

CENTRO UNIVERSITÁRIO FEEVALE

JONATAS CABERLON

UMA SOLUÇÃO BASEADA EM FERRAMENTAS *ORACLE*,  
PARA O DESENVOLVIMENTO DE UM SISTEMA DE *BUSINESS*  
*INTELLIGENCE*

Novo Hamburgo, novembro de 2007.

JONATAS CABERLON

UMA SOLUÇÃO BASEADA EM FERRAMENTAS *ORACLE*,  
PARA O DESENVOLVIMENTO DE UM SISTEMA DE *BUSINESS*  
*INTELLIGENCE*

Centro Universitário Feevale  
Instituto de Ciências Exatas e Tecnológicas  
Curso de Ciência da Computação  
Trabalho de Conclusão de Curso

Professor orientador: Juliano Varella de Carvalho

Novo Hamburgo, novembro de 2007.

## AGRADECIMENTOS

Gostaria de agradecer primeiramente aos meus pais que, com muita luta, puderam me dar a oportunidade de cursar uma faculdade.

Aos colegas de faculdade, por estarem comigo nos momentos mais difíceis do curso.

Aos colegas de trabalho, que de alguma forma, trouxeram contribuição a esse trabalho.

E ao professor-orientador Juliano, pelo seu brilhante apoio no desenvolvimento desse trabalho.

## RESUMO

O cenário globalizado atual exige uma intensa busca pela excelência na atuação das empresas no mercado, assim como a adaptação às constantes mudanças de cenários econômicos. Para isso, o uso de sistemas de informação que auxiliem a tomada de decisão é de suma importância. As empresas, para reagir aos concorrentes, clientes, fornecedores, mudanças sociais e tecnológicas, constroem sistemas para auxiliar neste ambiente dinâmico. Portanto, a implantação de um sistema de *Business Intelligence* poderá ser fundamental para o processo decisório da empresa. A Box Print, empresa de embalagens, possui todos os seus processos informatizados com um grande volume e variabilidade de dados. Porém, não há sistema a nível gerencial, que traga em um mesmo ambiente, consultas e relatórios que auxiliem a tomada de decisão de gerentes e diretores. Utilizando tecnologia *Oracle*, esse trabalho tem por objetivo desenvolver um sistema de *Business Intelligence* para um *Data Mart* comercial, que primará em facilitar a obtenção dos dados necessários para o processo decisório a nível gerencial.

Palavras-chave: *Business Intelligence*. OLAP. *Data Warehouse*. *Oracle*. Sistema de Apoio à Decisão (SAD)

## ABSTRACT

The current globalized scenery demands an intense search for the excellence in company's performance at the market as well as adaptation to the constant changes of economic scenes. Because of this, the use of systems of information that helps in the decision taking is of utmost importance. The companies, to react to the competitors, clients, suppliers, social and technologies changes, build systems to helps on this dynamic work's place. Therefore, the implantation of Business Intelligence System can be fundamental for this decision process of the company. Box Print, packaging company, has all of its computerized process with a great amount and variability of data. However, there is no system at a managerial level, that brings at same work's place, consultation and reports that helps the decision taking of managers and directors. Using Oracle technology, this final paper has as objective to develop a system of Business Intelligence for a Data Mart commercial, which will ease the attainment of necessary data to the decisive process at a managerial level.

Keywords: Business Intelligence. OLAP. Data Warehouse. Oracle. Decision Support Systems (DSS)

## LISTA DE FIGURAS

Figura 1.1 – Processo de tratamento de dados.....	21
Figura 1.2 – A questão da não-volatilidade.....	22
Figura 1.3 – Composição básica de uma tabela de fatos.....	25
Figura 1.4 – Exemplo de uma tabela de dimensão.....	26
Figura 1.5 – Exemplo de um esquema em estrela.....	28
Figura 1.6 – Exemplo de um esquema em <i>snowflake</i> .....	29
Figura 1.7 – Cubo de dados.....	31
Figura 1.8 – Níveis de granularidade.....	32
Figura 1.9 – Tabelas agregadas.....	33
Figura 1.10 – Exemplo das funcionalidades <i>Drill-down</i> e <i>Roll-up</i> .....	37
Figura 2.1 – Exemplo de um <i>wizard</i> na criação de tabelas no OWB.....	43
Figura 2.2 – <i>Oracle Warehouse Builder</i> .....	44
Figura 2.3 – Componentes do <i>Oracle Discoverer</i> .....	50
Figura 2.4 – <i>Discoverer Administrator</i> .....	51
Figura 2.5 – Estrutura do <i>Discoverer Desktop</i> .....	52
Figura 2.6 – Exemplo de consulta com tabela de referência cruzada.....	53
Figura 3.1 – Estrutura organizacional da empresa.....	63
Figura 3.2 – Organização do gerenciamento de vendas.....	64
Figura 4.1 – Estrutura do sistema.....	68
Figura 4.2 – Modelagem do sistema.....	69
Figura 4.3 – Hierarquia Principal do sistema.....	70
Figura 4.4 – Dimensão SGC_DIM_CIDADE.....	71
Figura 4.5 – Tabela de fatos de faturamento.....	72
Figura 4.6 – Tabela de fatos de devolução.....	73
Figura 4.7 – Relatório “Faturamento por Segmento”.....	74

Figura 4.8 – Agregado SGC_AGR_SEGMENTO.....	75
Figura 4.9 – <i>Stored Procedure</i> de carga de dados.....	78
Figura 4.10 – Pastas do <i>Discoverer</i> .....	80
Figura 4.11 – Hierarquia da data de emissão das notas fiscais.....	81
Figura 5.1 – Tela Inicial do Sistema.....	83
Figura 5.2 – Tela “Faturamento Líquido por Empresa (Mensal/Valor)” .....	86
Figura 5.3 – Faturamento Líquido por Empresa (Detalhado).....	87
Figura 5.4 – Exemplo de gráficos do módulo tático.....	88
Figura 5.5 – Análise: “Faturamento Líquido por Empresa(Mensal-Valor)” .....	89
Figura 5.6 – Menu de Ações.....	90
Figura 5.7 – Tela de exportação.....	91
Figura 5.8 – Tela para envio de <i>e-mail</i> .....	92
Figura 5.9 – Ferramenta de <i>layout</i> .....	93
Figura 5.10 – Ferramenta de edição de <i>layout</i> .....	93
Figura 5.11 – Exemplo de <i>pivoting</i> .....	94
Figura 5.12 – Opções de formato.....	94
Figura 5.13 – Exemplo de <i>stoplight</i> .....	95
Figura 5.14 – Configuração de linhas e colunas.....	95
Figura 5.15 – Menu de <i>Drill-Down</i> e <i>Roll-up</i> .....	96
Figura 5.16 – Exemplo de <i>Drill-down</i> em gráficos.....	97
Figura 5.17 – Ferramentas de gráficos.....	97
Figura 5.18 – Tela com opção de parâmetros.....	98
Figura 5.19 – Tabelas do controle de acessos.....	99

## LISTA DE QUADROS

Quadro 1.1 – Comparação entre os dados de natureza operacional e informacional.	19
Quadro 1.2 – Diferenças entre DW e DM.....	34
Quadro 2.1 – Recursos das ferramentas de <i>Data Warehouse</i> .....	48
Quadro 2.2 – Comparativo de ferramentas <i>Data Warehouse</i> .....	48
Quadro 2.3 – Recursos das ferramentas OLAP.....	59
Quadro 2.4 – Comparação das ferramentas OLAP.....	60

## LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
BD	Banco de Dados
BI	<i>Business Intelligence</i>
DM	<i>Data Mart</i>
DOLAP	<i>Desktop On-line Analytic Processing</i>
DW	<i>Data Warehouse</i>
ER	Entidade-relacionamento
ERP	<i>Enterprise Resource Planning</i>
ETL	<i>Extraction, Transformation and Loading</i>
FK	<i>Foreign Key</i>
HOLAP	<i>Hybrid On-line Analytic Processing</i>
HTML	<i>HyperText Markup Language</i>
IAS	<i>Internet Application Server</i>
J2EE	<i>Java 2Platform Enterprise Edition</i>
JCP	<i>Java Community Process</i>
JDBC	<i>Java Database Connectivity</i>
JOLAP	<i>Java On-line Analytic Processing</i>
MOLAP	<i>Multidimensional On-Line Analytic Processing</i>
ODBC	<i>Open Database Connectivity</i>

OLAP	<i>On-line Analytic Processing</i>
OLTP	<i>On-line Transactional Processing</i>
OWB	<i>Oracle Warehouse Builder</i>
PDF	<i>Portable Document Format</i>
PK	<i>Primary Key</i>
PL/SQL	<i>Procedural Language/Structured Query Language</i>
RDL	<i>Report Definition Language</i>
ROLAP	<i>Relational On-Line Analytic Processing</i>
SAD	Sistema de Apoio à Decisão
SGBD	Sistema Gerenciador de Banco de Dados
SGBDR	Sistema Gerenciador de Banco de Dados Relacional
SGC	Sistema Gerencial Comercial
SQL	<i>Structured Query Language</i>
TI	Tecnologia da Informação
UIX	<i>User Interface XML</i>
WOLAP	<i>Web On-line Analytic Processing</i>
XML	<i>EXtensible Markup Language</i>
XMLA	<i>EXtensible Markup Language for Analysis</i>

## SUMÁRIO

<b>INTRODUÇÃO.....</b>	<b>13</b>
<b>1 BUSINESS INTELLIGENCE (BI).....</b>	<b>17</b>
1.1 Dados Operacionais x Dados Informativos .....	18
1.2 <i>Data Warehouse</i> .....	19
1.2.1 Características do DW .....	20
1.2.1.1 Orientado a assuntos.....	20
1.2.1.2 Integrado .....	20
1.2.1.3 Não-Volátil .....	21
1.2.1.4 Variável com o tempo.....	22
1.2.2 Componentes do DW .....	22
1.2.3 Modelagem de Dados.....	23
1.2.3.1 Tabela de Fatos .....	24
1.2.3.1.1 Atributos Aditivos, semi-aditivos e não aditivos .....	25
1.2.3.2 Tabelas de Dimensões .....	26
1.2.3.3 Técnicas de Modelagem .....	27
1.2.3.3.1 <i>Star Schema</i> .....	27
1.2.3.3.2 <i>Snowflake</i> .....	29
1.2.3.4 Cubo de dados .....	30
1.2.3.5 Granularidade.....	31
1.2.3.6 Agregados .....	32
1.2.4 <i>Data Mart</i> (DM).....	34
1.3 OLAP.....	35
1.3.1 Características.....	35
1.3.1.1 Funções Básicas.....	36
1.3.1.2 Operações OLAP.....	36
1.3.2 Arquiteturas OLAP .....	38
1.3.2.1 ROLAP .....	38
1.3.2.2 MOLAP .....	39
1.3.2.3 ROLAP x MOLAP, qual a melhor tecnologia OLAP? .....	40
<b>2 FERRAMENTAS.....</b>	<b>42</b>
2.1 Ferramentas DW .....	42
2.1.1 <i>Oracle Warehouse Builder</i> .....	42
2.1.1.1 A Ferramenta.....	42
2.1.1.2 Características Principais.....	43
2.1.1.2.1 Orientado a Projetos.....	43
2.1.1.2.2 Fontes de dados .....	44

2.1.1.2.3 Importação de metadados.....	45
2.1.2 Oracle Express.....	45
2.1.3 Microsoft DTS.....	46
2.1.4 Business Information Warehouse.....	47
2.1.5 Comparativo das ferramentas DW.....	48
2.2 Ferramentas OLAP.....	49
2.2.1 Oracle Business Intelligence.....	49
2.2.1.1 Oracle Discoverer.....	50
2.2.1.1.1 Discoverer Administrador.....	51
2.2.1.1.2 Discoverer Desktop.....	52
2.2.1.1.3 Discoverer Plus e Discoverer Viewer.....	55
2.2.1.2 Oracle Spreadsheet Add-In.....	55
2.2.1.3 Oracle BI Beans.....	55
2.2.2 Microsoft Analysis Services.....	56
2.2.3 Pentaho BI.....	57
2.2.4 OpenI.....	58
2.2.5 Comparativo das ferramentas OLAP.....	59
<b>3 ESTUDO DE CASO.....</b>	<b>61</b>
3.1 Estrutura da Empresa.....	61
3.1.1 Tecnologia.....	62
3.1.2 Estrutura Organizacional.....	63
3.2 Problemas Atuais e Necessidades.....	64
3.3 Por que Oracle.....	65
<b>4 ESTRUTURA DO SISTEMA IMPLEMENTADO.....</b>	<b>67</b>
4.1 Arquitetura do Sistema.....	67
4.2 Modelagem do Sistema.....	68
4.2.1 Granularidade.....	69
4.2.2 Dimensões e Fatos.....	70
4.2.2.1 Dimensões.....	71
4.2.2.2 Fato SGC_FAT_FATURAMENTO.....	71
4.2.2.3 Fato SGC_FAT_DEVOLUCAO.....	73
4.2.3 Agregados.....	73
4.2.3.1 Agregado SGC_AGR_SEGMENTO.....	75
4.2.3.2 Agregado SGC_AGR_ESTADO.....	76
4.2.3.3 Agregado SGC_AGR_GRUPO_ECONOMICO.....	76
4.3 Carga de Dados.....	77
4.4 Uso do Oracle Warehouse Builder.....	78
4.5 Metadados.....	79
4.5.1 Metadados criados.....	80
<b>5 BOX BI.....</b>	<b>82</b>
5.1 Tela Inicial.....	82
5.1.1 Menu Principal.....	84
5.2 Telas do Sistema.....	85
5.2.1 Funcionalidades do Sistema.....	89
5.2.1.1 Menu de Ações.....	90
5.2.1.1.1 Exportação para PDF e planilha Excel.....	90
5.2.1.1.2 Imprimir.....	91
5.2.1.1.3 Exportar.....	91
5.2.1.1.4 Enviar como e-mail.....	91

5.2.1.2 Ferramentas .....	92
5.2.1.2.1 <i>Layout</i> .....	92
5.2.1.2.2 <i>Formato</i> .....	94
5.2.1.2.3 <i>Stoplight</i> .....	94
5.2.1.2.4 <i>Linhas e Colunas</i> .....	95
5.2.1.2.5 <i>Drill-down e Roll-up</i> .....	95
5.2.1.2.6 Ferramentas dos Gráficos.....	97
5.2.1.2.7 Parâmetros.....	97
5.3 Controle de Acessos .....	98
5.3.1 Controle de Consultas .....	98
5.3.2 Controle de Informações .....	98
<b>CONCLUSÃO.....</b>	<b>100</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>102</b>

## INTRODUÇÃO

Uma empresa que possui grande variabilidade e volume de dados tem consigo a possibilidade de obter as melhores condições para as tomadas de decisão, que muitas vezes resultam nos rumos que ela deve tomar. Mas para que essas informações sejam utilizadas de forma positiva, é necessário que as mesmas estejam devidamente organizadas e padronizadas, para evitar que elas sejam conflitantes ou duvidosas, gerando decisões erradas.

Uma das grandes dificuldades é usufruir dos dados de modo eficiente, pois a extração destes é sempre complexa, já que estão muitas vezes divididos em diversos locais da base, e muitas vezes da própria empresa, acarretando demora para agrupá-los, interpretá-los e extrair conclusões precisas.

Esta realidade está presente na empresa (Box Print) em que o projeto será realizado. Atualmente, o volume e a variabilidade de dados nesta corporação são muito grandes. Existem consultas e relatórios utilizados dentro do sistema transacional da empresa, que geram dados gerenciais. Além disso, planilhas paralelas muitas vezes são geradas, através de dados operacionais, para que se obtenham informações a nível gerencial que possibilitem a tomada de decisão.

No cenário atual da empresa, sempre que algum diretor ou gerente tem a necessidade de ver informações que lhe possibilitem a tomada de decisão, ele acaba designando um funcionário para garimpar, dentro do sistema transacional, dados de várias consultas e relatórios.

Essa coleta implica em uma dedicação e tempo muito grandes, além disso, há sempre a probabilidade de que as informações não sejam confiáveis, pois o funcionário designado para a tarefa não possui a visão de um gerente ou diretor, podendo cometer o erro de adicionar ou até mesmo retirar dados que seriam importantes para a tomada de decisão.

Para resolver os problemas descritos, surge então a necessidade de desenvolver um sistema de informações que possibilite aos gerentes e diretores obter a informação de forma mais macro e com celeridade, interpretá-la e tomar decisões precisas.

Segundo Poloni (2000), os sistemas de informação constituem-se de qualquer sistema usado para fornecer informações, independente da sua utilização. E, de acordo com Laudon & Laudon (1999) os sistemas são divididos em quatro tipos: Sistemas de Processamento de Transações, Sistemas de Base de Conhecimento, Sistemas Gerenciais e Sistema de Suporte Estratégico.

Dentre as alternativas acima, destaca-se o Sistema de Apoio a Decisão (SAD), integrado dentro de sistemas gerenciais, pois conforme Rezende (1999, p.50), “auxiliam o executivo em todas as fases de tomada de decisão, principalmente, nas etapas de desenvolvimento, comparação e classificação de riscos, além de fornecer subsídios para a escolha de uma boa alternativa em seus negócios”.

Porém, para que um SAD traga resultados positivos para o projeto, é necessário contar com uma tecnologia adequada ao desenvolvimento de um sistema de informação desta natureza. Assim, surge a proposta de se trabalhar com o conceito de *Business Intelligence*.

Segundo Barbieri (2001), *BI-Business Intelligence*, pode ser entendido como a utilização de variadas fontes de informação para se definir estratégias de competitividade nos negócios.

Conforme já comentado, há grandes volumes de dados, mas existe a dificuldade de extração a partir dos mesmos, dificultando o processo de tomada de decisão. O BI traz como um dos seus objetivos principais justamente a definição de regras e técnicas para a formatação ideal desses volumes de dados, tendo em vista transformá-los em depósitos estruturados de informações, independentemente da sua origem. Além disso, cabe como objetivo do *BI* a visualização dos dados com o máximo de flexibilidade para o usuário

Conforme Fortulan e Filho apud Shim et al. (2002), os *Data Warehouses*, OLAP, *Data Mining e Web-SAD* surgiram no começo dos anos 90 como novas ferramentas para SAD, e formam a base dos sistemas de BI. A seguir, serão abordadas as ferramentas de *Data Warehouse (DW)* e OLAP, que farão parte desse projeto.

De acordo com Boghi e Shitsuka (2002), *DW* é um conjunto de tecnologias com o objetivo de converter uma grande quantidade de dados em informações utilizáveis. E para as organizações, o *DW* acaba transformando as fontes de dados operacionais em um ambiente que permite o uso estratégico dos dados.

Dentre suas principais características, o *DW* é em um banco de dados separado do banco de dados dos sistemas transacionais da empresa, que é desenhado para realizar tarefas analíticas utilizando dados de diferentes aplicações.

O *Data Mart (DM)* é uma espécie de *DW* em menor escala e com o escopo mais definido (vários *Data Marts* podem formar um *Data Warehouse*). Por ser menor, possibilita a análise multidimensional, com os cruzamentos de dados e visões previamente calculadas, com o objetivo de aumentar a velocidade na consulta das informações.

Bispo (1998) apresenta OLAP como uma ferramenta capaz de efetuar análises de dados com visão multidimensional do negócio, comparando-os por diversos ângulos. Ou seja, os dados são agregados em várias dimensões<sup>1</sup> para que os analistas possam interagir com o meio e visualizar possíveis informações, de maneira ágil e consistente, que representam determinada situação sob o ponto de vista do usuário.

Portanto, considerando as tecnologias apresentadas, a proposta do trabalho parte da idéia de construir um *Data Mart* de informações comerciais e desenvolver um sistema utilizando tecnologia OLAP para a visualização dos dados.

O projeto será realizado com ferramentas da *Oracle*, pois além de ser a maior empresa de software empresarial do mundo (COMUNIQUE-SE, 2007), suas tecnologias já estão inseridas como padrão de desenvolvimento de aplicações na empresa.

Para a exibição do sistema, será utilizado o *Oracle Portal*, que é a ferramenta oficial da *Oracle* para criação de sites e portais corporativos. A ferramenta OLAP será o *Oracle Discoverer*. Ela permite ao usuário desenvolver consultas da sua necessidade, gráficos, explorações e consultas na *web*, a partir de bases *DW* ou *Data Marts*.

E por fim, os processos de modelagem e carga de todos os dados armazenados pela empresa, serão feitos com o auxílio do *Oracle Warehouse Builder*, ferramenta de projeto de

---

<sup>1</sup> Segundo Barbieri(2001), dimensões são os pontos de entrada específicos de uma estrutura dimensional de dados.

banco de dados e ETL (extração, transformação e carregamento) de todas as informações armazenadas pelas empresas.

Com isso, acredita-se que o sistema possa suprir as necessidades de se obter informações gerenciais através de via única, com rapidez e de forma confiável, auxiliando a tomada de decisões gerenciais.

O trabalho está dividido em cinco capítulos. O primeiro aborda o embasamento teórico das tecnologias presentes no trabalho. O segundo aborda as diversas ferramentas de DW e OLAP. O terceiro apresenta o estudo de caso e o quarto e quinto capítulos detalham de que forma o projeto foi realizado e o que foi implementado.

## 1 *BUSINESS INTELLIGENCE* (BI)

Ter em mãos informações essenciais para tomada de decisão é cada vez mais importante no mundo competitivo dos negócios. A necessidade de excelência na atuação no mercado, bem como maior agilidade no processo de adaptação, é cada vez maior. Assim, lidar com informações para a tomada de decisão é um processo fundamental para a adaptação a essas constantes mudanças.

Fortulan e Filho apud Shim et al.(2002) dizem que os Sistemas de Apoio à Decisão são soluções computacionais desenvolvidas para apoiar a tomada de decisões complexas durante a resolução de problemas. Ferramentas clássicas de SAD compreendem componentes para gerenciamento de sofisticados bancos de dados, poderosas funções de modelagem e projetos de interface com o usuário, que permitem trabalhar interativamente com questões, relatórios e funções gráficas.

Porém, o termo SAD, segundo Fortulan e Filho apud Carlsson e Turban (2002), está sendo cada vez menos usado. Isso pode ser visto em artigos, revistas e até comerciais de sistemas. Segundo eles, no seu lugar tem sido cada vez mais freqüente o uso do termo *Business Intelligence*.

De acordo com Barbieri (2001), BI pode ser entendido como a utilização de variadas fontes de informação para se definir estratégias de competitividade nos negócios da empresa. Ou ainda, BI “representa a habilidade de se estruturar, acessar e explorar informações, normalmente guardadas em DW/DM (*Data Warehouse/Data Mart*), com o objetivo de desenvolver percepções, entendimentos, conhecimentos, os quais podem produzir um melhor processo de tomada de decisão” (BARBIERI. 2001 p. 5).

Ainda, cresce o reconhecimento de que BI está se tornando um componente fundamental na chamada segunda geração dos sistemas ERP, que aponta a necessidade de dar

suporte não apenas ao processamento de transações operacionais, mas também ao processamento de análises (BARBIERI, 2001).

Segundo Serra (2002), o termo BI, na verdade vem sendo utilizado desde a década de 70, quando alguns produtos de *Business Intelligence* foram fornecidos para os analistas de negócios, porém exigiam interações com os usuários exaustivas e intensas, e não apresentavam respostas em tempo hábil para a tomada de decisões, além de possuir alto custo de implantação. Mas, com o surgimento dos bancos de dados relacionais, dos PC's e das interfaces gráficas, aliados ao aumento da complexidade dos negócios, surgiram os novos produtos direcionados aos analistas de negócios.

Segundo Serra (2002), algumas das principais características de um sistema de *Business Intelligence* são:

1. Extrair e integrar dados de múltiplas fontes;
2. Fazer uso da experiência do usuário no negócio;
3. Analisar dados contextualizados;
4. Procurar relações de causa e efeito;
5. Transformar os registros obtidos em informação útil para o conhecimento empresarial.

Conforme Fortulan e Filho apud Shim et al. (2002), os *Data Warehouses*, *OLAP*, *Data Mining* surgiram no começo dos anos 90 como novas ferramentas para SAD, e formam a base dos sistemas de BI. OLAP e DW, tecnologias utilizadas no projeto, serão explicadas a seguir nesse capítulo.

### **1.1 Dados Operacionais x Dados Informativos**

Existem dois tipos de dados nas empresas: um é conhecido como dado operacional e o outro como dado informativo. Segundo Singn (2001), são considerados sistemas de dados operacionais todos os aplicativos que suportam de forma direta as funções críticas de negócio da empresa, e podem ser chamados também de sistemas de OLTP (*On-Line Transaction*

*Processing*). Conforme Costa e Anciães (2001), os dados operacionais são parte da infraestrutura corporativa: são detalhados, atualizáveis e não-redundantes.

Já os dados informativos suportam o processo de tomada de decisões e possuem algumas diferenças em relação ao dado operacional. Enquanto o acesso operacional significa o acesso atual de instâncias específicas de dados, o acesso informativo implica em acessar grandes volumes de dados para análises elaboradas, a fim de planejar e tomar decisões. Esses tipos de dados estão relacionados à tecnologia OLAP, que será explicada nos próximos capítulos.

Os dois tipos de dados constituem fontes importantes para o estabelecimento dos conceitos do BI. (BARBIERI, 2001)

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

<i>Características</i>	<i>Dados Operacionais</i>	<i>Dados Informativos</i>
Conteúdo	Valores correntes	Valores Sumariados, calculados, integrados de várias fontes
Organização dos dados	Por aplicação/sistema de informação	Por assuntos/negócios
Natureza dos dados	Dinâmica	Estática até a carga dos dados
Formato das Estruturas	Relacional, próprio para computação transacional	Dimensional, simplificado, próprio para atividades analíticas
Atualização dos dados	Atualização campo a campo	Acesso sem atualização
Uso	Altamente estruturado, processamento repetitivo	Desestruturado, com processamento analítico/heurístico
Tempo de resposta	Otimizado para poucos segundos	Análises mais complexas, com tempos de respostas maiores

Fonte: BARBIERI, 2001

## 1.2 Data Warehouse

Segundo Inmon (1997), um *Data Warehouse* (DW) é um conjunto de dados baseado em assuntos, integrado, não volátil e variável em relação ao tempo, de apoio às decisões gerenciais. É um conceito que não é novo, pois foi originalmente usado como proposta de solução da IBM, sendo chamado de “*information warehouse*”.

O *Data Warehouse* pode ser considerado, de forma clara e objetiva, a separação física entre os chamados sistemas de dados operacionais e os sistemas de suporte à decisão em uma organização (SINGN, 2001).

### **1.2.1 Características do DW**

Conforme Inmon (1997) um *Data Warehouse* deve apresentar as seguintes propriedades:

1. Orientado a assunto;
2. Integrado;
3. Não volátil;
4. Variável com o tempo.

As seções a seguir descrevem cada uma destas propriedades.

#### **1.2.1.1 Orientado a assuntos**

Conforme Harrison apud Inmon (1997), a questão de o DW ser orientado em assuntos refere-se ao fato de que o mesmo está organizado de maneira a descrever o desempenho dos negócios. Enquanto isso, os bancos de dados operacionais são orientados para os processos de negócios.

Um exemplo dessa propriedade pode ser vista nesse projeto, onde se quer construir um DW focado em vendas. Questões como “quem foi o maior vendedor do ano”, ou “qual cliente apresenta mais problemas de devolução” podem ser respondidas nessa base voltada unicamente a vendas.

#### **1.2.1.2 Integrado**

Muitas vezes, para a criação do *Data Warehouse*, é necessário armazenar dados de diversas fontes, como planilhas, arquivos de texto, bancos de dados, entre outros. Isso deve ficar em um formato consistente, mas para isso surge a necessidade de que os dados sejam tratados antes de serem carregados para a estrutura definitiva.

Nesse processo, surgem algumas tarefas importantes como resolver conflitos de nomes, converter dados para um tipo de medida única ou ainda padronizar formatos de datas. A figura 1.1 mostra um exemplo de tratamento de dados:

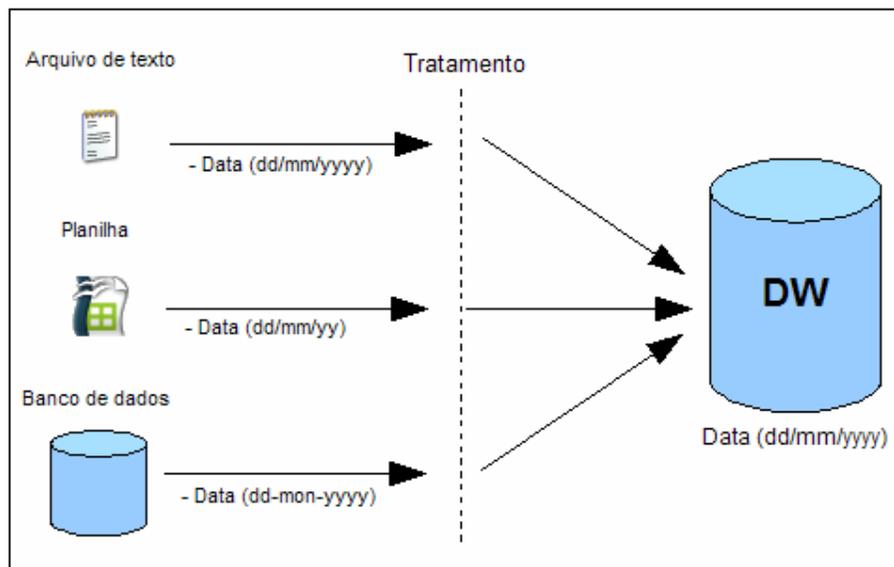


Figura 1.1 – Processo de tratamento de dados  
Adaptada de Inmon (1997)

Conforme Inmon (1997), não adianta de nada carregar os dados operacionais para o DW sem fazer a integração. Caso os mesmos cheguem ao *data warehouse* de forma não-integrada, não será possível a sua utilização como base para uma visão corporativa.

