

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

FERRAMENTA DE SUPORTE A DECISÃO
CARACTERIZADA POR CONSULTAS OLAP

DANIEL RICARDO BATISTON

BLUMENAU
2007

2007/1-03

DANIEL RICARDO BATISTON

**FERRAMENTA DE SUPORTE A DECISÃO
CARACTERIZADA POR CONSULTAS OLAP**

Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina Trabalho de Conclusão de Curso II do curso de Sistemas de Informação - Bacharelado.

Prof. Evaristo Baptista, MEng. - Orientador

**BLUMENAU
2007**

2007/1-03

FERRAMENTA DE APOIO A DECISÃO CARACTERIZADO POR CONSULTAS OLAP

Por

DANIEL RICARDO BATISTON

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: _____
Prof. Evaristo Baptista, Mestre – Orientador, FURB

Membro: _____
Prof. Oscar Dalfovo, Doutor – FURB

Membro: _____
Prof. Wilson Pedro Carli, Mestre – FURB

Blumenau, 05 de Julho de 2007

Dedico este trabalho a todos que, de alguma forma, me ajudaram em toda esta caminhada de conhecimento e vivência.

AGRADECIMENTOS

À Deus, que me abençoou e iluminou meu caminho para que chegasse até aqui.

À minha família, que mesmo longe, sempre esteve presente.

Aos meus amigos, pela amizade e companheirismo durante todo o curso.

Também a todos meus professores, que me transmitiram seus conhecimentos e suas experiências e em especial ao professor Evaristo Baptista, meu orientador pela atenção, sugestões e orientação durante a elaboração deste trabalho.

Finalmente, agradeço a todos aqueles que de alguma maneira, direta ou indireta contribuíram para a elaboração deste trabalho.

Muito obrigado!

As pessoas que vencem neste mundo são as que procuram as circunstâncias de que precisam e, quando não as encontram, as criam.

Bernard Shaw

RESUMO

A necessidade cada vez maior de possuir informações de forma rápida, clara e confiável para a tomada de decisões, faz com que as empresas invistam cada vez mais em sistemas de informação. Esta pesquisa, realizou um estudo mais aprofundado sobre tecnologia OLAP, além do desenvolvimento de uma ferramenta de auxílio na geração de cubos de decisões para exploração de dados utilizando os conceitos de *Data Mart* em bancos de dados relacionais. Isso permite aos usuários criar consultas e analisar rapidamente inúmeros cenários, gerar relatórios e descobrir tendências e fatos relevantes independente do tamanho, complexidade e fonte dos dados corporativos, tornando o analista um explorador da informação.

Palavras-chave: Tomada de decisões. OLAP. Cubo de decisões.

ABSTRACT

The ever increasing need to gather information in a fast, clear and reliable way for decision making drives companies to invest more and more in information systems. As a result of deep research of OLAP technology, and the development a tool that aids in the generation of cubes of decisions for exploration of data using Data Mart concepts in relational databases. This allow the users to create queries and to analyze countless scenarios quickly, this tool also allows users to generate reports and to discover tendencies and relevant facts independent of the size, complexity and source of the corporate data, turning the analyst into an explorer of the information.

Key-words: Decision making. OLAP. Cube of decisions.

LISTA DE ILUSTRAÇÕES

Figura 1 – Esquema estrela (<i>Star Schema</i>)	22
Figura 2 – Esquema <i>snowflake</i> (<i>Snowflake Schema</i>).....	22
Figura 3 – Análise multidimensional dos dados – Representação do cubo de decisão	23
Quadro 1 – Requisitos funcionais da ferramenta.....	27
Quadro 2 – Requisitos não funcionais da ferramenta.....	27
Figura 4 – Diagrama de Caso de Uso da ferramenta	29
Quadro 3 – Descrição do Use Case – Logar na Ferramenta.....	29
Quadro 4 – Descrição do Use Case – Manter Dicionário de Dados.....	30
Quadro 5 – Descrição do Use Case – Manter Usuários.....	31
Quadro 6 – Descrição do Use Case – Manter tabelas acessíveis ao usuário.....	31
Quadro 7 – Descrição do Use Case – Criar Consulta.....	31
Quadro 8 – Descrição do Use Case – Gravar Consulta.....	32
Quadro 9 – Descrição do Use Case – Visualizar código SQL.....	32
Quadro 10 – Descrição do Use Case – Abrir Consulta.....	33
Quadro 11 – Descrição do Use Case – Visualizar Consulta OLAP.....	33
Quadro 12 – Descrição do Use Case – Imprimir Dados.....	33
Quadro 13 – Descrição do Use Case – Exportar Dados.....	34
Quadro 14 – Descrição do Use Case – Visualizar Gráfico.....	34
Figura 5 – Diagrama de Classes	35
Figura 6 – Diagrama de Seqüência – Abrir Consulta	37
Figura 7 – Diagrama de Seqüência – Criar Consulta.....	37
Figura 8 – Diagrama de Entidade Relacionamento do Banco de Dados	39
Quadro 15 – Tabela de Consultas.....	40
Quadro 16 – Tabela de configuração das dimensões da consulta.....	40
Quadro 17 – Tabela de configurações dos parâmetros da consulta.....	41
Quadro 18 – Tabela de estruturas de tabelas do banco de dados.....	41
Quadro 19 – Tabela de estruturas de campos do banco de dados.....	42
Quadro 20 – Tabela de usuários.....	42
Quadro 21 – Tabela de configuração de acesso de cada usuário às tabelas de consulta.....	42
Figura 9 – Configuração do Banco de Dados.....	45
Figura 10 – Criação das estruturas do Banco de Dados.....	45

Figura 11 – Cadastro de Usuários e configurações de acesso	46
Figura 12 – <i>Login</i> da Ferramenta.....	46
Quadro 22 – Funções que relacionam a estrutura de campos existente no banco de dados	47
Figura 13 – Dicionário de Dados : Configuração de tabelas.	48
Figura 14 – Dicionário de Dados : Configuração de campos.	49
Figura 15 – Permissões de acesso às tabelas pelos usuários.....	49
Figura 16 – <i>Interface</i> Principal do Sistema	50
Quadro 23 – Função que monta a estrutura em árvore das consultas existente no banco de dados	51
Figura 17 – Etapa 1 : Seleção das tabelas utilizadas na consulta.....	52
Figura 18 – Etapa 2 : Seleção da tabela de fato	52
Figura 19 – Etapa 3 : Seleção de campos para a consulta	53
Figura 20 – Etapa 4 : Definição da posição inicial dos dados na consulta	53
Figura 21 – Etapa 5 : Definição de Agregações	54
Figura 22 – Etapa 6 : Criação e filtros para a consulta	54
Figura 23 – Etapa 7 : Definição de nome e descrições para a consulta.....	55
Figura 24 – Etapa 8 : Finalizando Nova Consulta.....	56
Figura 25 – Visualização de <i>script</i> SQL.....	56
Figura 26 – Objetos aplicados às configurações de filtro das consultas.....	57
Quadro 24 – Função de criação dinâmica dos objetos e parâmetros da consulta.....	59
Quadro 25 – Função de que cria dinamicamente as dimensões e campos calculados no Cubo de Decisão	60
Figura 27 – Consulta aplicada às funcionalidades OLAP - Cubo de Decisões	61
Figura 28 – Filtro de valores no Cubo de Decisões.....	61
Figura 29 – Impressão e Exportação de Dados.....	62
Figura 30 – Opções do Cubo de Decisão.....	62
Figura 31 – Gráficos dos dados consultados.	62

LISTA DE SIGLAS

DBA - *Data Base Administrator*

DM – *Data Mart*

DSS – *Decision Support Systems*

DW – *Data Warehouse*

FK – *Foreing Key*

MER – *Modelo de Entidade Relacionamento*

MOLAP – *Multi-dimensional OLAP*

OLAP – *On Line Analytical Processing*

PK – *Primary Key*

ROLAP – *Relational OLAP*

SQL – *Structured Query Language*

TI – *Tecnologia da Informação*

UML – *Unified Modeling Language*

SUMÁRIO

1 INTRODUÇÃO	13
1.1 CONTEXTUALIZAÇÃO	13
1.2 OBJETIVOS DO TRABALHO	15
1.3 ESTRUTURA DO TRABALHO	15
2 FUNDAMENTAÇÃO TEÓRICA	16
2.1 SISTEMA ATUAL E EMPRESA	16
2.2 CONCEITOS UTILIZADOS	17
2.2.1 Data Warehouse	18
2.2.2 Data Mart	18
2.2.3 OLAP	19
2.2.3.1 Arquitetura de ferramentas OLAP	20
2.2.4 Cubo de Decisão	22
2.3 TRABALHOS CORRELATOS	23
3 DESENVOLVIMENTO DO TRABALHO	25
3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO	25
3.2 ESPECIFICAÇÃO	28
3.2.1 Diagrama de Caso de Uso	28
3.2.2 Diagrama de Classe	34
3.2.3 Diagrama de Seqüência	36
3.2.4 Modelo Entidade Relacionamento – MER	38
3.2.5 Dicionário de Dados	39
3.3 IMPLEMENTAÇÃO	43
3.3.1 Ferramentas e Tecnologias Utilizadas	43
3.3.1.1 FireBird	43
3.3.1.2 HierCube	44
3.3.2 Operacionalidade da Implementação	44
3.3.2.1 Inicialização da Ferramenta	45
3.3.2.2 Visão do Administrador - Manutenção do Dicionário de Dados	47
3.3.2.3 Visão do Usuário - <i>Front End</i> Principal	50
3.3.2.4 Assistente de criação de consultas	51
3.3.2.5 Passagem de parâmetros	56

3.3.2.6 Visualização da consulta OLAP (Cubo de Decisões).....	59
3.4 RESULTADOS E DISCUSSÃO.....	63
4 CONCLUSÕES.....	64
4.1 CONSIDERAÇÕES.....	64
4.2 EXTENSÕES.....	65
4.3 DIFICULDADES.....	65
REFERÊNCIAS BIBLIOGRÁFICAS.....	67

1 INTRODUÇÃO

Este capítulo fornece uma visão geral dos assuntos abordados neste trabalho. Na contextualização, ainda tem-se o relato dos principais objetivos e a estrutura do trabalho.

1.1 CONTEXTUALIZAÇÃO

A tomada de decisão é a atividade máxima de qualquer líder e esse processo se baseia no conhecimento obtido através das informações disponíveis acerca do problema. Os administradores precisam estar muito bem informados, pois a informação é a base para toda e qualquer tomada de decisão.

Segundo Binder (1994, p. 1), a tomada de decisão consiste, basicamente, na escolha de uma opção entre diversas alternativas existentes, seguindo determinados passos que incluem o entendimento do problema, partindo da identificação de possíveis situações que exigem decisão, através da busca detalhada e precisa de informações, culminando na resolução de um problema de modo correto ou não.

Com o desenvolvimento da informática, as empresas, através de seus sistemas operacionais, vêm acumulando diariamente uma vasta quantidade de dados históricos de todas as transações e processos. Estes, por sua vez, quando manipulados e agrupados de forma coerente são transformados em informação.

Conforme Oliveira (1998, p. 3), a obtenção de informações decisórias das bases operacionais de empresas revelam enorme dificuldade, pois os bancos de dados que armazenam suas transações diárias, foram feitos para responder a questões simples, como totalizações e somatórios. Para resolver este problema, surgiu a filosofia *Data Warehouse* (DW), que, segundo Inmon (1997, p. 33), é um conjunto de dados baseado em assuntos, integrados, não-voláteis e variáveis em relação ao tempo, que tem por objetivo dar suporte aos processos de tomada de decisão.

O acesso a essas informações de forma compreensível para a tomada de decisões, tem se tornado crítico no mundo dos negócios. Se os usuários, não puderem obter respostas para

questões importantes do negócio, ou se desmotivarem pela dificuldade no entendimento e análise da informação, os dados armazenados se tornarão informações inúteis. Informações estratégicas nas mãos de um analista ou tomador de decisões podem representar melhores decisões e conseqüentemente uma enorme vantagem competitiva no mercado atual.

O sucesso do DW depende muito da ferramenta de consulta disponível. Para atender a esta demanda de informações na hora e do jeito que o tomador de decisão quer, se destacam as ferramentas para suporte à decisão, adotando soluções que passam pelo uso das técnicas de *On-line Analytic Processing* (OLAP) ou Cubo de Decisões. Estas possibilitam a construção de consultas *ad-hoc* que segundo Inmon, Terdeman e Imhoff (2001, p. 256) “são consultas com acesso casual único e tratamento dos dados segundo parâmetros nunca antes utilizados, geralmente executadas de forma interativa e heurística”.

Baseando-se em toda a informação obtida, a Baptistetti Consultoria Empresarial S/C Ltda, vem como um estudo de caso que auxiliará na realização deste trabalho. Essa empresa desenvolve e implanta softwares destinados a gestão de custos, gerando informações que ajudam os empresários a conhecer sua empresa e a tomar decisões futuras. Com o desenvolvimento personalizado que a empresa oferece aos seus clientes, a demanda por consultas flexíveis, parametrizáveis e de uso esporádico, resulta em uma grande demanda de trabalho para a equipe de desenvolvimento, pois, cada nova forma de análise dos dados, requerida por seus clientes, deve ser projetada e desenvolvida para depois ser incorporada e disponibilizada no sistema. Surge assim, a necessidade de criar mecanismos para agilizar desenvolvimento bem como a distribuição das consultas solicitadas.

Este trabalho teve por objetivo, o desenvolvimento de uma ferramenta para proporcionar agilidade e flexibilidade na construção de consultas e relatórios multidimensionais utilizando tecnologias de Cubo de Decisões para análise de dados e tomada de decisões. Essa ferramenta foi aplicada inicialmente aos bancos de dados dos sistemas da Baptistetti Consultoria, atendendo suas necessidades. Pode também ser configurável para a extração de dados de qualquer base de dados operacional ou *Data Mart* (DM) existente desenvolvido com banco de dados FireBird. Segundo Firebirdsql (2005), FireBird é um gerenciador de banco de dados relacional *open source* e multi-plataforma desenvolvido por uma comunidade independente de programadores a partir do código fonte do InterBase 6.0, liberado pela Inprise Corp (hoje Borland Software Corp) sob a licença *InterBase Public License V.1.0*. Sendo assim a ferramenta poderá ser utilizada por qualquer empresa que possua necessidades semelhantes, indiferente da área de atuação.

