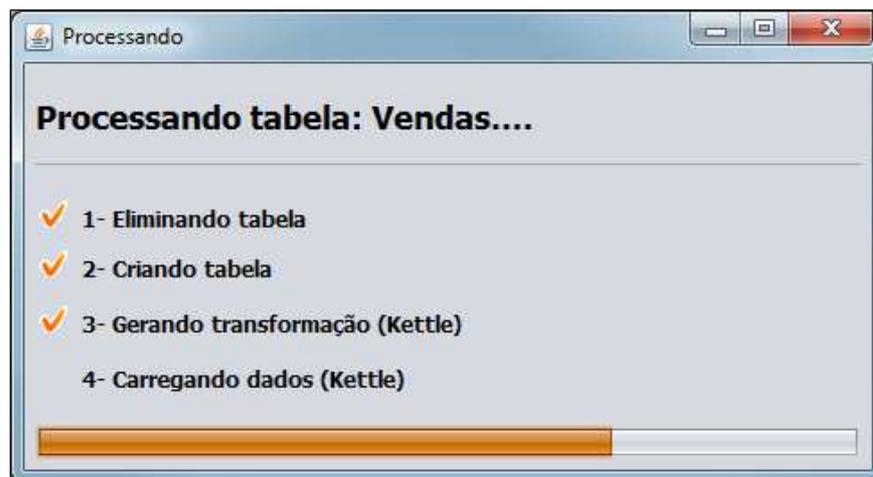


Figura 5.8 – Processamento de carga dos dados da tabela fato “Vendas”
Fonte: Do próprio autor

5.3 CRIAÇÃO DOS CUBOS DE DADOS

Conforme exposto anteriormente, é possível criar vários cubos de dados para um determinado DM. A fim de demonstrar este recurso, optou-se pela construção de dois cubos de dados para trabalhar com diferentes visões sobre o mesmo DM construído. As duas visões trabalham com a tabela fato de vendas e as dimensões de tempo, produtos e clientes. A diferença entre eles consiste nos campos utilizados e, principalmente, no agrupamento realizado, que foi diferente, permitindo criar uma navegação específica para cada situação no DM.

5.3.1 Vendas x Produto x Estado

O primeiro cubo construído tem como finalidade agrupar as vendas realizadas dos produtos agrupando os clientes por estado. Inicia-se a criação do cubo informando um nome para o mesmo e uma breve descrição para facilitar o entendimento do cubo construído. Após deve ser definido qual a tabela fato desse cubo e os campos que vão ser utilizados para a análise e consulta dessa visão dos dados. A figura 5.9 representa a criação desse cubo, e a escolha da tabela fato.



Figura 5.9 – Criação do cubo Vendas x Produto x Estado

Fonte: Do próprio autor

O próximo passo para criação do cubo de dados é definir quais as dimensões que vão fazer parte desta visão de dados. A primeira dimensão escolhida foi a dimensão “Tempo” que vai conter as informações sobre o período em que os fatos ocorreram, omitindo a informação de “dia” presente no DM, fazendo com que a navegação desse cubo tenha como nível mais baixo o mês (figura 5.10).



Figura 5.10 – Criação da dimensão “Tempo” do cubo Vendas x Produto x Estado

Fonte: Do próprio autor

A próxima dimensão presente no cubo é a dimensão “Produto”, que vai conter apenas a informação sobre a descrição do produto, ignorando demais informações presentes nesta tabela dimensão, conforme a figura 5.11.



Figura 5.11 – Criação da dimensão “Produto” do cubo Vendas x Produto x Estado

Fonte: Do próprio autor

É interessante destacar que o cubo pode ser construído de acordo com a necessidade de visualização dos dados pelos usuários. No caso do produto foram ignoradas de propósito as demais informações presentes no DM para exemplificar que este processo fica a critério do analista, sendo possível disponibilizar todas as informações, com diferentes ordens de navegação ou omitir certas informações em determinado cubo de dados.

A última dimensão a ser definida para este cubo foi a dimensão “Cliente” que é a dimensão que vai ser utilizada para a análise desses dados. Como a finalidade deste cubo é

visualizar as vendas de produtos por estado do cliente, é interessante que seja adicionado ao cubo além da informação do estado, a cidade em que o cliente se encontra, permitindo realizar operações de *drill-down* no DM.

Dessa forma o cubo deve obedecer obrigatoriamente à navegação entre estado e cidade. Para isso deve ser criada a estrutura de forma ordenada utilizando os botões com imagem de “seta para cima” e “seta para baixo”, criando um nível hierárquico entre eles. A ordem que os campos são adicionados no cubo de dados implica na ordem que serão criados os agrupamentos. Essa ordem pode ser alterada utilizando os botões com imagem de “flecha para cima” ou “flecha para baixo”. A figura 5.12 apresenta a criação da dimensão “Cliente”.