### 1.2.1.3 Não-Volátil

A questão do DW ser não-volátil implica na questão de que nesse tipo de sistema só são permitidos dois tipos de operações: a carga de dados operacionais para a base e as consultas, ou seja, sem alteração de dados. Isso explica uma das diferenças com relação aos ambientes operacionais, no qual é permitido incluir, excluir, alterar e consultar dados. A Figura 1.2 ilustra essa diferença:

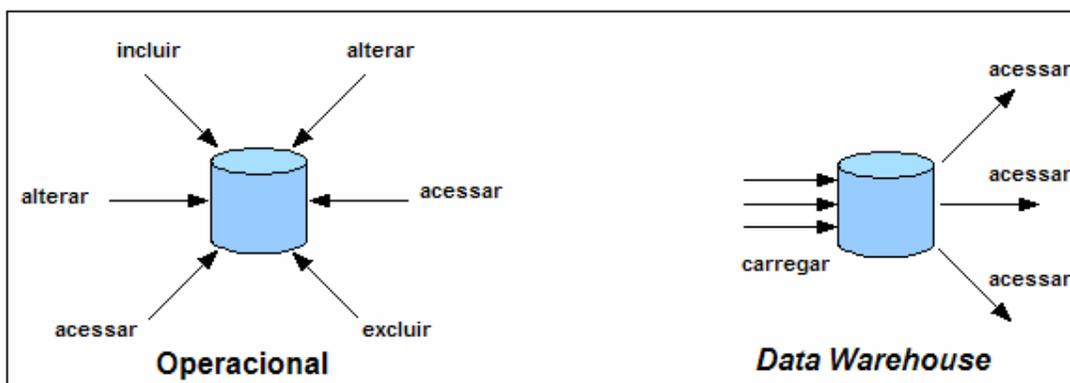


Figura 1.2 – A questão da não-volatilidade  
Adaptada de Inmon (1997)

#### 1.2.1.4 Variável com o tempo

Em ambientes operacionais, os dados podem ser visualizados somente de forma atual, isentando a questão do histórico, ou seja, não sabendo como estavam os mesmos antes da última atualização. No *Data Warehouse*, essa possibilidade muda, pois nesse ambiente os dados históricos são precisos, ou seja, pode-se saber como estava determinado dado em algum determinado momento.

Segundo Inmon (1997), o dado quando carregado no DW recebe, na sua chave, uma unidade de tempo e nunca mais é atualizado. Este tipo de armazenamento permite que os analistas de negócios façam análises de tendências, podendo visualizar as variações de uma determinada informação ao longo do tempo.

#### 1.2.2 Componentes do DW

Segundo Singn (2001), um *data warehouse* sempre apresenta os seguintes componentes:

**Dados atuais:** refletem os acontecimentos mais recentes, que sempre são de grande interesse para a organização;

**Dados antigos:** são acessados com menor frequência e armazenados em nível de detalhe consistente com o detalhe dos dados atuais. Ocupam grande volume de espaço físico;

**Dados sumariados:** a maior parte das consultas executadas no *Data Warehouse* são feitas em níveis altos de sumariação. São divididos em altamente sumariados, que são

compactados e de fácil acesso; e ligeiramente sumarizados que possuem um nível maior de detalhe;

**Metadados:** conforme Harrison (1998), a definição de metadados é “dados a respeito de dados”. Os mesmos são absolutamente críticos por descreverem os dados contidos dentro do DW e OLAP. Exemplos de metadados seriam: quando os dados foram atualizados pela última vez, ou qual o esquema de banco de dados, ou ainda, quais são as regras de agregação de dados.

### 1.2.3 Modelagem de Dados

Conforme Harrison (1998), o modelo entidade-relacionamento (ER)<sup>1</sup> produz um projeto otimizado para acessar dados registro por registro, no nível mais básico, que é o nível de transação ou ocorrências. Para esse tipo de sistema, são realizadas otimizações para uma eficiente criação, atualização e exclusão de registros específicos. E esses processos são realizados através da normalização<sup>2</sup>.

De acordo com Bispo (1998), no modelo entidade-relacionamento, os dados são divididos em diversas tabelas, que se relacionam entre si, formando um diagrama. Seu formato muitas vezes dificulta a interpretação e análise, mas é importante para a eficiência e o desempenho no ambiente operacional, no qual não são necessárias consultas que utilizem recursos excessivos para sua realização.

Para consultas maiores, ou seja, que necessitem uma análise de universo maior de dados, o modelo dimensional surge como melhor alternativa, principalmente pelo fato de economizar nas junções de diversas tabelas, e armazenando dados que facilitem a análise das informações.

O modelo dimensional surgiu para atender sistemas de processamento analítico, com consultas para planejamento tático e estratégico da empresa. Atualmente a utilização desses sistemas pelo nível operacional das empresas vem crescendo, auxiliando o processo de tomada de decisões diárias. Normalmente, ele atende um pequeno número de usuários que realizam consultas planejadas (relatórios pré-definidos) e *ad-hoc* (HOKAMA et al., 2004).

---

1 Proposta por Peter P. Chen em 1976, A modelagem Entidade-Relacionamento (ER) envolve identificando, os elementos de importância na organização (entidades), suas propriedades (atributos) e como eles estão relacionados uns aos outros (relacionamentos). O modelo resultante da informação é independente de qualquer armazenamento de dados ou método de acesso (LOBO, 1998).

2 Normalização é o processo de reunir todos os dados que serão armazenados em um certo banco de dados e separá-los em tabelas (MACHADO, 1996).

A modelagem dimensional permite modelar logicamente dados para melhorar o desempenho de consultas de grande porte. Nesse tipo de modelo, o tempo de resposta é maior, mas, para melhor desempenho nas consultas, há redundância planejada dos dados, compensando os gastos com armazenamento e atualização das informações. O resultado é uma estrutura simples, com modelos que refletem o processo de análise de negócios.

Essa estrutura é composta basicamente pelas tabelas de fatos e tabelas de dimensões. A tabela de fatos traz o resultado da consulta, ou seja, valores de medição. As restrições, objeções e questionamentos ficam nas tabelas de dimensões, que trazem informações textuais sobre o valor medido na tabela de fatos. (BARBALHO, 2003).

Os modelos dimensionais são compreensíveis, previsíveis, ampliáveis e resistentes ao grande uso de grupos de usuários de negócio, por se manter fiel à simplicidade, ter uma perspectiva voltada para as necessidades analíticas da empresa, e especialmente ao seu formato simétrico, em que todas as dimensões normalmente são iguais pontos de entrada na tabela de fatos (KIMBALL, 2002). Os modelos dimensionais são a base de muitos aprimoramentos de desempenho no SGBD, inclusive agregações e métodos de indexação avançados.

“A modelagem de dados é seguramente um dos fatores críticos de sucesso num projeto de *Data Warehouse*, e pode representar a fronteira entre o sucesso e o seu fracasso” (BARBIERI, 2001, p.74) .

### **1.2.3.1 Tabelas de fatos**

A tabela de fatos é a principal tabela de um modelo dimensional, onde as medições numéricas de interesse da empresa estão armazenadas (KIMBALL, 2002). A palavra "fato" representa uma medida dos processos que são modelados, como quantidades, valores e indicadores. Esse tipo de tabela registra os fatos que serão analisados e é composta por uma chave primária (formada por uma combinação única de valores de chaves de dimensão) e pelas métricas de interesse para o negócio.

De acordo com Hokama et al. (2004), a tabela de fatos é sempre esparsa, ou seja, possui um número relativamente pequeno de todas as combinações possíveis de valores de chaves. No contexto desse projeto, poderia ser dado como exemplo o fato de que um produto pode ser vendido por todos os representantes, comprado por todos os clientes, podendo ser

faturado em qualquer data. Por isso pode-se concluir que é algo extremamente esparsos, pois uma porcentagem muito pequena de todas as combinações possíveis de representantes, clientes, produtos e datas de faturamento aparecerá na tabela de fatos.

Segundo Bispo (1998), uma tabela de fatos contém vários fatos correspondentes a cada uma de suas linhas, sendo que cada fato pode armazenar uma ou mais medidas numéricas, que constituirão os valores da análise dimensional. Esse tipo de tabela normalmente armazena muito mais linhas que as tabelas de dimensões, e devem receber atenção pois podem ter um volume muito grande.

Conforme Kimball (1998), não se deve preencher uma linha da tabela fato com zeros para representar que nada aconteceu (por exemplo, que não houve faturamento para um cliente em data específica), pois isso faria com que a tabela de fatos crescesse demais.

Na figura 1.3, é ilustrada a composição de uma tabela de fatos, onde cada linha está representando um fato e as colunas chaves são herdadas das tabelas de dimensões. A tabela contém ainda os valores das medidas para o modelo em questão.

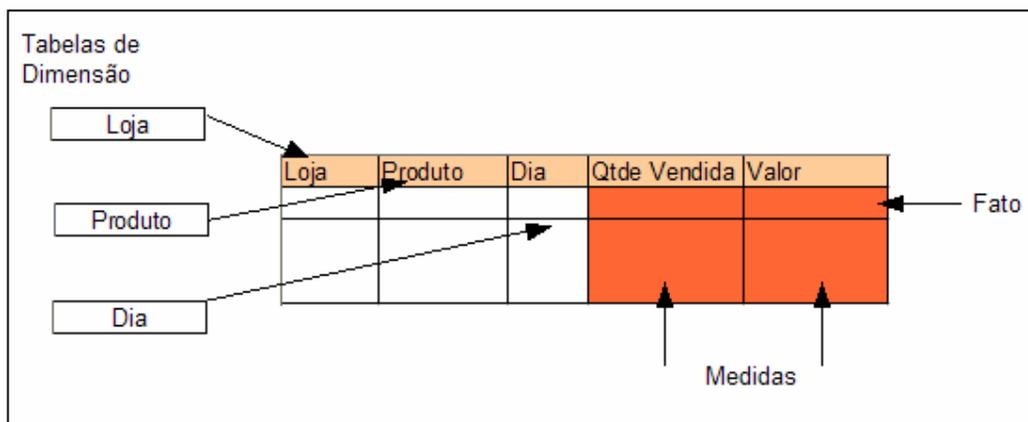


Figura 1.3 – Composição básica de uma tabela de fatos

Fonte: Barbieri (2001)

#### 1.2.3.1.1 Atributos Aditivos, Semi-aditivos e não-aditivos

Conforme já mostrado, as tabelas de fatos armazenam as medições métricas do negócio. Essas métricas, segundo Barbieri (2001), estão definidas em três tipos: aditivas, semi-aditivas e não aditivas. A seguir uma descrição de cada uma delas:

**Aditivos:** Quando os valores são passíveis de serem somados em todas as dimensões;

**Semi-Aditivas:** Quando a soma ou qualquer outro tratamento estatístico tiver sentido apenas em uma dimensão;

**Não-Aditivas:** Quando o valor não puder ser somado em nenhuma dimensão, ou produzir qualquer valor sem sentido válido.

### 1.2.3.2 Tabelas de Dimensões

As tabelas de dimensão são aquelas que armazenam as descrições textuais do negócio, onde cada uma dessas descrições textuais ajuda a definir um componente da respectiva dimensão. Exemplo disso seria que cada registro da dimensão cliente refere-se a um cliente específico (KIMBALL, 1998).

Segundo Barbieri (2001), as tabelas de dimensão têm uma relação de 1:N com a tabela de fatos, logo, possuem um número de registros bem menor. Possuem inúmeras colunas de informação e uma chave primária, que acaba participando das tabelas de fatos, como parte de sua chave múltipla.

É muito importante que os atributos das tabelas de dimensão sejam preenchidos com valores descritivos ao invés de códigos sem sentido, criptografados ou abreviados (KIMBALL, 2002). Por exemplo, em uma tabela de dimensão Alimentos, o campo “perceível” deve ser preenchido com valores como “É perceível” ou “Não é perceível” ao invés de usar simplesmente “S” e “N”. Em um relatório com a listagem de milhares de produtos, um valor descritivo tem muito mais utilidade do que códigos. Ao invés de usar uma aplicação para decodificar esses códigos e mostrar uma descrição, é melhor armazenar essas descrições no banco de dados, tornando a informação disponível ao usuário independentemente de seu aplicativo de acesso aos dados (HOKAMA et al., 2004).

A figura a seguir ilustra um exemplo de uma tabela de dimensão:

Cliente
Cod_cliente
Razao_social
PF_PJ
Endereco
Cod_cidade
Cond_Pgto
Telefone
Fax
Email

Figura 1.4 – Exemplo de uma tabela de dimensão

### 1.2.3.3 Técnicas de Modelagem

Harrison (1998) considera importante que as empresas iniciem seus projetos *Data Warehouse* com a escolha de um modelo lógico que ofereça maior desempenho possível e versatilidade funcional antes de ponderar os benefícios de uma maior eficiência no armazenamento dos dados. Dentre os cinco projetos citados a seguir, o esquema em estrela (*star schema*), é o mais indicado para iniciar devido à simplicidade do projeto de banco de dados e facilidade de compreensão, inclusive por usuários não técnicos.

Segundo Harrison (1998), os cinco modelos de projeto de *Data Warehouse* são:

1. Esquema em estrela (*star schema*);
2. Esquema em estrela parcial;
3. Esquema de *fact table* (tabela de fatos) particionada;
4. Esquema de tabela dimensional particionada;
5. Esquema *snowflake*.

Será explicado a seguir como funciona o *star schema* e o *snowflake*, por serem as técnicas mais conhecidas e mais utilizadas.

#### 1.2.3.3.1 Star Schema

Poe et al. (1998) explica que o esquema estrela é uma estrutura simples, com poucas tabelas e relacionamentos bem definidos e assemelha-se ao modelo de negócio, o que facilita a leitura e entendimento, não só pelos analistas, como por usuários finais não familiarizados com estruturas de banco de dados. Permite a criação de um banco de dados que facilita a execução de consultas complexas, podendo ser realizadas de modo eficiente e intuitivo pelo usuário.

Segundo Harrison (1998), o esquema estrela possui quatro propriedades que o diferenciam dos demais modelos de DW.

1. Dentro de cada categoria, existe uma única tabela de fatos histórica simples, contendo detalhes e dados a nível de sumário, armazenados nos níveis de estrutura indicado em cada tabela de dimensão;

2. A chave primária da tabela de fatos contém somente uma coluna chave de cada dimensão;
3. Cada chave é uma chave gerada pelo sistema;
4. Cada dimensão é representada por única tabela de fatos, usando também uma chave gerada pelo sistema.

A seguir, uma ilustração de um modelo em estrela:

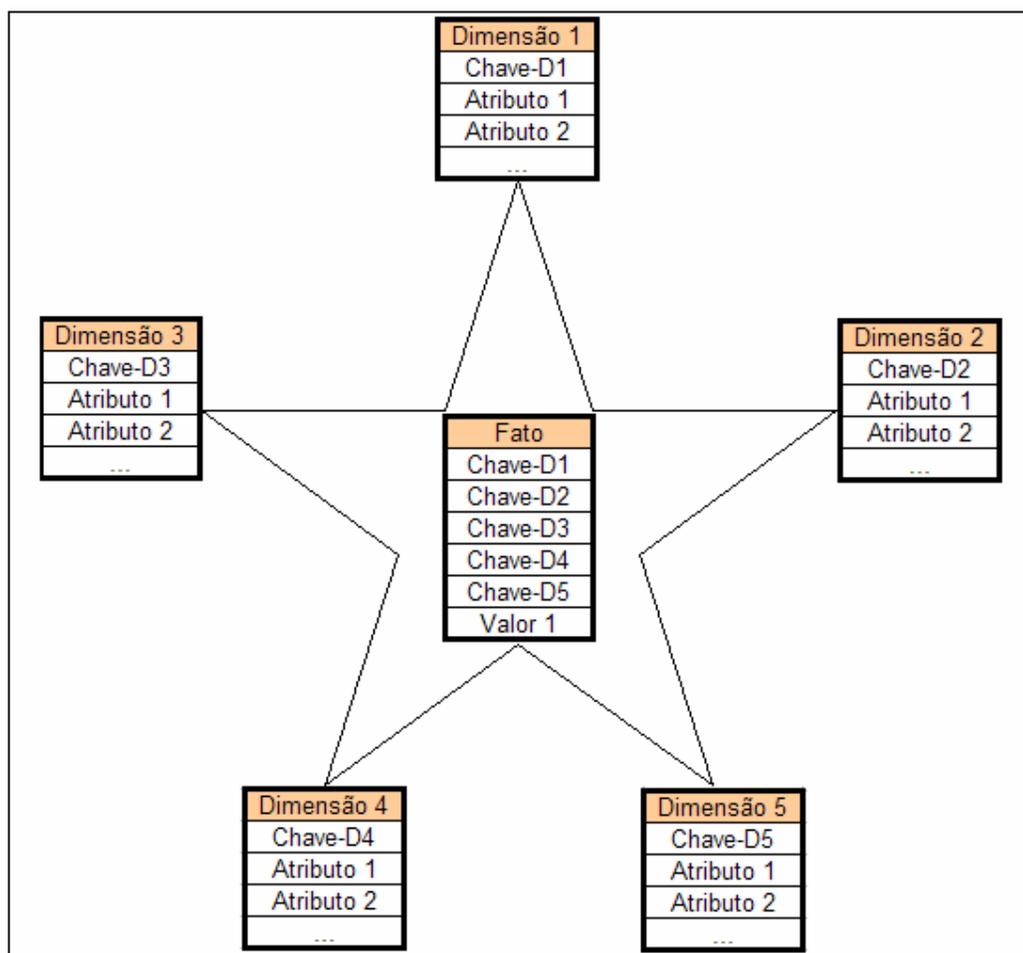


Figura 1.5 – Exemplo de um esquema em estrela  
Fonte: adaptada de Hokama et. Al (2004)

Como pode ser observado, o termo estrela está ligado à disposição das tabelas no modelo, que exibe uma tabela central, a de fatos, e diversas ligadas a ela, sendo as tabelas de dimensões.

Esse tipo de modelo oferece várias vantagens, incluindo um desempenho maior através do uso de chaves geradas pelo sistema, que reduzem o tamanho do índice. Segundo

Harrison (1998), o uso de uma tabela única por dimensão e de uma tabela de fatos simples por categoria assegura que as definições dos metadados possam ser usadas novamente, independentemente do nível de sumário ou fatos.

### 1.2.3.3.2 *Snowflake*

Segundo Machado (2000), o esquema *snowflake* (floco de neve) é uma variação do *star schema*, no qual todas as tabelas de dimensão são normalizadas na terceira forma normal, ou seja, são retirados das tabelas os campos que são funcionalmente dependentes de outros campos que não são chaves.

Na figura 1.6, pode-se visualizar que em *snowflake*, as tabelas de dimensão têm uma conexão lógica com as tabelas de fatos, através das chaves primárias. Há também tabelas menores, chamadas de extensões, que são usadas para armazenar descrições e decodificações para chaves e códigos nas tabelas maiores.

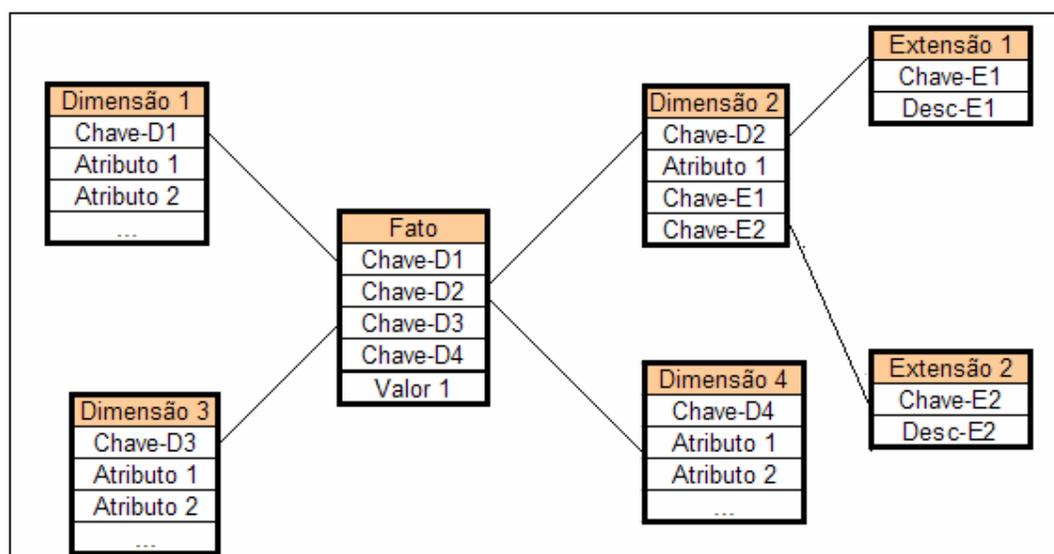


Figura 1.6 – Exemplo de um esquema *snowflake*

Fonte: Adaptada de Hokama et. Al (2004)

Segundo Singn (2001), o uso do esquema *snowflake* traz como desvantagens o aumento da complexidade da estrutura de dados, dificultando a compreensão do modelo por parte de usuários que trabalham diretamente com a estrutura física das tabelas. E essa complexidade pode trazer ainda, uma diminuição de performance nos processos que envolvem esse tipo de esquema.

Porém, o uso do *snowflake* pode ser indispensável em alguns casos em que, por exemplo, o uso de esquemas em estrela necessite espaço em disco muito maior ou suas tabelas dimensionais sejam muito grandes, onde assim causariam problemas de performance no sistema.

Kimball (1996) sugere aos projetistas "bem-intencionados" a resistirem a idéia de transformar *star schemas* em *snowflake*, em função da complexidade deste tipo de estrutura sobre o usuário final, enquanto que o ganho em termos de espaço de armazenamento seria pouco relevante.

#### 1.2.3.4 Cubos de Dados

Segundo Kimball (1998), através dos cubos de dados, quase todos os tipos de dados de negócio podem ser representados. Nessas estruturas, as células do cubo representam os valores medidos enquanto os lados representam as dimensões dos dados. Quando um cubo apresenta mais de três dimensões, são chamados de hipercubos.

Cubo é a estrutura multidimensional de dados que expressa a forma na qual os tipos de informações se relacionam entre si. É formado pela tabela de fatos e pelas tabelas de dimensão que a circundam e representam possíveis formas de visualizar e consultar os dados. O cubo armazena todas as informações relacionadas a um determinado assunto, de maneira a permitir que sejam montadas várias combinações entre elas, resultando na extração de várias visões sobre o mesmo tema.(HOKAMA et al. 2004, p. 49).

Conforme Gray e Watson (1999), os cubos são os principais objetos de um OLAP. São desenvolvidos com tecnologia que permite rápido acesso aos dados, sendo que normalmente são compostos por sub-conjuntos de um *Data Warehouse* e organizados e sumariados dentro de estruturas multidimensionais definidas por dimensões e medidas. Essas estruturas podem resultar em diversas matrizes esparsas que permitem trabalhar simultaneamente com diversos cenários definidos por combinações de dados, como produtos, clientes, períodos, etc. Esses cubos podem ser armazenados em modelos de bancos de dados ROLAP (OLAP Relacional), MOLAP (OLAP Multidimensional) ou HOLAP (OLAP Híbrido), que serão explicados nos próximos capítulos.

Na figura 1.7, pode-se visualizar a composição de um cubo, formado pelas células que compõem as medições (valores faturados) e as laterais representando dimensões de período, clientes e produtos.

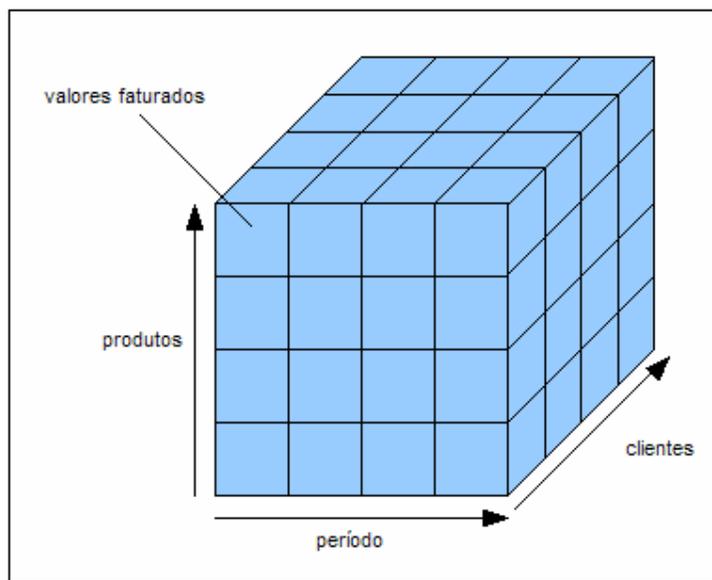


Figura 1.7 – Cubo de dados

Uma das técnicas que faz análise desse tipo de estrutura é o *slice-dice* (INMON et al., 1999), que significa cortar o cubo de dados em fatias permitindo rotacionar os lados do mesmo em qualquer sentido, possibilitando a combinação de quaisquer dimensões e a obtenção de informações correspondentes sobre vários enfoques. Essa técnica, dentre outras, será explicada com mais detalhes no capítulo relacionado a OLAP.

### 1.2.3.5 Granularidade

Segundo Inmon (1996), a granularidade refere-se ao nível de detalhe ou sumariação contida nas unidades de um DW. Quanto maior o detalhamento, menor a granularidade, quanto menor o detalhamento, maior a granularidade. O exemplo na figura 1.8 mostra isso. Caso seja visualizado o total de vendas dentro do mês, haverá um baixo nível de detalhes, mas uma granularidade alta(Figura 1.8[a]). Se esses valores forem divididos em dias, o nível de detalhe aumentará, porém a granularidade não (Figura 1.8[b]).

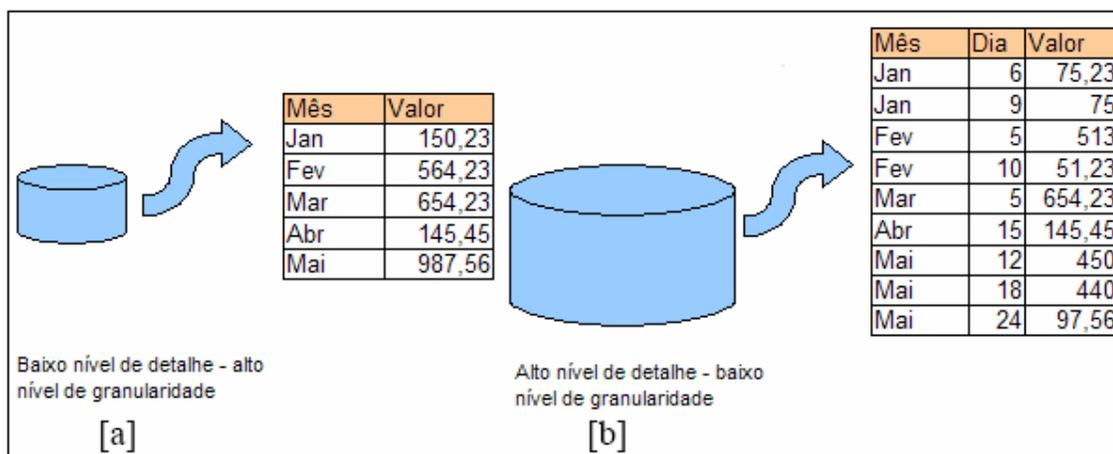


Figura 1.8 – Níveis de granularidade

Fonte: Adaptada de Inmom (1996)

A mais importante questão de projeto que o desenvolvedor do *data warehouse* precisa enfrentar diz respeito à definição da granularidade do *data warehouse*. Quando a granularidade de um *data warehouse* é apropriadamente estabelecida, os demais aspectos de projeto e implementação fluem tranquilamente; quando ela não é estabelecida, todos os outros aspectos se complicam (INMON, 1997, p.143).

Os níveis de granularidade dependem muito do total de linhas a serem carregados para o *Data Warehouse*. Por exemplo, se para um horizonte de um ano, o total de linhas for em torno de 10.000, qualquer técnica funcionará. Caso for maior, supondo-se 1.000.000 de linhas, seria necessário utilizar mais níveis de granularidade, os chamados “níveis duais de granularidade”. Esses níveis permitem que se processe eficientemente a enorme quantidade de solicitações, atendendo a qualquer questão que possa ser respondida. Essa é a melhor de todas as situações e deveria ser a opção de projeto padrão (INMON, 1997).

Inmon (1997) exemplifica esse tipo de método com um sistema bancário. Em um sistema desses, pode-se armazenar, por exemplo, os lançamentos individuais em contas correntes nos últimos 60 dias, e armazenar o histórico resumido desses lançamentos nos últimos 5 anos, com o valor total de lançamentos sumariados por mês. Tanto os dados resumidos, quanto os detalhados estarão disponíveis para o usuário.

### 1.2.3.6 Agregados

Dentro das aplicações transacionais, o número de registros a serem recuperados normalmente é reduzido. Ao contrário de ambientes DW, onde as consultas são bastante intensivas (*data-intensive*). As consultas em sistemas de apoio a decisão, segundo Meredith e Khader (1996), podem ser divididas em *data-intensive*, que acessam um número grande de

linhas e colunas, não importando se são recuperadas através de índices ou buscas por toda a tabela, e *data-selective*, que mesmo com poucas colunas têm critérios de seleção complexos.

Índices podem auxiliar nas consultas *data-selective*, mas para melhorar o desempenho de consultas do tipo *data-intensive* são necessárias outras técnicas. Dentre elas o particionamento, que reduz o número de dados a serem varridos pelo sistema e a agregação, que pré-calcula as medidas necessárias para que o sistema acesse menos registros.

Conforme Kimball (1998), os agregados são sumários armazenados, que devem existir juntamente com os registros base do sistema, criados para aumentar o desempenho das consultas. Esses ganhos de performance podem chegar muitas vezes até a um fator de 100 a 1000 vezes maior.