1.2 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é desenvolver uma ferramenta de auxílio na geração de Cubos de Decisões para exploração de dados utilizando os conceitos de *Data Mart* em bancos de dados relacionais.

Os objetivos específicos do trabalho são:

- a) proporcionar flexibilidade na criação de consultas e relatórios utilizando a tecnologia de Cubo de Decisões, aplicados à *Data Mart* existentes;
- b) permitir a criação de consultas não estruturadas;
- c) tornar a ferramenta genérica e compatível para a utilização com qualquer base de dados operacional existente desenvolvida com banco de dados FireBird.

1.3 ESTRUTURA DO TRABALHO

Este trabalho está organizado em quatro capítulos descritos a seguir.

O primeiro capítulo apresenta a introdução e os objetivos pretendidos com a elaboração do trabalho.

O segundo capítulo descreve o sistema atual e a empresa que é o objeto do estudo de caso deste trabalho juntamente com a fundamentação teórica e conceitos utilizados.

O terceiro capítulo apresenta o desenvolvimento do trabalho, com sua especificação, técnicas e ferramentas mais relevantes ao desenvolvimento. Também apresenta a ferramenta desenvolvida, demonstrando todos os seus módulos com telas, códigos-fonte que servem para ilustrar as tecnologias por trás das especificações.

O quarto capítulo finaliza o trabalho, apresentando as conclusões e sugestões.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta etapa do trabalho é apresentado um breve histórico da empresa e uma síntese do sistema atual que servirá como base para o desenvolvimento da ferramenta proposta, bem como conceitos utilizados neste trabalho.

Também são abordados aspectos da ferramenta a ser desenvolvida e são apresentados trabalhos correlatos ao assunto.

2.1 SISTEMA ATUAL E EMPRESA

A Baptistetti Consultoria Empresarial S/C Ltda, fundada em 1999, começou seu trabalho no Oeste Catarinense e, rapidamente, expandiu suas atividades para os estados da Região Sul, São Paulo e Minas Gerais. Segundo Baptistetti (2006), a empresa atua no segmento de consultoria de custos e desenvolvimento de software baseada em modernos conceitos de gestão estratégica de resultado, que auxiliam no controle de processos existentes, agilizando assim o fluxo de informação interna e a coleta de dados para tomada de decisões. Isso leva aos gestores, informações antecipadas e precisas, sobre qual produto, processo ou atividade maximiza o seu lucro.

Os sistemas desenvolvidos pela empresa, agem em diversas áreas da empresa, envolvendo o gerenciamento e controle da produção, estoque, materiais, fornecendo análises e cálculos de custos como:

- a) lucratividade por unidade de negócio;
- b) lucratividade de clientes, representantes e regiões;
- c) *mix* de produção e vendas que maximiza o lucro;
- d) rentabilidade dos produtos de acordo com os fatores de restrições de produção (gargalos);
- e) produtividade e eficiência dos produtos.

Além de simulações como:

- a) preço de venda, custo do produto, margem de lucro;
- b) resultados futuros pelo critério de valor presente;
- c) pedidos de venda sob alteração de inúmeras variáveis;

d) impacto dos juros (custo de capital para a empresa) no custo do produto.

Seus funcionários desenvolvem atividades especializadas, utilizando tecnologia de ponta em nível mundial. A empresa conta com uma área de Tecnologia da Informação (TI), que desenvolve as ferramentas e sistemas aplicados aos seus clientes e com uma equipe de consultores e gestores de custos que coordenam e implantam esses sistemas, fazem treinamentos, análises de custos e auxiliam na tomada de decisões.

Os sistemas da empresa tem sua estrutura voltada ao ambiente Windows sendo desenvolvidos em linguagem de programação Delphi utilizando banco de dados relacional FireBird. Suas rotinas de cálculo geram e mantêm informações históricas de produção, estoque, vendas, custos e análises de resultados financeiros importantes para o processo de tomada de decisão. Mas existe uma série de questões que não podem ser respondidas por uma simples consulta, em virtude da própria limitação da sintaxe do *Structured Query Language* (SQL), como por exemplo montar consultas em forma de matriz com as informações sendo visualizadas de uma forma horizontal relacionadas com as diversas dimensões no eixo das colunas, ao invés de uma lista vertical de dados com colunas pré-definidas na consulta, ou quando há necessidade de totalizar quantidades em diferentes níveis de agregação.

Para obter melhores resultados nas necessidades do cliente em que a Baptistetti Consultoria presta seus serviços, os consultores de custos aplicam técnicas de análise de dados diferenciadas para cada cliente de acordo com cada problema. Isso gera uma demanda muito grande de desenvolvimento de relatórios e análises para a equipe de TI responsável por esse desenvolvimento, aumentando o custo e o tempo destinado a essa tarefa além do retorno ao cliente ser mais demorado.

Por isso é preciso criar alternativas para tirar mais proveito das informações geradas pelo sistema, fazendo com que a tomada de decisão se torne mais simples e rápida, bem como o desenvolvimento de novas consultas sejam feitas sem a intervenção de analistas e programadores, dando a liberdade do próprio consultor de custos montar consultas parametrizadas conforme a necessidade do cliente final.

2.2 CONCEITOS UTILIZADOS

A seguir serão apresentados conceitos como: Data Warehouse, Data Marte e OLAP, sendo os mais relevantes para a idealização deste trabalho.

2.2.1 Data Warehouse

Segundo Inmon (1997), um *Data Warehouse* (DW) pode ser definido como um banco de dados especializado, o qual integra e gerencia o fluxo de informações a partir dos bancos de dados corporativos e fontes de dados externas à empresa. Seu conjunto de dados é baseado em assuntos que apóiam as decisões gerenciais e é construído para que tais dados possam ser armazenados e acessados de forma que não sejam limitados por tabelas e linhas, estritamente relacionais. A função do DW é tornar as informações corporativas acessíveis para o seu entendimento, gerenciamento e utilização.

Conforme Oliveira (1998), um DW é um banco de dados que armazena dados sobre as operações da empresa extraídos de uma fonte única ou múltipla, e transformando-os em informações úteis, oferecendo um enfoque histórico, para permitir um suporte efetivo à decisão.

2.2.2 Data Mart

Segundo Italiano (2003), *Data Marts* (DM) existem para responder questões que as pessoas tem sobre os negócios. São bases de informações consolidadas, integradas e não voláteis, para apoiar os processos de tomada de decisões estratégicas, táticas e também operacionais de organizações.

Conforme Oliveira (1998), um *Data Mart* é um *Data Warehouse* de menor porte, construído para armazenar dados ligados a um determinado aspecto do negócio da empresa, e, aproximadamente 70 a 80% de todos os *Data Warehouse* correntemente em produção são, na verdade, *Data Marts*.

Os *Data Marts* normalmente tem características analíticas, ou seja, são utilizados por analistas de negócios, gerentes e executivos para analisar todas as possibilidades de cruzamento de dados, além da segmentação e outras análises disponíveis para se obter informações necessárias.

2.2.3 OLAP

O termo OLAP – *Online Analytical Processing* – refere-se a um conjunto de tecnologias voltadas ao acesso e análise *ad-hoc* de dados. Fornece para as organizações um método de acessar, visualizar e analisar dados corporativos com alta flexibilidade e performance.

Conforme O'Brien (2003), OLAP é a capacidade dos sistemas de apoio à decisão permitir aos gerentes e analistas examinarem e manipularem interativamente enormes quantidades de dados detalhados e consolidados a partir de múltiplas perspectivas.

De acordo com Inmon et. all. (1999), OLAP é um conjunto de funcionalidades que tenta facilitar a análise multidimensional. A análise multidimensional é a habilidade de manipular dados que tenham sido agregados em várias categorias ou dimensões. Tem o propósito de auxiliar o usuário a sintetizar informações empresariais através da visualização comparativa, personalizada e também por meio da análise de dados históricos e projetados.

O OLAP proporciona as condições de análise de dados *on-line* necessárias para responder às possíveis torrentes de perguntas dos analistas, gerentes e executivos. As ferramentas baseadas nesta tecnologia são as aplicações que os usuários finais têm acesso para extrair os dados de suas bases e construir os relatórios capazes de responder as questões gerenciais (DWBRASIL, 2005).

Sendo assim, o objetivo final de uma ferramenta OLAP é transformar dados em informações capazes de dar suporte a decisões gerenciais de forma amigável e flexível ao usuário final e em tempo hábil. Através de um estilo de navegação e pesquisa simples, usuários podem rapidamente analisar inúmeros cenários, gerar relatórios e descobrir tendências e fatos relevantes independente do tamanho, complexidade e fonte dos dados corporativos.

O processo interativo de criar, administrar, analisar e gerar relatórios sobre os dados são percebidos e manipulados como se estivessem armazenados em um *array* multidimensional (DATE, 2000).

A tecnologia OLAP envolve comparações entre períodos, percentual de diferença, médias, somas acumulativas como também funções estatísticas. O resultado deste tipo de análise é, através do comportamento de determinadas variáveis de tempo, descobrir tendências e com isso transformar os dados transacionais em informação estratégica.

As principais vantagens de uma ferramenta OLAP, referem-se as suas características

de permitir a visualização das informações de várias formas, conforme a necessidade de detalhamento.

Segundo Cielo (2000), as principais características OLAP são:

- a) *drill across*: é fazer com que duas ou mais tabelas de fato que compartilham dimensões sejam combinadas num único relatório permitindo ao usuário alternar um nível intermediário dentro da mesma dimensão. Por exemplo: a dimensão período é composta por ano, semestre, trimestre, mês e dia. O usuário estará executando um *drill across* quando ele passar diretamente para o semestre, mês ou dia;
- b) *drill down*: permite aumentar o nível de detalhe da informação, diminuindo o grau de granularidade. Fazer um *drill down* significa obter mais informações sobre os dados que estão sendo apresentados, seja descendo numa hierarquia ou adicionando dimensões que complementem a análise dos dados;
- c) *drill up*: ao contrário do *drill down*, possibilita aumentar o grau de granularidade, diminuindo o detalhamento da informação;
- d) *drill through*: ocorre quando o usuário passa de uma informação contida em uma dimensão para outra;
- e) *slice and dice*: é uma das principais características de uma ferramenta OLAP. Corresponde a técnica de mudar a ordem das dimensões mudando assim a orientação segundo a qual os dados são visualizados. Altera linhas por colunas de maneira a facilitar a compreensão dos usuários.

2.2.3.1 Arquitetura de ferramentas OLAP

Existem várias maneiras de se classificar uma ferramenta OLAP quanto à sua arquitetura. A mais relevante para este trabalho diz respeito ao acesso e localização dos dados a serem utilizados na análise. Segundo GSI (2004), esse tipo de ferramenta classifica-se em: *Multi-dimensional OLAP (MOLAP)* e *Relational OLAP (ROLAP)*.

Toda ferramenta OLAP provê uma visão multidimensional dos dados, não importando de onde os extrai. Ferramentas MOLAP acessam bancos de dados multidimensionais, ou seja, ao invés de acessarem tabelas relacionais, essas ferramentas extraem informações de estruturas de dados pré-calculados, geralmente referenciados como “cubos”, que contém todas as respostas possíveis dentro de um determinado âmbito.

Bancos de dados multidimensionais apresentam limitações quanto a sua escalabilidade,

já que mudanças no modelo dimensional requerem a re-organização do banco de dados. A criação de índices, fundamentais para garantir a performance do sistema, acabam também restringindo o tamanho dos bancos de dados utilizados. Outra limitação apresentada segundo GSI (2004), é a de não suportar a criação *ad-hoc* de visões multidimensionais.

As ferramentas MOLAP se aplicam melhor no caso onde o usuário já tem definidas as dimensões e os conceitos com os quais ele quer trabalhar, ou seja, tem o escopo de sua análise de dados já definida.

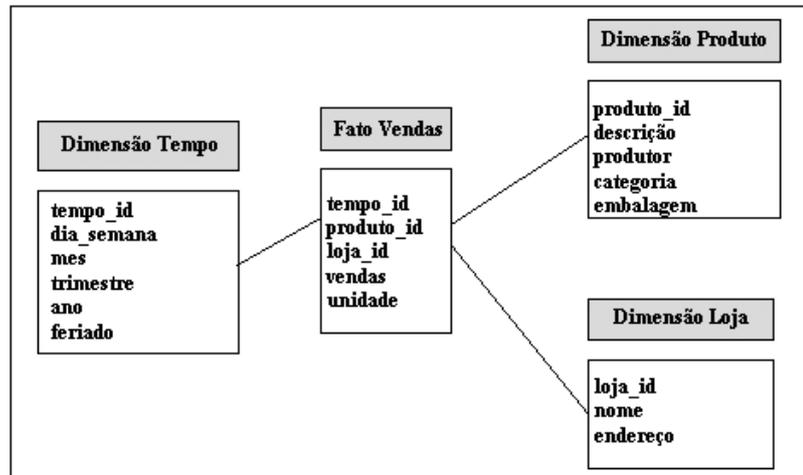
Conforme Leitão (2000), ferramentas ROLAP não utilizam os cubos pré-calculados. À medida que o usuário monta sua consulta em uma *interface* gráfica, a ferramenta acessa os metadados ou qualquer outros recursos que possua, para gerar um consulta em SQL e fazer com que ela seja executada em uma base de dados relacional, tal como Firebird, MySQL, Oracle entre outras.

A característica principal das ferramentas OLAP é o poder de fazer qualquer consulta, visto que não está limitada ao conteúdo de um “cubo” e a capacidade de navegar nos dados até atingir o nível de detalhe mais baixo, ou seja, de menor granularidade.

As ferramentas ROLAP atendem melhor o usuários que não têm um escopo de análise bem definido. Uma desvantagem desta arquitetura é o seu tempo de resposta, por vezes muito longo.

Conforme GSI (2004) a tecnologia OLAP pode ser aplicada às estruturas de banco de dados desenvolvidas com esquema estrela e *snowflake*.