Figura 5.12 – Criação da dimensão “Cliente” do cubo Vendas x Produto x Estado
Fonte: Do próprio autor

Após definidas todas as informações do cubo de dados, automaticamente é gerado o arquivo XML contendo as informações deste cubo para a análise e consulta utilizando a ferramenta OLAP.

5.3.2 Vendas x Grupo Produto x Cliente

O segundo cubo a ser construído para este estudo de caso vai permitir a análise das vendas realizadas pelos clientes tendo uma visão mais detalhada para os produtos que foram vendidos, com agrupamento pelo grupo de produtos.

As etapas para criação do cubo são praticamente as mesmas do cubo criado anteriormente. A maior diferença é que a dimensão produto vai conter as informações sobre o grupo e o subgrupo de produtos, conforme a figura 5.13.



Figura 5.13 – Criação da dimensão “Produto” do cubo Vendas x Grupo Produto x Cliente
Fonte: Do próprio autor

Neste cubo de dados a dimensão “Tempo” vai permitir a navegação de todo o período, ou seja, vai conter informações do ano, mês e dia. A dimensão cliente vai conter apenas a informação da descrição do cliente.

5.4 ANÁLISE DOS CUBOS DE DADOS

Após a construção do DM, carga dos dados e criação dos cubos de dados, está tudo pronto para que o gestor possa analisar os dados usando uma ferramenta OLAP, para que este possa interpretar os dados e transformá-los em informações para auxiliar o processo decisório. Conforme indicado anteriormente, as consultas OLAP podem ser realizadas em qualquer ferramenta OLAP compatível com o Mondrian. No estudo de caso foi utilizado o JPivot para realizar algumas consultas e análises considerando estes dois cubos de dados construídos, utilizando a base de dados já carregada anteriormente no DM.

5.4.1 Análise do cubo Vendas x Produto x Estado

Utilizando a ferramenta JPivot para navegar no cubo de dados criado, foi possível identificar nesta visão que os maiores compradores de produtos desta empresa no ano de 2010 estão situados no estado do Rio Grande do Sul, seguido pelo estado de São Paulo e posteriormente as vendas para o Exterior (sigla de estado “EX”). A figura 5.14 demonstra a análise realizada, incluindo um gráfico de barras vertical.

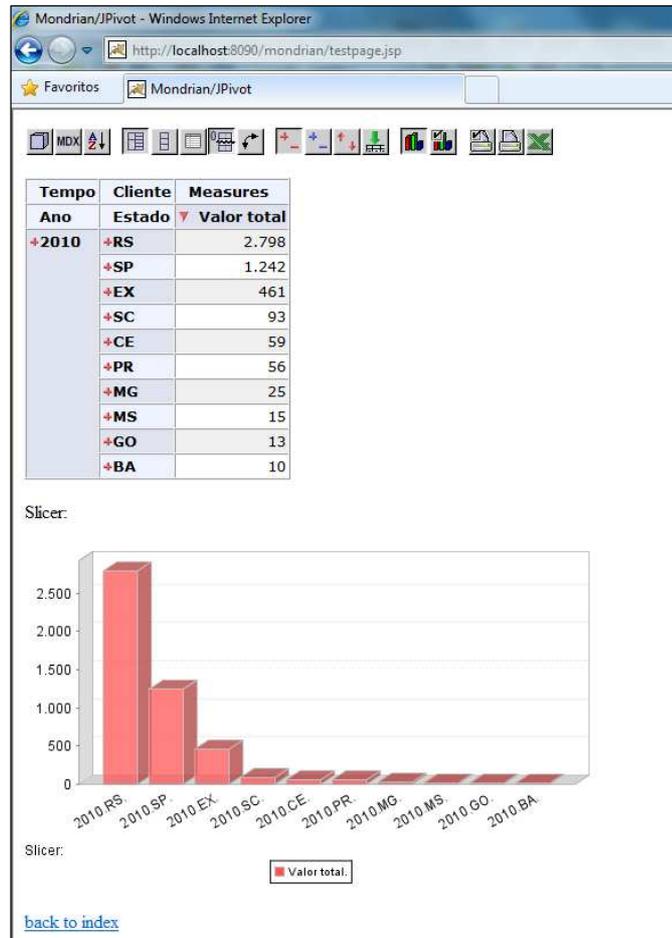


Figura 5.14 – Análise realizada no cubo Vendas x Produto x Estado
Fonte: Do próprio autor

Outras análises podem ser realizadas, como por exemplo, uma navegação *drill-down* no estado de São Paulo para identificar em quais as cidades onde mais ocorreram essas vendas (figura 5.15). Esta informação pode levar o tomador de decisão a talvez investir em mais uma unidade da empresa nessa região ou proximidades ou até a repensar novas estratégias para regiões com volume de vendas menor.