“O agregado é um registro de tabela de fatos que representa o resumo de nível básico da tabela de fatos. Um registro da tabela de fatos agregado está sempre associado a um ou mais registros de tabela de dimensões agregadas” (KIMBALL, 1998, p. 191).

Um dos problemas levantados nos agregados é a questão de que essa solução agride de certa forma os processos tradicionais de ausência de redundância, estabelecidos nos preceitos dos projetos de banco de dados. Além de ocuparem mais espaço, pela razão de exigirem uma coleção de tabelas de fatos ou de dimensões, que passam a ser dedicadas ao armazenamento de dados pré-processados (BARBIERI, 2001).

A figura 1.9 mostra como podem ser utilizados agregados através de tabelas de fatos. A tabela “FatoVenda” tem duas agregações, a “agregada1”, onde os dados estão sumariados por representante e a “agregada2”, onde estão sumariados por cliente.

<b>FatoVenda</b>	<b>Agregada1</b>	<b>Agregada2</b>
<u>Cod_data</u>	<u>Cod_data</u>	<u>Cod_data</u>
<u>Cod_Cliente</u>	<u>Cod_Representante</u>	<u>Cod_Cliente</u>
<u>Cod_Representante</u>	<u>Cod_Produto</u>	<u>Cod_Produto</u>
Valor1	Valor1	Valor1
Valor2	Valor2	Valor2
Valor3	Valor3	Valor3

Figura 1.9 – Tabelas agregadas  
Fonte: Adaptada de Hokama et. Al (2004)

### 1.2.4 Data Mart (DM)

Segundo Barbieri (2001), *Data Mart* (DM) é um depósito de dados que atende a áreas específicas da empresa e objetiva auxiliar o processo decisório gerencial. Sobre DW e DM, Barbieri (2001, p.50) cita: “ambos podem ser definidos como espécies do mesmo tipo, ficando a diferença entre os dois centrada no escopo do projeto e nos limites de suas abrangências”.

Na definição de Serra (2002), *Data Mart* é um pequeno *Data Warehouse* que fornece suporte à decisão para um pequeno grupo de pessoas. Pelo fato de ser mais focado em uma área específica, o *Data Mart* é atrativo por possuir baixo custo e tempo menor de implementação, com crescentes avanços tecnológicos. Segundo Serra (2002, p. 136), “os *Data Marts* podem servir como veículo de teste para empresas que desejam explorar os benefícios do *Data Warehouse*”.

Apesar das vantagens de custo e desempenho, como os DM's diferem de departamento para departamento, o desenvolvimento de DM's independentes, sem um planejamento global acarreta a fragmentação de dados de uma organização e inibe a utilização de informações de forma integrada na corporação, podendo fazer surgir e proliferar à falta de integração e compartilhamento de dados entre os sistemas transacionais (FORTULAN e FILHO apud GRAY E WATSON, 1998).

O quadro 1.2, aponta as diferenças básicas entre *Data Warehouse* e *Data Marts*:

Quadro 1.2 – Diferenças entre DW e DM.

<i>Data Warehouse</i>	<i>Data Mart</i>
<ul style="list-style-type: none"> <li>- Utilização altamente imprevisível, aplicações não estruturadas, analíticas;</li> <li>- Tempo de resposta pode ser de segundos ou minutos;</li> <li>- Dados relacionais;</li> <li>- Informações organizadas por área de análise, normalmente históricas;</li> <li>- Usuários finais: gerência, consumidores de informação.</li> </ul>	<ul style="list-style-type: none"> <li>- Tipo de <i>Data Warehouse</i> em que os dados estão mais próximos aos usuários;</li> <li>- Menores e mais fáceis de serem gerenciados;</li> <li>- Tomada de decisão pode ser em nível departamental;</li> <li>- Dados relacionais ou multidimensionais.</li> </ul>

Fonte: Adaptado de Serra(2002)

De acordo com Poe et al. (1998), as organizações podem escolher esse tipo de arquitetura, os *data marts*, para o início do desenvolvimento de um projeto piloto, limitado a

uma área de negócios específica, provendo uma oportunidade de aprendizagem e futura integração em um projeto global único que seria um DW corporativo. Outras características marcantes dos DM's são a rapidez na implementação, o baixo custo, o controle local em vez de centralizado e a redução do tempo de resposta a consultas, tornando a questão do custo-benefício muito favorável, contrastando com o longo trabalho de modelagem, tempo de desenvolvimento e recursos financeiros exigidos pelo DW.

Fortulan e Filho apud Gray e Watson (1998) reforçam essas idéias ao afirmar que os altos custos de implementação de um DW limitam o seu uso por empresas de grande porte, as quais muitas vezes não estão dispostas a correr riscos no investimento em algo que não se tem certeza do sucesso e, conseqüentemente, o retorno do investimento, tornando os *data marts*, uma alternativa reduzida e de baixo custo.

### **1.3 OLAP**

A sigla OLAP (*On-Line Analytic Processing*) significa Processamento Analítico *On-Line*, sendo o inverso do conceito de OLTP (Processamento de Transações *On-Line*). Segundo Kimball (1998), OLAP é um termo inventado para descrever uma abordagem dimensional para suporte à decisão. Porém, a filosofia OLAP, contudo, ainda necessita de critérios mais específicos para ser aceita como padrão de comparação para sistemas de suporte à decisão.

Harrison (1998) explica que OLAP é um rótulo, antes mesmo de ser considerada uma tecnologia, aplicando-se a todas as funcionalidades analíticas requeridas para a criação de informações que sejam úteis a partir de dados de DW. OLAP permite exercer as mais diversas funcionalidades de análise dos dados através das dimensões do *Data Warehouse*.

#### **1.3.1 Características**

O OLAP, segundo Santos (2000), é um paradigma muito poderoso para análise estratégica de dados de bancos de dados de grandes organizações. Dentre as principais características deste tipo de análise, as que podem ser consideradas chave são: grandes volumes de dados; apoio explícito para a dimensão temporal; apoio para vários tipos de agregação; análise de longo alcance, no qual tendências de dimensões globais são mais importantes que detalhes de atributos de dados individuais.

Segundo Harrison (1998), o OLAP possui capacidades computacionais que podem ser divididas entre as quatro a seguir:

1. Gerar consultas e relatórios, proporcionando formatações até mesmo a cargo do próprio usuário final;
2. Análise multidimensional, permitindo acessar qualquer dimensão do DW;
3. Análise estatística, possibilitando calcular médias e outras medidas mais sofisticadas como regressão, correlação, fatoração e agrupamentos;
4. *Data Mining*, fazendo identificação de modelos e relações e algoritmos de aprendizado para trabalhar com previsões.

#### **1.3.1.1 Funções Básicas**

De acordo com Harrison (1998) os aplicativos ou ferramentas OLAP em geral executam cinco funções básicas:

1. **Interface:** as telas e métodos usados para direcionar instruções internas a outras funções baseadas nas seleções dos usuários;
2. **Consulta:** a lógica do aplicativo usada para gerar o código SQL;
3. **Processo:** a lógica do aplicativo que executa a análise de dados no conjunto de resultados retornado pela consulta ao banco de dados;
4. **Formato:** a lógica do aplicativo requerida para rotular propriamente linhas e colunas de dados e criar um arquivo padrão;
5. **Exibição:** apresentação do arquivo formatado, como relatório ou gráfico, para visualização pelo usuário.

#### **1.3.1.2 Operações OLAP**

Como já explicado, o OLAP permite ao usuário navegar nas mais diversas dimensões do *data warehouse*. Isso possibilita a solicitação às mais variadas consultas, de acordo com as sumariações desejadas, ficando a cargo deste usuário a definição de cada uma das visões que melhor se adequar à resposta esperada.

Kimball (1996), Inmon (1997, 1999), Lopes (2006) trazem como algumas das principais operações a serem realizadas através da tecnologia OLAP:

**Pivoting:** Serve para adicionar, remover ou rearranjar as dimensões das tabelas, utilizando simplesmente o *mouse*, arrastando e soltando o botão.

**Slice and dice:** É a descrição da habilidade de fatiar e cortar o cubo separando partes de um cubo. O uso do *slice and dice* permite rotacionar os lados de um cubo de dados (dimensões) em qualquer sentido, possibilitando assim, a combinação de quaisquer dimensões e a obtenção de informações correspondentes sobre o enfoque desejado.

**Drill down e roll-up (ou drill-up):** *Drill-down* serve para que usuário tenha a possibilidade de ter uma visão mais detalhada de um conjunto de dados, navegando, por exemplo, em uma hierarquia de dados pré-definida. *Roll-up* é o processo inverso do *drill-down*, ou seja, exibe os dados de uma forma mais macro, mais agrupados ou sumariados. A figura 1.10 representa os processos de *drill-down* e *roll-up* em um exemplo de hierarquia de tempo.

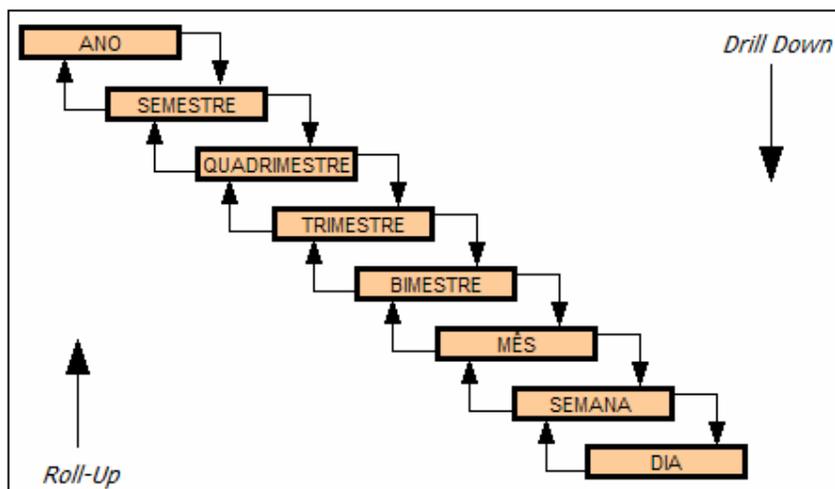


Figura 1.10 – Exemplos de operações de *Drill Down* e *Roll-Up*

**Drill-across e Drill-through:** São variações das operações de *drill-down* e *roll-up*. *Drill-across* diz respeito a navegação em uma dimensão avançando para os níveis intermediários, enquanto o *drill-through* está relacionado a navegar a um ponto de detalhe menor que o existente na dimensão.

**Tabelas cruzadas:** as tradicionais planilhas eletrônicas. A diferença reside no fato de que os dados são apresentados em planilhas com mais de duas dimensões, normalmente quatro ou mais.

**Consultas *ad-hoc*:** segundo Inmon (1997) são consultas com acesso casual único e tratamento dos dados segundo parâmetros nunca antes utilizados, geralmente executados de forma interativa e heurística. De forma mais sucinta, consultas *ad-hoc* estão ligadas ao fato de o usuário gerar consultas de acordo com suas necessidades, relacionando informações de uma maneira única e própria.

**Geração de *queries*:** possibilidade de o usuário montar *queries* próprias, através de uma ferramenta amigável e transparente, sobre a qual é necessário um conhecimento mínimo de informática por parte do usuário, a fim de obter as informações que deseja. Basicamente consiste em gerar o acesso ao *Data Warehouse/Data Mart* para obter a informação e analisá-la (LOPES, 2006).

Além dessas operações básicas, diversas ferramentas possuem funções que atuam como indicadores. Exemplos disso são semáforos e sinalizadores, além dos indicadores de tendências, relatórios de exceção, previsões, projeções, simulações, entre outras (WAGNER, 2003).

### 1.3.2 Arquiteturas OLAP

Segundo Anzanello (2002), de acordo com o método de armazenamento de dados para a aplicação OLAP, será elaborada a arquitetura ideal para a aplicação. Os principais métodos para esse fim são MOLAP, ROLAP, DOLAP e HOLAP, complementados por uma tendência chamada JOLAP. Destes, cada um tem uma função específica e deve ser utilizada quando melhor atender às necessidades de análise pela ferramenta de OLAP. A seguir serão destacados ROLAP e MOLAP, por serem considerados pela maioria dos autores consultados os mais importantes desse grupo de métodos.

#### 1.3.2.1 ROLAP

Segundo Kimball (1998), o ROLAP (Relacional OLAP - *Relational OLAP*) constitui-se de um conjunto de interfaces de usuário e aplicações que dá ao banco de dados relacional características dimensionais.

Com relação ao funcionamento de ROLAP, Bispo (1998), afirma que as ferramentas ROLAP podem efetuar o processamento para realizar as consultas ou outras funções em um ambiente dimensional de duas formas:

1. Efetuar o processamento dos dados no servidor da base de dados, ou seja, o servidor OLAP gera os comandos SQL em múltiplos passos e as tabelas temporárias necessárias para o devido processamento das consultas;

2. Executar comandos SQL para recuperar os dados, mas fazer todo o processamento, incluindo junções e agregações, no servidor OLAP.

De acordo com Fernandes (2004), o desenvolvimento de aplicações ROLAP pressupõe que não é criada mais nenhuma estrutura fixa de dados, para além da estrutura de dados do *data warehouse*. As *queries* (consulta a um banco de dados) são elaboradas na própria ferramenta ROLAP e sempre que o utilizador pede um relatório contendo determinada *query*, esta é lançada no motor de dados que suporta o *Data Warehouse*, que enviará posteriormente uma resposta à ferramenta ROLAP a fim de esta apresentar os referidos resultados.

Segundo Figueiredo (1998), uma das vantagens o uso de uma solução ROLAP consiste na utilização de uma tecnologia estabelecida, de arquitetura aberta e padronizada como é a relacional, beneficiando-se da diversidade de plataformas, escalabilidade e paralelismo de *hardware*.

### **1.3.2.2 MOLAP**

Segundo Kimball (1998), MOLAP (*Multidimensional OLAP*), também chamado de Banco de Dados Multidimensional, constitui-se de um conjunto de interfaces de usuário, aplicações e banco de dados, com tecnologia proprietária, que possui características eminentemente dimensionais. Esse tipo de banco de dados armazena os dados em cubos que podem conter múltiplas dimensões, adicionando ainda a variável de tempo às dimensões.

Através da análise multidimensional, é possível comparar qualquer parte do negócio com qualquer outra parte, definindo diversos tipos de análise, sempre que necessários. Isso, sem a obrigatoriedade de ter que projetar um novo banco de dados para cada análise realizada. Assim há a possibilidade de gerar inúmeras análises de forma intensa, em um curto espaço de tempo (BISPO, 1998).

Por ser uma tecnologia complexa, principalmente quanto às questões de armazenamento e visualização de dados, o MOLAP exige um trabalho muito maior com todo o processo de engenharia de banco de dados, tamanho, estruturação, processo de carga, indexação, otimização, entre outros (HOKAMA et al., 2004).

### **1.3.2.7 ROLAP x MOLAP, qual a melhor tecnologia OLAP?**

Essa é uma dúvida que acaba gerando muitas discussões, pois cada uma delas tem suas vantagens e desvantagens. Porém, é necessário analisar muito bem o contexto em que será utilizada a tecnologia OLAP para decidir qual delas usar.

Bispo apud Gentia e Software (1998), fez levantamentos de pontos importantes entre os dois tipos de tecnologias OLAP que podem auxiliar na tomada de decisão de qual delas usar:

**Volume de dados:** Ferramentas ROLAP podem gerenciar volumes maiores que MOLAP, mas o desempenho e manutenção podem se tornar limitados. Mas o que vem ocorrendo é a criação de subconjuntos de volume de dados grandes, tornando vantajosos ambos os tipos de tecnologia;

**Aplicações de constantes atualizações:** O processo de atualização de dados é bastante semelhante entre os dois tipos de tecnologia;

**Análise real de dados operacionais:** Não é recomendável efetuar análises complexas de forma multidimensional em bancos transacionais. Mas, caso isso seja necessário, os tempos de resposta entre ROLAP e MOLAP são equivalentes;

**Disponibilidade dos dados:** Os dados em ROLAP podem ser acessados por um horizonte maior de ferramentas. Mas para que os tempos de respostas sejam aceitáveis, os dados são armazenados em tabelas relacionais de certa forma numerosas, e às vezes não muito compreensíveis para o usuário;

**Velocidade na carga de dados:** a carga dos dados implica em uma série de processos, que vai desde a leitura dos dados, até uma validação final dos mesmos. Tudo isso ocorre mantendo-se o banco de dados *on-line* e consistente. Portanto, é mais provável que o processo se torne mais rápido no MOLAP do que no ROLAP.

**Performance em cálculos e recuperação de dados:** O ROLAP necessita de mais processamento para executar as tarefas do que o MOLAP, porém um ROLAP bem projetado pode atingir um desempenho equiparável ao de um MOLAP.

**Análise multidimensional:** Mesmo ferramentas ROLAP que utilizam indexação para simular bancos multidimensionais não atingem a mesma performance de ferramentas MOLAP. Com relação aos cálculos multidimensionais, as ferramentas ROLAP superaram limitações nesse ponto através de múltiplos passos de SQL, mas mesmo assim não tem a mesma quantidade de cálculos que o MOLAP.

**Análises simples:** MOLAP não é necessário para consultas mais simples, que podem ser geradas através das ferramentas OLAP ou geradores de relatórios que acompanham os bancos de dados relacionais.

**Plataformas de servidores:** Alguns produtos ROLAP possuem maior disponibilidade em plataformas diferentes do que a maioria dos produtos MOLAP;

**Custo:** MOLAP implica em bancos de dados especiais e licenças mais caras. Porém o fato de necessitar de menos espaço em disco, menos processamento e menos refinação, fazem com que o custo do MOLAP possa se tornar acessível mesmo com as taxas embutidas na compra desse tipo de tecnologia.

Gorla (2003), também fez um estudo nessa comparação e acabou chegando a algumas conclusões. A indicação para utilizar MOLAP seria para usuários que consigam pré-definir todas as suas necessidades de informação e que não precisem dos dados numa base de atualização diária. Sendo assim MOLAP poderia ser usado naqueles sistemas em que a atualidade dos dados não é uma questão vital para o negócio e se admite uma defasagem entre os dados reais e os das análises que podem ser de um dia a uma semana, ou mais.

Com relação a ROLAP, Gorla (2003) recomenda para usuários que precisem analisar informações de mercado com elevado nível de variação e que tenham uma constância maior de atualização. Ainda nos casos em que a volatilidade dos dados é constante, os sistemas ROLAP apresentam melhores respostas.

Assim, após esse estudo teórico sobre BI, no próximo capítulo serão apresentadas ferramentas de OLAP e DW.

## 2 FERRAMENTAS

Nesse capítulo, será realizado um estudo de algumas ferramentas de *Business Intelligence*. Será feita uma divisão em duas seções. Na primeira serão abordadas algumas ferramentas de *Data Warehouse* existentes no mercado, e na segunda, ferramentas de OLAP.

### 2.1 Ferramentas DW

As ferramentas que serão mostradas a seguir foram analisadas através de manuais técnicos, sites relacionados e trabalhos acadêmicos que utilizaram algumas das ferramentas, que são: *Oracle Warehouse Builder*, *Oracle Express*, *Microsoft DTS* e *Business Information Warehouse*.

Ao final do capítulo será feito um breve comparativo entre as ferramentas DW descritas, apontando as principais diferenças entre elas, em um quadro com os pontos fundamentais que compõem esse tipo de produto.

#### 2.1.1 Oracle Warehouse Builder

A seguir, será explicado com um nível maior de detalhes o funcionamento da ferramenta *Oracle Warehouse Builder* (OWB). Esse detalhamento maior foi realizado por se tratar da ferramenta que, apesar de não ter sido utilizada integralmente, como se cogitou no TC1, serviu como aprendizado e auxílio nos processos de modelagem e carga do sistema.

##### 2.1.1.1 A ferramenta

O *Oracle Warehouse Builder* é uma ferramenta de *Business Intelligence* da empresa *Oracle Corporation*, desenvolvida em 1999 e que hoje está na versão 10g. Ela traz uma

solução integrada para modelar e desenvolver *data warehouses*, *data marts* e aplicações de *Business Intelligence*.

Ela possui componentes que efetuam o processo de ETL e, além disso, é considerada o ponto central da integração com as ferramentas do pacote *Oracle BI*, provendo integração com utilitários de consultas *ad-hoc* e funcionalidades de bancos relacionais (ORACLE, 2003).

### 2.1.1.2 Características principais

Conforme Marques (2002), o OWB é um aplicativo, de interface *Java*, que opera em múltiplas camadas: interface com o usuário, gerador de código, integradores, interface de aplicativo *Java* e o repositório de metadados. Essas ferramentas acabam tornando um produto de referência no mercado na questão de carregamento e manutenção do DW.

Uma das vantagens do OWB em relação às outras ferramentas de ETL é que a maioria das atividades necessárias para se modelar um *warehouse* são feitas através de assistentes, os chamados *wizards*, que orientam todos os passos necessários para se definir o *data warehouse* (GONÇALVES, 2002). E isso, traz um grande ganho de produtividade, além de facilitar o uso para o desenvolvedor. A figura 2.1 mostra um exemplo das telas de *wizard* existentes no OWB, o assistente de criação de uma tabela.

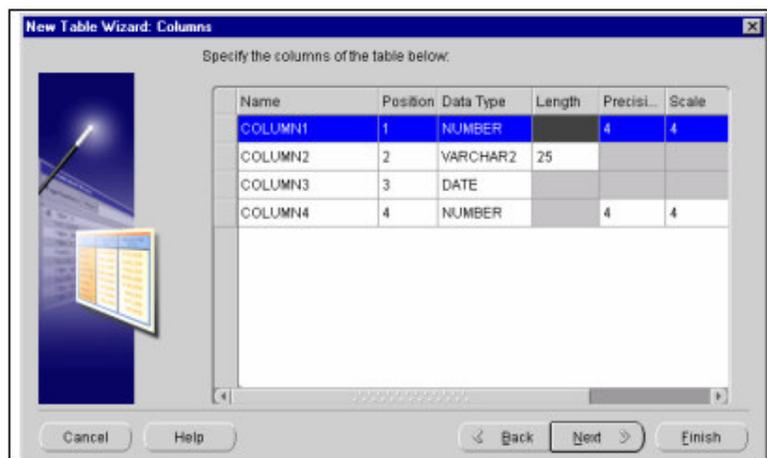


Figura 2.1 – Exemplo de um *wizard* na criação de tabelas no OWB  
Fonte: Oracle (2003)

#### 2.1.1.2.1 Orientado a Projetos

Sempre que for iniciada uma nova modelagem e construção de um *warehouse* é necessário que se crie um novo projeto. Por isso, pode-se afirmar que o OWB é uma ferramenta orientada a projeto. As informações existentes em cada projeto são armazenadas

no próprio repositório do OWB através de um SGBD *Oracle*. Segundo Marques (2002), as informações armazenadas no repositório de metadados podem ser utilizadas fora da ferramenta, possibilitando uma integração com o *Oracle Business Intelligence*.

Na figura 2.2 pode-se observar a tela principal do OWB, mostrando um novo projeto, denominado de “MY\_PROJECT”, com os principais itens que o compõe.

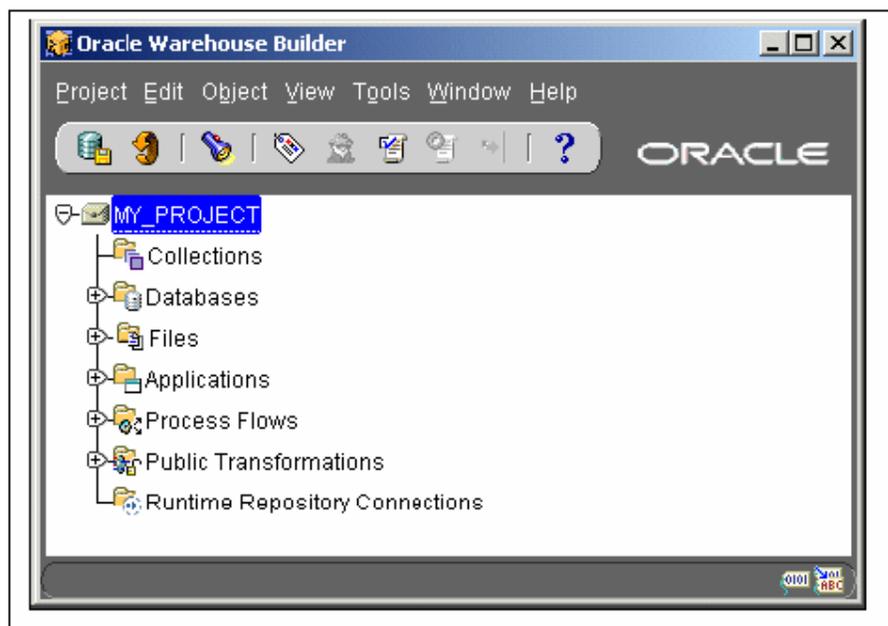


Figura 2.2 – *Oracle Warehouse Builder*  
Fonte: Oracle (2003)

#### 2.1.1.2.2 Fontes de dados

Conforme a *Oracle* (2003), o OWB pode criar módulos de dados de diferentes fontes, tais como:

1. Bancos de dados *Oracle*;
2. Bancos de dados não-*Oracle*;
3. Arquivos de texto;
4. Aplicações em SAP.

Conforme Gonçalves (2002), uma restrição existente no *Oracle Warehouse Builder* é que existem apenas três interfaces de integração que são disponibilizadas na própria ferramenta: interface com bancos de dados *Oracle*, arquivos de texto e aplicações SAP.

As outras ferramentas de interface que podem ser utilizadas são *gateways* desenvolvidos pela *Oracle* para acessar outros bancos de dados. Porém, para adquirir esses produtos há um custo adicional. Segundo a *Oracle*, *drivers* ODBC também podem ser utilizados para prover conexões com bancos de dados relacionais de outros fornecedores.

Com relação aos tipos de dados oferecidos pela ferramenta, o OWB trabalha com objetos dimensionais e relacionais. Dentro dos objetos relacionais estão incluídas tabelas, *views*<sup>1</sup>, *materialized views*<sup>2</sup> e *sequencies*<sup>3</sup>. Os objetos multidimensionais incluem dimensões e cubos de dados (ORACLE, 2003).

### 2.1.1.2.3 Importação de Metadados

Uma das opções que merece destaque na ferramenta é a possibilidade do usuário importar metadados de estruturas de banco de dados transacionais e/ou de *data warehouses* já existentes. Esta técnica, que é conhecida na maioria das ferramentas CASE como engenharia reversa, tendo como principal objetivo documentar bases de dados que foram construídas sem a utilização de uma ferramenta CASE. Além disso, a importação de metadados no OWB facilita principalmente a modelagem das bases transacionais que são utilizadas para a extração de dados, pois com a importação dos dados para o DW, o usuário não necessita definir de forma manual todos os objetos que serão utilizados na extração (GONÇALVES, 2002).

### 2.1.2 Oracle Express

Adquirida pela *Oracle* em 1995 da *Information Resource Inc.*, não é apenas uma ferramenta que desempenha o papel de *data warehouse*. Além de trabalhar com o processo de ETL, o *Oracle Express* atua como ferramenta de OLAP. Na versão *Oracle Express Server*, atua como servidor de banco de dados multidimensional. Na versão *Oracle Personal Express*, desempenha a função de DOLAP, executando na versão cliente.

Segundo Barbieri (2001), como MOLAP, o *Oracle Express* armazena os dados como uma planilha multidimensional, com aspectos de otimização de espaços para valores esparsos,

---

1 *Views* são apresentações personalizadas de dados através de um SQL consultando em uma ou mais tabelas (ORACLE, 2003).

2 *Materialized Views* são tabelas de *views* atualizáveis, que possuem dados vindos através de uma consulta SQL de uma ou mais tabelas (ORACLE, 2003).

3 *Sequences* são objetos do banco de dados que geram listas de valores únicos (ORACLE, 2003).

estratégias especiais de indexação, que se diferem daquelas usadas nos gerenciadores relacionais, nos quais as linhas são armazenadas com as suas chaves adjacentes dos dados.

A ferramenta trabalha nos três estilos de cliente-servidor: completa, média e *web*. Na primeira, os dados podem ser transferidos do servidor para o cliente, onde as consultas rodarão na versão local, o *Oracle Personal Express*. Na média, o cliente acessa os dados através do servidor, e a terceira, o cliente utiliza o *browser* como único instrumento para o acesso dos dados no servidor.

### 2.1.3 Microsoft DTS

O *Microsoft Data Transformation Services* (DTS) é um produto da empresa *Microsoft*, composto de um conjunto de ferramentas gráficas e objetos programáveis com o objetivo de extrair, transformar e consolidar dados de fontes diversas em simples ou múltiplo destino.

Conforme Barbieri (2001), a manipulação de dados pode ser efetuada através de passos, agrupados em pacotes, que podem ser automatizados e ter também o seu fluxo controlado, por desvios e condicionantes. Dentro de um passo, é possível executar um SQL<sup>4</sup>, um *script* em *Java*, *Perl* ou *Visual Basic*, qualquer programa externo ou ainda recuperar um outro pacote de ações, inclusive um envio de *e-mail*.

Segundo Youness (2000), o DTS permite referenciar *scripts* do banco de dados, procedimentos armazenados e programas externos responsáveis pela execução da validação dos dados que serão migrados ao DW.