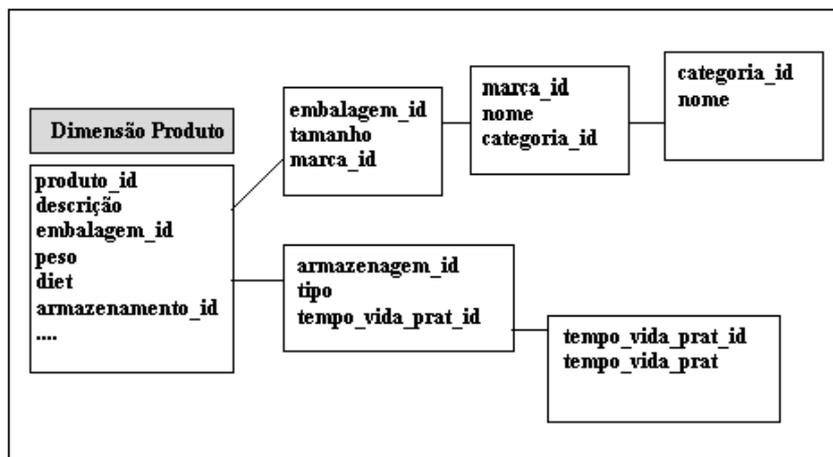
O esquema estrela (*Star Schema*) é a representação do modelo dimensional em bancos de dados relacionais e tem como característica uma tabela dominante no seu centro chamada de fato (*fact table*) contendo múltiplos relacionamentos para as outras tabelas. As outras tabelas chamadas de dimensão (*dimension table*), possuem um único relacionamento para a tabela central. A Figura 1 ilustra o esquema estrela.



Fonte: GSI (2004).

Figura 1 – Esquema estrela (*Star Schema*)

No esquema *snowflake* (*Snowflake Schema*), as tabelas de dimensão descrevem os fatos. É feita a normalização das tabelas eliminando a redundância e diminuindo o espaço em disco. Assim uma dimensão contém uma ou mais hierarquias que podem ser decompostas em tabelas separadas. A Figura 2 ilustra uma estrutura *snowflake*.



Fonte: GSI (2004).

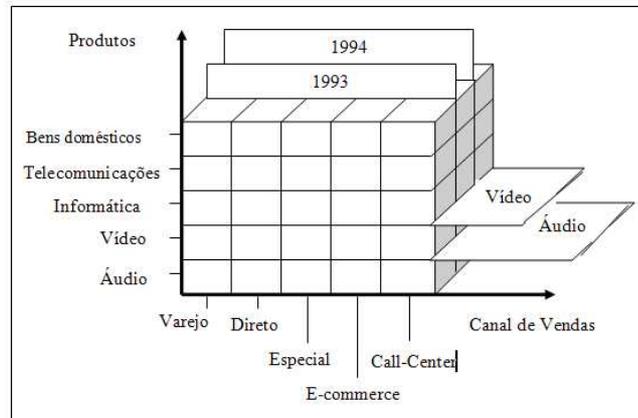
Figura 2 – Esquema *snowflake* (*Snowflake Schema*)

2.2.4 Cubo de Decisão

Decision Cube (Cubo de Decisão), refere-se a um conjunto de componentes de suporte a decisões que podem ser utilizados para cruzar tabelas de um banco de dados, gerando visões através de planilhas ou gráficos. Envolve o cálculo, quando da carga dos dados, de dados que o usuário virá a solicitar, mas que podem ser derivados de outros dados. Quando o usuário solicita os dados, estes já estão calculados, agregados em um Cubo de Decisão (INMON, et.

all. 1999, p. 193).

A Figura 3 ilustra como o cubo de decisão pode ser usado para visualizar a análise multidimensional dos dados de vendas nas três dimensões selecionadas.



Fonte: Inmon (1999, p. 194).

Figura 3 – Análise multidimensional dos dados – Representação do cubo de decisão

Conforme Cielo (2000), os cubos são massas de dados que retornam das consultas feitas ao banco de dados e podem ser manipulados e visualizados por inúmeros ângulos.

As dimensões, podem ser associadas à figura geométrica de um cubo, sendo assim, as dimensões são os elementos que caracterizam os fatos, dando diferentes perspectivas ao negócio. São associadas à quantidade de opções de visualização dos dados que o usuário poderá obter manipulando o cubo de decisão.

2.3 TRABALHOS CORRELATOS

No que diz respeito à utilização de tecnologias de Cubo de Decisões, destacam-se algumas ferramentas comerciais como o Oracle Discoverer, que faz parte das ferramentas *Decision Support Systems* (DSS) da Oracle e o Analysis Services que compõe as Soluções de *Business Intelligence* da Microsoft.

Segundo Oracle (2005), o Oracle Discoverer se destaca não como sendo apenas uma ferramenta OLAP, mas pode funcionar também como uma ferramenta de consulta a qualquer banco de dados relacional criando consultas *ad-hoc*, livrando usuários de ter de conhecer SQL, trabalha com uma visão multidimensional dos dados tendo como enfoque a construção de metadados baseado no que já está definido no banco de dados relacional. Esses metadados, são armazenados em um banco de dados relacional único mas podem ser divididos em áreas

de negócio (*business areas*), ao qual o sistema pode permitir ou restringir o acesso do usuário mediante configurações.

Segundo Microsoft (2005), Analysis Services em conjunto com o banco de dados SQL Server disponibiliza recursos para que os analistas de bases de dados, técnicos de desenvolvimento e outros técnicos construam bases de dados e cubos OLAP rapidamente e com pouca necessidade de formação adicional.

As duas ferramentas citadas tem parcela significativa no mercado de ferramentas para análise de dados, mas trabalham com bancos de dados proprietários, o que torna a aquisição do software cara e utilizável apenas por empresas com elevado poder financeiro.

Outros trabalhos já foram desenvolvidos na área de aplicação da tecnologia de Cubo de Decisões, entre eles estão: Martins (2005) desenvolveu uma ferramenta utilizando a tecnologia OLAP voltada à administração de materiais em indústrias gráficas.

Corradi (2002) propôs um sistema de informação com análises pré-definidas utilizando o cubo de decisão aplicado à comercialização de planos de previdência privada.

O trabalho aqui proposto, distingui-se dos anteriores por ser focado não no desenvolvimento de relatórios específicos e pré-definidos pelo usuário e desenvolvedor, mas sim na ferramenta genérica que ajudará a criar estas análises rapidamente e na hora que se tornarem necessárias, com qualquer estrutura de base de dados existente.

3 DESENVOLVIMENTO DO TRABALHO

A seguir serão apresentadas as tecnologias e ferramentas utilizadas para o desenvolvimento e aplicação deste trabalho.

Para o desenvolvimento foram utilizados conceitos da linguagem *Unified Modeling Language* (UML) aplicadas à ferramenta Enterprise Architect 6.5, além do banco de dados Firebird e a ferramenta de desenvolvimento Delphi, as quais serão descritas no item Ferramentas e Tecnologias e Utilizadas.

3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Este trabalho desenvolveu uma ferramenta *Desktop* OLAP genérica que possibilita os desenvolvedores, analistas e usuários finais, criarem consultas utilizando tecnologia de Cubo de Decisões, para a obtenção rápida de informações que atendam suas necessidades na hora e no momento necessário.

A ferramenta foi desenvolvida para ser aplicada a qualquer base de dados criada com o gerenciador de banco de dados FireBird, tornando possível sua utilização não apenas nos sistemas deste estudo, mas sim a qualquer estrutura de dados existente. Isso será feito através da definição prévia pelos analistas e desenvolvedores do banco de dados, de um dicionário de dados, onde conterà informações de onde e como localizar as informações dentro do banco de dados.

Com o dicionário de dados criado, o usuário passa a ter, juntamente com um assistente de criação de consultas de cubo de decisão, condições de construir consultas personalizadas utilizando os atributos do banco de dados, salvando e reutilizando as consultas posteriormente. O próprio assistente citado acima, deixa transparente para o usuário final a estrutura e relacionamentos possíveis entre os dados no banco de dados fazendo com que não haja a necessidade de conhecimentos na linguagem SQL por parte do usuário.

Sendo assim não há carga de dados para tabelas secundárias, pois a ferramenta utiliza os conceitos de *Data Mart* explorando logicamente os relacionamentos definidos no banco de dados para a obtenção das informações diretamente das tabelas existentes.

No assistente de criação de Cubo de Decisões, o usuário seleciona as tabelas de fato

dentre as definidas no dicionário de dados, bem como indica as dimensões, filtros e agregações iniciais da consulta. A ferramenta utiliza as configurações previamente definidas no dicionário de dados e também as tabelas de configuração internas do banco de dados FireBird para montar logicamente os relacionamentos entre os dados selecionados pelo usuário e gera assim as instruções de consulta SQL ao banco de dados.

As consultas geradas são aplicadas à tecnologia OLAP permitindo a visualização multidimensional dos dados. A *interface* de consulta fornece ampla liberdade de combinação dos elementos disponíveis, desta forma o usuário pode analisar os dados de diferentes ângulos, como se estivesse com um cubo em suas mãos e fosse alterando de posição para visualizar um outro lado. Todos os parâmetros da consulta podem ser alterados em tempo de execução bem como agregações e filtros.

Espera-se com isso, reduzir o tempo de resposta ao cliente, pois as consultas serão criadas tanto pela equipe de desenvolvimento como, *in loco* pelo consultor de custos ou pelo próprio usuário final. Outro fator esperado é a diminuição do tempo e custo gastos pela equipe de TI da Baptistetti Consultoria no desenvolvimento e manutenção desses relatórios já que até então eram criados formulários específicos para cada consulta solicitada. Além disso, a intenção é fornecer novos recursos e formas de análise de dados utilizando-se dos conceitos e recursos de tecnologias OLAP citados no item 2.2.3 deste trabalho.

O Quadro 1 apresenta os requisitos funcionais da ferramenta e sua rastreabilidade, ou seja, vinculação com o(s) caso(s) de uso associado(s).

Requisitos Funcionais	Caso de Uso
RF01: A ferramenta deve controlar o acesso através de usuário e senha.	UC01
RF02: A ferramenta deve permitir ao usuário administrador criar e manter um dicionário de dados, traduzindo nomes da base de dados existente para nomes significativos para o usuário final.	UC02
RF03: A ferramenta deve permitir ao administrador cadastrar e manter os usuários que terão acesso à ferramenta.	UC03
RF04: A ferramenta deve permitir que o usuário administrador identifique quais tabelas e campos cada usuário poderá acessar.	UC04
RF05: A ferramenta deve permitir que o usuário crie uma nova consulta selecionando tabelas, atributos e configurando as dimensões do cubo de	UC05

decisão, bem como seus filtros e agregações.	
RF06: A ferramenta deve permitir que o usuário grave consultas criadas.	UC06
RF07: A ferramenta deve permitir ao usuário visualizar o código SQL gerado.	UC07
RF08: A ferramenta deve permitir que o usuário abra uma consulta existente.	UC08
RF09: A ferramenta deve permitir ao usuário visualizar as informações, no formato de Cubo de Decisões, possibilitando a reorganização das dimensões, ordenação e filtro de valores e exploração detalhada das informações.	UC09
RF10: A ferramenta deve permitir ao usuário imprimir as informações geradas na consulta.	UC10
RF11: A ferramenta deve permitir ao usuário exportar as informações geradas na consulta para arquivos texto e arquivos do Microsoft Excel.	UC11
RF12: A ferramenta deve permitir ao usuário gerar gráficos dos dados explorados.	UC12

Quadro 1 – Requisitos funcionais da ferramenta.

O Quadro 2 lista os requisitos não funcionais da ferramenta.

Requisitos Não Funcionais
RNF01: A ferramenta deverá ser compatível para funcionar com bancos de dados FireBird 1.0 ou superior.
RNF02: A ferramenta deverá ser construída utilizando a linguagem de programação Delphi 7.
RNF03: A ferramenta deverá permitir que os usuários tenham acesso através dos perfis: <ul style="list-style-type: none"> a) administrador: pode utilizar todas as funções da ferramenta; b) usuário: pode abrir consultas existentes ou utilizar os recursos de criação de consultas de cubo de decisões para criar e gravar consultas personalizadas.

Quadro 2 – Requisitos não funcionais da ferramenta.

3.2 ESPECIFICAÇÃO

Neste item são apresentadas as atividades desempenhadas na fase de especificação como: Diagrama de Casos de Uso, Diagrama de Classe, Diagrama de Seqüência, Modelo de Entidade Relacionamento (MER) e Dicionário de Dados desenvolvidos com base em *Unified Modeling Language* (UML). Nos itens a seguir estas atividades são apresentadas detalhadamente.

O MER apresentado neste trabalho contempla a estrutura das tabelas que a ferramenta proposta necessita para o seu funcionamento. Não serão demonstradas as tabelas que servirão como fontes de dados para as consultas, pois a ferramenta sendo genérica pode funcionar com banco de dados de qualquer aplicação sem levar em conta sua natureza, finalidade, tamanho ou estrutura operacional do sistema envolvido. Estendendo-se assim o seu funcionamento, desde a um banco de dados de estrutura simples, até bases de dados operacionais da empresa ou *Data Marts* específicos.

3.2.1 Diagrama de Caso de Uso

Um caso de uso é a especificação de uma seqüência de interações entre um sistema e os agentes externos que utilizam este sistema. Um caso de uso deve definir o uso de uma parte da funcionalidade de um sistema sem revelar a estrutura e o comportamento internos desse sistema (BEZERRA, 2002, p. 46).

Furlan (1998), descreve os diagramas de caso de uso como uma descrição da visão externa do sistema e suas interações com o mundo através de atores. Esse diagrama representa uma visão abrangente da funcionalidade intencional ocasionada pela interação de um tipo de requisição de usuário.

No trabalho realizado, os atores foram definidos como sendo os administradores do banco de dados e usuários finais, ou seja, quem utilizará as funcionalidades implementadas na ferramenta de consulta. O diagrama de caso de uso apresentado na Figura 4 ilustra estes usuários e as ações realizadas por eles.