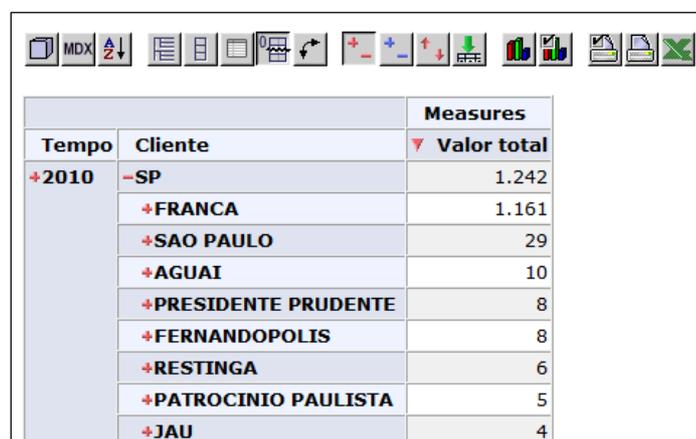


Figura 5.15 – Operação *drill-down* realizada no estado de São Paulo
Fonte: Do próprio autor

5.4.2 Análise do cubo Vendas x Grupo Produto x Cliente

O outro cubo de dados montado tem como finalidade identificar as vendas realizadas para os clientes no ano de 2010, agrupando pelo grupo de produto. Para essa análise os grupos de produtos foram renomeados para nomes genéricos como “Grupo 36”, “Grupo 37” para manter a confidencialidade da base de dados utilizada para a análise.

A análise realizada mostra no ano de 2010 os cinco grupos de produtos que mais tiveram valor de venda. Essa análise utilizou a função de *ranking* selecionando apenas os cinco com as maiores vendas e a ordenação pelo valor total das vendas. Um gráfico de barras vertical foi utilizado para demonstrar essa análise conforme demonstra a figura 5.16.

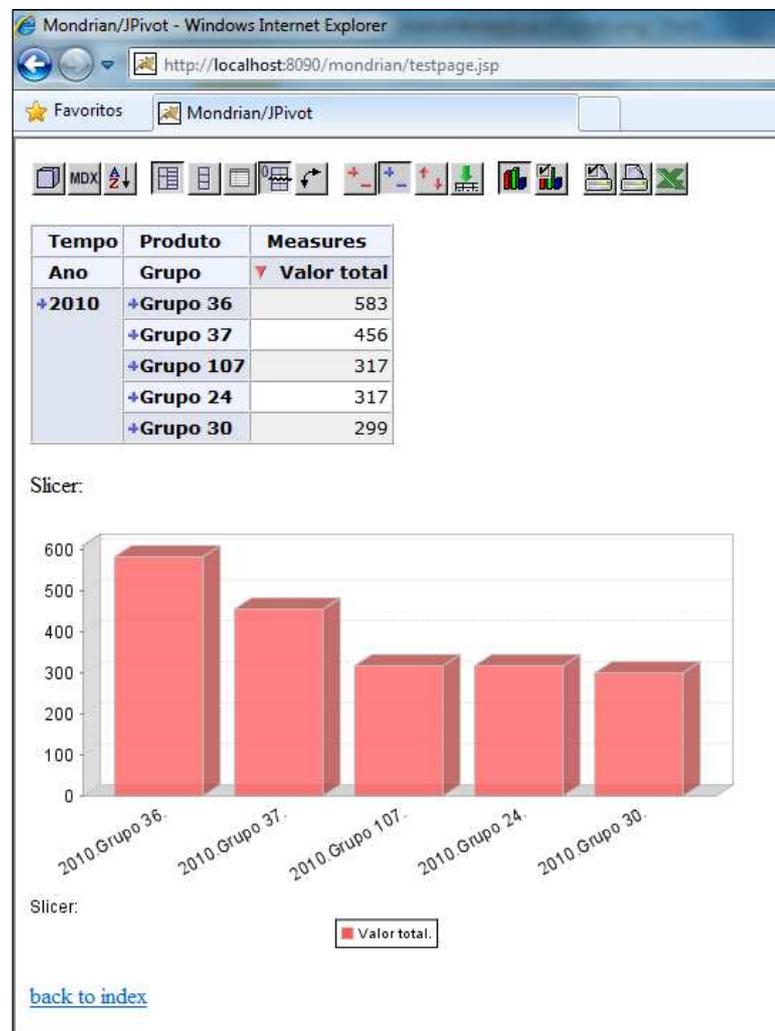


Figura 5.16 – Análise realizada no cubo Vendas x Grupo Produto x Cliente

Fonte: Do próprio autor

O cubo de dados ainda permite outras análises e consultas OLAP conforme as funções disponibilizadas pela barra de ferramentas do JPIVOT.