Segundo Peterson et al. (1999), as principais funcionalidades apresentadas pelo *Microsoft DTS* são:

1. Suporte a múltiplas fontes de dados como fonte e destino;
2. Importação de dados em arquivo texto para o *SQL Server*;

---

<sup>4</sup> SQL é uma linguagem, criada pela IBM, usada para a definição e manipulação de dados em um banco de dados relacional (WIKIPEDIA, 2007)

3. Operação de manipulação sobre cada registro de dado extraído da fonte usando funções *scripts VBScript*<sup>5</sup>, *JScript*<sup>6</sup>, ou *PerlScript*<sup>7</sup>;
4. Transferência de objetos do *SQL Server* de um servidor para outro;
5. Transferência do esquema e dados de um banco relacional para o *SQL Server*;
6. Registro de como os dados foram manipulados;

### 2.1.4 Business Information Warehouse

O *Business Information Warehouse* é o produto da empresa SAP, que tem por objetivo a disponibilização de um ambiente separado do produto de ERP<sup>8</sup> da mesma empresa, o ERP-SAP R/3, para o tratamento exclusivo de informações gerenciais. É uma ferramenta que já vem com alguns itens pré-configurados, como *templates*<sup>9</sup> de relatórios, modelos definidos, além de diversos procedimentos de ETL (BARBIERI, 2001).

A ferramenta é composta por três módulos: Explorador de Negócios (*Business Explorer*), Servidor BW (*BW Server*) e os Extratores de dados.

No *Explorer*, os objetivos são a análise e produção de relatórios, a partir de um vasto repertório de relatórios pré-formatados, ou relatórios dimensionais, além de possibilitar manipulação de dados através de planilhas do *Microsoft Excel*<sup>10</sup>.

Os extratores de dados atuam de duas formas: para bases SAP e bases não-SAP. Com relação às bases SAP, os dados são carregados transparentemente até o repositório de dados DW, através de processos já disponibilizados pela ferramenta. Para bases não-SAP, a solução é utilizar ferramentas de terceiros, que estão disponíveis no mercado, porém para bases Oracle, a SAP possui um mecanismo de extração nativa.

---

5 *Visual Basic Scripting Edition*. É um subconjunto da linguagem de programação do *Visual Basic* da *Microsoft*, otimizado para a programação relacionada à Web (WIKIPEDIA, 2007).

6 *Jscript* é a denominação dada pela *Microsoft* para a linguagem de *scripts Java* chamada *JavaScript*. (MACHADO, 2000)

7 *PerlScript* é um *script ActiveX* desenvolvido pela empresa *ActiveState* (WIKIPEDIA, 2007).

8 Os ERPs são sistemas de informação integrados adquiridos na forma de pacotes comerciais de *software* com a finalidade de dar suporte à maioria das operações de uma empresa (BERVIAN apud SOUZA e ZWICKER, 2000).

9 *Template* é um documento sem conteúdo, com apenas a apresentação visual (apenas cabeçalhos, por exemplo) e instruções sobre onde e qual tipo de conteúdo deve entrar a cada parcela da apresentação (WIKIPEDIA, 2007).

10 Ferramenta para criação e edição de planilhas de cálculo do pacote *Office* da *Microsoft* (MICROSOFT, 2007)

O *BW Server* é o componente que efetivamente realiza os processamentos das consultas e demais solicitações. Ele trata os dados multidimensionais através de *drill-down*, *roll-up* e *drill-across*. Possui ainda um *cache* dos relatórios mais utilizados, que pode acabar otimizando as requisições repetidas (BARBIERI, 2001).

### 2.1.5 Comparativo das ferramentas DW

No quadro 2.1 estão alguns dos recursos principais das ferramentas de *Data Warehouse* que foram apresentadas nesse trabalho. Os itens foram retirados de uma relação feita por Gonçalves (2002). Nesse levantamento, alguns itens como desempenho e escalabilidade foram analisados. Porém, estes itens não foram analisados nesse projeto pelo fato de que isso não seja o escopo principal do trabalho, mas uma apuração de recursos que deveriam ser testados em cada uma das ferramentas.

Quadro 2.1 – Recursos das ferramentas de *Data Warehouse*

<i>Nr</i>	<i>Descrição</i>
1	Extrair dados de diversas fontes
2	Executar em ambiente multi-plataforma
3	Permitir compactação de dados
4	Permitir criptografia de dados
5	Conter módulo para administração
6	Permitir gerenciamento centralizado
7	Conter repositório de metadados
8	Permitir <i>schedule</i> de tarefas
9	Disponibilizar interface gráfica para montar cargas de dados
10	Disponibilizar uma linguagem para transformação dos dados
11	Gerar código através de diagramas gráficos
12	Permitir importar metadados

Fonte: Adaptado de Gonçalves (2002)

Assim, o seguinte quadro pode ser desenvolvido, através de manuais técnicos e casos de uso, comparando os recursos das ferramentas descritas nesse trabalho:

Quadro 2.2 – Comparativo de ferramentas *Data Warehouse*

<i>Ferramenta /Recurso</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>	<i>12</i>
Oracle Warehouse Builder	X	X			X	X	X	X	X		X	X

<b><i>Ferramenta /Recurso</i></b>	<b><i>1</i></b>	<b><i>2</i></b>	<b><i>3</i></b>	<b><i>4</i></b>	<b><i>5</i></b>	<b><i>6</i></b>	<b><i>7</i></b>	<b><i>8</i></b>	<b><i>9</i></b>	<b><i>10</i></b>	<b><i>11</i></b>	<b><i>12</i></b>
Oracle Express	X	X			X	X	X	X	X		X	X
Microsoft DTS	X				X	X	X	X	X		X	X
Business Information Warehouse	X	X			X	X	X		X		X	X

Com relação ao item 2 não estar presente na ferramenta *Microsoft DTS*, deve-se ao fato da mesma não ser executada em sistemas operacionais não-*Windows*. Os itens 3 e 4 e 10 não foram localizados em nenhuma das documentações das ferramentas, portanto, chegou-se a conclusão de que as funcionalidades não estão presentes. O mesmo ocorre no item 8 para a ferramenta *Business Information Warehouse*, no qual não foi identificado nenhum recurso de *schedule* de tarefas.

## **2.2 Ferramentas OLAP**

Nessa seção serão mostradas algumas das ferramentas de análise OLAP disponíveis no mercado. As ferramentas a serem estudadas são: *Oracle Business Intelligence*, *Microsoft Analysis Services* e as ferramentas *open source: Pentaho e OpenI*.

Assim como realizado com o DW, ao final do capítulo será feito um breve comparativo entre as ferramentas de OLAP descritas, trazendo as principais diferenças entre elas, num quadro com os pontos fundamentais que compõem esse tipo de produto.

### **2.2.1 Oracle Business Intelligence**

Como o próprio nome já indica, o *Oracle Business Intelligence (Oracle BI)* é a solução para *Business Intelligence* da *Oracle Corporation*. É um produto bastante robusto, que atende a toda uma gama de requisitos analíticos das empresas, que inclui consultas *ad-hoc*, geração de relatórios e análise, ETL e desenvolvimento de aplicativos de BI. Suas principais ferramentas são (ORACLE BI, 2006):

***Oracle Discoverer*** – Consulta, geração de relatórios e análise com recursos de formatação de dados;

***Oracle Spreadsheet Add-In*** – Funcionalidade para acessar dados do DW através de planilhas do *Microsoft Excel*;

***Oracle BI Beans*** – Desenvolvimento personalizado de aplicativos de BI;

**Oracle Warehouse Builder** – ferramenta de ETL já explicada anteriormente.

Com essas ferramentas, o *Oracle Business Intelligence* torna-se um produto completo, com acesso integrado ao banco de dados, eliminando necessidades de integração entre componentes de BI de fornecedores diferentes.

Dentre as ferramentas que compõem o pacote *Oracle Business Intelligence*, o *Oracle Discoverer*, usado no projeto, será explicado a seguir.

### 2.2.1.1 Oracle Discoverer

O *Oracle Discoverer* é a ferramenta de análise OLAP do pacote *Oracle BI*, que é baseada na arquitetura ROLAP e atende às necessidades de inteligência empresarial, como consultas *ad-hoc*, criação de relatórios e gráficos, explorações e consultas na *web*, através de DW ou *Data Marts*. Seu acesso aos bancos de dados é feito através do SQL Net, no caso de um SGBD *Oracle*, ou via ODBC para outros SGBD's (ORACLE 2005a).

Além de atender bancos relacionais, a última versão do produto, 10g, traz a opção de análise em ambientes multidimensionais, desde que esteja baseada em um banco de dados *Oracle* na versão *Enterprise Edition*, com a opção OLAP instalada. A figura 2.3 ilustra um esquema mostrando a estrutura da ferramenta *Oracle Discoverer*, mostrando seus componentes principais (ORACLE 2005a).

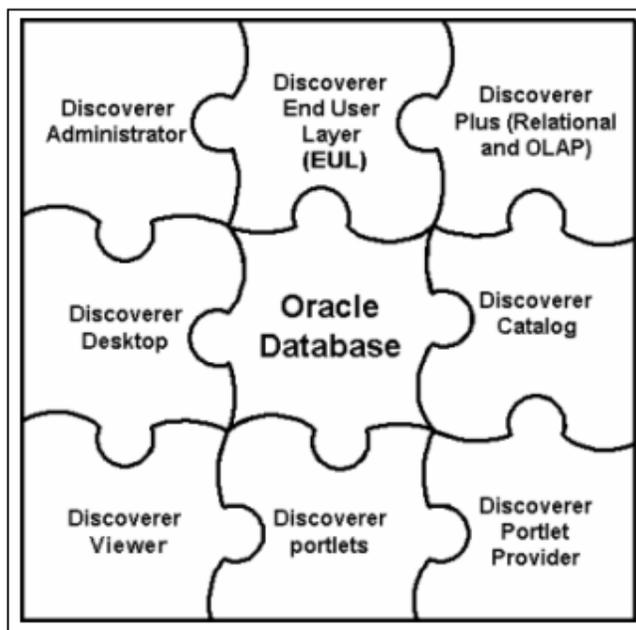


Figura 2.3 – Componentes do *Oracle Discoverer*  
Fonte: Oracle (2005a)

### 2.2.1.1.1 Discoverer Administrator

O *Discoverer Administrator* é o componente responsável por gerenciar o ambiente de dados que será utilizado para as consultas dos usuários finais. Nele, há uma camada lógica de metadados chamada *End User Layer* (EUL), onde são criadas as hierarquias de dados, tabelas sumariadas, lista de valores e as áreas de negócios, chamadas de *Business Areas* (ORACLE 2005a).

Como o próprio nome aponta, as áreas de negócios são conjuntos de informações (tabelas ou visões) relacionadas entre si e agrupadas logicamente, com intuito de atender a uma finalidade comum de negócios e necessidades de informação do usuário. Por exemplo, poderia ser criada dentro de uma EUL, uma área de negócios voltada para um DM de vendas e outra para um DM relacionado aos dados de produção (ORACLE 2005a).

Cada *Business Area* pode ser composta de um ou mais *folders*, que são um conjunto de dados retirados do BD, que podem ser uma tabela, *view*, junções de tabelas ou até mesmo um SQL personalizado. Cada *folder* é composto de itens, que podem ser colunas da tabela ou *view*, ou ainda fórmulas desenvolvidas para um fim específico. Os itens podem ter sua máscara e nomes alterados, de forma que, quando visualizados pelo usuário final, estejam da forma mais amigável possível. Na figura 2.4 há uma representação do *Discoverer Administrator*, composta por uma *Business Area* com alguns *folders* (ORACLE 2005a).

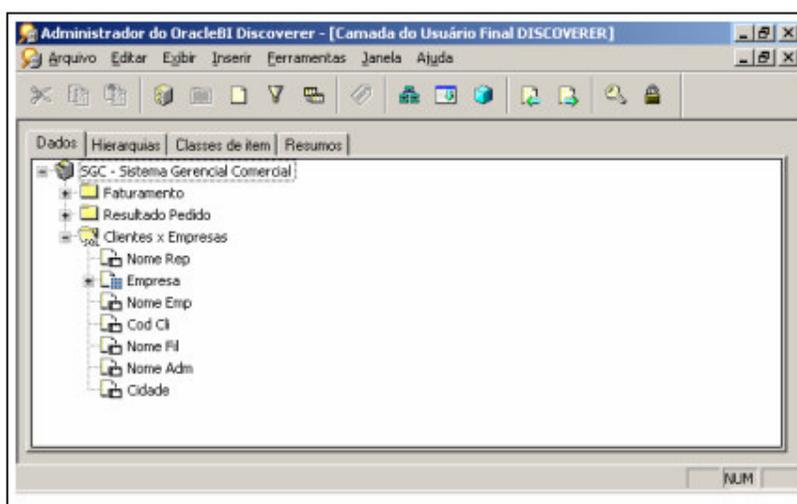


Figura 2.4 – *Discoverer Administrator*

Fonte: Adaptado de Oracle (2005a)

No *Administrator*, existem funcionalidades para dar permissão aos usuários para que visualizem ou não determinada *Business Area*. Ou ainda, permitir que certos usuários criem ou compartilhem consultas baseadas em determinadas *Business Areas* (ORACLE 2005a).

Uma das funções do *Administrator* que merece destaque é a criação de hierarquias. As hierarquias podem ser criadas através de uma lógica de dados ou datas. Por exemplo, é possível criar uma hierarquia que possui como nível máximo o cliente, num nível mais baixo o produto e em seguida o pedido. Ou ainda uma hierarquia com ano, semestre, mês e dia. Essas montagens servirão para a utilização do *drill-down* e *roll-up* nas consultas do usuário final (ORACLE 2005a).

#### 2.2.1.1.2 *Discoverer Desktop*

Esse é o componente responsável pela interação com o usuário final, que possibilita desenvolver ou mesmo consultar relatórios baseados nas *Business Areas* disponibilizadas a ele através do *Discoverer Administrator*. É uma aplicação que deve ser instalada na máquina do cliente e pode rodar em *Windows*, *Solaris*, *Linux* e *HP-UX*, mas não pode ser rodada na web (ORACLE, 2005b).

A ferramenta possui uma interface bastante semelhante a um programa de planilha de cálculo, pois assim como, por exemplo, um arquivo do *Microsoft Excel*, um documento do *Discoverer*, que é chamado de *workbook*, pode conter diversas páginas (*worksheets*) com assuntos diferentes (ORACLE, 2005b). A estrutura que compõe a ferramenta pode ser visualizada na figura 2.5.



Figura 2.5 – Estrutura do *Discoverer Desktop*  
Fonte: Adaptado de Meira e Souza (2002)

A *worksheet* possui dois tipos principais de visualização de dados: tabela e tabelas com referência cruzada. Na tabela os itens são exibidos em linhas e colunas, como se estivesse visualizando uma tabela diretamente do banco de dados. A tabela com referência

cruzada permite uma visão multidimensional dos dados, na qual o usuário pode cruzar linhas e colunas, com identações de dados. Cada formato permite ainda a inclusão dos chamados itens de página, que são parâmetros que permitem novas consultas em uma mesma *worksheet* sem uma nova requisição ao banco de dados (ORACLE, 2005b). A figura 2.6 mostra uma consulta formatada em tabela de referência cruzada, com itens de página.

The screenshot shows a software interface with a menu bar (Arquivo, Editar, Exibir, Folha, Formatar, Ferramentas, Gráfico, Janela, Ajuda) and a toolbar. The main window title is 'Área de Trabalho do Oracle Business Intelligence Discoverer - [SGC00100 - Sistema Gerencial Comercial]'. Below the toolbar, there's a header for 'Sistema Gerencial Comercial' with 'Faturamento por Representante' and 'Data: 11/05/07 Pág.: 1 à 1'. A filter bar shows 'Ano: 2007', 'Segmento: <Tudo>', 'Tipo de Pedido: <Tudo>', 'Mercado: <Tudo>', and 'Administrador: 2-EDUARDO'. The main data area is a table with columns for months (01/2007, 02/2007, 03/2007, 04/2007, 05/2007) and a 'Total' column. Rows list representatives from AA to DD. Red arrows point to the filter bar ('Itens de página'), the table rows ('Linhas'), and the table columns ('Colunas').

Representante	Valor Faturado					Total	
	Mês	> 01/2007	> 02/2007	> 03/2007	> 04/2007		> 05/2007
> Representante AA		99.999.999	99.999.999	99.999.999	99.999.999	99.999.999	99.999.999
> Representante AB		99.999.999	99.999.999	99.999.999	99.999.999	99.999.999	99.999.999
> Representante BA		99.999.999	99.999.999	99.999.999	99.999.999	99.999.999	99.999.999
> Representante BB		99.999.999	99.999.999	99.999.999	99.999.999	99.999.999	99.999.999
> Representante XX		99.999.999	99.999.999	99.999.999	99.999.999	99.999.999	99.999.999
> Reoresentante XY		99.999.999	99.999.999	99.999.999	99.999.999	99.999.999	99.999.999
> Reoresentante YY		99.999.999	99.999.999	99.999.999	99.999.999	99.999.999	99.999.999
> Representante YX		99.999.999	99.999.999	99.999.999	99.999.999	99.999.999	99.999.999
> Representante XZ		99.999.999	99.999.999	99.999.999	99.999.999	99.999.999	99.999.999
> Representante ZX		99.999.999	99.999.999	99.999.999	99.999.999	99.999.999	99.999.999
> Reoresentante ZZ		99.999.999	99.999.999	99.999.999	99.999.999	99.999.999	99.999.999
> Representante CC		99.999.999	99.999.999	99.999.999	99.999.999	99.999.999	99.999.999
> Representante CD		99.999.999	99.999.999	99.999.999	99.999.999	99.999.999	99.999.999
> Representante DC		99.999.999	99.999.999	99.999.999	99.999.999	99.999.999	99.999.999
> Representante DD		99.999.999	99.999.999	99.999.999	99.999.999	99.999.999	99.999.999
Total		99.999.999	99.999.999	99.999.999	99.999.999	99.999.999	99.999.999

Figura 2.6 – Exemplo de consulta com tabela de referência cruzada

O usuário pode criar sua consulta de forma bastante simples, bastando selecionar os dados que deseja visualizar da *Business Area* desejada. Assim o usuário não precisa conhecer conceitos de banco relacional ou garimpar dados em diversos locais diferentes. A ferramenta possui uma série de funcionalidades que ajudam a enriquecer a consulta, dentre elas (ORACLE, 2005b):

**Gráficos:** é possível desenvolver gráficos através das consultas criadas pelo usuário;

**Exportar dados:** exportar os dados para uma planilha em *Excel* ou nos mais diversos formatos de arquivo, como pdf<sup>11</sup>, html<sup>12</sup>, ou arquivo de texto formatado;

11 Criado pela Adobe, o formato PDF (*Portable Document Format*) é uma especificação disponível publicamente usada por entidades de padronização do mundo inteiro para a distribuição e a troca mais seguras e confiáveis de documentos eletrônicos (ADOBE, 2007).

12 HTML (*HyperText Markup Language*, que significa Linguagem de Marcação de Hipertexto) é uma linguagem de marcação utilizada para produzir páginas na *Web* (WIKIPEDIA, 2007).

**Totais:** há ferramentas de totalização de dados, que incluem funções de sumariação, percentual, valor máximo e mínimo em um conjunto de dados;

**Itens calculados:** o usuário pode criar itens através de cálculos que podem ser entre colunas existentes na consulta;

**Condições:** o usuário pode criar suas próprias condições para o resultado da pesquisa. Nesse componente, por exemplo, o usuário pode informar que deseja visualizar somente dados que estejam com o valor maior que o indicado por ele. É possível criar condições dinâmicas, que devem ser parametrizadas a cada re-consulta;

**Formatação de dados:** pode-se formatar a máscara, tamanho e cor da fonte dos dados;

**Exceções:** funcionalidade que permite ao usuário destacar dados dentro da consulta. Exemplo: em uma consulta de faturamento por cliente, o usuário pode indicar que deseja visualizar na cor vermelha todos aqueles clientes que possuem faturamento inferior a R\$ 10.000,00.

Com relação a análise OLAP, o *Discoverer* possui as seguinte funcionalidades:

***Drill-down e Roll-up:*** através de hierarquias criadas no *Administrator*, é possível navegar através dos diversos níveis de dados;

***Slice e Dice:*** permite que as consultas possam ser visualizadas de forma particionada ou fatiada com base nas dimensões;

***Pivoting:*** permite, através de clicar com o *mouse*, transformar linhas em colunas e vice-versa;

***Sorts:*** o usuário pode ordenar a consulta pela informação que desejar;

**Filtros:** através dos componentes de condição e parametrização, é possível filtrar somente os dados, exibindo somente o desejado.

O usuário possui ainda condições de agendar as suas consultas, ou seja, executá-las em determinado horário para que quando ele necessite das informações, não tenha a necessidade de aguardar que as mesmas sejam buscadas do banco de dados. Além disso, há a

possibilidade de salvar os *workbooks* tanto no banco de dados quanto em arquivos, que podem ser compartilhados com outros usuários (ORACLE, 2005b).

#### **2.2.1.1.3 Discoverer Plus e Discoverer Viewer**

O *Discoverer Plus* é nada mais do que a versão Web do *Discoverer Desktop*. A ferramenta apresenta as mesmas funcionalidades da versão desktop, porém roda através de um *browser* (ORACLE, 2005c).

O *Discoverer Viewer* é a ferramenta que permite ao usuário visualizar e salvar as consultas desenvolvidas através do *Discoverer Desktop* e *Discoverer Plus*, porém, não permite ao mesmo desenvolver as suas próprias consultas. O *Discoverer Viewer* ainda possibilita ao usuário ajustar seus relatórios usando parâmetros específicos e permite a conversão e publicação de relatórios gerados usando diferentes formatos como HTML, PDF ou XML (ORACLE, 2005d).

#### **2.2.1.2 Oracle Spreadsheet Add-In**

O *Oracle Spreadsheet Add-In* é um componente que permite ao usuário acessar os dados do OLAP Oracle através do *Microsoft Excel*. Quando o *Spreadsheet* é instalado no cliente, é gerado um item de menu dentro do *Excel* chamado "*OracleBI*". Nele, os dados podem ser selecionados e carregados dentro da planilha, permitindo ao usuário analisar e formatá-los da maneira que desejar. Para a seleção de dados não é necessário nenhum tipo de conhecimento técnico da estrutura do banco dados, pois os dados virão formatados de acordo com as *Business Areas* criadas no *Discoverer Administrator*.

O componente está disponível somente para usuários que possuam *Windows* e em banco de dados *Oracle* que possuam a opção OLAP instalada.

#### **2.2.1.3 Oracle BI Beans**

O *Oracle BI Beans* é um componente integrado a ferramenta da *Oracle JDeveloper*, que é utilizada para desenvolvimento de aplicações em Java. No *BI Beans*, os desenvolvedores podem desenvolver aplicações personalizadas de *Business Intelligence*, incluindo as mais diversas funcionalidades OLAP.

*Oracle BI Beans* inclui *beans*<sup>13</sup> de apresentação (gráfico e *crossstab*), beans de dados (construtores de consulta e cálculos) e serviços de persistência, que podem ser implantados tanto em aplicações cliente HTML, quanto em aplicações cliente *Java*.

### 2.2.2 *Microsoft Analysis Services*

O *Microsoft Analysis Services* é a ferramenta OLAP da *Microsoft Corporation*, que vem acoplada ao banco de dados da mesma empresa, o *SQL-Server*. Oferece uma visão integrada dos dados empresariais para relatórios, análise OLAP, *scorecards* de indicadores chave de desempenho chave (*key performance indicator* - KPI) e *data mining* (MICROSOFT, 2007). É uma ferramenta que suporta MOLAP, ROLAP, HOLAP e DOLAP. Conforme Barbieri (2001), é dividida em cinco componentes básicos:

1. *PivotTable Service*, que pode ser conectado via *Excel*, *Office 2000* e versões posteriores ou ferramentas de terceiros;
2. Métodos para acesso aos dados, baseado no padrão OLE/DB-OLAP<sup>14</sup>;
3. Máquina para serviços e armazenamento OLAP;
4. Gerenciador OLAP;
5. *DTS- Data Transformation Service* (já explicado anteriormente).

A ferramenta é dividida em quatro camadas principais: camada cliente, camada servidora, camada de gerência e acesso e camada de retaguarda.

A camada cliente contém os componentes que fazem interfaces com o usuário e podem ser as próprias planilhas *Excel* e produtos de terceiros. Um dos recursos que merece destaque nessa camada é o *PivotTable Service*, que manipula os dados e os armazena em cubos ou fatias, permitindo que as solicitações sejam feitas no próprio cliente, sem a necessidade de acesso constante ao servidor, formando um DOLAP.

---

13 *Beans* são componentes reutilizáveis da plataforma Java para criação de aplicações mais sofisticadas (JAVA, 2007)

14 O *Microsoft OLE DB* para OLAP é um conjunto de objetos e interfaces que estendem a habilidade do OLE DB para prover acesso ao armazenamento de dados multidimensionais (MICROSOFT, 2007).

Na camada servidora está a máquina de serviços OLAP, onde as requisições são recebidas e são realizados os acessos e tratamentos das mesmas. Os dados podem ser armazenados tanto em esquema em estrela (*star schema*) como em esquema *snowflake*. Um dos diferenciais da ferramenta está relacionado ao tratamento dos cubos de dados. No *Analysis Services* é possível particionar um cubo lógico em diversos cubos físicos, inclusive em diferentes servidores. Além disso, é possível criar cubos virtuais, uma espécie de *view* multidimensional, que pode ser formado pela junção de diversos cubos (BARBIERI, 2001).

A camada de gerência, chamada também de *OLAP Manager* é aquela que gerencia o ambiente OLAP, tendo interfaces para a criação de tabelas e definições de agregados. Um dos seus componentes principais é o *Cube Browser*, que permite ao usuário acessar de forma interativa os dados das tabelas, cubos e metadados. Sua linguagem padrão para acesso de dados é o MDX<sup>15</sup>.

A camada de retaguarda é o DTS (*Data Transformation Services*), ferramenta de ETL, que já foi explicada no capítulo 2.1.3.

### **2.2.3 Pentaho BI**

A seguir será apresentada a ferramenta *Pentaho BI*. As informações da mesma foram retiradas do seu site oficial ([www.pentaho.org](http://www.pentaho.org)).

O projeto *Pentaho BI* é uma iniciativa da comunidade de *Open Source* para fornecer às empresas uma solução que supra as necessidades de sistemas de *Business Intelligence*. Pelo fato de ser *Open Source*, inovações e novas implementações são constantes na ferramenta, agregando funcionalidades e inserindo melhorias.

Na plataforma *Pentaho BI*, o elemento central na arquitetura da *Pentaho Open BI Suite*, é centrada em processos cujo controle central é realizado através de um mecanismo de *workflow*. Este mecanismo utiliza definições de processos para determinar os processos de BI que serão executados na plataforma. Os processos podem ser personalizados e novos podem ser facilmente incorporados.

---

15 A MDX é uma linguagem de expressão multidimensional definida na especificação OLE DB para OLAP (MICROSOFT, 2007)

A ferramenta integra *workflow*, regras de negócios, notificação e entrega de informações, escalonamento, auditoria, integração de aplicações navegação por conteúdo, interfaces ao usuário, ferramentas de projeto e administração de relatórios, análise, *dashboards* e componentes de mineração de dados.

A arquitetura do *Pentaho* é dividida em Servidor *Pentaho*, e o *Design Studio*, baseado no *Eclipse*<sup>16</sup>.

O Servidor *Pentaho* trabalha em acordo com o padrão de servidor web *J2EE Server*, tais como *Apache*<sup>17</sup>, *JBOSS AS*<sup>18</sup>, *WebSphere*<sup>19</sup>, *WebLogic*<sup>20</sup> e *Oracle AS*<sup>21</sup>. Os conteúdos apresentados ao usuário podem ser em XML, HTML, ou mostrados na tela pelos *JSR-168 portlets*<sup>22</sup>. O servidor *Pentaho Server* fornece ainda os mecanismos e componentes para gerar relatórios, análises, regras de negócios, e-mail e fluxos de processos.

O *Pentaho Studio*, que é baseado no *Eclipse*, provê ferramentas amigáveis de projeto de relatórios, *dashboards* e visões analíticas; um processo de projeto de *workflow*; editor de regras de negócios; console de mineração de dados para preparação de dados; e ferramentas de modelagem OLAP. Essa ferramenta é multi-plataforma, desenvolvida em *Java* e é instalada na máquina dos desenvolvedores e dos administradores do sistema.

Dentre suas principais vantagens, o *Pentaho* traz uma fácil personalização de suas funcionalidades e integração com sistemas externos. Além claro, de possuir um custo bastante menor que a maioria das soluções de *Business Intelligence* existentes no mercado.

### 2.2.4 *OpenI*

A seguir será tratada da ferramenta de *Business Intelligence OpenI*, no qual todas as informações referentes a ela foram retiradas do seu site oficial ([www.openi.org](http://www.openi.org)).

O *OpenI* é uma ferramenta de análise OLAP, *open-source*, que é disponibilizada como uma aplicação *Web J2EE*, e suporta servidores OLAP compatíveis com o padrão

---

16 *Eclipse* - Ferramenta de desenvolvimento de aplicações JAVA - [www.eclipse.org](http://www.eclipse.org)

17 *Apache* – [www.apache.org](http://www.apache.org)

18 *JBOSS AS* – [www.jboss.org](http://www.jboss.org)

19 *WebSphere* - [www.ibm.com/software/websphere](http://www.ibm.com/software/websphere)

20 *WebLogic*- [www.bea.com](http://www.bea.com)

21 *Oracle AS* - [www.oracle.com/appserver/index.html](http://www.oracle.com/appserver/index.html)

22 *JSR-168 Portlets* são aplicações em tecnologia *Java* baseadas em componentes *web*, gerenciado por um *portlet container* que processa requests e gera conteúdo dinâmico (OLIVEIRA, 2004).

XMLA, incluindo o *Microsoft Analysis Services* e *Mondrian*. Possui a implementação padrão executando no servidor *Web Apache Tomcat*. Ainda pode publicar relatórios analíticos baseados na *Web* a partir de três tipos de servidores de dados: Servidores OLAP, servidores de bancos de dados relacionais e servidores de *data mining*.

O OpenI está dividido em três categorias principais:

**Componente de conexão:** A função dos conectores é "falar" o idioma nativo das fontes de dados de análise. Para os bancos de dados relacionais, o *OpenI* utiliza o JDBC. Para os servidores OLAP, é utilizado o XMLA como o protocolo padrão de comunicação. Para servidores RDBMS, é utilizado o padrão ODBC. Para os serviços de *data mining*, o *OpenI* é integrado com a plataforma *open source* de *data mining* chamada *The R project*<sup>23</sup>.