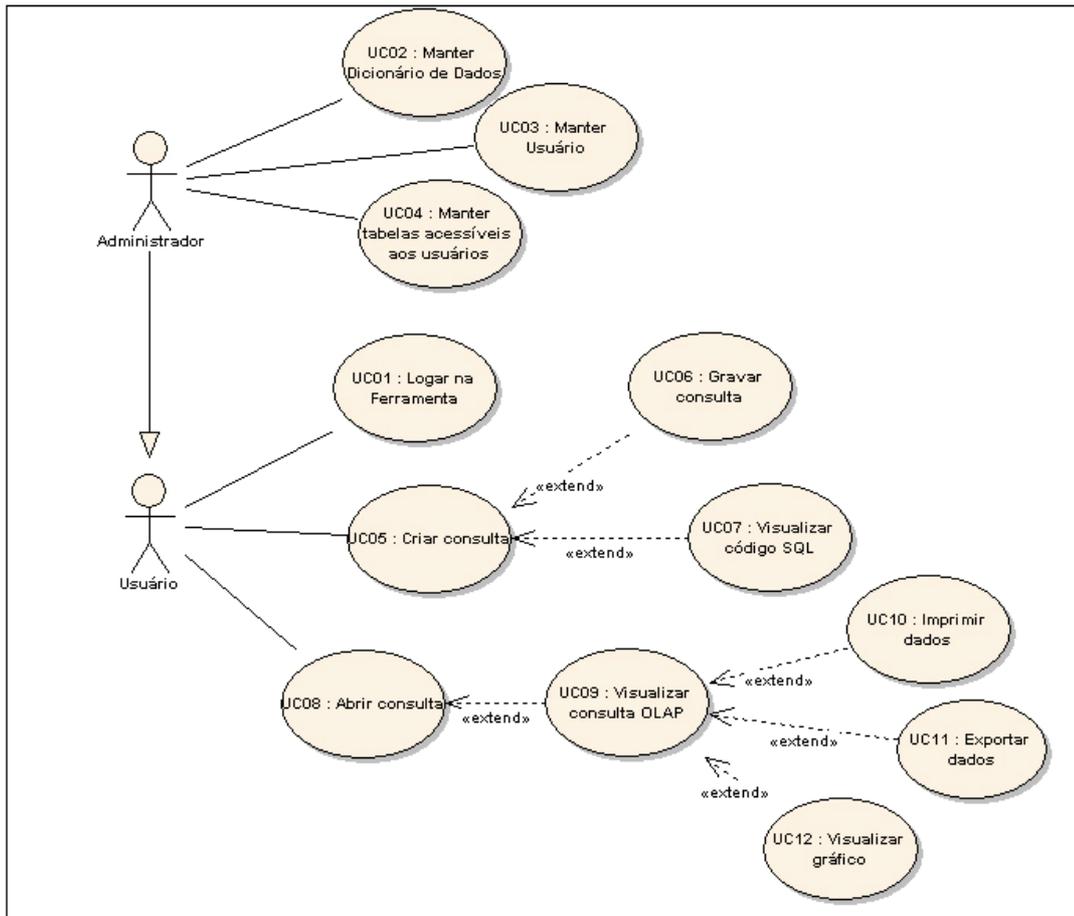
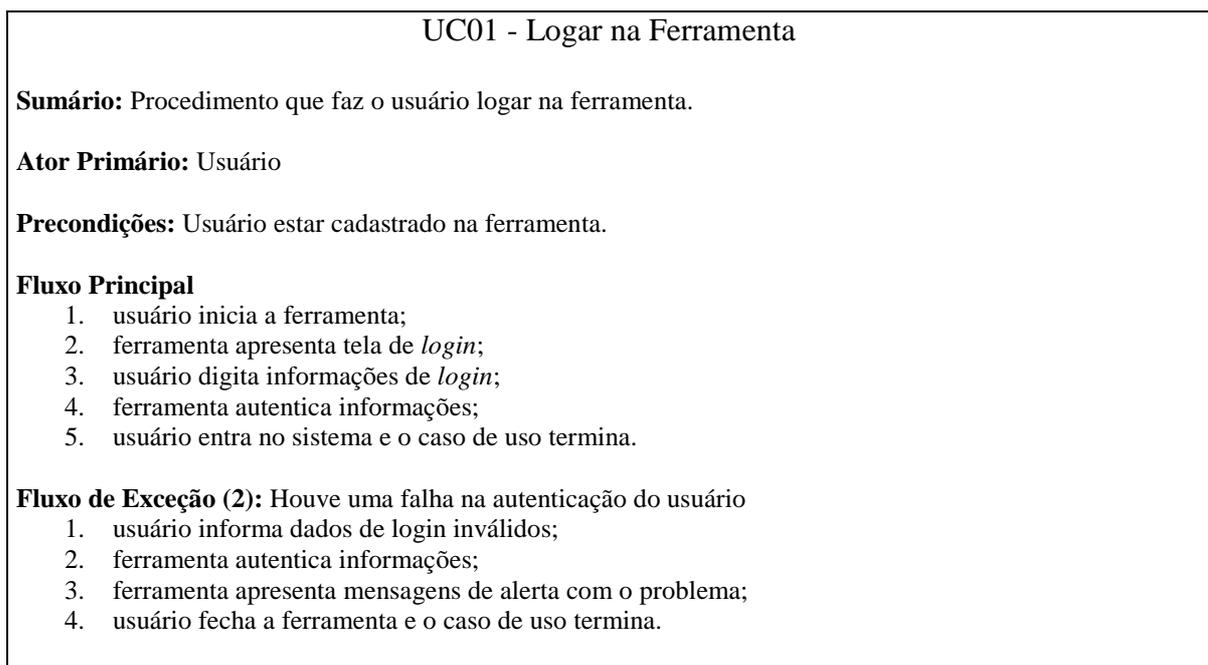


Figura 4 – Diagrama de Caso de Uso da ferramenta

No Quadro 3 mostra-se a ação logar na ferramenta realizada pelo ator usuário.



Quadro 3 – Descrição do Use Case – Logar na Ferramenta.

O Quadro 4 especifica a ação Manter Dicionário de Dados do caso de uso do ator

administrador, demonstrando os passos que o mesmo deve executar para realizar tal ação dentro da ferramenta.

UC02 - Manter Dicionário de Dados
<p>Sumário: Neste procedimento o Administrador acessa o Dicionário de Dados para configurar a estrutura de tabelas, campos e informações que serão utilizada na criação das consultas.</p>
<p>Ator Primário: Administrador</p>
<p>Precondições: Administrador autenticado na ferramenta.</p>
<p>Fluxo Principal</p> <ol style="list-style-type: none"> 1. administrador entra na ferramenta com a sua senha própria; 2. administrador seleciona o menu do cadastro de Dicionário de Dados; 3. ferramenta se comunica com o banco de dados e obtém sua estrutura de tabelas e campos; 4. ferramenta apresenta a estrutura obtida ao Administrador em forma de grade de dados; 5. administrador seleciona as tabelas que estarão utilizadas para a criação de consultas; 6. ferramenta mostra os campos de cada tabela selecionada; 7. administrador informa os atributos de cada campo; 8. administrador salva as informações; 9. administrador sai do cadastro e o caso de uso termina.

Quadro 4 – Descrição do Use Case – Manter Dicionário de Dados.

O Quadro 5 especifica a ação Manter Usuários do caso de uso do ator administrador, demonstrando os passos que o mesmo deve executar para realizar tal ação dentro da ferramenta.

UC03 - Manter Usuários
<p>Sumário: Este procedimento é responsável pelo cadastramento de novos usuários da ferramenta. Esta funcionalidade é composta pelas opções: Novo, Gravar, Excluir e Cancelar. Um usuário é composto por um <i>login</i>, senha, nome, definição de perfil e um conjunto de permissões.</p>
<p>Ator Primário: Administrador</p>
<p>Precondições: Administrador autenticado na ferramenta.</p>
<p>Fluxo Principal: Cadastrar Usuário</p> <ol style="list-style-type: none"> 1. administrador acessa o Cadastro de Usuário; 2. administrador seleciona opção Novo; 3. administrador informa dados do novo usuário; 4. administrador seleciona opção Gravar; 5. administrador sai do cadastro e o caso de uso termina.
<p>Fluxo de Exceção (2): Alterar Usuário</p> <ol style="list-style-type: none"> 1. administrador acessa o Cadastro de Usuário; 2. administrador seleciona o usuário a ser alterado; 3. administrador altera dados do usuário selecionado; 4. administrador seleciona opção Gravar; 5. administrador sai do cadastro e o caso de uso termina.
<p>Fluxo de Exceção (3): Apagar Usuário</p> <ol style="list-style-type: none"> 1. administrador acessa o Cadastro de Usuário; 2. administrador seleciona o usuário a ser alterado;

3. administrador seleciona opção Excluir;
4. administrador confirma a exclusão do usuário;
5. administrador sai do cadastro e o caso de uso termina.

Quadro 5 – Descrição do Use Case – Manter Usuários.

O Quadro 6 mostra a ação Manter tabelas acessíveis ao usuário realizada pelo ator administrador.

UC04 - Manter tabelas acessíveis ao usuário

Sumário: Este procedimento é responsável por definir quais tabelas de fonte de dados cada usuário poderá acessar na construção de consultas.

Ator Primário: Administrador

Precondições: Administrador autenticado na ferramenta. Usuários já devem estar cadastrados na ferramenta.

Fluxo Principal

1. administrador acessa o Cadastro de Usuário;
2. administrador seleciona opção “Configurar permissões de acesso à tabelas”;
3. ferramenta mostra todas as tabelas de fontes de dados configuradas no dicionário de dados;
4. administrador seleciona as tabelas que o usuário selecionado poderá acessar;
5. administrador sai do cadastro e o caso de uso termina.

Quadro 6 – Descrição do Use Case – Manter tabelas acessíveis ao usuário.

O Quadro 7 mostra a ação Criar Consulta realizada pelo ator usuário.

UC05 - Criar Consulta

Sumário: Neste procedimento a ferramenta mostra um assistente passo a passo que permite ao usuário criar suas consultas não estruturadas seguindo uma seqüência lógica de utilização. A ferramenta por sua vez interage com o dicionário de dados e com a estrutura do banco de dados deixando a interligação de tabelas, e geração de código de consulta ao banco de dados transparente para o usuário.

Ator Primário: Usuário

Precondições: Usuário autenticado na ferramenta. Dicionário de Dados estar configurado. Usuário ter permissões de acesso as tabelas do banco de dados.

Fluxo Principal

1. usuário acessa a ferramenta;
2. usuário seleciona opção de Criar Nova Consulta;
3. ferramenta mostra informações da estrutura de banco de dados que o usuário tem permissão de acessar;
4. usuário seleciona tabelas e campos da consulta;
5. usuário informa dimensões da nova consulta;
6. usuário informa agregações para a nova consulta;
7. usuário informa filtros da nova consulta;
8. usuário informa título, subtítulo e descrições para a consulta;
9. ferramenta gera instruções SQL de consulta ao banco de dados;
10. usuário testa a consulta e o caso de uso termina.

Fluxo de Exceção (2): Erro na geração de instruções consulta ao banco de dados

1. ferramenta mostra mensagem de possíveis erro;
2. usuário seleciona opção Anterior para retornar aos passos anteriores e identificar possíveis erros.

Quadro 7 – Descrição do Use Case – Criar Consulta.

O Quadro 8 mostra a ação Gravar Consulta realizada pelo ator usuário.

UC06 - Gravar Consulta
<p>Sumário: Este procedimento é responsável por gravar no banco de dados a consulta criada pelo usuário juntamente com suas configurações.</p>
<p>Ator Primário: Usuário</p>
<p>Precondições: Consulta estar criada.</p>
<p>Fluxo Principal</p> <ol style="list-style-type: none"> 1. usuário seleciona opção Salvar; 2. ferramenta mostra lista de menus já existentes; 3. usuário seleciona um dos menus existentes; 4. usuário informa o número identificador da consulta; 5. usuário seleciona a opção Salvar Consulta; 6. ferramenta grava a consulta no banco de dados; 7. usuário sai da tela de salvamento e o caso de uso termina.

Quadro 8 – Descrição do Use Case – Gravar Consulta.

O Quadro 9 mostra-se a ação Visualizar código SQL realizada pelo ator usuário.

UC07 - Visualizar código SQL
<p>Sumário: Este procedimento permite visualizar as instruções SQL de consulta ao banco de dados geradas pelo assistente de criação de consultas.</p>
<p>Ator Primário: Usuário</p>
<p>Precondições: Consulta estar criada.</p>
<p>Fluxo Principal</p> <ol style="list-style-type: none"> 1. usuário seleciona opção Visualizar Código SQL; 2. ferramenta mostra código SQL gerado; 3. usuário sai da tela de visualização e o caso de uso termina.

Quadro 9 – Descrição do Use Case – Visualizar código SQL.

O Quadro 10 mostra a ação Abrir Consulta realizada pelo ator usuário.

UC08 - Abrir Consulta
<p>Sumário: Este procedimento lê as informações de configurações da consulta selecionada, criando dinamicamente uma tela de passagem de parâmetros para os filtros existentes.</p>
<p>Ator Primário: Usuário</p>
<p>Precondições: Usuário autenticado na ferramenta.</p>
<p>Fluxo Principal</p> <ol style="list-style-type: none"> 1. usuário inicia a ferramenta; 2. ferramenta mostra lista de consultas existentes; 3. usuário seleciona uma consulta; 4. usuário seleciona opção Abrir Consulta; 5. ferramenta lê configurações de consulta; 6. ferramenta cria tela de passagem de parâmetros criando dinamicamente componentes visuais

- vinculados aos filtros existentes na consulta;
- 7. usuário digita informações de filtros da consulta;
- 8. usuário seleciona a opção Executar e o caso de uso termina.

Quadro 10 – Descrição do Use Case – Abrir Consulta.

O Quadro 11 mostra a ação Visualizar Consulta OLAP realizada pelo ator usuário.

UC09 - Visualizar Consulta OLAP

Sumário: Este procedimento cria dinamicamente o Cubo de Decisão de acordo com os parâmetros definidos na consulta e mostra ao usuário uma tela com funcionalidades de manipulação das informações em formato multidimensional.

Ator Primário: Usuário

Precondições: Consulta estar aberta. Informações de parâmetros estarem digitadas.

Fluxo Principal

1. usuário executa consulta;
2. ferramenta lê configurações da consulta junto ao banco de dados;
3. ferramenta cria tela de visualização de consulta;
4. ferramenta cria dinamicamente o Cubo de Decisões com base nos parâmetros gravados na consulta;
5. usuário visualiza Cubo de Decisões;
6. usuário manipula dimensões, filtros, agregações das informações e o caso de uso termina.

Quadro 11 – Descrição do Use Case – Visualizar Consulta OLAP.