CONCLUSÃO

Os sistemas de BI são reconhecidos no mercado pela sua capacidade de reunir e consolidar informações oriundas dos diferentes sistemas de uma empresa. Desta forma, o BI torna-se uma importante ferramenta para os gestores que desejam obter vantagens competitivas através da tomada de decisões rápidas e acertadas.

Visando reunir as funcionalidades do ERP SIGER, com os benefícios de uma solução de BI, este trabalho apresentou a proposta de uma solução completa de BI integrada ao ERP SIGER. A solução é composta pela ferramenta proposta para construção do DM, em conjunto com o Kettle para o processo de extração, transformação e carga dos dados, e com o Mondrian para as consultas OLAP.

Desta forma, facilmente podem ser construídos os cubos de dados para atender os diversos ramos de atividades que utilizam o ERP SIGER. Sendo assim, as informações contidas no ERP SIGER estarão disponíveis para os gestores das empresas no momento em que forem necessárias, facilitando a tomada de decisões.

O estudo de caso realizado demonstra que a solução atende diversas situações, tendo flexibilidade para obtenção dos dados contidos no ERP SIGER, onde puderam ser analisados a fim de obter informações importantes para o processo decisório.

Durante o desenvolvimento do trabalho, foram identificadas funcionalidades que podem ser desenvolvidas como trabalhos futuros para enriquecer a solução, mas que para este trabalho não foram abordadas pelo tempo necessário para análise e desenvolvimento, bem como não foram previstas no escopo do anteprojeto.

Dentre as principais funcionalidades estão:

- Criação de um portal para visualização e agrupamento dos cubos de dados, permitindo que usuário possa realizar um *login*, com permissões para os usuários acessarem apenas alguns cubos de dados, a fim de evitar que informações confidenciais sejam acessadas por qualquer usuário;
- Permitir ao usuário salvar as consultas realizadas no cubo de dados, para que seja possível reutilizar as consultas quando necessário;
- Permitir a utilização de arquivos texto separados por vírgula, utilizando as informações da mesma forma como é feito para os dicionários de dados.

REFERÊNCIAS

- BARBIERI, Carlos. **Business Intelligence: Modelagem & Tecnologia**. Rio de Janeiro, RJ: Axcel Books, 2001. 424 p.
- BINDER, Fábio Vinicius. **Sistemas de Apoio à Decisão**. São Paulo, SP: Érica, 1994. 98p.
- DAVENPORT, Thomas H. **Missão Crítica: obtendo vantagens competitivas com os sistemas de gestão empresarial**. Porto Alegre, RS: Bookman, 2002. 293 p.
- HABERKORN, Ernesto Mário. **Gestão empresarial com ERP**. São Paulo, 2004. 674 p.
- INMON, Willian H. **Como construir o Data Warehouse**. 2. ed. Rio de Janeiro, RJ: Campus, 1997. 388 p.
- INMON, Willian H; TERDEMAN, R. H.; IMHOFF, Claudia. **Data Warehousing: como transformar informações em oportunidades de negócios**. São Paulo, SP: Berkely, 2001. 266 p.
- INMON, William H.; WELCH, J. D.; GLASSEY, Katherine L. **Gerenciando Data Warehouse**. São Paulo, SP: Makron Books, 1999. 375 p.
- KIMBALL, Ralph. **Data Warehouse Toolkit**. São Paulo, SP: Makron Books, 1998. 388 p.
- MACHADO, Felipe Nery Rodrigues. **Projeto de Data Warehouse: uma visão multidimensional**. 1. ed. São Paulo, SP: Érica, 2000. 248 p.
- MICROSOFT, **SQL Server Integration Services**. Disponível em: <<http://www.microsoft.com/sqlserver/2005/en/us/integration-services.aspx>>. Acesso em: 20 jun. 2010.
- MONDRIAN, **Mondrian Project**. Disponível em: <<http://mondrian.pentaho.org>>. Acesso em: 21 jun. 2010.
- PENTAHO. **Pentaho BI Suite**. Disponível em: <<http://www.pentaho.com>>. Acesso em: 18 abr. 2010.
- RECH. **Sistema Integrado Gestão Empresarial Rech (SIGER)**. Rio Grande do Sul: Novo Hamburgo. Disponível em: <<http://www.rech.com.br>>. Acesso em: 10 mai. 2010.
- SADIG. **Sadig BI**. Disponível em: <<http://www.sadig.com>>. Acesso em: 17 abr. 2010.
- SINGH, Harry. **Data Warehouse**. São Paulo, SP: Makron Books, 2001. 382 p.