**Componente de Relatório:** O *OpenI* utiliza linguagens específicas de definição de relatório (RDL - *Report Definition Language*) para definir e explorar os relatórios criados na plataforma. Para os relatórios em base de dados relacionais, ele utiliza a RDL .jrmxl do *JasperReports*<sup>24</sup>. Para relatórios OLAP e *data mining*, ele implementa sua própria RDL baseada em XML.

**Componente de interface com o usuário:** O componente de interface com o usuário do *OpenI* reúne outros projetos em uma única plataforma, tornando-a amigável aos usuários não-técnicos, como os analistas de negócio. São utilizados componentes dos projetos *Jpivot*<sup>25</sup> e *Jfreechart*<sup>26</sup>, unificando-os como um consistente *framework Web* de navegação.

### 2.2.5 Comparativo das ferramentas OLAP

Baseado em Klein (1999), pode-se criar o seguinte quadro com recursos importantes nas ferramentas OLAP:

Quadro 2.3 – Recursos das ferramentas OLAP

<i>Nr</i>	<i>Característica</i>	<i>Descrição</i>
1	Múltiplas camadas	Camada do usuário final, camada do administrador, camada para Web e camada de metadados
2	Interface amigável	Navegação, apresentação visual alternativa, guia para formulação

<sup>23</sup> *The R project* - <http://www.rproject.org>

<sup>24</sup> *JasperReports* - <http://jasperreports.sourceforge.net>

<sup>25</sup> *JPivot* - <http://jpivot.sourceforge.net>

<sup>26</sup> *JfreeChart* - <http://www.jfree.org/jfreechart>

<i>Nr</i>	<i>Característica</i>	<i>Descrição</i>
		de consultas etc.
3	Múltiplos usuários	Possuir uma arquitetura do tipo cliente/servidor
4	Flexibilidade de consulta	Operações de “rollup”, “drill”, “slice e dice”, “ranking”, “filtros”, “pivoting”, alinhamento de dimensões, manipulação de hierarquias assimétricas e incompletas etc.
5	Manipulação de dados não-convencionais	Capacidade de a ferramenta lidar, além dos dados convencionais (numéricos, datas e strings), com dados não-convencionais no cubo.
6	Funções Matemáticas	Agregações, cálculos procedurais e com cruzamento de dimensões, precedência de fórmulas, matrizes esparsas etc.
7	Manipulação de hierarquias	Estabelecimento de hierarquias nas dimensões, possibilitando que um membro pai represente a consolidação de membros filhos.
8	Funcionalidades de manutenção	Gerenciamento de atualização das tabelas sumariadas
9	Funcionalidades para WEB	Aumento da disponibilidade do ambiente através da Web

Assim, com as ferramentas apresentadas nesse trabalho, foi possível levantar uma análise das mesmas, baseado em manuais técnicos e estudos de caso, criando-se o seguinte quadro de comparação entre elas:

Quadro 2.4 – Comparação das ferramentas OLAP

<i>Ferramenta/recurso</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>
<i>Oracle Business Intelligence</i>	X	X	X	X	X	X	X	X	X
<i>Microsoft Analysis Services</i>	X	X	X	X		X	X	X	X
<i>Pentaho</i>	X	X	X	X		X	X		X
<i>OpenI</i>		X	X	X	X		X		X

Com relação aos itens faltantes, esses não possuem referências de sua existência nas documentações oficiais das ferramentas analisadas.

## 3 ESTUDO DE CASO

Na concepção de Simon (2007), o estudo de caso é uma técnica de estudo, em que é realizada uma pesquisa sobre um caso particular, para tirar conclusões sobre princípios gerais daquele caso específico.

Nesse capítulo será realizado o estudo de caso da aplicação de um sistema de *Business Intelligence* para a construção do *Data Mart* comercial na empresa Box Print, utilizando ferramentas *Oracle*.

O capítulo passará por uma explicação da empresa em que o projeto foi aplicado, sua estrutura, forma de organização e como a tecnologia está inserida nela. A seguir será mostrado o cenário das informações disponíveis para a tomada de decisão, quais suas deficiências e problemas para a sua busca e análise. Por fim, será tratado de forma mais ampla o porquê da utilização de tecnologias *Oracle* nesse projeto.

### **3.1 Estrutura da Empresa**

A Box Print é uma empresa do ramo de embalagens, que fabrica produtos na área de cartões diversos, microondulados e ondulados, para os setores de calçados, eletrodomésticos, farmacêuticos, gêneros alimentícios, congelados, autopeças, cutelaria, cosméticos, perfumaria e outros. Atende, além do mercado interno, clientes localizados nos demais países da América Latina e Estados Unidos.

A empresa possui quatro unidades fabris, todas localizadas no Rio Grande do Sul, e um escritório central de vendas, localizado na cidade de São Paulo-SP. A unidade matriz localiza-se em Campo Bom-RS, e é nesse local que a estrutura principal de tecnologia está instalada.

### 3.1.1 Tecnologia

Todos os processos da empresa são informatizados. São mais de oitocentos funcionários que lidam diariamente com os sistemas desenvolvidos pelo setor de TI da empresa, que possui uma equipe de oito pessoas.

A plataforma padrão de aplicação e desenvolvimento de sistemas é *Oracle*. Ela está inserida desde os SGBD's até as aplicações para o usuário final. A empresa possui um total de seis SGBD's que possuem as seguintes funções:

**Banco de produção:** atende aos diversos sistemas transacionais da empresa. Dentre eles pode-se destacar o sistema comercial, compras, administrativo, controle de estoque, qualidade e controle e planejamento de produção. Atualmente possui a versão 9i do SGBD Oracle

**Banco de desenvolvimento:** banco de dados utilizado pelo setor de TI para o desenvolvimento de aplicações e novos sistemas. É uma cópia do banco de produção, que é atualizada sempre que necessário. Possui a versão 10g do SGBD Oracle.

**Banco de modelagem:** possui o repositório dos modelos ER desenvolvidos através da ferramenta *Oracle Designer*, dos sistemas transacionais da empresa. Possui a versão 10g do SGBD Oracle.

**Sistema de clientes:** possui os dados a serem acessados através do sistema *BoxNet*, utilizado por clientes e representantes comerciais para consultar informações de pedidos e notas fiscais. Possui a versão 10g do SGBD *Oracle*.

**Site Corporativo e Intranet:** banco de dados da ferramenta *Oracle Portal*, no qual é utilizado para armazenar dados do sistema de *intranet* e do site da empresa ([www.boxprint.com.br](http://www.boxprint.com.br)). Possui a versão 10g do SGBD *Oracle*.

**Sistema de representantes:** banco de dados com informações relacionadas ao sistema utilizado pelos representantes da empresa para enviar e receber orçamentos e pedidos. Esta máquina intermedia a comunicação entre os computadores utilizados pelos representantes e o banco de produção, contendo informações do que é enviado ou recebido por ambas as pontas existentes no sistema. Cabe ainda, ressaltar que o sistema utilizado pelos

representantes possui um SGBD Oracle 8i instalado em suas respectivas máquinas e as aplicações também são desenvolvidas com tecnologia *Oracle*.

### 3.1.2 Estrutura Organizacional

A empresa possui no seu maior cargo o diretor-presidente da empresa, seguido pelos diretores de suas respectivas áreas, passando pelos gerentes, que por fim gerenciam setores específicos dentro da organização. A figura abaixo pode demonstrar resumidamente a estrutura organizacional da empresa Box Print, com detalhamento no setor de TI.

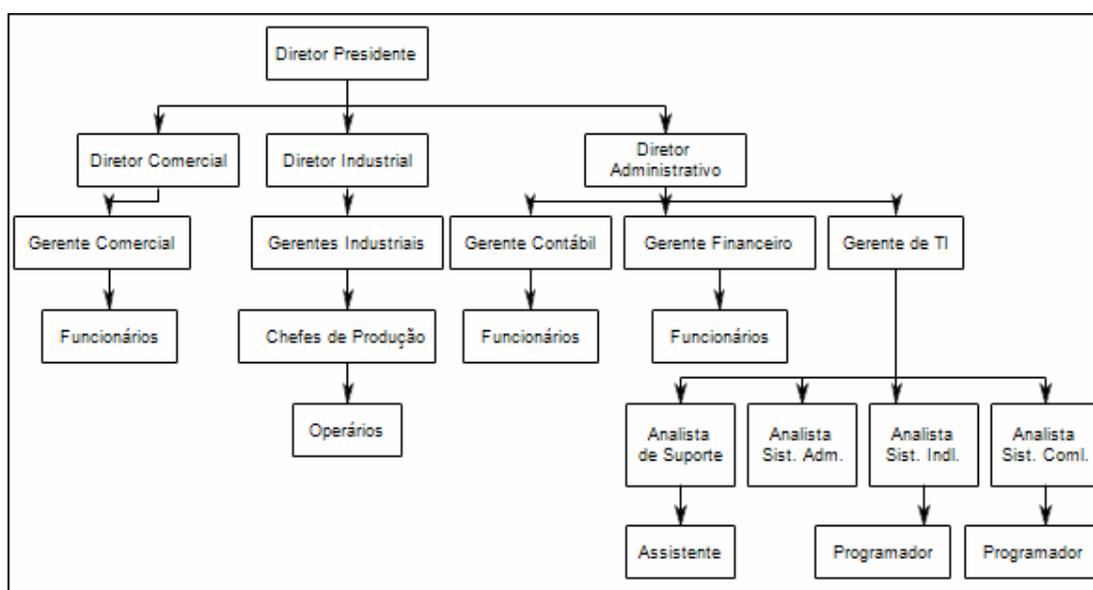


Figura 3.1 – Estrutura organizacional da empresa

O setor comercial, que é o escopo do trabalho, possui uma estrutura organizacional que tem em seu cargo principal o diretor comercial. Ele tem em uma de suas responsabilidades coordenar o trabalho dos administradores de venda. Estes por sua vez, são os responsáveis em gerenciar um grupo determinado de representantes de venda, que possuem clientes específicos a serem atendidos. Essa estrutura pode ser resumidamente representada na figura 3.2.

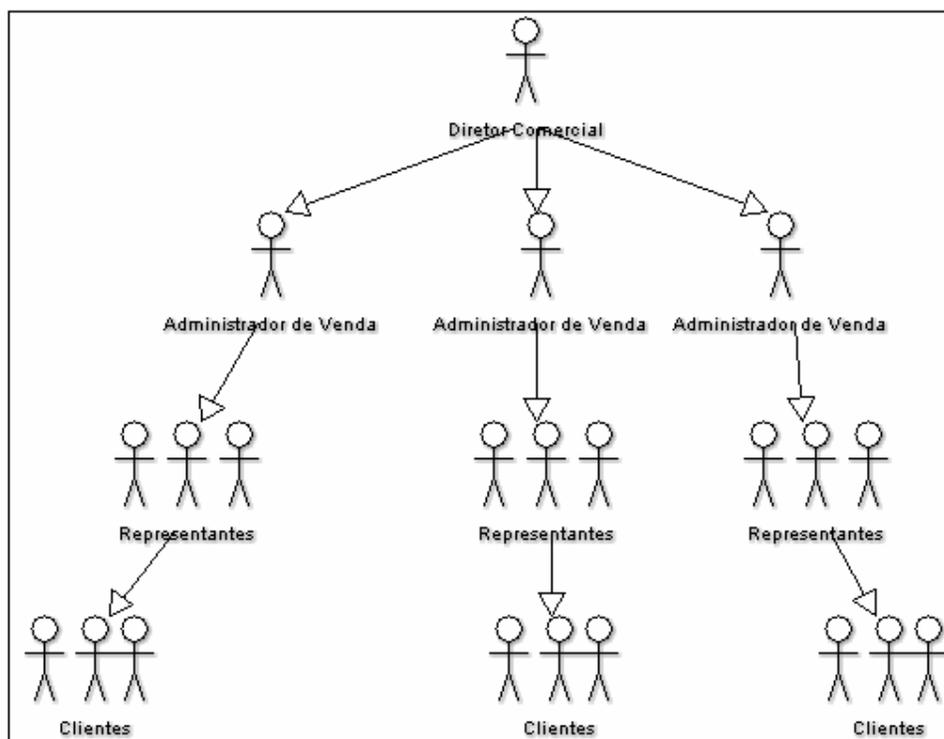


Figura 3.2 – Organização do gerenciamento de vendas

### 3.2 Problemas Atuais e Necessidades

A tecnologia *Oracle* foi incorporada na Box Print no ano de 2000, através de uma migração de um antigo sistema transacional desenvolvido em *Cobol*<sup>1</sup>. Desde então, a base de dados desse sistema está se tornando cada vez mais sólida, com um volume de dados cada vez maior, tornando-se assim uma fonte muito rica para as mais diversas requisições e, por consequência, ambiente adequado para aplicação de tecnologias que buscam informação e conhecimento.

Porém a empresa não possui atualmente, um sistema que reúna esse grande volume de dados para fins de análise gerencial que auxiliem a tomada de decisão. O que existe são diversos relatórios e consultas, desenvolvidos para fins específicos, através das ferramentas *Oracle Forms Builder* (desenvolvimento de programas) e *Oracle Reports Builder*. (desenvolvimento de relatórios). Essas aplicações estão disponíveis através dos sistemas transacionais da empresa.

1 COBOL - Linguagem originada de um aprimoramento do ARGOL, criada na década de 70 e batizada em homenagem a Blaise Pascal (WIKIPEDIA, 2007). Site Oficial: <http://www.cobolportal.com>

Porém, pelo fato de serem aplicações que possibilitam a análise gerencial e são executadas através do banco de dados transacional, acabam tendo uma geração dos resultados para o usuário de forma demorada, consumindo muito recurso do servidor, prejudicando muitas vezes o desempenho normal do mesmo.

Muitas vezes, essas formatações de informações existentes atualmente na empresa não atendem totalmente as necessidades do usuário. Com isso, muitas vezes o gerente ou diretor acaba designando um funcionário para garimpar, dentro do sistema transacional, dados de várias consultas e relatórios, implicando em uma dedicação e tempo muito grandes. Além disso, há sempre a probabilidade de que as informações possam não ser confiáveis, uma vez que o usuário, que não possui uma visão gerencial, pode cometer o erro de adicionar ou até mesmo retirar dados que seriam importantes para a tomada de decisão.

E essa realidade está localizada em todos os setores da empresa, inclusive no setor comercial. E é exatamente nessa área que este trabalho de desenvolvimento de um sistema de *Business Intelligence* será realizado. Essa escolha foi tomada baseando-se no fato de que apesar de a empresa possuir relatórios gerenciais nos diversos setores, o setor comercial possui um número mínimo desse tipo de informação e seria um ponto inicial para um futuro sistema gerencial totalmente integrado, envolvendo todas as áreas da empresa. Além do que, esse setor é extremamente estratégico para a empresa, pois através de uma análise gerencial, possibilitará um mapeamento do desempenho de administradores de venda e representantes.

### **3.5 Por que Oracle**

A escolha de utilizar ferramentas *Oracle* para o desenvolvimento do sistema deu-se ao fato de que essa tecnologia é a padrão da empresa em que o projeto será implementado. Pois já é uma tecnologia aprovada e homologada dentro da empresa, e a continuação da sua utilização nesse projeto foi dada como natural em função dessa aprovação.

Pois, como as bases de dados existentes são *Oracle*, é possível ter uma integração mais segura e com muito mais recursos disponíveis quando as ferramentas que processam as informações dessas bases são desenvolvidas pela mesma empresa.

Acredita-se que a utilização de ferramentas de outros fabricantes para as tarefas de criação da estrutura de dados até a sua visualização por parte do usuário traria menos opções de funcionalidades, sem contar que a integração com tecnologias diferentes é sempre um

obstáculo a mais no desenvolvimento de um projeto do porte desse que está sendo proposto nesse trabalho.

Além do mais, pelo fato de *Oracle* ser uma tecnologia paga, há um serviço de suporte técnico muito eficaz, pois caso surja alguma dificuldade com uso das ferramentas ou problemas nas mesmas, o atendimento ao cliente sempre traz soluções que acabam auxiliando na resolução dessas dificuldades e problemas.

No próximo capítulo será apresentado o sistema elaborado através desse estudo de caso, mostrando de que maneira ele foi desenvolvido e de que forma atendeu as necessidades da empresa Box Print.

## 4 ESTRUTURA DO SISTEMA IMPLEMENTADO

Para atender as necessidades de análise de informações gerenciais, foi desenvolvido um sistema de *Business Intelligence* através de ferramentas da *Oracle*. As ferramentas escolhidas para a geração de relatórios e publicação dos mesmos foram as mesmas previstas no projeto inicial, *Oracle Discoverer* e *Oracle Portal*. A exceção foi na modelagem e carga de dados, onde havia sido cogitada a utilização do *Oracle Warehouse Builder*, mas que acabou sendo substituído por *Oracle Designer 6i*, *scripts* em PL/SQL e recursos do próprio SGBD *Oracle*. Essa e outras questões serão abordadas com maior detalhamento no decorrer desse capítulo.

### 4.1 Arquitetura do Sistema

Conforme havia sido projetado inicialmente, o DM foi construído utilizando dados unicamente do banco de dados transacional da empresa. Esses dados, juntamente com os metadados e demais relatórios, foram armazenados em um novo banco de dados relacional *Oracle*, chamado de ORAOLAP.

Para a construção dos relatórios em *Discoverer*, foi instalado um servidor *Oracle Internet Application Server* (IAS), chamado de “BI Box Print”. Nele estão instaladas as aplicações *Oracle Discoverer Plus*, para a construção de relatórios, e *Oracle Discoverer Viewer*, para a visualização dos mesmos.

A publicação desses relatórios foi realizada utilizando-se o *Portlet Provider*, que é um recurso do *Oracle Portal* que tem a finalidade de publicar, através de *portlets*<sup>1</sup>, o que foi desenvolvido no *Oracle Discoverer*. Assim, a estrutura do sistema pode ser graficamente representada pela figura 4.1.

---

<sup>1</sup> *Portlets* são blocos do *Oracle Portal* que podem receber diversos tipos de conteúdo, desde código HTML, até programas em JAVA, ou *Discoverer*, como é o caso dessa aplicação (PORTAL, 2007).

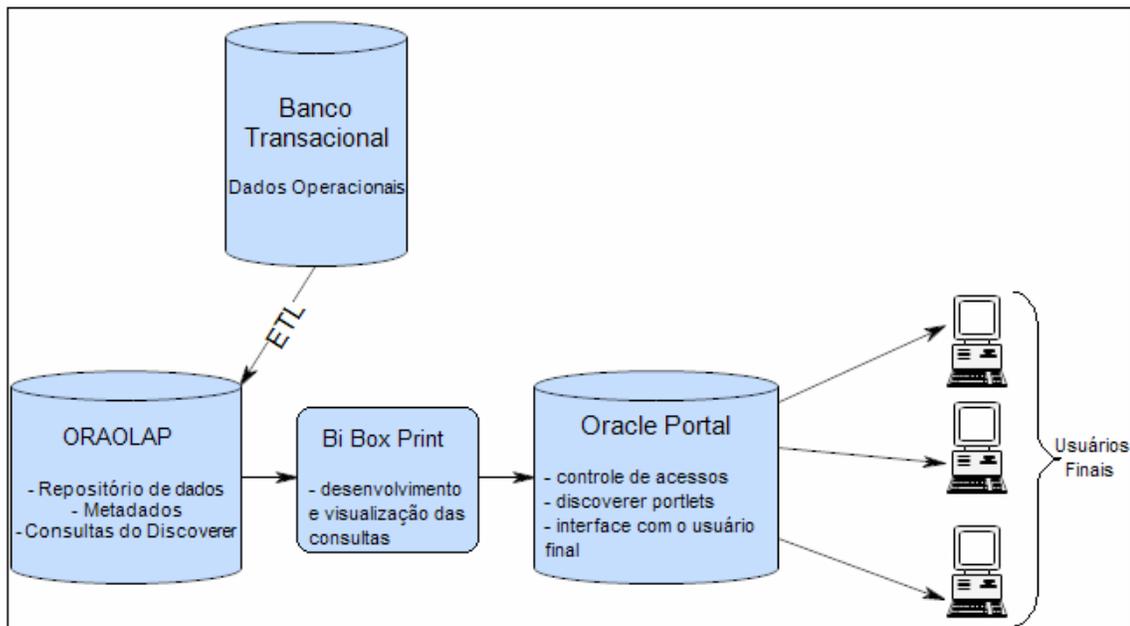


Figura 4.1 – Estrutura do Sistema

## 4.2 Modelagem do sistema

A modelagem do sistema foi desenvolvida através do *Oracle Designer*, ferramenta da *Oracle* utilizada para desenvolvimento e modelagem de sistemas. Através dela é possível desenvolver todo um sistema cliente-servidor, do modelo ER até os programas e relatórios para o usuário final (ORACLE, 2000a).

Assim, o modelo proposto para o DM apresenta uma modelagem em *star schema*, utilizando a arquitetura ROLAP de armazenamento de dados. Como pode ser visto na figura 4.2, há doze tabelas de dimensão que formam duas tabelas de fatos, SGC\_FAT\_FATURAMENTO para o módulo de faturamento e SGC\_FAT\_DEVOLUCAO para o módulo de devolução. Ambas as tabelas possuem as medições como atributos aditivos, ou seja, campos numéricos que tem a possibilidade de serem somados passíveis à dimensão relacionada. Há ainda três agregados, que foram criados para atender consultas específicas a um determinado usuário do sistema.

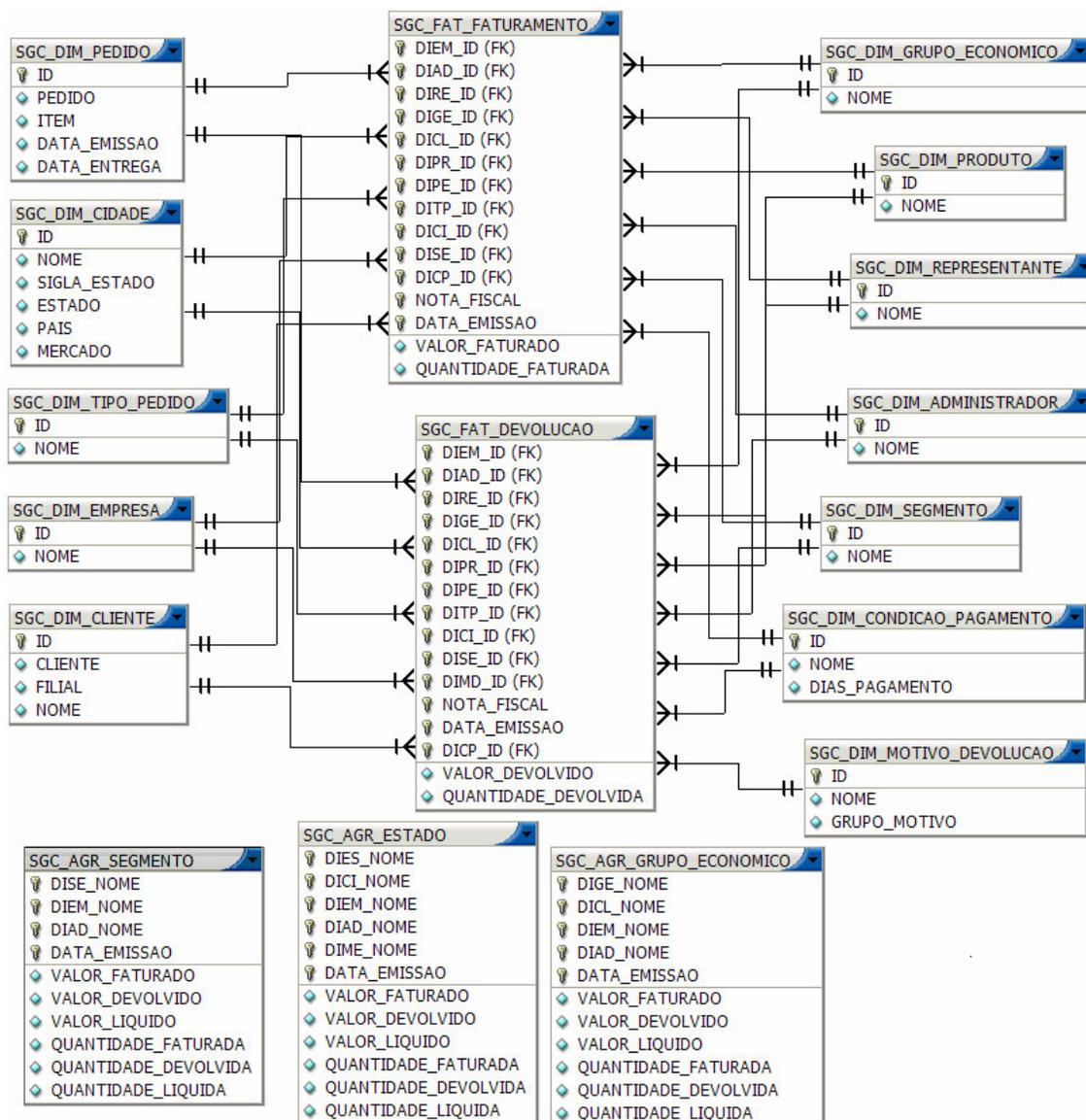


Figura 4.2 Modelagem do Sistema

Um detalhe a ser observado na modelagem é de que não há uma dimensão com atributos da data de emissão de uma nota fiscal, como por exemplo, mês, ano, semana ou demais dados relacionados a mesma. Essa dimensão não foi necessária, pois esses atributos de uma data podem ser gerados através dos metadados do *Discoverer*. Maiores detalhes serão explicados posteriormente.

#### 4.2.1 Granularidade

A granularidade foi um fator bastante discutido na análise de requisitos, em conjunto com os usuários que utilizariam o sistema. Como havia a necessidade de visualizar os níveis

mais baixos da hierarquia de dados, ou seja, visualizar qual a nota fiscal faturada ou devolvida, foi aplicada uma granularidade baixa.

Dessa forma, a preocupação com relação ao desempenho do sistema fez com que fosse definido que os dados a serem consultados ficariam restritos às notas fiscais emitidas ou devolvidas nos últimos três anos, ou seja, o ano atual e os dois anteriores. Caso fosse necessário visualizar dados mais antigos, seriam criadas consultas com níveis menores de detalhamento.

Assim, a tabela de faturamento, que apresenta maior volume do que a de devolução possui uma média de 130.000 registros, e testes de desempenho mostraram que a performance das consultas e nem mesmo o processo de carga de dados foram afetados por esse volume de dados.

Para que se compreenda melhor de que maneira esse detalhamento aplicado ao sistema funciona, a figura 4.3 representa graficamente a hierarquia principal utilizada através das consultas disponibilizadas ao usuário.

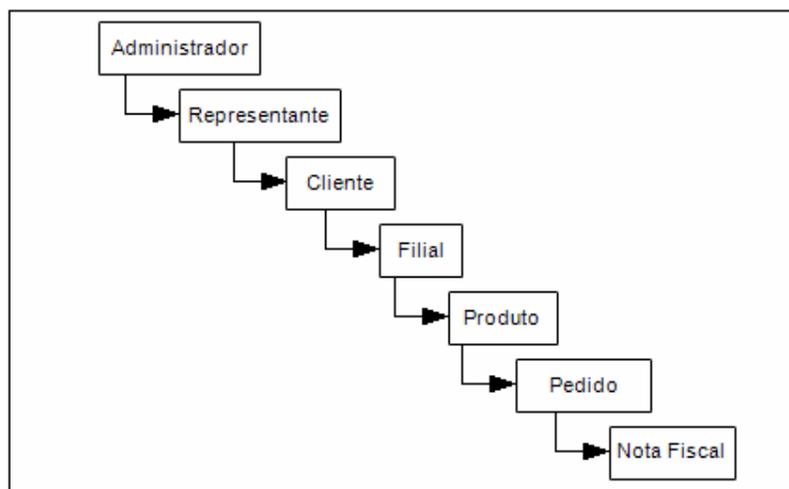


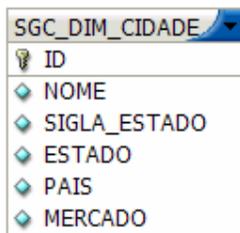
Figura 4.3 Hierarquia principal do sistema

#### 4.2.2 Dimensões e Fatos

Como pode ser visualizado na figura 4.2, foram desenvolvidas duas tabelas de fatos e doze de dimensão, que terão suas características principais apresentadas a seguir.

#### 4.2.2.1 Dimensões

Como já explicado, as dimensões são diretamente ligadas às tabelas de fatos através de suas chaves primárias, formando assim o *star-schema* da modelagem dimensional. Para que seja explicado de que forma está composta cada dimensão, será utilizada como exemplo a dimensão SGC\_DIM\_CIDADE, na qual estão armazenadas as informações da cidade em que houve o faturamento ou de onde veio a devolução, observada na figura 4.4.



SGC_DIM_CIDADE
ID
NOME
SIGLA_ESTADO
ESTADO
PAIS
MERCADO

Figura 4.4 – Dimensão SGC\_DIM\_CIDADE

Como pode ser observado, essa e as demais dimensões, além dos outros objetos do banco de dados, possuem como prefixo em sua nomenclatura a sigla “SGC”. Essa sigla quer dizer “Sistema Gerencial Comercial”, ou seja, representa que todos os objetos com esse prefixo fazem parte desse módulo do BI. Além do “SGC”, há a sigla “DIM”, que representa ser uma tabela de dimensão, assim como as tabelas de fato possuem a sigla “FAT” na sua nomenclatura.