O Quadro 12 mostra-se a ação Imprimir Dados realizada pelo ator usuário.

UC10 - Imprimir Dados

Sumário: Este procedimento permite visualizar e imprimir as informações do Cubo de Decisão

Ator Primário: Usuário

Precondições: Consulta OLAP ter sido gerada.

Fluxo Principal

1. usuário seleciona opção Imprimir;
2. ferramenta mostra tipos de impressão disponíveis;
3. usuário seleciona tipo de impressão;
4. ferramenta gera impressão das informações do Cubo de Decisão;
5. usuário visualiza Impressão;
6. usuário seleciona opção imprimir e o caso de uso termina.

Quadro 12 – Descrição do Use Case – Imprimir Dados.

No Quadro 13 mostra-se a ação Exportar Dados realizada pelo ator usuário.

UC11 - Exportar Dados

Sumário: Este procedimento permite exportar dados do Cubo de Decisão para formatos de arquivos utilizados por aplicativos como Microsoft Word , Microsoft Excel ou no formato HTML.

Ator Primário: Usuário

Precondições: Consulta OLAP ter sido gerada.

Fluxo Principal

1. usuário seleciona opção Exportar;
2. ferramenta mostra tipos de exportação de dados disponíveis;
3. usuário seleciona tipo de exportação de dados;
4. ferramenta mostra tela para salvamento do arquivo de exportação de dados;
5. usuário informa nome e local do arquivo de exportação;
6. ferramenta gera exportação das informações no formato escolhido pelo usuário e o caso de uso termina.

Quadro 13 – Descrição do Use Case – Exportar Dados.

O Quadro 14 mostra a ação Visualizar Gráfico realizada pelo ator usuário.

UC12 - Visualizar Gráfico

Sumário: Neste procedimento são gerados gráficos das informações visualizadas no Cubo de Decisão.

Ator Primário: Usuário

Precondições: Consulta OLAP ter sido gerada.

Fluxo Principal

1. usuário seleciona a opção Gráficos;
2. ferramenta gera gráficos das informações visualizadas no cubo de decisão;
3. usuário visualiza gráfico;
4. usuário altera parâmetros de visualização do gráfico e o caso de uso termina.

Quadro 14 – Descrição do Use Case – Visualizar Gráfico.

3.2.2 Diagrama de Classe

Scott (2000, p. 57) descreve o diagrama de classes como sendo parte central do desenvolvimento de um software. Afirma também que o diagrama de classes descreve os tipos de objetos declarativos do modelo, como classes, tipos e seus respectivos conteúdos bem como vários tipos de relacionamento estático que existem entre eles. A Figura 5 demonstra o diagrama de classes da ferramenta proposta.

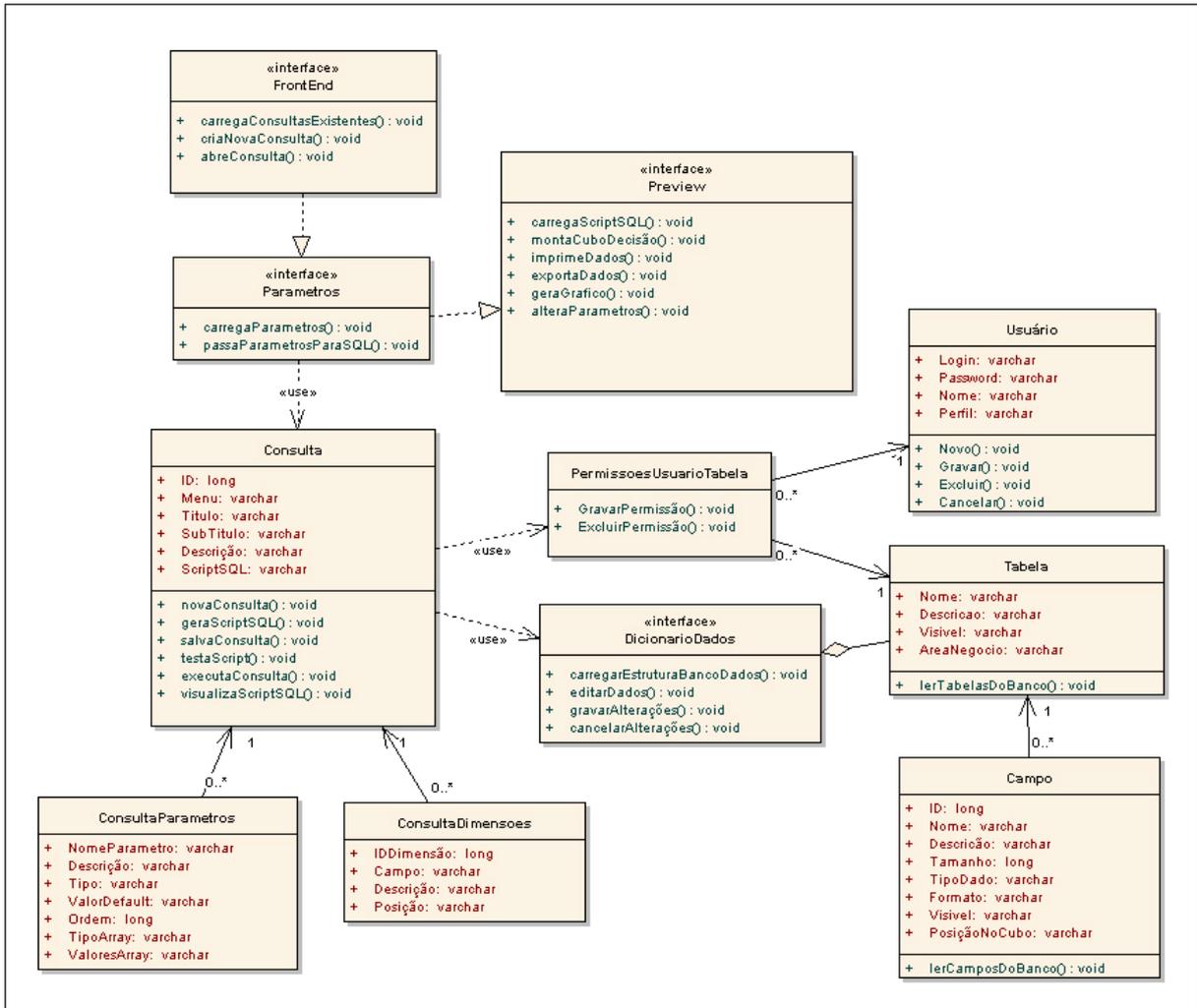


Figura 5 – Diagrama de Classes

A classe FrontEnd fornece a *interface* inicial da ferramenta bem como as funções principais de manutenção de cadastros, criação e utilização de consultas.

A classe Parametros é utilizada para criar em tempo de execução os parâmetros definidos na consulta associados a um componente visual da *interface* deixando a entrada de valores mais rápida e fácil para o usuário.

A classe Preview é utilizada para a construção do cubo de decisões de acordo com os parâmetros da consulta além de prover as funcionalidades de manipulação da informação no formato multidimensional.

A funcionalidade da classe Consulta é criar o assistente que o usuário irá utilizar para a geração de novas consultas.

Na classe ConsultaParametros é feita a gravação dos parâmetros da consulta.

A classe ConsultaDimensões é utilizada na gravação das configurações de dimensões que serão mostradas na visualização do cubo de decisão.

Na classe Usuário são armazenadas as informações do usuário que são utilizadas no

controle de acesso das funcionalidades e informações da ferramenta.

As classes Tabela e Campo mantém as informações de estrutura do banco de dados utilizado.

A funcionalidade da classe DicionárioDados é registrar as configurações e parametrizações das entidades e campos utilizados para a criação de consultas.

A classe PermissoesUsuarioTabela é utilizado para definir o acesso de cada usuário às informações do banco de dados.

3.2.3 Diagrama de Seqüência

Medeiros (2004, p. 147) afirma que o diagrama de seqüência é um dos diagramas de interação existentes que pode ser usado para mostrar a evolução de uma dada situação em determinado momento do software. Mostrar uma dada colaboração entre duas ou mais classes e pode, também, ser usado para mostra a tradução de um Caso de Uso desde a interação com o usuário até a finalização daquele dado processo.

São apresentados abaixo os diagramas de seqüência mais relevantes ao trabalho desenvolvido. Eles demonstram visualmente a seqüência das ações que o software tomará frente a interação de um ator em abrir uma consulta (Figura 6) e criar uma nova consulta (Figura 7).

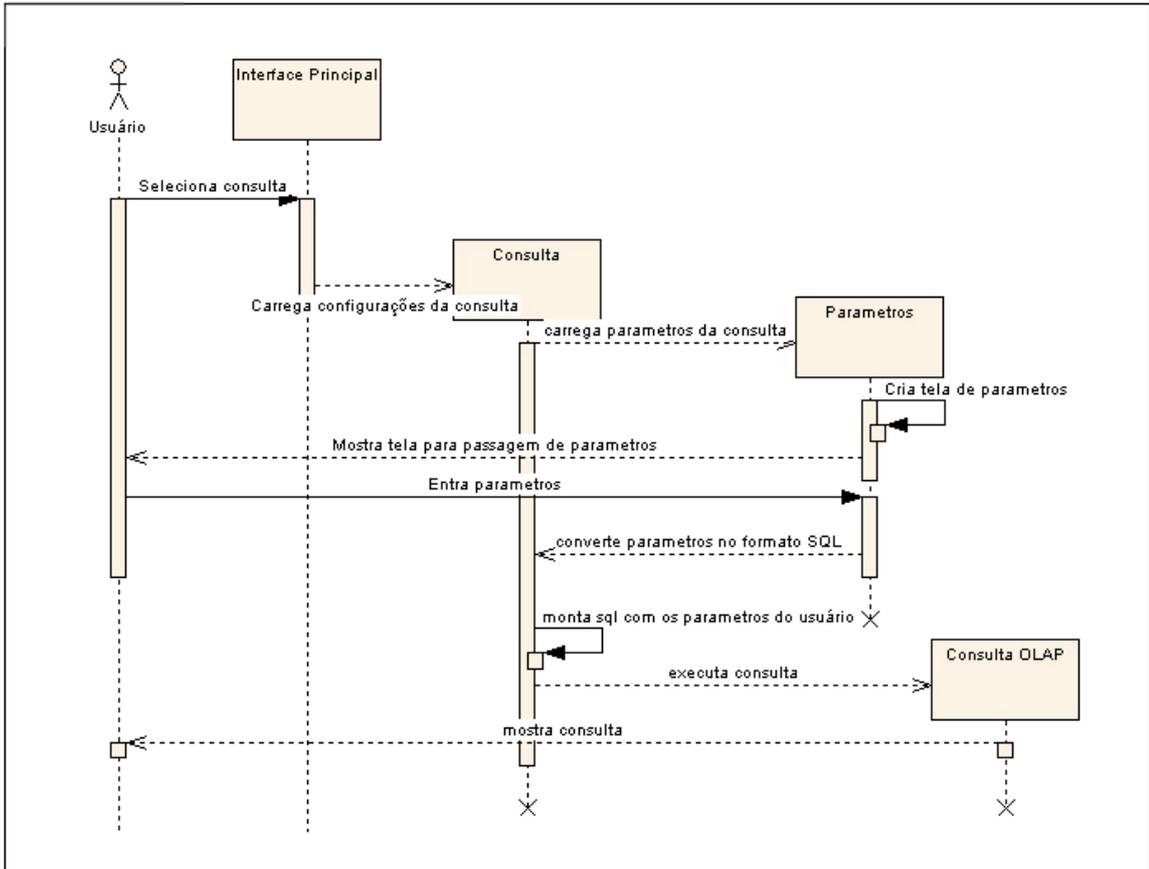


Figura 6 – Diagrama de Seqüência – Abrir Consulta

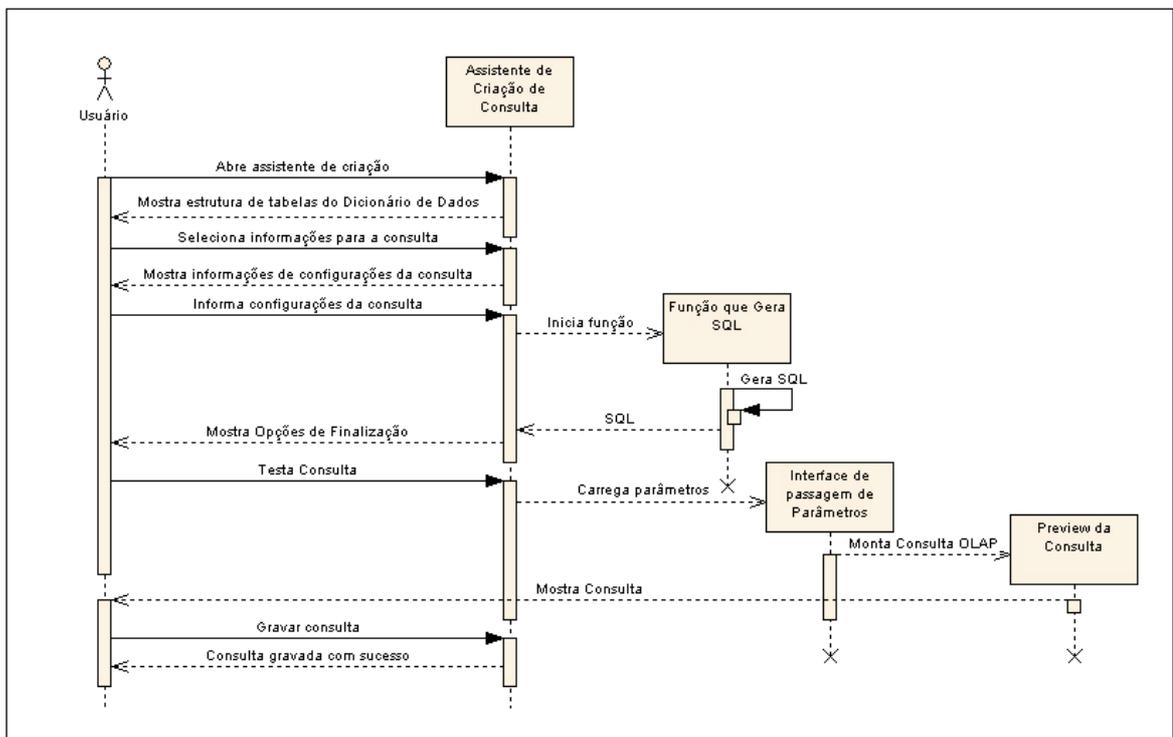


Figura 7 – Diagrama de Seqüência – Criar Consulta

3.2.4 Modelo Entidade Relacionamento – MER

De acordo com Yourdon (1990), o modelo de entidades e relacionamentos pode ser definido como um modelo que descreve a diagramação dos dados armazenados de um sistema em alto nível de abstração. Os principais componentes de um modelo de entidades e relacionamentos são:

- a) entidades: são representados por um retângulo em um diagrama de entidades-relacionamentos. Ele representa uma coleção ou um conjunto de objetos, entidades do mundo real;
- b) relacionamentos: são interligações feitas entre os objetos e representam um conjunto de conexões entre objetos descritos por um losango;
- c) notação alternativa para relacionamentos: descreve os tipos de relacionamentos existentes entre os objetos, podendo ser um-para-um, um-para-muitos ou muitos-para-muitos. O relacionamento mostra tanto a cardinalidade como a ordinalidade. A notação de uso é descrita através de uma seta com ponta dupla para indicar um relacionamento um-para-muitos, enquanto a seta de ponta singela é empregada para indicar relacionamentos um-para-um entre objetos.