Cada dimensão criada possui como chave primária um campo único, como recomenda Kimball (1998). A chave é o campo “ID”, que nesse caso, é de formato numérico, e possui os mesmos valores utilizados no sistema transacional. Além do ID, cada dimensão possui uma série de atributos. No caso da SGC\_DIM\_CIDADE, informações como o nome da cidade (NOME), sigla do estado (SIGLA\_ESTADO), ESTADO e o país da cidade (PAIS) estão presentes. Além dessas, há uma informação que é o mercado, no qual, se a cidade for de fora do país, o valor gravado é “Mercado Externo”, se for uma cidade brasileira, é gravado “Mercado Interno”.

#### 4.2.2.2 Fato SGC\_FAT\_FATURAMENTO

A tabela de fatos SGC\_FAT\_FATURAMENTO possui informações relacionadas às notas fiscais de faturamento e pode ser graficamente visualizada através da figura 4.5.

Column Name	Type
DIEM_ID	(FK)
DIAD_ID	(FK)
DIRE_ID	(FK)
DIGE_ID	(FK)
DICL_ID	(FK)
DIPR_ID	(FK)
DIPE_ID	(FK)
DITP_ID	(FK)
DICI_ID	(FK)
DISE_ID	(FK)
DICP_ID	(FK)
NOTA_FISCAL	
DATA_EMISSAO	
VALOR_FATURADO	
QUANTIDADE_FATURADA	

Figura 4.5 – Tabelas de fatos de faturamento

Como se pode observar, a tabela possui onze FKs (chaves-estrangeiras), que são as chaves primárias das dimensões. Dessa maneira, a chave primária é composta por elas e as colunas de `NOTA_FISCAL` e `DATA_EMISSAO`. Essas colunas estão incorporadas diretamente na tabela de fatos e não em uma outra tabela de dimensão. Isso ocorre pois se fosse criada uma tabela de dimensão para armazenar os atributos da nota fiscal, que é o caso dessas duas colunas, o sistema teria uma tabela a mais com o mesmo volume da `SGC_FAT_FATURAMENTO`, com o mesmo número de registros, que seria extremamente redundante, logo, desnecessário.

Além da chave, a tabela possui ainda duas medidas, `VALOR_FATURADO` e `QUANTIDADE_FATURADA`. As colunas relacionadas às chaves-estrangeiras são as seguintes:

`DIEM_ID` – Id da tabela `SGC_DIM_EMPRESA`

`DIAD_ID` – Id da tabela `SGC_DIM_ADMINISTRADOR`

`DIRE_ID` – Id da tabela `SGC_DIM_REPRESENTANTE`

`DIGE_ID` – Id da tabela `SGC_DIM_GRUPO_ECONOMICO`

`DICL_ID` – Id da tabela `SGC_DIM_CLIENTE`

`DIPR_ID` – Id da tabela `SGC_DIM_PRODUTO`

`DIPE_ID` – Id da tabela `SGC_DIM_PEDIDO`

DITP\_ID – Id da tabela SGC\_DIM\_TIPO\_PEDIDO

DICI\_ID – Id da tabela SGC\_DIM\_CIDADE

DISE\_ID – Id da tabela SGC\_DIM\_SEGMENTO

DICP\_ID – Id da tabela SGC\_CONDICAO\_PAGAMENTO

#### 4.2.2.3 Fato SGC\_FAT\_DEVOLUCAO

Essa tabela de fatos possui as informações das notas fiscais de devolução e pode ser graficamente visualizada através da figura 4.6.

SGC_FAT_DEVOLUCAO	
DIEM_ID (FK)	
DIAD_ID (FK)	
DIRE_ID (FK)	
DIGE_ID (FK)	
DICI_ID (FK)	
DIPR_ID (FK)	
DIPE_ID (FK)	
DITP_ID (FK)	
DICI_ID (FK)	
DISE_ID (FK)	
DIMD_ID (FK)	
DICP_ID (FK)	
NOTA_FISCAL	
DATA_EMISSAO	
VALOR_DEVOLVIDO	
QUANTIDADE_DEVOLVIDA	

Figura 4.6 – Tabelas de fatos de devolução

Essa tabela possui as mesmas características da tabela SGC\_FAT\_FATURAMENTO, ou seja, a composição de sua chave primária é através das chaves das dimensões e das colunas de NOTA\_FISCAL e DATA\_EMISSAO. As dimensões que a compõem também são as mesmas, porém com uma coluna a mais, que é a DIMD\_ID, id da tabela SGC\_DIM\_MOTIVO\_DEVOLUCAO. Estão presentes também as duas colunas de medidas, VALOR\_DEVOLVIDO E QUANTIDADE\_DEVOLVIDA.

#### 4.2.3 Agregados

O uso de agregados foi necessário em razão de que o *layouts* solicitados pelo diretor comercial para três relatórios não poderiam ser desenvolvidos através do *Discoverer*

utilizando as tabelas de fatos e dimensões diretamente. Para que se compreenda melhor a necessidade de *layout*, a figura 4.7 exibe o relatório “Faturamento por Segmento”, que foi um dos relatórios que tem por base um agregado de dados:

Faturamento Trimestral por Segmento										
Empresa <Tudo> Administrador <Tudo>										
	1o Trimestre		2o Trimestre		3o Trimestre		4o Trimestre		Vlr Líq	%
	Vlr Líq	%								
01-Alimentício		7		7		7		7		28
2007	1	-50%	1	-50%	1	-50%	1	-50%	4	-50%
2006	2	-50%	2	-50%	2	-50%	2	-50%	8	-50%
2005	4		4		4		4		16	
02-Calçado		7		7		7		7		28
2007	1	-50%	1	-50%	1	-50%	1	-50%	4	-50%
2006	2	-50%	2	-50%	2	-50%	2	-50%	8	-50%
2005	4		4		4		4		16	
03- Eletro-Eletrônico		7		7		7		7		28
2007	1	-50%	1	-50%	1	-50%	1	-50%	4	-50%
2006	2	-50%	2	-50%	2	-50%	2	-50%	8	-50%
2005	4		4		4		4		16	
04-Químico Farmacêutico		7		7		7		7		28
2007	1	-50%	1	-50%	1	-50%	1	-50%	4	-50%
2006	2	-50%	2	-50%	2	-50%	2	-50%	8	-50%
2005	4		4		4		4		16	
05-Metal-Mecânico		7		7		7		7		28
2007	1	-50%	1	-50%	1	-50%	1	-50%	4	-50%
2006	2	-50%	2	-50%	2	-50%	2	-50%	8	-50%
2005	4		4		4		4		16	
<b>TOTAL</b>		35		35		35		35		140
2007	5	-50%	5	-50%	5	-50%	5	-50%	20	-50%
2006	10	-50%	10	-50%	10	-50%	10	-50%	40	-50%
2005	20		20		20		20		80	

Figura 4.7 – Relatório “Faturamento por Segmento”

Como se pode observar, esse relatório faz um cruzamento de meses com segmentos, sendo que cada segmento está dividido entre os anos de 2005, 2006 e 2007. O motivo pelo qual um agregado fosse desenvolvido foi a coluna TOTAL, posicionada na parte inferior da consulta. A operação de totalizar os dados do *Discoverer* faria apenas com que uma única linha, acumulando todos os valores, fosse exibida na parte inferior do relatório. Porém a necessidade era exatamente como a figura 4.6 mostra no destaque, uma linha chamada TOTAL dividida entre os três anos apresentados.

Para que isso fosse possível, foi criado um agregado de dados chamado SGC\_AGR\_SEGMENTO, em destaque na figura 4.8. Os dados carregados nessa tabela são provenientes das tabelas de fatos. O diferencial com relação às tabelas de fatos é que além das informações vindas delas, o agregado possui registros gerados através da carga, que são os somatórios das medições. Nesses registros, a coluna DISE\_NOME tem o valor “TOTAL” e

os demais atributos mantêm suas descrições, o que faz com que, por exemplo, haja um total para cada administrador de venda (coluna DIAD\_NOME) ou data de emissão. Assim, a descrição “TOTAL” existente no relatório da figura 4.7 não é um totalizador, e sim registros da tabela que possuem na descrição do segmento o valor “TOTAL”.

SGC_AGR_SEGMENTO	
🔑	DISE_NOME
🔑	DIEM_NOME
🔑	DIAD_NOME
🔑	DATA_EMISSAO
◆	VALOR_FATURADO
◆	VALOR_DEVOLVIDO
◆	VALOR_LIQUIDO
◆	QUANTIDADE_FATURADA
◆	QUANTIDADE_DEVOLVIDA
◆	QUANTIDADE_LIQUIDA

Figura 4.8 – Agregado SGC\_AGR\_SEGMENTO

Portanto, os agregados criados foram SGC\_AGR\_SEGMENTO, SGC\_AGR\_ESTADO E SGC\_AGR\_CLIENTE. Para isto, foram criadas tabelas separadas da modelagem dimensional, onde cada registro do agregado possui colunas de métricas, com valores e quantidades faturadas e devolvidas, além das informações de forma líquida, ou seja, a métrica de faturamento descontada a métrica de devolução. A seguir cada um dos agregados será detalhadamente explicado.

#### 4.2.3.1 Agregado SGC\_AGR\_SEGMENTO

Como já explicado, esse agregado foi criado para contemplar os relatórios de Faturamento por Segmento, utilizando-se dos dados das tabelas de fatos SGC\_FAT\_FATURAMENTO E SGC\_FAT\_DEVOLUCAO, além das dimensões SGC\_DIM\_EMPRESA, SGC\_DIM\_ADMINISTRADOR e SGC\_DIM\_SEGMENTO e apresenta, além das colunas métricas, as seguintes:

DISE\_NOME – Descrição do segmento;

DIEM\_NOME – Descrição da empresa;

DIAD\_NOME – Descrição do administrador de venda;

DATA\_EMISSAO – Data de Emissão da nota fiscal.

#### **4.2.3.2 Agregado SGC\_AGR\_ESTADO**

Esse agregado foi criado para contemplar os relatórios de Faturamento por Estado, utilizando-se dos dados das tabelas de fatos SGC\_FAT\_FATURAMENTO E SGC\_FAT\_DEVOLUCAO, além das dimensões SGC\_DIM\_EMPRESA, SGC\_DIM\_ADMINISTRADOR e SGC\_DIM\_ESTADO e apresenta, além das colunas métricas, as seguintes:

DIES\_NOME – Descrição do estado;

DICI\_NOME – Descrição da cidade;

DIEM\_NOME – Descrição da empresa;

DIAD\_NOME – Descrição do administrador de venda;

DIME\_NOME – Descrição do mercado;

DATA\_EMISSAO – Data de emissão da nota fiscal.

#### **4.2.3.3 Agregado SGC\_AGR\_GRUPO\_ECONOMICO**

Agregado criado para as consultas de Faturamento por Grupo Econômico e Clientes, utilizando além das tabelas de fatos SGC\_FAT\_FATURAMENTO e SGC\_FAT\_DEVOLUCAO, as dimensões SGC\_DIM\_EMPRESA, SGC\_DIM\_ADMINISTRADOR, SGC\_DIM\_CLIENTE e SGC\_DIM\_GRUPO\_ECONOMICO. Apresenta, além das métricas, as seguintes colunas:

DIGE\_NOME – Descrição do grupo econômico

DICI\_NOME – Descrição do cliente;

DIEM\_NOME – Descrição da empresa;

DIAD\_NOME – Descrição do administrador de venda;

DATA\_EMISSAO – Data de Emissão da nota fiscal.

### 4.3 Carga de dados

O processo de carga do DM foi desenvolvido utilizando a linguagem de programação nativa dos bancos de dados *Oracle*, o PL/SQL<sup>3</sup>. Através dela foram desenvolvidas *stored procedures*<sup>4</sup> que executam todo o processo de ETL.

O processo de carga de dados é executado uma vez ao dia, através de um *job*<sup>5</sup>, no horário da noite e possui a ação de limpar toda a base de dados e inserir os dados novamente. Essa medida de limpar e carregar toda a base foi tomada por ser considerado que, dessa maneira, o processo não exigiria o trabalho de fazer comparações das tabelas do DM com as tabelas do sistema transacional, verificando o que deve ser inserido, atualizado ou não, que acarretaria um processamento e tempo muito maior.

Outra medida tomada para uma boa performance, foi a utilização da opção *NOLOGGING* na criação de todas as tabelas e índices<sup>6</sup>. Através dessa opção, as transações executadas nesses objetos não serão registradas nos arquivos de *redo log*<sup>7</sup> (ORACLE, 2000 c).

Ainda com relação ao ganho de performance, antes de iniciar a exclusão e inserção dos dados, todos os índices são apagados e todas as *primary keys (PK)* e *foreign keys (FK)* desabilitadas. Isso permite que a exclusão dos dados seja executada através do comando *TRUNCATE TABLE*, que limpa toda a tabela de forma imediata, processando de maneira mais rápida que o convencional comando *DELETE*, pois executa sem opção de *rollback*<sup>8</sup> e sem atualização dos índices da tabela.

Com os índices, PKs e FKs das tabelas a serem carregadas desabilitados, a inserção dos dados torna-se muito mais rápida, uma vez que não há a necessidade de atualizar os

---

<sup>3</sup> PL/SQL é a extensão de linguagem procedural do Oracle para SQL. PL/SQL permite combinar instruções SQL e construções procedurais (ORACLE, 2000c)

<sup>4</sup> *Stored procedures* ou procedimentos armazenados são procedimentos criados através do dicionário de dados do BD formado por construções em PL/SQL ou SQL (ORACLE, 2000c).

<sup>5</sup> JOB é uma tarefa que é agendada para executar de período em período automaticamente no banco de dados (ORACLE, 2002).

<sup>6</sup> Índices são objetos que podem acelerar a recuperação de linhas usando um ponteiro. Seu objetivo é reduzir a necessidade de E/S do disco, usando um caminho indexado para localizar dados rapidamente (ORACLE, 2000b).

<sup>7</sup> Arquivos de *Redo logs* são utilizados para armazenar registros das alterações efetuadas no banco de dados, a fim de ativar a recuperação dos dados se houver falhas (ORACLE, 2000c).

<sup>8</sup> Um segmento de *rollback* é uma estrutura utilizada para salvar o antigo valor quando um processo altera os dados de um banco de dados. Ele armazena a localização dos dados e os dados na forma em que existiam antes da modificação (ORACLE, 2000c).

índices cada vez que um registro é inserido. O processo de recriar os índices e habilitar as *constraints* é realizado ao final da carga de cada uma das tabelas de dimensões e de fatos.

A *stored procedure* de carga dos dados é chamada de SGC\_PRC\_CARGA\_DADOS, que pode ser graficamente representada através da figura 4.9:

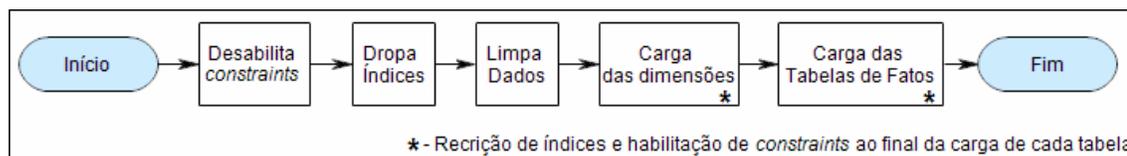


Figura 4.9 – *Stored procedure* de carga dos dados

Quando os processos começaram a ser construídos através de *stored procedures*, o tempo médio de execução da carga de dados era de mais de uma hora de duração. Com o uso da opção *NOLOGGING* e da implementação das rotinas de desabilitar *constraints* e apagar índices, unidas com o método de exclusão *TRUNCATE TABLE*, o processo passou a ter uma duração média de 20 minutos.

#### 4.4 Uso do *Oracle Warehouse Builder*

Quando a análise de requisitos foi realizada, o *Oracle Warehouse Builder* foi definido como ferramenta a ser usada para o processo de ETL e modelagem do sistema. Essa definição partiu da premissa que se tratava da ferramenta *da Oracle* que tem especificamente a função de administrar um repositório de dados, possibilitando então realizar as tarefas de ETL e modelagem.

O OWB chegou a ser utilizado durante um período de duas semanas, para que fosse possível aprender como utilizá-lo e para realmente comprovar se a sua utilização traria ganhos efetivos ao projeto.

Com a ferramenta, pode-se criar algumas tabelas de dimensões e de fatos, e também alguns processos de mapeamento de informações, que inseriam nas tabelas criadas de acordo com a fonte de dados indicada. Entretanto, a ferramenta não foi utilizada e a causa disso deve-se a basicamente dois pontos: usabilidade e necessidade.

A usabilidade do OWB é de certo modo complexa, pois é uma ferramenta burocrática, que exige uma série de detalhes para criação ou modificação de qualquer objeto criado através dela. Para que o seu uso fosse eficaz e produtivo, a necessidade de um

conhecimento apurado da ferramenta, como por exemplo, através de um curso seria necessário. Para isso, deveria haver uma janela de tempo muito maior para o desenvolvimento do projeto, o que é impossível, por se tratar de um projeto de curto a médio prazo.

Ainda quanto à usabilidade, uma das dificuldades que se teria com o uso da ferramenta seria com relação à edição dos processos de mapeamento das tabelas. O OWB cria tais processos através de *stored procedures* com códigos PL/SQL extremamente complexos, o que acarretaria muita dificuldade quando surgisse a necessidade de se fazer alterações para contemplar as exceções que normalmente surgem no decorrer do desenvolvimento do sistema. Além disso, os processos de mapeamento criados pelo OWB não apresentaram desempenho melhor do que os utilizados atualmente na carga de dados. A carga da dimensão SGC\_DIM\_ADMINISTRADOR foi testada com o OWB, e levava mais de três minutos para ser finalizada. Já com as *stored procedures* o tempo resume-se a poucos segundos.

Com relação à necessidade, o fato é que empresa trabalha unicamente com base de dados *Oracle* que, como já colocado anteriormente, é a tecnologia homologada e aprovada, que dificilmente será substituída por outra. Dessa maneira, o uso de uma ferramenta do porte do OWB torna-se desnecessário, por se tratar de um aplicativo que tem como principal característica a integração de fontes de dados diversas, algo não-presente nesse projeto. Dessa maneira, todos os objetos de banco de dados componentes do sistema podem ser administrados por qualquer ferramenta de gerenciamento de banco de dados, como o *SQL Developer* da *Oracle* ou o *Toad*, da *Quest*<sup>9</sup>.

#### 4.5 Metadados

Através do *Discoverer Administrator* foi desenvolvida a camada lógica de metadados do sistema. Para organizar e personalizar o DM foi criada uma *Business Area*, chamada “Sistema Gerencial Comercial”. Nela, seis pastas foram criadas, cada uma delas referenciando uma *view* do banco de dados.

Todos os metadados são armazenados em tabelas do banco de dados próprias do *Discoverer*. Nelas estão armazenados dados de todas as hierarquias, listas de itens, permissões de acessos e pastas criadas. Além disso, todas as consultas criadas através do *Discoverer Plus*

---

<sup>9</sup> Toad – Disponível em [www.toadsoft.com](http://www.toadsoft.com).

ou *Desktop* são armazenadas nessas tabelas, salvo aquelas que foram criadas em arquivo, que assim não ficam no banco de dados, mas sim em arquivos do tipo DIS, nativos do *Discoverer*.

#### 4.5.1 Metadados criados

Foram desenvolvidas seis pastas baseadas em *views* do banco de dados para contemplar as consultas necessárias do sistema. A pasta “Fato Faturamento” referencia a *view* que possui dados de faturamento, ou seja, as medidas, número da nota fiscal, e data de emissão da tabela de fatos SGC\_FAT\_FATURAMENTO, além das descrições dos dados dimensionais, através das dimensões que fazem parte dessa tabela. A pasta “Fato Devolução” tem a mesma construção da anterior, porém busca dados de devolução, através da tabela de fatos SGC\_FAT\_DEVOLUCAO.

A pasta “Fato Faturamento x Devolução”, em destaque na figura 4.10, é baseada em uma *view* que cruza os dados de faturamento e devolução, possibilitando uma análise de faturamento líquido, ou seja, diminuindo do valor ou quantidade faturada o que foi devolvido.

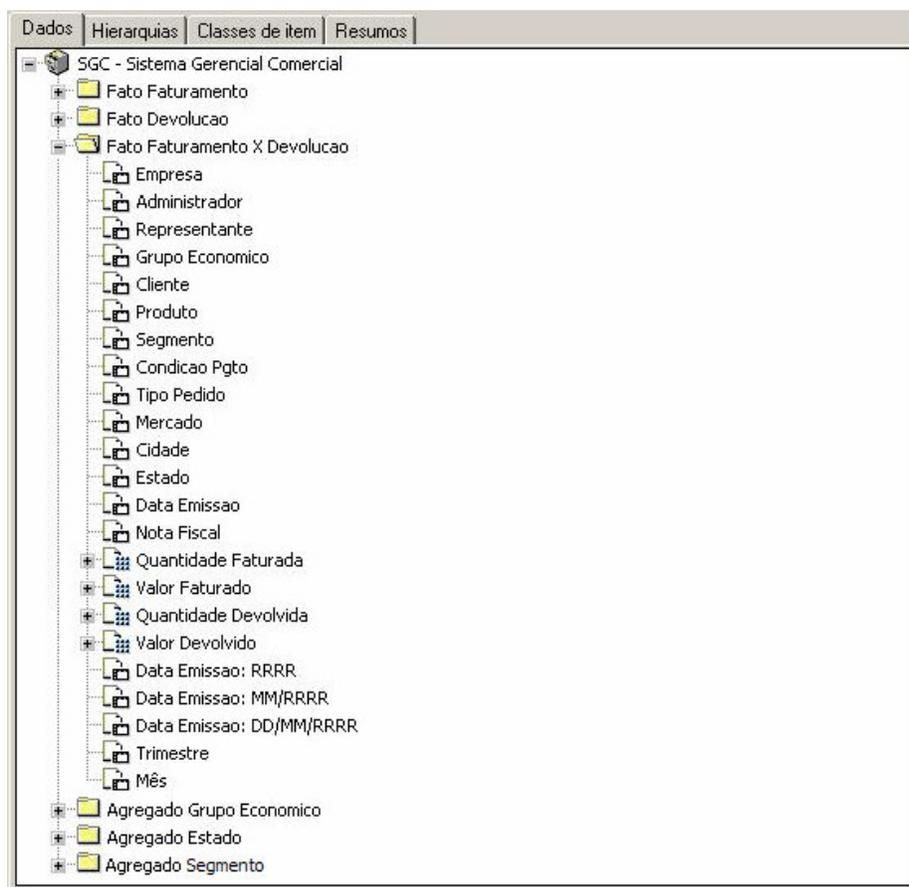


Figura 4.10 – Pastas do *Discoverer*

As outras três pastas existentes na *Business Area* são relacionadas aos agregados de dados, já explicados anteriormente.

Como já comentado, não há a existência de uma dimensão que possui os dados das datas, ou seja, informações como dia, semana, mês e ano da emissão de uma nota fiscal. Isso não foi necessário, pois, sempre que uma nova pasta do *Discoverer* é criada, há uma opção em que todas as colunas de datas possam ser hierarquizadas.

Dessa maneira, além da coluna “data de emissão”, o *Discoverer* gera por padrão novas colunas com máscaras diferentes. Uma coluna para ano, com o formato “aaaa”, uma para mês, no formato “mm/aaaa”, e outra sendo o próprio dia, no formato “dd/mm/aaaa”. É possível que o usuário personalize essas hierarquias, alterando os formatos de máscara ou incluindo novos níveis.

No caso desse sistema, foi desenvolvida uma hierarquia que possui além de ano, mês e dia, a informação do trimestre em que a data está presente, escrito de maneira literal. Assim, a data de emissão pode ser visualizada, além da hierarquia padrão, conforme a figura 4.11:

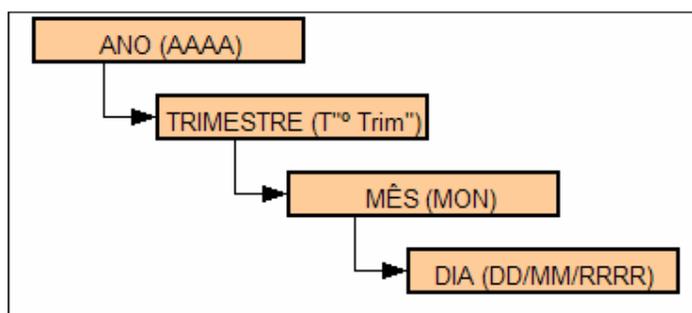


Figura 4.11 - Hierarquia da data de emissão das notas fiscais

Além desta hierarquia, foi desenvolvida a principal do sistema, já explicada anteriormente. Foram criadas ainda mais duas. A primeira delas permite ao usuário visualizar as informações em nível de estado e fazer o *drill down* visualizando por nível de cidade. A segunda está relacionada à consulta de devoluções. Ao visualizar as informações por motivos de devolução como, por exemplo, “problemas de cadastro”, é possível que o usuário saiba mais especificamente qual o motivo, abrindo então para “problemas de cadastro de preço” ou “cliente de faturamento informado incorretamente”.

No capítulo a seguir, será explicado como o sistema BOX BI foi disponibilizado ao usuário, mostrando sua disposição gráfica e de que modo foi administrada a questão de usuários e permissões de acessos de cada um.

## 5 BOX BI

Como já explicado, o sistema em questão é chamado de BOX BI, nome dado através da união dos principais conceitos que o envolvem, *Business Intelligence*, e a palavra “Box”, que populariza a empresa Box Print. O BOX BI foi disponibilizado essencialmente para pessoas envolvidas no setor comercial da empresa: diretor comercial, gerentes comerciais das unidades da empresa e administradores de venda. Além destes, a diretoria administrativa também possui acesso ao sistema.

O sistema é acessado através do navegador de internet, de duas formas distintas. A primeira digitando o endereço <http://portal.boxprint.ind.br/bi>; a segunda, através de um *link* disponibilizado na *intranet* da empresa, que estará visível apenas para os usuários que podem utilizar o BOX BI.

### 5.1 Tela Inicial

Ao entrar no sistema, os usuários possuem uma tela inicial, que traz de forma resumida, as principais informações do BOX BI, através de tabelas ou gráficos. Ela está dividida em três blocos: geral, equipe e estatísticas. No “geral”, está presente o faturamento geral do ano corrente, sem nenhuma parametrização. No bloco “equipe”, estão exibidas informações sobre a equipe de cada gerente ou administrador de venda. E por último, no bloco “estatísticas”, são exibidos alguns gráficos relacionados a faturamento ou devoluções de forma mais detalhada. A figura 5.1 detalha a tela inicial.

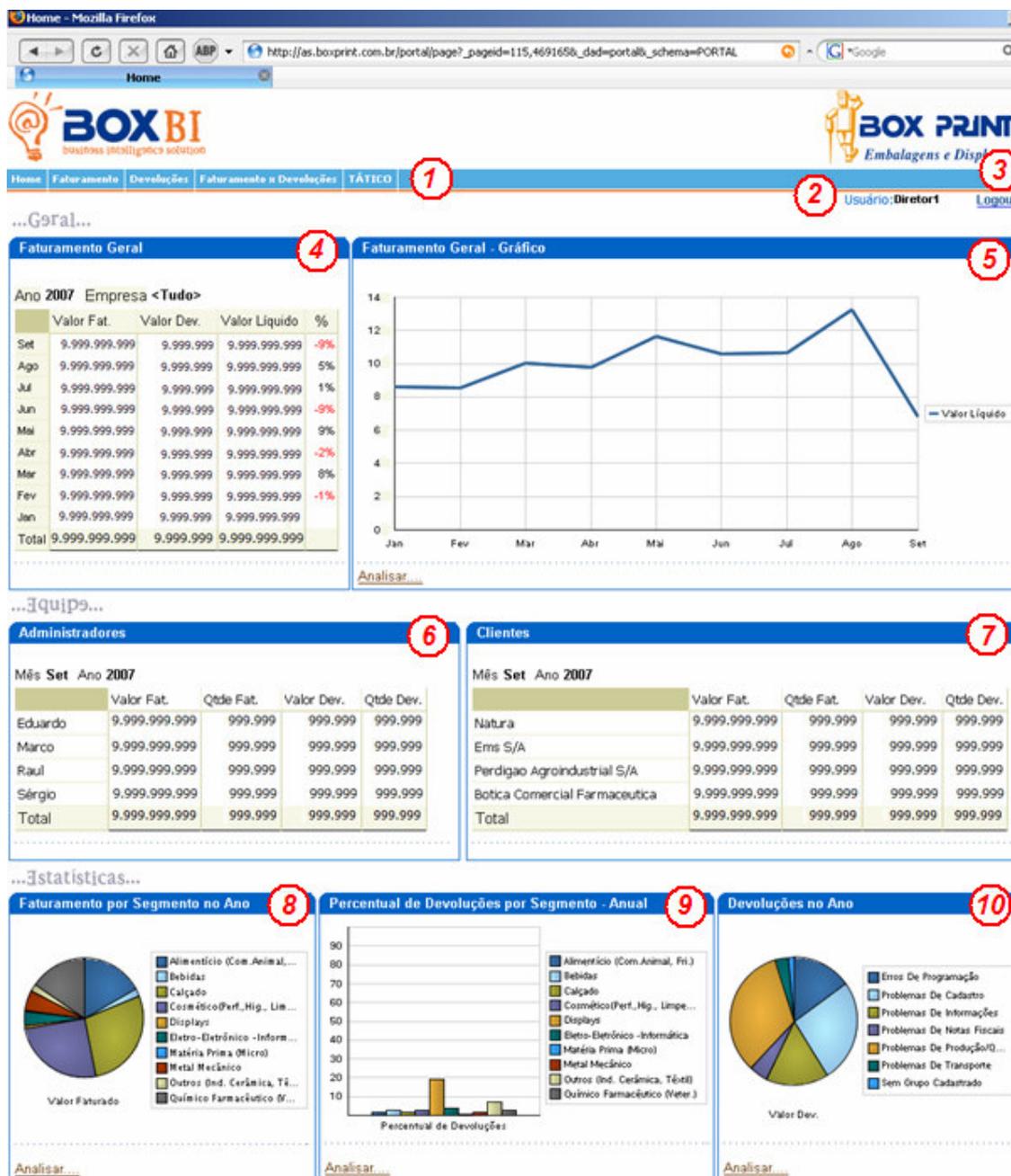


Figura 5.1 – Tela inicial do sistema

Os seguintes itens podem ser visualizados na tela inicial:

1. **Menu principal:** exhibe as telas disponíveis para cada usuário. Será explicado com mais detalhamento posteriormente;
2. **Nome do usuário conectado;**
3. **Link de logout:** para desconectar do sistema;

4. **Faturamento geral:** mostra o faturamento no decorrer do ano atual agrupado por meses, com um totalizador e um percentual comparando quanto o mês cresceu ou decresceu com relação ao mês anterior, de acordo com o valor líquido;

5. **Faturamento geral – gráfico:** representa graficamente o valor líquido através dos meses do ano atual;

6. **Administradores:** exibido somente para o usuário da diretoria, no qual exibe, do mês corrente as informações de faturamento dos administradores de venda. Quando os usuários conectados são os próprios administradores de venda, as informações são exibidas no mesmo formato, porém com as informações dos quatro representantes da sua competência com maior faturamento no período.