A Figura 8 a seguir representa o Modelo Entidade Relacionamento das tabelas de configurações utilizadas pela ferramenta para guardar informações pertinentes ao seu funcionamento.

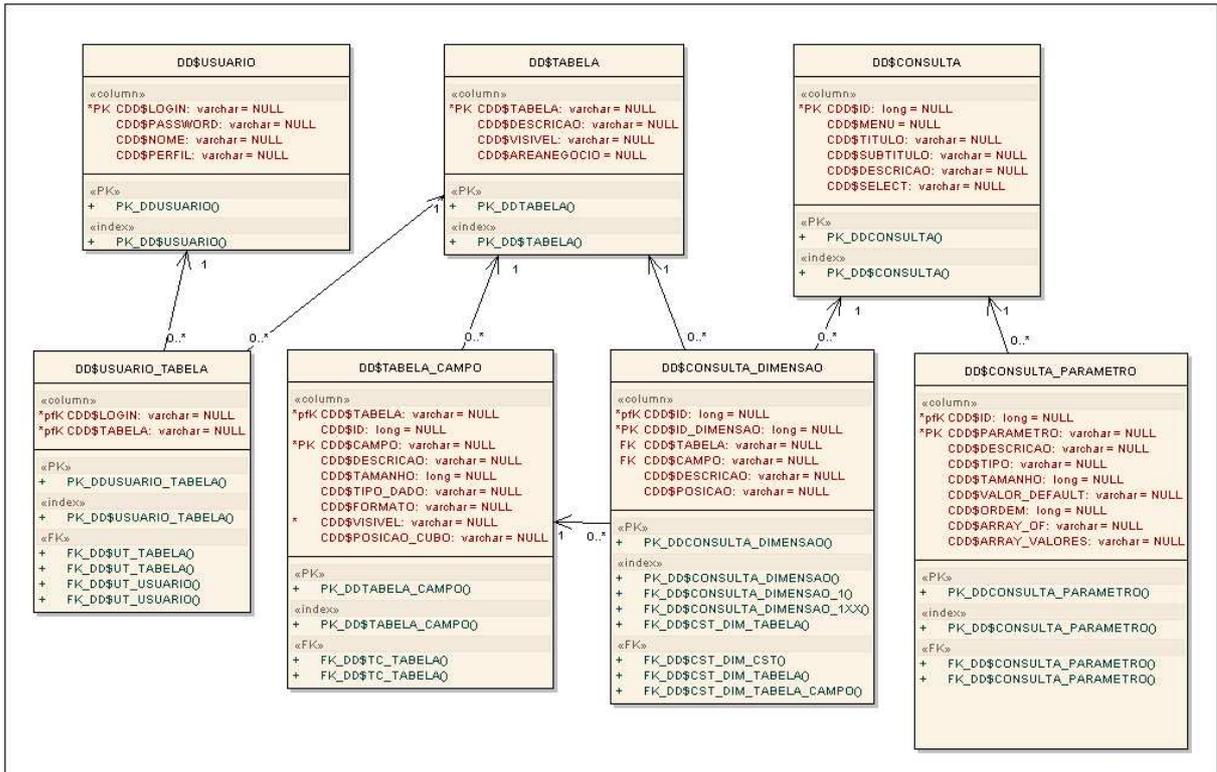


Figura 8 – Diagrama de Entidade Relacionamento do Banco de Dados

3.2.5 Dicionário de Dados

O dicionário de dados possui a definição dos dados mencionados no MER, tais como, as entidades e seus atributos, incluindo detalhes do modelo físico. Pode ser visto como um depósito central que descreve e define o significado de toda a informação usada na construção de um sistema

Segundo Reyes, Amaro, Patrício (2000), um dicionário de dados é uma descrição de características e atributos relevantes para um projeto ou trabalho em particular. É usado para o armazenamento das informações no levantamento do projeto, também para o controle da coleta de objetos e suas características e atributos. É formado por uma lista de características a serem coletadas.

Abaixo é possível visualizar o dicionário de dados da ferramenta proposta demonstrado nos quadros com a seguinte ordem:

- a) O Quadro 15 apresenta a tabela de Consultas que armazena as consultas criadas pelo usuário;
- b) O Quadro 16 apresenta a tabela de configuração das dimensões da consulta;

- c) O Quadro 17 apresenta a tabela de configurações dos parâmetros da consulta;
- d) O Quadro 18 apresenta a tabela de estruturas de tabelas do banco de dados;
- e) O Quadro 19 apresenta a tabela de estruturas de campos do banco de dados;
- f) O Quadro 20 apresenta a tabela de usuários onde são cadastrados os usuários que podem ter acesso a ferramenta;
- g) O Quadro 21 apresenta a tabela de configuração de acesso de cada usuário às tabelas de consulta.

Tabela		Descrição			
DD\$CONSULTA		Guarda as consultas criadas pelos usuários			
PK	FK	Campo	Tipo	NN	Descrição
1		CDD\$ID	INTEGER	<input checked="" type="checkbox"/>	Nr que identifica a consulta
		CDD\$NIVEL_MENU	VARCHAR(100)	<input type="checkbox"/>	Caminho no formato "menu\submenu\" onde a consulta irá aparecer na árvore de consultas existentes no módulo principal da ferramenta
		CDD\$TITULO	VARCHAR(100)	<input type="checkbox"/>	Título da consulta
		CDD\$SUBTITULO	BLOB	<input type="checkbox"/>	Subtítulo da consulta
		CDD\$DESCRICA0	BLOB	<input type="checkbox"/>	Descrição e observações pertinentes à consulta
		CDD\$SELECT	BLOB	<input type="checkbox"/>	Script SQL de consulta
		CDD\$TIPO	CHAR(1)	<input type="checkbox"/>	Identifica o tipo da consulta / quem criou a consulta S = Sistema U = Usuário

Quadro 15 – Tabela de Consultas.

Tabela		Descrição			
DD\$CONSULTA_DIMENSAO		Configurações das dimensões e campos calculados da consulta			
PK	FK	Campo	Tipo	NN	Descrição
1	F	CDD\$ID	INTEGER	<input checked="" type="checkbox"/>	Nr que identifica a consulta
2		CDD\$ID_DIMENSAO	INTEGER	<input checked="" type="checkbox"/>	Nr que identifica a Dimensão da consulta
		CDD\$CAMPO	VARCHAR(31)	<input type="checkbox"/>	Nome físico do campo
		CDD\$DESCRICA0	VARCHAR(50)	<input type="checkbox"/>	Descrição que aparecerá para o usuário
		CDD\$POSICAO	CHAR(1)	<input type="checkbox"/>	Posição inicial no da Dimensão no Cubo de Decisões

Quadro 16 – Tabela de configuração das dimensões da consulta.

Tabela		Descrição			
DD\$CONSULTA_PARAMETRO		Configurações de Filtros e Parâmetros da consulta			
PK	FK	Campo	Tipo	NN	Descrição
1	F	CDD\$ID	INTEGER	<input checked="" type="checkbox"/>	Nr que identifica a consulta
2		CDD\$PARAMETRO	VARCHAR(50)	<input checked="" type="checkbox"/>	Nome do Parâmetro
		CDD\$DESCRICA0	VARCHAR(50)	<input type="checkbox"/>	Descrição do parâmetro que aparecerá para o usuário na tela de passagem de parâmetros

		CDD\$TIPO	VARCHAR(20)	<input type="checkbox"/>	Identifica o tipo de objeto que será criado dinamicamente no formulário de passagem de parâmetros INTEGER = Caixa de texto com formatação para números inteiros FLOAT = Caixa de texto com formatação para números reais DATE = Caixa de texto com formatação de datas STRING = Caixa de texto PERÍODO = Painel de seleção de meses/anos com campos texto para inicial data inicial e final PERIODO_DATA_INI = Painel de seleção de meses/anos onde o parâmetro será a data inicial do mês selecionado PERIODO_DATA_FIM = Painel de seleção de meses/anos onde o parâmetro será a data final do mês selecionado ARRAY = Conjunto de valores pré-definidos
		CDD\$TAMANHO	INTEGER	<input type="checkbox"/>	Tamanho máximo dos valores
		CDD\$VALOR_DEFAULT	VARCHAR(15)	<input type="checkbox"/>	Valor padrão para o parâmetro/filtro
		CDD\$ORDEM	INTEGER	<input type="checkbox"/>	Ordem de visualização dos objetos no formulário de passagem de parâmetros
		CDD\$ARRAY_OF	VARCHAR(15)	<input type="checkbox"/>	Tipo do dado para lista de valores pré-definidos STRING = Valores alfanuméricos INTEGER = Números Inteiros FLOAT = Números Reais
		CDD\$ARRAY_VALORES	VARCHAR(255)	<input type="checkbox"/>	Lista de valores pré-definidos separados por vírgula (,) ex. Item1, Item2, Item3

Quadro 17 – Tabela de configurações dos parâmetros da consulta.

Tabela		Descrição			
DD\$TABELA		Estrutura de Tabelas do Banco de Dados. é gravada toda a estrutura das tabelas para facilitar a configuração de seus atributos dentro da ferramenta			
PK	FK	Campo	Tipo	NN	Descrição
		CDD\$TABELA	VARCHAR(31)	<input checked="" type="checkbox"/>	Nome físico da tabela
		CDD\$DESCRICA0	VARCHAR(50)	<input type="checkbox"/>	Nome fantasia que será mostrado para o usuário final
		CDD\$VISIVEL	CHAR(1)	<input type="checkbox"/>	Identifica se a tabela estará visível ou não para o Usuário Final S = Sim N = Não
		CDD\$GRUPO	VARCHAR(50)	<input type="checkbox"/>	Identifica a Área de Negócio a qual a tabela pertence.

Quadro 18 – Tabela de estruturas de tabelas do banco de dados.

Tabela		Descrição			
DD\$TABELA_CAMPO		Estrutura de Campo de cada Tabelas do Banco de Dados. é gravada toda a estrutura dos campos para facilitar a configuração de seus atributos dentro da ferramenta			
PK	FK	Campo	Tipo	NN	Descrição

		CDD\$ID	INTEGER	<input type="checkbox"/>	Nr que indica a ordem dos campos na estrutura do banco de dados
		CDD\$CAMPO	VARCHAR(31)	<input checked="" type="checkbox"/>	Nome físico do Campo
		CDD\$DESCRICAÇÃO	VARCHAR(50)	<input type="checkbox"/>	Descrição do campo que aparecerá para o Usuário Final
		CDD\$TAMANHO	INTEGER	<input type="checkbox"/>	Tamanho do campo extraído da estrutura do Banco de Dados
		CDD\$TIPO_DADO	CHAR(1)	<input type="checkbox"/>	Tipo do dados N = Numérico T = Texto (Alfanumérico) D = Data
		CDD\$FORMATO	VARCHAR(10)	<input type="checkbox"/>	Máscara que será aplicada a formatação de dados do tipo número e datas ex. #.###,00 ou dd/mm/yyyy
		CDD\$VISIVEL	CHAR(1)	<input checked="" type="checkbox"/>	Indica se o campo estará visível para o Usuário Final
		CDD\$POSICAO_CUBO	CHAR(1)	<input type="checkbox"/>	Indica o formato que o valor pode assumir no Cubo de Decisão D = Campo como uma Dimensão que poderá ser manipulada pelo usuário

Quadro 19 – Tabela de estruturas de campos do banco de dados.

Tabela		Descrição			
DD\$USUARIO		Cadastro de Usuários que podem acessar a ferramenta			
PK	FK	Campo	Tipo	NN	Descrição
		CDD\$LOGIN	VARCHAR(15)	<input checked="" type="checkbox"/>	Login
		CDD\$PASSWORD	VARCHAR(15)	<input type="checkbox"/>	Senha
		CDD\$NOME	VARCHAR(50)	<input type="checkbox"/>	Nome do Usuário
		CDD\$PERFIL	CHAR(1)	<input type="checkbox"/>	Identifica o perfil do usuário A = Administrador U = Usuário Final

Quadro 20 – Tabela de usuários.

Tabela		Descrição			
DD\$USUARIO_TABELA		Cadastro das Tabelas que o usuário poderá utilizar para criar suas consultas			
PK	FK	Campo	Tipo	NN	Descrição
		CDD\$LOGIN	VARCHAR(15)	<input checked="" type="checkbox"/>	Login do Usuário
		CDD\$TABELA	VARCHAR(31)	<input checked="" type="checkbox"/>	Nome da Tabela

Quadro 21 – Tabela de configuração de acesso de cada usuário às tabelas de consulta.

3.3 IMPLEMENTAÇÃO

Nesta seção são apresentadas as técnicas e ferramentas utilizadas para implementação do sistema desenvolvido, tais como Delphi, Firebird, HierCube, Enterprise Architect.

3.3.1 Ferramentas e Tecnologias Utilizadas

A busca por técnicas e ferramentas de desenvolvimento que possibilitassem atender aos objetivos de flexibilidade e generalização da ferramenta proposta resultou nas principais tecnologias e aplicações descritas abaixo que foram utilizadas na implementação deste projeto.

3.3.1.1 FireBird

Segundo Firebirdsqli (2005), o Firebird é um banco de dados Cliente/Servidor *Open Source*, compatível como padrão SQL-ANSI-92, originado a partir do código da versão *Open Source* do Interbase® 6.0 de Julho de 2000, quando a Borland liberou os fontes de seu produto.

Cantu (2005), comenta que devido ao descontentamento da comunidade de desenvolvedores pelo modo como a Borland estava conduzindo o processo de abertura do código de seu banco de dados InterBase e também por ter anunciado que lançaria no mercado novas versões não gratuitas do seu banco de dados, um grupo de desenvolvedores dentre eles alguns dos idealizadores do InterBase como Ann Harrison e Jim Starkey, assumiram o projeto de identificar e corrigir inúmeros *bugs* da versão original criando um novo banco de dados baseado no código do InterBase 6.0, que foi chamado de Firebird. Com isso o Firebird vem evoluindo e ganhando novas versões, sendo utilizado por seus usuários com a garantia de ser um produto que está em pleno desenvolvimento e que será sempre *Free e Open Source*.

Firebird é um banco de dados que oferece excelente concorrência de acesso, alta

performance, e uma poderosa linguagem com suporte a *stored procedures* e *triggers* de bancos. Hoje ele roda em mais de 10 plataformas/sistemas operacionais e praticamente não requer manutenção.

3.3.1.2 HierCube

Conforme RadarSoft (2007), HierCube é um conjunto de componentes visuais desenvolvido pela empresa RadarSoft com o objetivo de disponibilizar recursos OLAP para o desenvolvimento de aplicações com alto nível de processamento de dados e análise de dados.

A biblioteca utiliza os conceitos de MOLAP juntamente com auto desenvolvimento OLAP para criar visões multidimensionais de dados otimizando memória e tempo de processamento. Seu desempenho não depende da capacidade do cubo e da profundidade de dimensões hierárquicas e também não requer um servidor OLAP. Trabalha tanto com a filosofia “estrela” como a “*Snowflake*”.

3.3.2 Operacionalidade da Implementação

Esta seção apresenta o sistema desenvolvido descrevendo as funcionalidades de cada tela.

A descrição da operacionalidade da ferramenta encontra-se dividida em 6 partes:

- a) primeira parte: refere-se à instalação e inicialização da ferramenta podendo ser aplicada não somente a base de dados deste estudo mas em qualquer base de dados desenvolvido com FireBird;
- b) segunda parte: apresenta as ações e configurações pertinentes ao administrador de banco de dados que darão condições para que a ferramenta seja funcional;
- c) terceira parte: mostra o *Front End* Principal da ferramenta e suas funcionalidades que são o ponto de partida para a utilização da ferramenta pelos usuários finais;
- d) quarta parte: descreve o Assistente que ajudará usuários a criar novas consultas e extrair informações do banco de dados;
- e) quinta parte: fala sobre a passagem de parâmetros e a forma com que o sistema cria dinamicamente os componentes necessários para que isso aconteça facilitando a digitação das informações por parte do usuário;

- f) sexta parte: mostra as funcionalidades da consulta em multi-dimensões, utilizando a tecnologia OLAP e os recursos de cubo de decisões para visualização das informações em diferentes ângulos bem como os recursos de impressão, exportação de dados entre outros.

3.3.2.1 Inicialização da Ferramenta

Ao ser iniciada pela primeira vez, a ferramenta apresenta uma tela de configuração e conexão com o banco de dados (Figura 9), na qual o usuário teve identificar o banco de dados que se deseja trabalhar.



Figura 9 – Configuração do Banco de Dados.

Como a ferramenta foi projetada para o funcionamento em qualquer banco de dados Firebird, ao ser iniciada, ela identifica se existe no banco de dados a estrutura necessária para o seu funcionamento, estrutura esta apresentada anteriormente no Modelo Entidade Relacionamento. Se esta não existir, a ferramenta mostra uma tela (Figura 10) permitindo que o usuário atualize o banco de dados criando assim esta estrutura. Caso o usuário não aceite fazer a atualização, o sistema será fechado avisando ao usuário que não é possível executar a ferramenta no banco de dados selecionado sem que este procedimento seja feito.

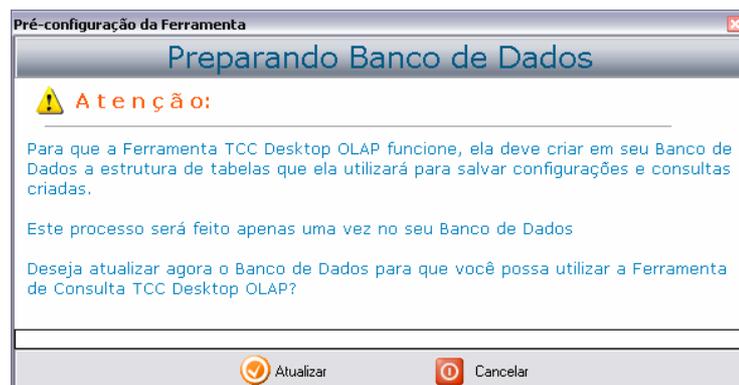


Figura 10 – Criação das estruturas do Banco de Dados

Como no momento da instalação não existem usuários cadastrados a ferramenta alerta através de mensagens que os passos iniciais de cadastro de usuários e cadastro do dicionário de dados devem ser realizados antes de começar a operar a ferramenta para criação de consultas, sendo que o dicionário de dados é parte fundamental na criação destas.

O usuário administrador então cadastra os usuários que podem acessar o sistema junto ao menu “Cadastro de Usuários” (Figura 11).

Nesta tela são solicitadas apenas informações básicas do usuário para o controle do acesso ao sistema. Também apresenta a lista de tabelas acessíveis de cada usuário. Podendo ser configurado no módulo de Dicionário de Dados visto a seguir.

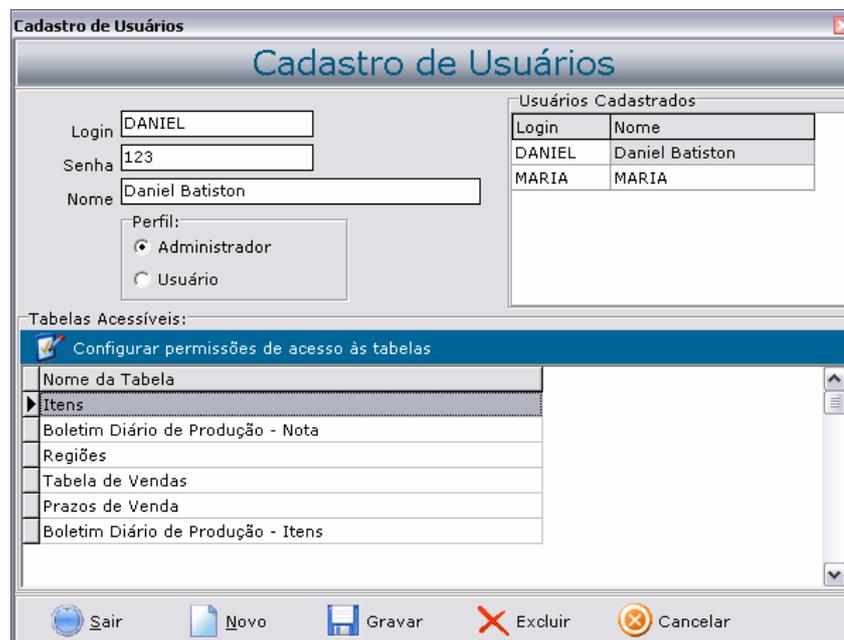


Figura 11 – Cadastro de Usuários e configurações de acesso

Ao serem cadastrados os usuários, as configurações de segurança e acesso serão aplicadas a partir da próxima vez que a ferramenta for iniciada apresentando uma tela de *login* para efetuar o acesso à ferramenta (Figura 12).

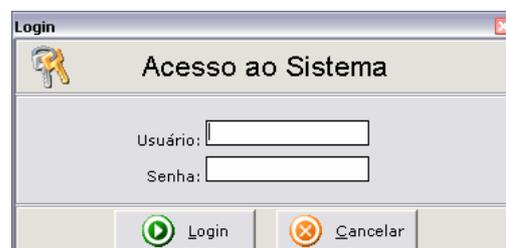


Figura 12 – Login da Ferramenta

Logo após efetuar o *login* na aplicação o usuário é levado à tela inicial com suas respectivas funcionalidades de acordo com o seu perfil, onde somente os usuários administradores terão acesso à manutenção de usuários e dicionário de dados. As demais funcionalidades serão aplicadas a todos os perfis de usuários.

3.3.2.2 Visão do Administrador - Manutenção do Dicionário de Dados

O cadastro principal da ferramenta diz respeito ao Dicionário de Dados a ser configurado por um analista, desenvolvedor ou DBA (*Data Base Administrator*) que tenha o conhecimento de cada parte do banco de dados e consiga traduzir a estrutura técnica das tabelas em nomes de fácil compreensão para o usuário final.

Ao iniciar a manutenção do cadastro do Dicionário de Dados, a ferramenta automaticamente carrega toda a estrutura do banco de dados com suas tabelas e campos relacionados evitando que o administrador tenha que preencher estas informações manualmente reduzindo o tempo para esta configuração.

Um exemplo de código fonte utilizado para listar a estrutura do banco de dados pode ser visto no Quadro 22. Esta função é um exemplo de como a ferramenta interage com o banco de dados, utilizando suas tabelas de configurações internas para extrair as informações de estrutura de tabelas e campos que serão utilizadas na montagem do dicionário de dados da ferramenta.

```

create procedure DD$LISTA_TABELAS_CAMPOS
returns (
  TABELA_NOME char(31),
  CAMPO_ID integer,
  CAMPO_NOME char(31),
  CAMPO_TIPO char(31),
  CAMPO_TAMANHO smallint,
  CAMPO_NULL smallint,
  CAMPO_CHECK blob sub_type 0 segment size 80)
as
begin
  /* Retorna todas as tabelas e seus respectivos campos com informacoes de tipo, tamanho, valor not null, e
  checks */
  For select DD$LISTA_TABELAS.TABELA_NOME
    from DD$LISTA_TABELAS
    order by TABELA_NOME
    into :TABELA_NOME
  do
  begin
    for
      select
        R.RDB$FIELD_ID,
        R.RDB$FIELD_NAME,
        T.RDB$TYPE_NAME,
        F.RDB$FIELD_LENGTH,
        R.RDB$NULL_FLAG,
        F.RDB$VALIDATION_SOURCE
      from RDB$RELATION_FIELDS R
        join RDB$FIELDS F on F.RDB$FIELD_NAME = R.RDB$FIELD_SOURCE
        join RDB$TYPES T on F.RDB$FIELD_TYPE = T.RDB$TYPE
      where
        (R.RDB$RELATION_NAME=:TABELA_NOME) and (T.RDB$FIELD_NAME='RDB$FIELD_TYPE')
    into
      :CAMPO_ID,
      :CAMPO_NOME,
      :CAMPO_TIPO,
      :CAMPO_TAMANHO,
      :CAMPO_NULL,
      :CAMPO_CHECK
    do
    begin
      suspend;
    end
  end
end
end

```

Quadro 22 – Funções que relacionam a estrutura de campos existente no banco de dados

Não é necessário informar os relacionamentos e interligações entre as tabelas do banco de dados pois a própria ferramenta utilizará as informações da estrutura do banco de dados para montar de forma lógica estes relacionamentos de acordo com a integridade referencial definida pelos desenvolvedores do banco de dados. Este é um requisito fundamental para o funcionamento da ferramenta, pois o banco de dados deve estar normalizado da forma mais correta possível para que o assistente gere informações consistentes para a criação das consultas.

A Figura 13 mostra as Configurações de Tabelas onde são relacionadas todas as tabelas do sistema onde o administrador deve informar um pseudônimo de fácil compreensão que identifique cada entidade, selecionar as tabelas que estarão visíveis para os usuários finais bem como sua “Área de Negócio”, que é uma descrição com o objetivo de agrupar as tabelas de acordo com sua finalidade ou setores que às utilizam dentro da empresa, facilitando sua identificação, como por exemplo: Vendas, Compras, Contabilidade, Produção, Financeiro e etc.



Figura 13 – Dicionário de Dados : Configuração de tabelas.

Na Figura 14 pode-se visualizar as Configurações de Campos. Uma vez definida a visibilidade de cada tabela, o administrador passa a configurar nesta aba os atributos de cada campo ligado às tabelas, tais como:

- descrição: apelido/nome que identifica o campo e será apresentado nas consultas;
- visível: identifica se o campo pode ou não ser utilizado para a criação de consultas;
- tipo do Dado: identifica o tipo do dado entre as opções: Texto, Número, Data;
- tamanho: mostra o tamanho máximo do campo definido na estrutura do banco de

dados;

- e) posição no Cubo: identifica se na criação da consulta o campo será utilizado como uma Dimensão da consulta ou como um Valor calculado pela consulta;
- f) formato: configuração de máscara para o campo que será automaticamente formatado na visualização da consulta, pode definir por exemplo casas decimais para dados numéricos, formato de datas entre outros.



Figura 14 – Dicionário de Dados : Configuração de campos.

O administrador pode configurar as tabelas que cada usuário terá acesso para a criação das consultas na tela de Permissões visualizada na Figura 15.