7. **Clientes:** exibe informações do mês corrente dos quatro clientes com maior faturamento no período;

8. **Faturamento por segmento no ano:** exibe em forma de gráfico, o faturamento líquido no ano;

9. **Percentual de devoluções por segmento – anual:** exibe o percentual que representam as devoluções em cada um dos segmentos. O período é o ano atual.

10. **Devoluções no ano:** exibe o total de devoluções no ano, dividido pelos motivos de devolução.

### 5.1.1 Menu principal

O menu principal possui todas as telas disponíveis e está dividido em faturamento, devoluções, faturamento x devoluções e tático.

No faturamento, estão disponíveis as telas que possuem as informações do faturamento bruto da empresa, e este pode ser visualizado da seguinte forma: por empresa; por administrador; por representante; por cliente; por produto; por segmento; por estado; por tipo de pedido; por condição de pagamento.

No item devolução, as informações de devolução estão disponíveis da seguinte forma: por empresa; por motivo de devolução; por administrador; por representante; por cliente; por produto; por segmento; por estado; por tipo de pedido.

No item faturamento x devoluções são exibidas informações do faturamento líquido da empresa, além de telas com a informação de faturamento e devoluções de forma discriminada. Essas informações podem ser vistas da seguinte forma: por empresa; por administrador; por representante; por cliente; por produto; por segmento; por estado; por tipo de pedido; por condição de pagamento.

O item “tático” está disponível apenas para a diretoria, nele estão os relatórios que foram criados através dos agregados, já explicados anteriormente. As telas disponíveis são de faturamento líquido por cliente, por segmento e por estado.

A seguir será explicado de forma detalhada como são as telas disponíveis e de que forma podem ser analisadas.

## **5.2 Telas do sistema**

Conforme já explicado, as consultas podem ser acessadas através do menu principal do sistema. Cada uma delas tem como padrão o formato de tabela de referência cruzada, e podem ser graficamente representadas pela figura 5.2, que exibe a tela “Faturamento Líquido por Empresa (Mensal/Valor)”, que, a partir de agora, servirá como exemplo para explicar as características e funcionalidades das consultas disponíveis no sistema.

	Valor Faturado									
	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	TOTAL
Unidade Box Print	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	9.999.999,99
Unidade Bpg	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	9.999.999,99
Unidade Igrejinha	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	9.999.999,99
Unidade Microondulados	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	9.999.999,99
<b>MÉDIA</b>	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	
<b>TOTAL</b>	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99

Figura 5.2 – Tela “Faturamento Líquido por Empresa (Mensal/Valor)”

Como pode ser observado, a tela de consultas possui sete abas, onde cada uma delas representa uma maneira de representar a consulta. Ou seja, quando a consulta “Faturamento Líquido por Empresa (Mensal/Valor)” é acessada, existem sete formatos para o mesmo tipo de informação, que são eles:

**Mensal-Valor:** exibe os valores divididos em meses;

**Trimestral-Valor:** exibe os valores divididos em trimestres;

**Anual-Valor:** exibe os valores divididos em anos;

**Mensal-Quantidade:** exibe as quantidades divididas em meses;

**Trimestral - Quantidade:** exibe as quantidades divididas em trimestres;

**Anual - Quantidade:** exibe as quantidades divididas em anos;

**Detalhado:** Essa opção está disponível para o módulo “Faturamento x Devoluções” e exibe as informações de quantidade e valor de forma discriminada, dentro de um mês, como mostra a figura 5.3:



Figura 5.3 - Faturamento Líquido por Empresa (Detalhado)

Quanto ao módulo tático, os formatos possíveis são apenas os relacionados a valores, juntamente com o formato detalhado. Ainda há a opção “gráficos” que possui dois gráficos disponíveis: de desempenho e de participação. No exemplo da figura 5.4, ambos podem ser visualizados. O desempenho dos maiores clientes exibe um gráfico do tipo “linha” que mostra, mês a mês, como foi o faturamento líquido de cada um deles. O gráfico “participação dos maiores clientes” é no formato “pizza” e mostra qual o percentual que os maiores clientes representam no total do faturamento líquido. Ambos mostram os cinco maiores clientes da Box Print, dentro do ano atual, mas há a possibilidade do usuário parametrizar a consulta, alterando o número de clientes ou o período a ser visualizado, através do *link* “Analisar”, que será explicado posteriormente.

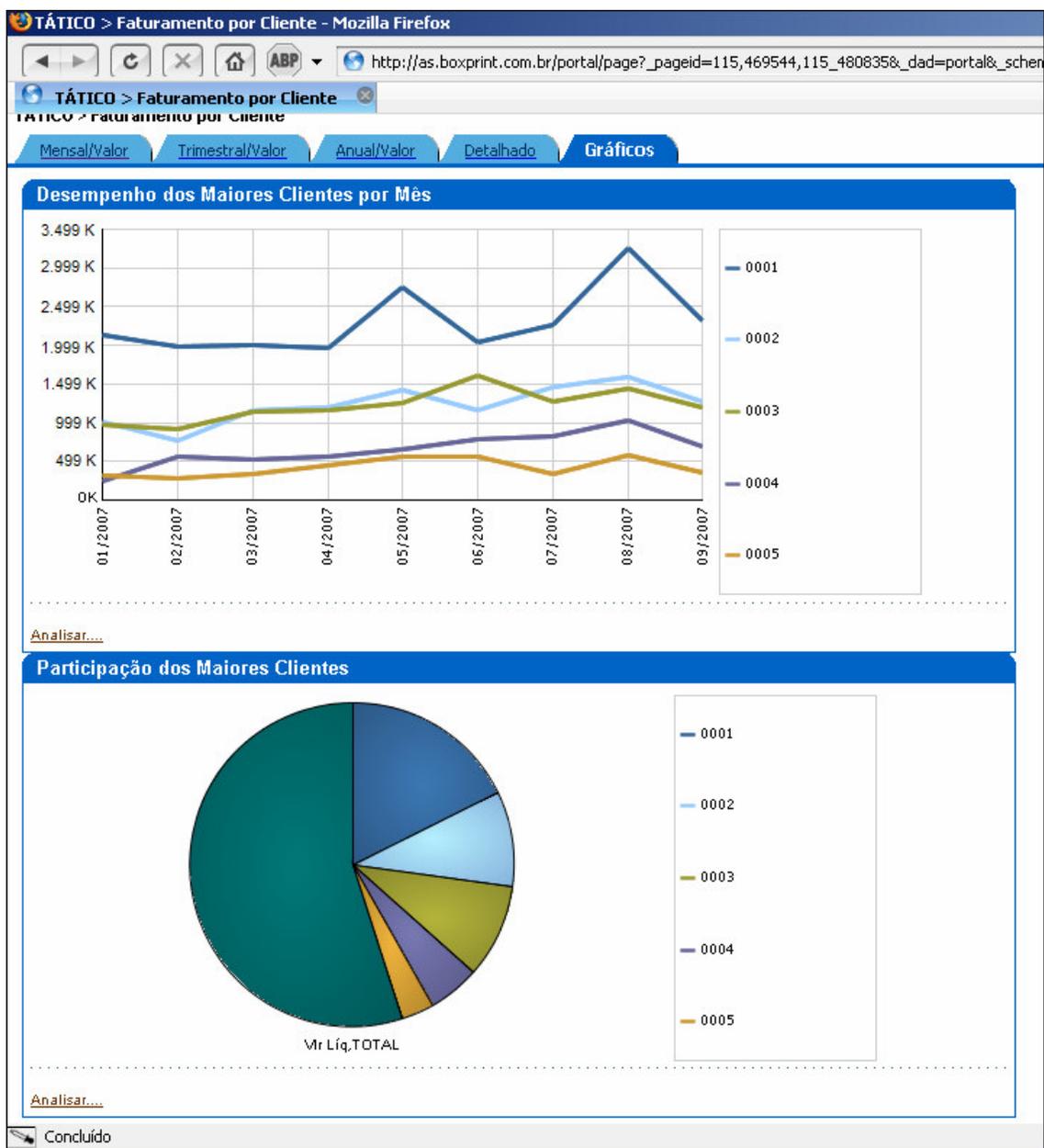


Figura 5.4 – Exemplo de gráficos do módulo tático

Cada uma dessas formas de visualização das consultas são *portlet*s criados através do *Oracle Portal*. No momento em que um *portlet* com consultas é exibido, não há busca de informações ao banco de dados. Todo o *portlet* possui um processo que atualiza de hora em hora as informações da consulta relacionada. Assim, quando a tela é exibida, o que é visualizado é uma tela estática, com informações já carregadas para o *portlet* anteriormente.

Para que as telas possam ser analisadas com as funcionalidades OLAP ou efetuar qualquer outro tipo de ação, é necessário clicar no *link* “Analisar”, que está disponível no

canto inferior esquerdo de cada um dos *portlets*. A seguir, todas as possibilidades de análise e demais ações serão explicadas.

### 5.2.1 Funcionalidades do sistema

Ao clicar no *link* “analisar” de uma consulta, uma nova janela do *browser* é aberta, e pode ser graficamente representada pela figura 5.5, que será explicada através dos indicadores colocados na mesma.

The screenshot displays the Oracle BI Discoverer interface for a report titled "Faturamento Líquido por Empresa (Valor-Mensal)". The interface includes a navigation menu on the left with options like "Gerar PDF", "Gerar Planilha Excel", and "Enviar como e-mail". The main area shows a filter section with dropdowns for "Ano" (2007), "Segmento" (<Tudo>), "Mercado" (<Tudo>), and "Tipo Pedido" (<Tudo>). Below the filters is a table with columns for months (Jan to Set) and a "TOTAL" column. The table rows include "Empresa", "Unidade Box Print", "Unidade Bpg", "Unidade Igrejinha", "Unidade Microcondulados", "MÉDIA", and "TOTAL". The values in the table are all "999.999,99".

Mês	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	TOTAL
Empresa										
Unidade Box Print	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	9.999.999,99
Unidade Bpg	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	9.999.999,99
Unidade Igrejinha	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	9.999.999,99
Unidade Microcondulados	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	9.999.999,99
MÉDIA	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	
TOTAL	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99

Figura 5.5 – Análise: “Faturamento Líquido por Empresa (Mensal-Valor)”

Assim os seguintes itens podem ser visualizados:

1. **Título da consulta;**
2. **Tabela de referência cruzada** com as informações da consulta;
3. **Salvar e retornar ao portal:** o usuário pode formatar a consulta da sua maneira e, caso utilize essa opção, toda a vez que retornar a consulta, ela estará da maneira que ele formatou;
4. **Cancelar e retornar ao portal:** fecha a tela de análise sem salvar;
5. **Menu de ações:** possui as opções de exportação, envio por e-mail e impressão das consultas. Serão explicados com maiores detalhes a seguir.

6. **Ferramentas:** possui as possibilidades de formatação da consulta. Serão explicadas com maiores detalhes a seguir.

7. **Itens de página:** parâmetros que permitem novas consultas em uma mesma tela sem uma nova requisição ao banco de dados (ORACLE, 2005b).

8. **Links de drill-down e roll-up:** ao serem clicados permitem executar essas operações OLAP. Serão explicadas com maior detalhamento a seguir.

### 5.2.1.1 Menu de Ações

O menu de ações possui as maneiras possíveis de o usuário compartilhar ou até mesmo editar a sua consulta. A figura 5.6 representa graficamente o menu.

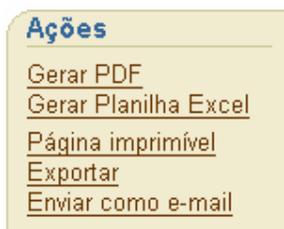


Figura 5.6 – Menu de ações

Como pode ser observado, existem possibilidades de exportar a consulta para os mais diversos formatos, imprimir e ainda enviar por *e-mail*. Há ainda *links* diretos para exportação dos formatos de arquivo PDF e planilhas Excel.

#### 5.2.1.1.1 Exportação para PDF e planilha Excel

Os dois *links* de acesso rápido para exportação exibidos no menu de ações foram inseridos através de modificações nos arquivos de configuração do servidor BI. Os arquivos de configuração estão no formato *UIX*<sup>1</sup>. Assim o arquivo que possui o *layout* da tela de análise de consultas teve o incremento dos *links* “Gerar PDF” e “Gerar Planilha Excel” (BI BLOG, 2006). Ao serem clicados, imediatamente o arquivo já é gerado e pronto para ser salvo, impresso, ou como o caso da planilha, ser editado pelo usuário.

---

<sup>1</sup> *UIX (User Interface XML)* é um conjunto de tecnologias que constituem um *framework* para o desenvolvimento de aplicações *web*, com foco principal na camada de apresentação ao usuário de uma aplicação (ORACLE *UIX*, 2007).

### 5.2.1.1.2 Imprimir

A impressão das consultas é feita através da geração de um arquivo PDF, porém, a diferença da impressão da simples exportação para o formato PDF está nas possibilidades da página a ser impressa receber diversas formatações. Ao clicar no *link* “Página imprimível” uma nova página é carregada onde ajustes como tamanho e orientação da página, tamanho das margens, possibilidades de imprimir ou não cabeçalhos e rodapés, ou até tamanho das colunas podem ser realizados.

### 5.2.1.1.3 Exportar

Ao clicar no *link* exportar uma nova página é aberta exibindo uma lista com todos os formatos possíveis de exportação, que podem ser visualizados na figura 5.7:

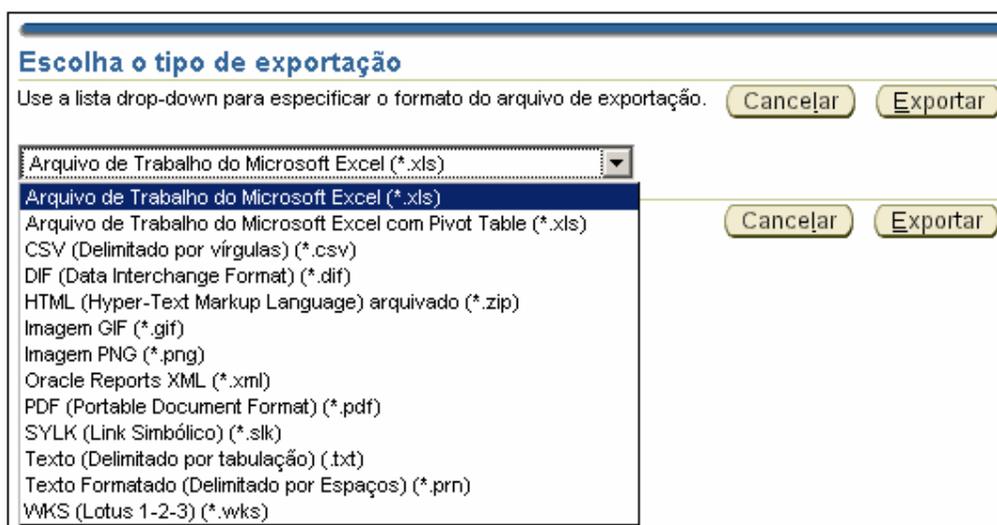


Figura 5.7 – Tela de exportação

Assim, ao selecionar o formato desejado, basta clicar no botão exportar que a mensagem “Exportação realizada com sucesso” será exibida, com um *link* para que a visualização do arquivo seja realizada.

### 5.2.1.1.4 Enviar como e-mail

Quando o *link* “enviar como e-mail” é clicado, uma nova página é exibida contendo o mesmo conteúdo da página de exportação, onde o formato do arquivo a ser enviado como anexo é solicitado. Ao selecionar o formato, uma próxima página é carregada, contendo os

campos de remetente, destinatário, CC, CCo, assunto e texto do e-mail, como mostra a figura 5.8:

Figura 5.8 – Tela para envio de *e-mail*

Dessa forma, não é necessário que o usuário tenha configurado em sua máquina nenhum tipo de programa de envio de *e-mails*, bastando apenas preencher os campos obrigatórios e clicar no botão “Finalizar”.

### 5.2.1.2 Ferramentas

Como já explicado, o usuário tem a possibilidade de formatar a sua consulta de diversas maneiras. Isso é possível através das ferramentas disponíveis a ele que são: *layout*, *formatos*, *stoplight* e *linhas e colunas*.

#### 5.2.1.2.1 *Layout*

Através dessa utilidade, o usuário tem a condição de utilizar a operação de *pivoting*. Como mostra a figura 5.9, é possível mover ou inverter linhas ou colunas, de acordo com a necessidade do usuário. Para isso, basta selecionar entre “mover” ou “trocar” no campo 1 da figura, indicando nos campos 2 e 3 quais as informações envolvidas na operação.

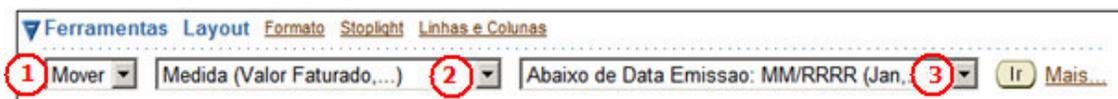


Figura 5.9 – Ferramenta de *Layout*

É possível ainda trocar a posição das linhas ou colunas utilizando o *link* “Mais ...”, que exhibe uma nova tela possibilitando que o usuário altere o layout, como mostra a figura 5.10:

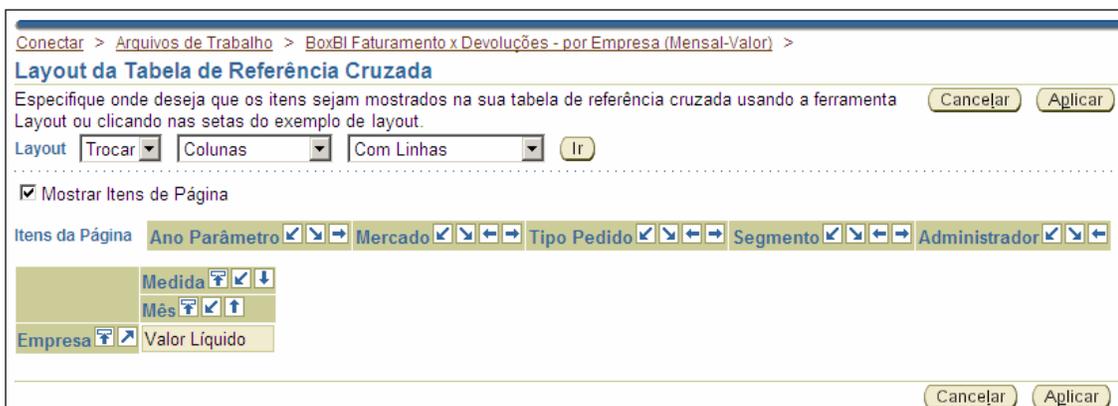


Figura 5.10 – Tela de edição de *Layout*

Tanto nessa tela, como na formatação rápida da tela de consulta, é possível passar qualquer um dos campos para ser um item de página, que desempenha a operação de *slice and dice*. Para que isso seja possível, basta clicar em uma das setas existentes em cada um dos campos, indicando para qual posição o campo deve ir.

No exemplo da figura 5.11, a coluna “Mercado”, que estava posicionada como um item de página, foi movida para ficar abaixo da linha “Empresa”, fazendo com que os valores que eram por empresa, passassem a ser divididos dentro de cada uma delas por mercado.

▼ Tabela de Referência Cruzada

▶ Ferramentas [Layout](#) [Formato](#) [Stoplight](#) [Linhas e Colunas](#)

Mover [Medida (Valor Faturado,...)] [Abaixo de Data Emissao: MM/RRRR (Jan,...)] Ir Mais...

Itens da Página Ano 2007 Segmento <Tudo> Administrador <Tudo> Tipo Pedido <Tudo>

		Valor Faturado									
▶ Mês		▶ Jan	▶ Fev	▶ Mar	▶ Abr	▶ Mai	▶ Jun	▶ Jul	▶ Ago	▶ Set	TOTAL
▶ Empresa	▶ Mercado										
	▶ Unidade Box Print	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	9.999.999,99
	▶ Mercado Externo	111.111,11	111.111,11	111.111,11	111.111,11	111.111,11	111.111,11	111.111,11	111.111,11	111.111,11	1.111.111,11
	▶ Mercado Interno	888.888,88	888.888,88	888.888,88	888.888,88	888.888,88	888.888,88	888.888,88	888.888,88	888.888,88	8.888.888,88
	▶ Unidade Bpg	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	9.999.999,99
	▶ Mercado Interno	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	9.999.999,99
	▶ Unidade Igrejainha	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	9.999.999,99
	▶ Mercado Interno	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	9.999.999,99
	▶ Unidade Microondulados	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	9.999.999,99
	▶ Mercado Externo	111.111,11	111.111,11	111.111,11	111.111,11	111.111,11	111.111,11	111.111,11	111.111,11	111.111,11	1.111.111,11
	▶ Mercado Interno	888.888,88	888.888,88	888.888,88	888.888,88	888.888,88	888.888,88	888.888,88	888.888,88	888.888,88	8.888.888,88
	MÉDIA	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	
	TOTAL	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99

Figura 5.11 – Exemplo de *Pivoting*

### 5.2.1.2.2 Formato

O item “formato” traz ao usuário a condição de formatar as células da consulta. Basta selecionar o campo desejado e clicar em uma das opções: negrito; itálico; sublinhado; fundo, que altera a cor do fundo da célula; fonte, para formatação da fonte; ou ainda criar formatos condicionais. O formato condicional serve para o usuário destacar informações dentro de uma regra, como por exemplo, destacar em vermelho os meses em que o valor faturado foi abaixo de 1 milhão. A figura 5.12 demonstra as opções de formato:

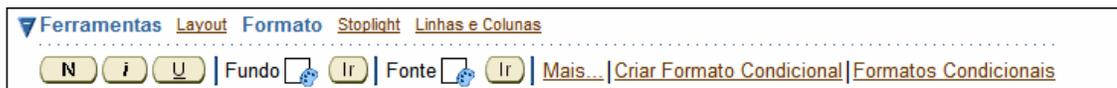


Figura 5.12 – Opções de formato

### 5.2.1.2.3 Stoplight

O *stoplight* é uma maneira mais rápida de se criar uma formatação condicional. Nele, as células são formatadas como se fosse um semáforo, ou seja, de acordo com o valor, as mesmas ficam em verde, amarelo ou vermelho. Os parâmetros que determinam as faixas para cada cor são informados manualmente pelo usuário.

Nesses campos, o usuário indica a partir de que valor para baixo as células ficam em vermelho, e para cima em verde. A figura 5.13 mostra um exemplo de formatação com *stoplight*.

▼ Tabela de Referência Cruzada

▼ Ferramentas [Layout](#) [Formato](#) [Stoplight](#) [Linhas e Colunas](#)

Formato  | Inaceitável  - Aceitável  - Desejável  Ir [Formatos Condicionais](#)

DICA A faixa aceitável está entre valores inaceitáveis e desejáveis.

Itens da Página Ano  Tipo Pedido

Segmento  Administrador

		Valor Faturado									
		Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	TOTAL
Empresa											
▶	Unidade Box Print	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	9.999.999,99
▶	Unidade Bpg	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	9.999.999,99
▶	Unidade Igrejinha	111.111,11	111.111,11	111.111,11	111.111,11	111.111,11	111.111,11	111.111,11	111.111,11	111.111,11	1.111.111,11
▶	Unidade Microondulados	888.888,88	888.888,88	888.888,88	888.888,88	888.888,88	888.888,88	888.888,88	888.888,88	888.888,88	8.888.888,88
MÉDIA		999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	
TOTAL		9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99	9.999.999,99

Figura 5.13 – Exemplo de *stoplight*

O usuário tem ainda a condição de alterar as cores do *stoplight*, não se limitando ao formato de semáforo.

#### 5.2.1.2.4 Linhas e Colunas

A opção de linhas e colunas faz com que o usuário possa visualizar um número determinado de linhas e de colunas. Os limites estão restritos de 1 a 999, como mostra a figura 5.14:

▼ Tabela de Referência Cruzada

▼ Ferramentas [Layout](#) [Formato](#) [Stoplight](#) [Linhas e Colunas](#)

Linhas  Colunas  Ir

(1 - 999) (1 - 999)

Figura 5.14 – Configuração de linhas e colunas

#### 5.2.1.2.5 Drill-down e Roll-up

As operações de *drill-down* e *roll-up* são realizadas através das setas existentes ao lado do título de cada uma das informações existentes na consulta. Ao clicar na seta, além da opção de realizar a operação através das hierarquias configuradas em metadados, é possível fazer o *drill down* ou *roll-up* para qualquer um dos demais itens disponíveis ao usuário. A figura 5.15 mostra o menu que é exibido ao clicar na seta da operação de *drill*. Nele, os campos pertencentes à hierarquia são mostrados em destaque, e os demais itens podem ser abertos na opção “Fazer *Drill* para Itens Relacionados”.

							Valor Faturado
Ano	Jan	Fev	Mar	Abr	Mai	Jun	
Administrador							
Eduardo	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99	999.999,99
Administrador							
Representante							
Grupo Economico							
Cliente							
Produto							
Pedido							
Nota Fiscal							
Todas as Hierarquias ▶							
Fazer Drill para Itens Relacionados ▶							
1.2.48.18							

- Empresa
- Representante
- Grupo Economico
- Cliente
- Produto
- Pedido
- Segmento
- Condicao Pagto
- Estado
- Cidade
- Data Emissao
- Nota Fiscal
- Data Emissao: MM/RRRR
- Data Emissao: DD/MM/RRRR
- Trimestre

Figura 5.15 – Menu de *Drill-Down* e *Roll-up*

Essa funcionalidade também existe na visualização de gráficos. No exemplo da figura 5.16, há o gráfico de devoluções por motivo (figura 5.16[a]), nele é possível clicar em uma das fatias que representam os motivos e visualizar de modo mais detalhado. Ao clicar no motivo de devoluções “Problemas de Informações” (1), foi gerado um novo gráfico (Figura 5.16[b]), mostrando quais os problemas de informações que causaram devoluções. É possível retornar ao gráfico original, clicando no *link* na parte superior da legenda (2). Diferentemente do *drill* das tabelas, nos gráficos só é possível realizar essas operação através das hierarquias configuradas no *Discoverer Administrator*.

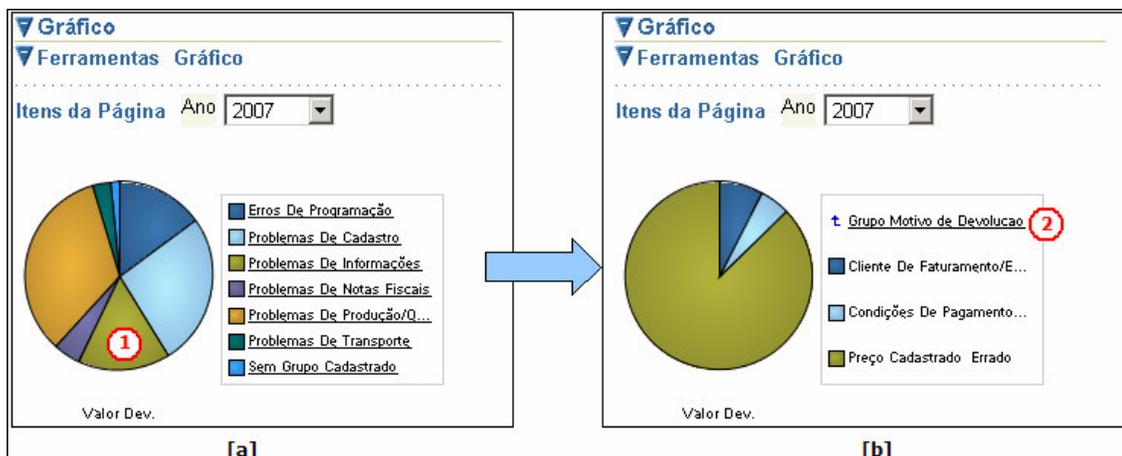


Figura 5.16 – Exemplo de Drill-down em gráficos

### 5.2.1.2.6 Ferramentas dos Gráficos

Nos gráficos, o usuário também possui a opção de formatações. Alterar as dimensões e também do gráfico, aplicar efeito de 3D, de gradiente e ainda alterar o tipo do gráfico. Na figura 5.17 está representado graficamente o menu de opções de formatação de gráficos.

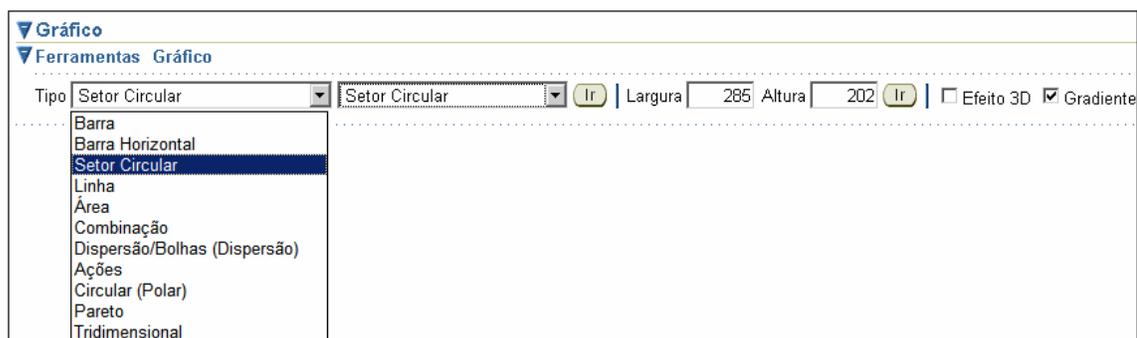


Figura 5.17 – Ferramentas de gráficos

### 5.2.1.2.7 Parâmetros

Além das ferramentas apresentadas, algumas consultas possuem parâmetros, que fazem a função de filtro. O gráfico “desempenho dos maiores clientes”, visualizado na figura 5.18, possui essa opção com parâmetros de período e número de clientes a serem visualizados. O usuário preenche os campos correspondentes e clica no botão de “Ir”. Dessa forma a consulta ao banco de dados se restringe de acordo com os valores determinados pelo usuário.

**Parâmetros**

Selecione valores para os parâmetros a seguir.

Indica campo obrigatório

Data Inicial: '01/01/2005'  
Informe a Data Inicial do Período (Exemplo: 18/10/2007)

Data Final: '31/12/2007'  
Informe a data final do período (Exemplo: 18/10/2007)

Posição: '5'  
Informe até qual posição do Ranking você deseja visualizar

Ir

Figura 5.18 – Tela com opção de parâmetros

### 5.3 Controle de acessos

O BOX BI será acessado por quatro grupos de usuários: diretoria, administradores de venda, gerentes das empresas do grupo Box Print e funcionários da área de TI. Cada um desses grupos terá características exclusivas de acesso, que serão explicadas a seguir.

#### 5.3.1 Controle de consultas

O controle de consultas é realizado através da permissão de um usuário ou grupo de usuários visualizarem os *portlets* disponíveis no sistema. Ou seja, quando um *portlet* é criado, seja qual o seu conteúdo, é possível determinar quem poderá visualizá-lo.

O menu do sistema é composto por *portlets*, sendo que cada um deles é um dos módulos de consultas. Assim, os grupos de administradores de venda e gerentes não terão acesso às consultas do módulo “tático”. Já a diretoria e informática terão acesso total ao conteúdo do sistema. As demais telas não sofrem nenhum tipo de controle, sendo liberadas de forma igual a todos os usuários.

#### 5.3.2 Controle de informações

Além do controle de que telas podem estar disponíveis para determinado grupo de usuários, foi necessário implementar um controle de quais informações dentro de uma mesma consulta cada usuário pode visualizar.

Os administradores de venda só podem visualizar as informações que correspondem a sua equipe, já os gerentes de unidades só podem visualizar os dados da unidade pela qual é o responsável. Diretoria e informática acessam todas as informações.

Isso foi possível criando uma estrutura composta por duas tabelas no banco de dados, nas quais estão armazenados os dados dos usuários e quais unidades ou equipes de venda o mesmo pode visualizar. A figura 5.19 representa graficamente essa estrutura.

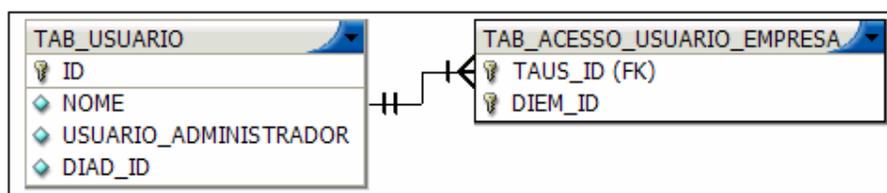


Figura 5.19 – Tabelas do controle de acessos

A tabela “TAB\_USUARIO” possui o código (ID) e nome do usuário (NOME), além da informação se ele é um administrador do sistema (USUARIO\_ADMINISTRADOR) e qual o código de administrador de venda dele (DIAD\_ID), caso seja um. A tabela “TAB\_ACESSO\_USUARIO\_EMPRESA” armazena quais as empresas do grupo Box Print o usuário tem permissão de visualizar (DIEM\_ID).

Para que os dados dessas tabelas sejam utilizados para o controle de informações foi criada uma função no banco de dados, que foi incorporada às *views* utilizadas no sistema. Essa função, chamada SGC\_FNC\_RETORNA\_ACESSO\_REGISTRO é executada registro a registro, ou seja, a cada registro lido na *view*, é feita a verificação se o usuário tem a permissão de visualizá-lo. Para a função, são passados os parâmetros de qual o usuário está conectado, qual a unidade do registro e qual o administrador de venda do mesmo. Assim, é feita a consulta nas tabelas verificando se o usuário pode visualizá-lo. Se a mesma retornar o valor de “S” a visualização é permitida, caso contrário, não é permitida.

Assim, os usuários dos gerentes possuem cadastradas, na tabela TAB\_ACESSO\_USUARIO\_EMPRESA, apenas as empresas que os mesmos podem visualizar. Os administradores de venda têm o código da equipe de venda vinculado ao seu usuário. A diretoria possui todas as empresas cadastradas e a informática está com os usuários marcados como administradores do sistema.

## CONCLUSÃO

Através de tudo que foi apresentado ao final desse trabalho de conclusão, pode-se perceber o grau de valia que possuem as tecnologias de *Business Intelligence*, trazendo um grande auxílio para a busca dos objetivos traçados no início do projeto.

Inicialmente, pôde-se estudar DW e OLAP, através de um embasamento teórico, o qual possibilitou a compreensão de como esses conceitos podem ser aplicados para o desenvolvimento desse projeto. Através do estudo de DW, criou-se uma modelagem dimensional utilizando a técnica *star-schema*, que trouxe simplicidade e ao mesmo tempo ganho de performance ao sistema como um todo. Ainda, mesmo utilizando uma única fonte de dados, foi possível aplicar o uso de ETL, criando uma base de dados abrangente em termos de conteúdo e agilidade nas consultas.

Os conceitos de OLAP puderam ser compreendidos de forma mais clara também com a aplicação da ferramenta *Oracle Discoverer*, que traz consigo diversas funcionalidades dessa tecnologia, tais como *drill-down* e *roll-up*, filtros e *pivoting*.

As tecnologias de BI puderam então ser aplicadas através do *Oracle Discoverer*, que apresenta um gama de recursos muito grande e que pôde ser explorada nesse projeto. Mesmo não sendo utilizada, a ferramenta *Oracle Warehouse Builder* contribuiu para o entendimento de como funcionam os processos ETL, servindo de auxílio para a construção dos *scripts* em PL/SQL, que executam os mesmos.

Assim, com esses conceitos e ferramentas foi possível implementar o sistema BOX BI, que teve como principal objetivo fazer com que as informações que cheguem aos gerentes e diretores sejam confiáveis para as suas tomadas de decisão, eliminando relatórios paralelos que poderiam trazer a incerteza na veracidade das informações.

O sistema foi apresentado à diretoria e aprovado por ela. Dessa forma, o diretor comercial já está utilizando o BOX BI e incentivando o seu uso para os administradores de venda e gerentes. O número de usuários que já estão trabalhando com a ferramenta só não é maior, porque um dos servidores, o “Bi Box Print”, ainda é um *desktop*, e por isso não atenderia a demanda de mais usuários. Porém, após a aprovação do sistema pela diretoria, um servidor será adquirido, possibilitando crescimento de usuários previstos no início do projeto.

Mesmo assim, as informações já estão sendo disponibilizadas para os demais usuários pelo setor de TI, através desse serviço que cumpre o papel de centralizar relatórios e demais consultas, eliminando envios de planilhas eletrônicas geradas através do banco transacional. Com isso foi possível diminuir a redundância, tempo e garantir segurança nas informações para a tomada de decisão.

Para o futuro, abre-se a oportunidade de disponibilizar o sistema para demais tipos de usuários, como representantes comerciais, possibilitando a eles terem um gerenciamento de seus clientes.

Além de outros usuários, há a possibilidade de expandir o BOX BI dentro da empresa, desenvolvendo novos *data-marts* para áreas que também possuem carência em informações de cunho gerencial, como suprimentos ou recursos humanos, trazendo a elas a possibilidade de tomada de decisão de forma rápida e segura.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ADOBE. *Adobe PDF*. 2007. Disponível em [www.adobe.com/br/products/acrobat/adobepdf.html](http://www.adobe.com/br/products/acrobat/adobepdf.html). Acesso em 10 jun. 2007.
- ANZANELLO, Cynthia Aurora. **OLAP: conceitos e utilização**. Porto Alegre: 2002. 7p. Artigo - Instituto de Informática, UFRGS. Disponível em: [http://www.inf.ufrgs.br/~clesio/cmp151/cmp15120021/artigo\\_cynthia.pdf](http://www.inf.ufrgs.br/~clesio/cmp151/cmp15120021/artigo_cynthia.pdf). Acesso em 02 mai. 2007.
- APACHE. *Apache Software Foundation*. Disponível em [www.apache.org](http://www.apache.org). Acesso em 10 jun. 2007.
- AUDY, Rejane Blomberg; ENDRES, Ana Cristina Trois; MALVEZZI, Maria Luiza Falsarella. **Case BI - Informações Gerenciais em Hospital de Referência**. Porto Alegre: 2003. 11p. Artigo – Hospital de Clínicas de Porto Alegre. Disponível em <http://inf.unisul.br/~ines/workcomp/cd/pdfs/2420.pdf>. Acesso em 15 mar. 2007
- BARBALHO, Patrícia. Descubra o *Data Warehouse*: produtividade e rapidez. **Revista SQL Magazine**. Rio de Janeiro: nº 3, ano 1, p. 34 – 38, 2003
- BARBIERI, Carlos. **Business Intelligence: modelagem e tecnologia**. Rio de Janeiro: Axcel Books, 2001. 424 p.
- BERVIAN, Andreia Eliana. **Critérios para a decisão de personalização de processos na implantação de Sistemas ERP**. São Leopoldo: 2004. 69 p. Trabalho de conclusão de curso apresentado como requisito parcial para a obtenção do título de Bacharel em Informática- Universidade do Vale do Rio dos Sinos. Disponível em [http://inf.unisinos.br/alunos/arquivos/TC\\_AndreiaBervian.pdf](http://inf.unisinos.br/alunos/arquivos/TC_AndreiaBervian.pdf). Acesso em 10 jun. 2007

- BEVILACQUA, J. F.; BITU Y. A. **Business Intelligence (BI) e a abordagem de Gestão Balanced Scorecard (BSC) na Organização**. Brasília: 2003. 54p. Monografia - Pós Graduação MBA em Sistemas de Informação – Universidade Católica de Brasília. Disponível em [http://www.eln.gov.br/Conhecimento/GestaodoConhecimento/Monografia\\_MBA\\_180703.pdf](http://www.eln.gov.br/Conhecimento/GestaodoConhecimento/Monografia_MBA_180703.pdf). Acesso em 15 mar. 2007
- BI BLOG. **Add A Quick Export Link to the Worksheet Page**. 2006. Disponível em <http://oraclebi.blogspot.com/2006/03/add-quick-export-link-to-worksheet.html>. Acesso em 18 ago. 2007.
- BISPO, Carlos Alberto Ferreira. **Uma análise da nova geração de sistemas de apoio à decisão**. São Carlos: 1998. 160 p. Dissertação de Mestrado - Escola de Engenharia de São Carlos, Universidade de São Paulo. Disponível em: [http://www.teses.usp.br/teses/disponiveis/18/18140/tde-04042004-152849/publico/dissertacao\\_carlos.pdf](http://www.teses.usp.br/teses/disponiveis/18/18140/tde-04042004-152849/publico/dissertacao_carlos.pdf). Acesso em: 10 mar. 2007.
- BOGHI, C.; SHITSUKA R.. **Sistemas de Informação: Um Infoque Dinâmico**. São Paulo-SP: Érica, 2002. 288p.
- CARLSSON, C.; TURBAN, E. **DSS: directions for the next decade**. *Decision Support Systems*, v. 33, n. 2, p. 105-110, 2002.
- CIELO, Ivã. **Arquiteturas OLAP**. 2000. Disponível em <http://www.datawarehouse.inf.br/artigos/olap2.asp>. Acesso em 04 mai. 2007.
- COMUNIQUE-SE. **Comunique-se – O Portal da comunicação**. 2007. Disponível em [http://www.comunique-se.com.br/produtos/saladeimprensa/oracle/show.asp?\\_mat=32295&\\_ed=602&\\_tar=PK&\\_sec=om](http://www.comunique-se.com.br/produtos/saladeimprensa/oracle/show.asp?_mat=32295&_ed=602&_tar=PK&_sec=om). Acesso em 15 jul. 2007
- COSTA, André Fernandes; ANCIÃES, Felipe Curvello. **Aspectos de criação e carga de um ambiente de Data Warehouse**. Rio de Janeiro: 2001. 111 p. Projeto de Diplomação (Bacharelado em Informática) – Departamento de Ciência da Computação, Universidade Federal do Rio de Janeiro. Disponível em: <<http://dataware.nce.ufrj.br:8080/dataware/publicacoes/dataware/fisico/projetosFinais/datawarehousing/COSTA-ANCIAES-2001.pdf>>. Acesso em: 19 abr. 2007.

- DWBRASIL. **Histórico do Data Warehouse**. 2003. Disponível em [http://www.dwbrasil.com.br/html/artdw\\_hist.html](http://www.dwbrasil.com.br/html/artdw_hist.html). Acesso em 19 abr. 2007.
- ECLIPSE. **Eclipse Foundation**. Disponível em: <http://www.eclipse.org>. Acesso em: 10 jun 2007.
- FERNANDES, Sérgio Antônio de Sousa. **O projeto de Data Warehouses: A tecnologia ROLAP versus MOLAP**. Lisboa-Portugal: 2004. 13 p. Artigo – Instituto Superior Técnico – Lisboa – Portugal. Disponível em: [http://berlin.inesc-id.pt/cadeiras/pfsi/PFSI2003/SEMINARIO/pdfs/Data\\_Warehouse-Sergio\\_Fernandes.pdf](http://berlin.inesc-id.pt/cadeiras/pfsi/PFSI2003/SEMINARIO/pdfs/Data_Warehouse-Sergio_Fernandes.pdf). Acesso em: 29 abr. 2007.
- FIGUEIREDO, Adriana Maria C.M. MOLAP x ROLAP: Embate de Tecnologias para Data warehouse. **Developer's Magazine**. Rio de Janeiro: n° 18 p. 24, 1998
- FORTULAN, Marcos Roberto. Filho; Eduardo Vila Gonçalves. **Uma Proposta de Aplicação de um Business Intelligence no Chão-de-Fábrica**. São Carlos-SP: 2005. 12p. Artigo – G & P: Gestão e Produção. Disponível em: [www.scielo.br/pdf/gp/v12n1/a06v12n1.pdf](http://www.scielo.br/pdf/gp/v12n1/a06v12n1.pdf) . Acesso em 01 mai. 2007
- GENTIA SOFTWARE. **OLAP for enterprise**. Disponível em [www.gentia.com/products/gseolap.exe](http://www.gentia.com/products/gseolap.exe). Acesso em 11 mar. 2007
- GONÇALVES, Marcio Elias. **Uma Ferramenta de Extração de Dados para Data Warehouse Baseada em Agentes Distribuídos**. Florianópolis: 2002. 112 p. Dissertação de Mestrado - Universidade Federal de Santa Catarina. Disponível em <http://www.datainfo.inf.br/marcio/download/artigos/mestrado.pdf>. Acesso em 06 mai. 2007.
- GORLA, Nasimhaiah. **Features to consider in a Data Warehouse System**. 2003. Disponível em <http://portal.acm.org/citation.cfm?id=948389>. Acesso em 04 mai. 2007
- GRAY, Paul, WATSON, Hugh J. **Decision Support in the Data Warehouse**. New Jersey: Prentice Hall PTR, 1998. 416 p.
- GRAY, P.; WATSON, H. J. **The new DSS: data warehouses, OLAP, MDD and KDD**. 1999. Disponível em: <http://hsb.baylor.edu/ramsover/ais.ac.96/papers/graywats.htm>. Acesso em: 01 mai. 2007.

HARRISON, Thomas H. *Intranet Data Warehouse*. Tradução: Daniel Vieira. São Paulo: Berkeley Brasil, 1998. 359 p.

HOKAMA, Daniele Del Bianco et al. **A modelagem de dados no ambiente *Data Warehouse***. São Paulo: 2004. 121 p. Projeto de Diplomação (Bacharelado em Sistemas de Informação) – Faculdade de Computação e Informática, Universidade Presbiteriana Mackenzie. Disponível em:  
<<http://meusite.mackenzie.com.br/rogerio/tgi/2004ModelagemDW.pdf>>. Acesso em: 29 abr. 2007.

INMON, Willian H. *Building the Data Warehouse*. New York: John Wiley & Sons. 1996. 401 p.

INMON, Willian H. **Como construir um *Data Warehouse***. Tradução da Segunda Edição. Rio de Janeiro-RJ: Campus, 1997. 388 p.

INMON, W.H, WELCH, J. D., GLASSEY, K. L. **Gerenciando *Data Warehouse***. São Paulo: Makron Books, 1999. 375p.

JAVA. *Java Technology*. Disponível em: <http://java.sun.com>. Acesso em: 10 jun. 2007.

JBOSS. *JBOSS Community driven*. Disponível em [www.jboss.org](http://www.jboss.org). Acesso em 10 jun. 2007.

JCP. *Java Community Process*. Disponível em: <http://www.jcp.org>. Acesso em: 10 jun. 2007.

JFREECHART. *A free Java chart library*. Disponível em: <http://www.jfree.org/jfreechart>. Acesso em: 10 jun. 2005.

JPIVOT. *A JSP based OLAP*. Disponível em: <http://jpivot.sourceforge.net>. Acesso em 10 jun. 2007.

KIMBALL, Ralph. *Data Warehouse Toolkit*. São Paulo: Makron Books, 1996. 388 p.

KIMBALL, Ralph, REEVES, Laura, ROSS, Margy, THORNTHWAITE, Warren. **The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing and Deploying Data Warehouses**. New York: John Wiley & Sons, Inc., 1998. 771 p.

KIMBALL, Ralph, ROSS, Margy. **The Data Warehouse Toolkit (Segunda Edição)**. Rio de Janeiro: Editora Campus, 2002. 494 p.

KLEIN, Cc LAWRENCE ZORDAM. **A Tecnologia Relacional - Objeto em um Ambiente de Data Warehouse**. Rio de Janeiro: 1999. 191 p. Tese Submetida como Requisito Parcial para a Obtenção do Grau de Mestre em Ciências em Sistemas e Computação - Instituto Militar de Engenharia. Disponível em <http://dataware.nce.ufrj.br:8080/dataware/publicacoes/dataware/fisico/teses/datawarehousing/KLEIN-1999.pdf>. Acesso em 15/05/2007

LAUDON, K. C; LAUDON, J. P. **Sistemas de Informação com internet**. 4. ed. Rio de Janeiro: LTC, 1999. 408 p.

LOBO, Adilton. **Modelagem de um estudo de caso utilizando a ferramenta entidade/relacionamento, baseada no modelo de Peter Chen**. Joinville, SC: 1998. Disponível em: <http://pages.udesc.br/~r4al/artentre.htm>. Acesso em: 06 jun. 2007.

LOPES, Alison Zille. **As tendências e preocupações da Tecnologia da Informação no Brasil**. Lavras – MG: 2006. 68 p. Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para a obtenção do título de Bacharel em Ciência da Computação - Departamento de Ciência da Computação da Universidade Federal de Lavras. Disponível em [http://www.comp.ufla.br/monografias/ano2006/As\\_tendencias\\_e\\_preocupacoes\\_da\\_tecnologia\\_da\\_informacao\\_no\\_Brasil.pdf](http://www.comp.ufla.br/monografias/ano2006/As_tendencias_e_preocupacoes_da_tecnologia_da_informacao_no_Brasil.pdf). Acesso em 27 abr. 2007

MACHADO, Felipe. N. R.; ABREU, M. P. **Projeto de Banco de Dados: uma visão prática**. São Paulo: Érica, 1996. 320 p.

MACHADO, Felipe N. R. **Projeto de Data Warehouse: uma visão multidimensional**. São Paulo: Érica, 2000. 251 p.

MARQUES, Valmir Ferreira. **Analisando os Dados de Melhoramento Genético da Raça Nelore com Data Warehousing e Data Mining**. São Carlos-SP: 2002. 134 p. Dissertação de Mestrado na Área de Ciências da Computação e Matemática Computacional – Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo. Disponível em <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-28022003-161537/publico/Dissertacao-Valmir.pdf>. Acesso em 06 mai. 2007.

MEIRA, Carlos Alberto Alves; SOUZA, Tatiana Aparecida Lima de. **Oracle Discoverer: Guia do Usuário**. 2002. Campinas-SP: 2002. 54 p. Ministério da Agricultura, Pecuária e Abastecimento. Disponível em <http://www.cnptia.embrapa.br/modules/tinycontent3/content/2002/doc21.pdf>. Acesso em 15 mar. 2007.

MEREDITH, Mary E; KHADER Aslam. **Divide and Aggregate: desingning large warehouses**. 1996. Disponível em <http://www.dbpd.com/vault/khader.htm>. Acesso em 01 mai. 2007

MICROSOFT. **Analysis Services**. 2005. Disponível em <http://www.microsoft.com/brasil/sql/technologies/analysis/default.mspix>. Acesso em 08 mai. 2007.

MICROSOFT. **Microsoft Corporation**. 2007. Disponível em [www.microsoft.com](http://www.microsoft.com). Acesso em 10 jun. 2007.

MONDRIAN. **Pentaho Analysis Services: Mondrian Project**. 2007. Disponível em <http://mondrian.pentaho.org/>. Acesso em 14 mai. 2007

OPENI. **OpenI**. 2005. Disponível em <http://portuguese.osstrans.net/software/openi.html>, Acesso em 08 mai. 2007.

OLIVEIRA, Eric C M. **Tecnologia de Portais Java e a JSR 168**. 2004. Disponível em [http://www.linhadecodigo.com.br/artigos.asp?id\\_ac=494](http://www.linhadecodigo.com.br/artigos.asp?id_ac=494). Acesso em 10 jun. 2007.

ORACLE. **Introducing Oracle Desingner**. 2000. Disponível em [http://download.oracle.com/otn\\_hosted\\_doc/designer/misc/276931/dsgnr\\_tutch01\\_65.htm](http://download.oracle.com/otn_hosted_doc/designer/misc/276931/dsgnr_tutch01_65.htm). Acesso em 25 ago 2007.

ORACLE. **Introdução ao Oracle: SQL e PL/SQL – Volume 1**. São Paulo – SP. 2000. 372 p.

ORACLE. **Enterprise DBA Parte 1A: Administração e Arquitetura – Volume 1**. São Paulo - SP. 2000. 489 p.

ORACLE. **Jobs**. 2002. Disponível em [http://download.oracle.com/docs/cd/B10501\\_01/em.920/a96670/ch\\_jobs.htm#520](http://download.oracle.com/docs/cd/B10501_01/em.920/a96670/ch_jobs.htm#520). Acesso em 06 set 2007.

ORACLE. *Oracle® Warehouse Builder - User's Guide - 10 g Release 1 (10.1)*. 2003. Disponível em [http://download-east.oracle.com/docs/pdf/B12146\\_01.pdf](http://download-east.oracle.com/docs/pdf/B12146_01.pdf), Acesso em 19 mar. 2007

ORACLE. *Oracle® Business Intelligence Discoverer Desktop - User's Guide - 10g Release 2 (10.1.2.1) for Windows*. 2005. Disponível em [http://download-east.oracle.com/docs/pdf/B13917\\_03.pdf](http://download-east.oracle.com/docs/pdf/B13917_03.pdf), Acesso em 12 mai. 2007

ORACLE. *Oracle® Business Intelligence Discoverer Administrator - User's Guide - 10g Release 2 (10.1.2.1) for Windows*. 2005. Disponível em [http://download-east.oracle.com/docs/pdf/B13917\\_04.pdf](http://download-east.oracle.com/docs/pdf/B13917_04.pdf), Acesso em 12 mai. 2007

ORACLE. *Oracle® Business Intelligence Discoverer Plus - User's Guide - 10g Release 2 (10.1.2.1) for Windows*. 2005. Disponível em [http://download-east.oracle.com/docs/pdf/B13915\\_04.pdf](http://download-east.oracle.com/docs/pdf/B13915_04.pdf), Acesso em 12 mai. 2007

ORACLE. *Oracle® Business Intelligence Discoverer Viewer - User's Guide - 10g Release 2 (10.1.2.1) for Windows*. 2005. Disponível em [http://download-east.oracle.com/docs/pdf/B13987\\_04.pdf](http://download-east.oracle.com/docs/pdf/B13987_04.pdf), Acesso em 12 mai. 2007

ORACLE. *Oracle apresenta nova versão de ferramenta de design de banco de dados*. 2006. Disponível em [http://www.oracle.com/global/br/corporate/press/2006\\_oct/oracle\\_nova\\_ferramenta\\_banco\\_dados.html](http://www.oracle.com/global/br/corporate/press/2006_oct/oracle_nova_ferramenta_banco_dados.html). Acesso em 23 mar. 2007

ORACLE. *Oracle Corporation*. 2007. Disponível em [www.oracle.com](http://www.oracle.com). Acesso em 29 mai. 2007.

ORACLE AS. *Oracle Application Server*. Disponível em [www.oracle.com/appserver/index.html](http://www.oracle.com/appserver/index.html) . Acesso em 10 jun. 2007.

ORACLE BI. *Oracle Business Intelligence*. 2006. Disponível em <http://www.oracle.com/bi/index.html>. Acesso em 23 mar. 2007.

ORACLE UIX. *Oracle9i XML Developer's Kits Guide – XDK -Release 2 (9.2) – Introduction to UIX*. 2007. Disponível em

[http://download.oracle.com/docs/cd/B10501\\_01/appdev.920/a96621/adx26uix.htm](http://download.oracle.com/docs/cd/B10501_01/appdev.920/a96621/adx26uix.htm). Acesso em 05 out. 2007.

PENTAHO. *Open Source Business Intelligence*. Disponível em: <http://www.pentaho.org>. Acesso em: 18 mai. 2007

PETERSON et al. *Microsoft Olap Unleashed*. Sams Publishing, 1999. 949p.

POE, Vidette, KLAUER, Patricia, BROBST, Stephen. *Building a Data Warehouse for Decision Support*. New Jersey. Prentice-Hall, Inc, 1998. 285 p.

POLONI, E. G. F. *Administrando sistemas de informação*. São Paulo: Futura, 2000. 272 p.

PORTAL. *Oracle Application Server Portal*. 2007. Disponível em <http://www.oracle.com/global/pt/pmes/documentation/oracle%20application%20server%20portal1.pdf>. Acesso em 01 out. 2007.

REZENDE, D. A. *Engenharia de software e sistemas de informação*. RJ: Brasport, 1999. 344 p.

SANTOS, Rodrigo Gonçalves. *Utilização de técnicas de Data Mining na Busca de conhecimento na web*. Pelotas-RS: 2000. 119 p. Monografia apresentada ao Curso de Bacharelado em Informática do Instituto de Física e Matemática da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Bacharel em Informática - Instituto de Física e Matemática da Universidade Federal de Pelotas. Disponível em <http://www.ufpel.edu.br/prg/sisbi/bibct/acervo/info/2000/Mono-RodrigoSantos.pdf>. Acesso em 02 mai. 2007.

SERRA, Laércio. *A essência do Business Intelligence*. São Paulo: Berkeley, 2002. 296 p.

SHIM, J. P. Et al . *Past, present, and future of decision support technology*. *Decision Support System*, v. 33, n. 2, p. 111-126, 2002.

SIMON, Inre. Um Estudo de Caso: A Produção e Disseminação da Literatura Acadêmica. Disponível em: <http://www.ime.usp.br/~is/ddt/mac339-01/aulas/www.linux.ime.usp.br/hvila/mac339/tema8.html>. Acesso em 12 jun. 2007.

SINGN, Harry S. *Data Warehouse: Conceitos, Tecnologias, Implementação e Gerenciamento*. São Paulo: Makron Books, 2001. 382 p.

SOUZA, César Alexandre de; ZWICKER, Ronaldo. Ciclo de vida de sistemas ERP. **Caderno de pesquisas em administração**, São Paulo, v. 1, n. 11, 1 trim./2000.

SPRAGUE, R. H.; WATSON, H. J. **Sistemas de Apoio à Decisão: Colocando a Teoria em Prática**. Rio de Janeiro: Campus, 1991. 498 p.

THE R PROJECT. *The R Project for Statistical Computing*. Disponível em: <http://www.rproject.org>. Acesso em: 10 jun. 2007.

WAGNER, Cláudio Arruda. **Estudo para implantação de um Data Warehouse em um ambiente empresarial**. Florianópolis: 2003. 99 p. Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação - Universidade Federal de Santa Catarina. Disponível em <http://teses.eps.ufsc.br/defesa/pdf/16068.pdf>. Acesso em 03 mai. 2007.

WEBLOGIC. *WebLogic – Business Software*. Disponível em [www.bea.com](http://www.bea.com). Acesso em 10 jun. 2007.

WEBSPHERE. *IBM WebSphere Software*. Disponível em [www.ibm.com/software/websphere](http://www.ibm.com/software/websphere). Acesso em 10 jun.2007.

WIKIPEDIA. A enciclopédia livre. 2007. Disponível em <http://pt.wikipedia.org>. Acesso em 10 jun. 2007.

YOUNESS, S. *Professional Data Warehousing with SQL Server 7.0 and OLAP Services*. Wrox Press Ltd, 2000. 604 p.