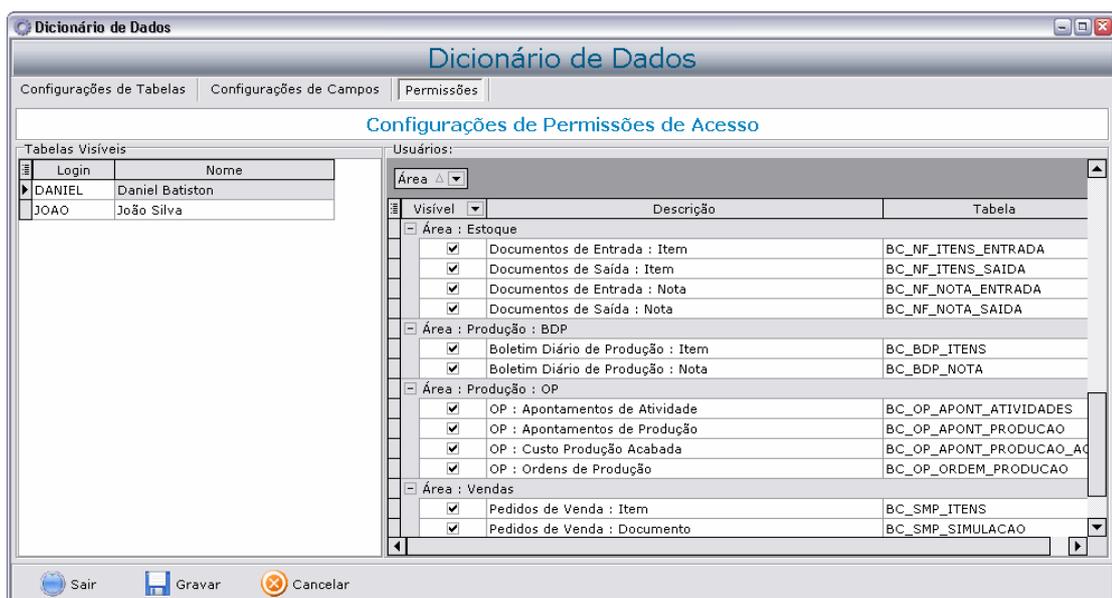


Figura 15 – Permissões de acesso às tabelas pelos usuários

3.3.2.3 Visão do Usuário - *Front End* Principal

Após o *login* ser efetuado é mostrada para o usuário a *interface* inicial da ferramenta vista na Figura 16.

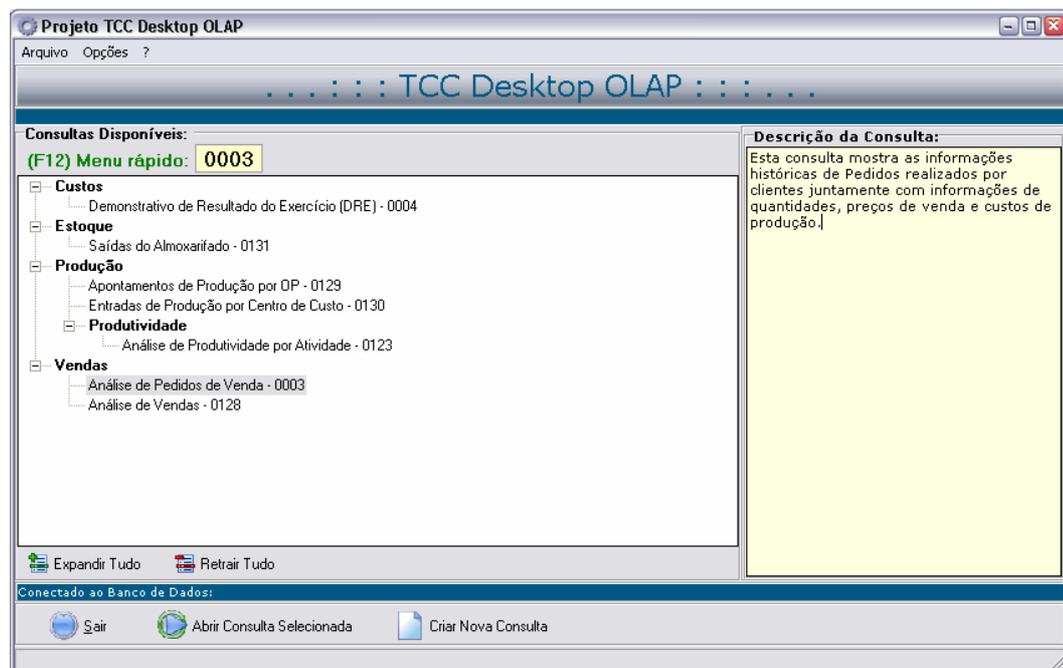


Figura 16 – *Interface* Principal do Sistema

A ferramenta irá mostrar todas as consultas disponíveis criadas anteriormente em forma de árvore que torna a visualização mais eficiente.

Parte do código de construção da árvore de consultas existentes pode ser visto no Quadro 23.

```

Procedure TFormPrincipal._ListaConsultas;
Var
  vTreeNode: Array[ 0..50 ] Of TTreeNode;
  vSTR, vStrContBarra, vCaminhoMenu: String;
  vAdicionar: boolean;
Begin
  Tree.Items.Clear;
  DD_CONSULTA.Close;
  DD_CONSULTA.Open;
  DD_CONSULTA.First;

  While Not DD_CONSULTA.Eof Do
  Begin
    vCaminhoMenu := DD_CONSULTACDDNIVEL_MENU.Value;
    Begin
      With Tree.Items Do
      Begin
        vStrContBarra := '';

        For ii := 1 To length( vCaminhoMenu ) Do
        Begin
          If copy( vCaminhoMenu, ii, 1 ) = '\' Then
          Begin
            vStrContBarra := vStrContBarra + '\';
            vContador := length( vStrContBarra );
          End;
        End;
      End;
    End;
    vAdicionar := true;
  End;

```

```

For iii := 1 To tree.items.Count Do
Begin
  Try
    If tree.items.Item[ iii - 1 ].Level = vContador - 1 Then
      If tree.items.Item[ iii - 1 ].text = vSTR Then
        Begin
          vAdicionar := false;
        End;
      Except
        vAdicionar := false;
      End;
    End;
  End;

  If vAdicionar = true Then
    If vContador = 1 Then
      Begin
        vTreeNode[ vContador - 1 ] := Add( Nil, vSTR );
        vTreeNode[ vContador - 1 ].SelectedIndex := -1;
      End
    Else
      Begin
        vTreeNode[ vContador - 1 ] := AddChild( vTreeNode[ vContador - 2 ], vSTR );
        vTreeNode[ vContador - 1 ].SelectedIndex := DD_CONSULTACDDID.AsInteger;
      End;

      vTreeNode[ vContador - 1 ].ImageIndex := -1;

      If ( DD_CONSULTACDDTIPO.AsString = 'S' ) And ( vTreeNode[ vContador - 1
].SelectedIndex <> -1 ) Then
        vTreeNode[ vContador - 1 ].ImageIndex := 1;
      If ( DD_CONSULTACDDTIPO.AsString = 'U' ) And ( vTreeNode[ vContador - 1
].SelectedIndex <> -1 ) Then
        vTreeNode[ vContador - 1 ].ImageIndex := 2;

      vSTR := '';
    End
  Else
    Begin
      vstr := vstr + copy( vCaminhoMenu, II, 1 );
    End
  End;
End;
End;
DD_CONSULTA.Next;
End;
End;

```

Quadro 23 – Função que monta a estrutura em árvore das consultas existente no banco de dados

Cada consulta é salva com um número identificador que é mostrado no final da sua descrição na árvore de consultas. Este número pode ser digitado no campo “Menu rápido”, para abrir rapidamente as consultas mais utilizadas pelo usuário. O botão “Abrir Consulta selecionada” faz o mesmo processo com a consulta que estiver selecionada na árvore de consultas.

O quadro “Descrição da Consulta” mostra os comentários e descrições pertinentes a consulta selecionada que foram digitadas pelo usuário que fez a consulta para melhor identificar as informações que serão obtidas ao ser executada a consulta.

O botão “Criar Nova Consulta” inicia o assistente de criação de consultas da ferramenta.

3.3.2.4 Assistente de criação de consultas

A idéia principal do Assistente de Criação de Consultas é auxiliar a criação de novas

consultas através de etapas definidas em que o usuário irá selecionar/configurar um atributo da consulta tornando o processo o mais transparente possível para o usuário.

O assistente inicia selecionando todas as tabelas que podem ser acessadas pelo usuário que está utilizando a ferramenta. Isto é definido no cadastro de usuários. Nesta fase inicial, Etapa 1 (Figura 17), o usuário seleciona as Tabelas que ele irá utilizar na criação da consulta.

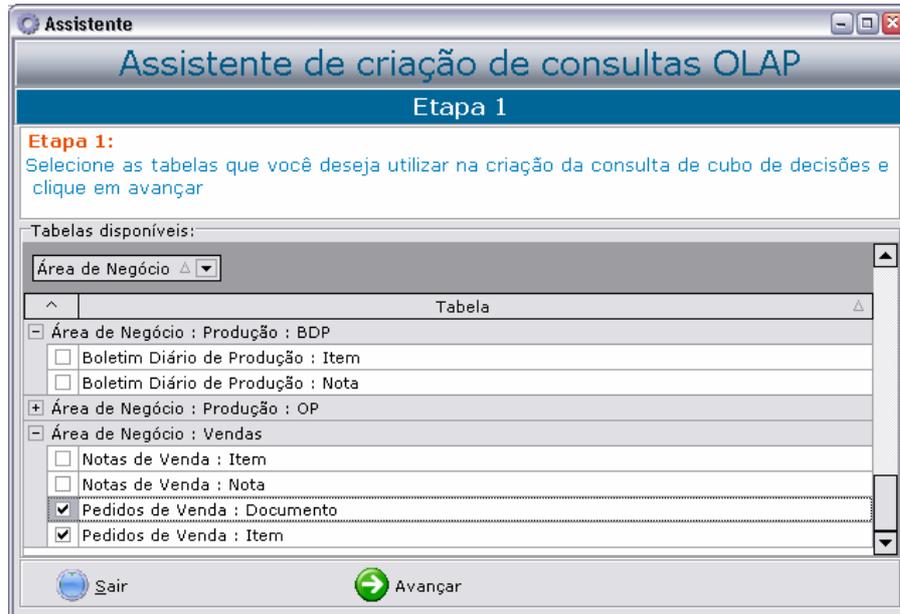


Figura 17 – Etapa 1 : Seleção das tabelas utilizadas na consulta

Na Etapa 2 (Figura 18), o usuário deve informar a tabela de Fato, ou seja a tabela que contém as informações mais pertinentes da consulta para que a ferramenta identifique os relacionamentos entre as outras tabelas de detalhe. Ele também pode informar se a consulta mostrará todas as informações ou apenas um número determinado de registros.

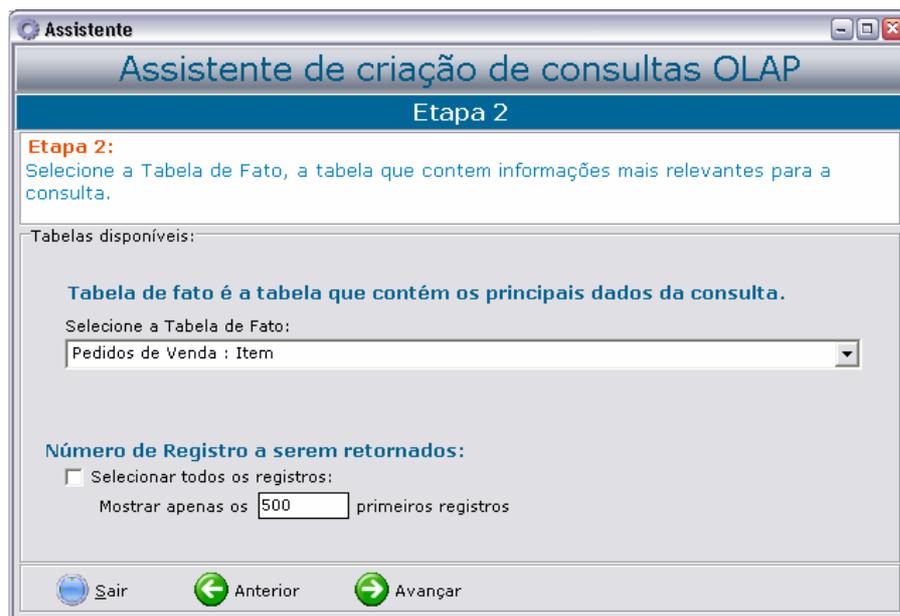


Figura 18 – Etapa 2 : Seleção da tabela de fato

A ferramenta então mostra na Etapa 3 (Figura 19), todos os campos visíveis para as tabelas selecionadas anteriormente, permitindo que o usuário escolha os dados que aparecerão na nova consulta.

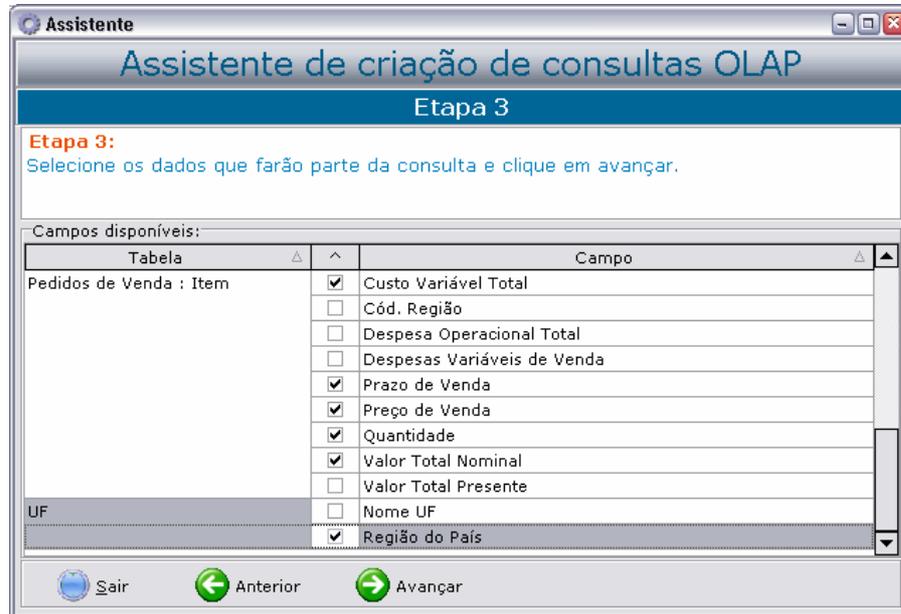


Figura 19 – Etapa 3 : Seleção de campos para a consulta

Na etapa 4 (Figura 20), a ferramenta relaciona todos os campos selecionados pelo usuário e definidos com o parâmetro “Posição no Cubo” igual a “Dimensão” no dicionário dados indicando que ele pode ser manipulado como uma dimensão na consulta. Neste momento o usuário pode escolher em que posição ele deseja que a informação apareça inicialmente na consulta, pois essa posição poderá ser alterada no momento da visualização.

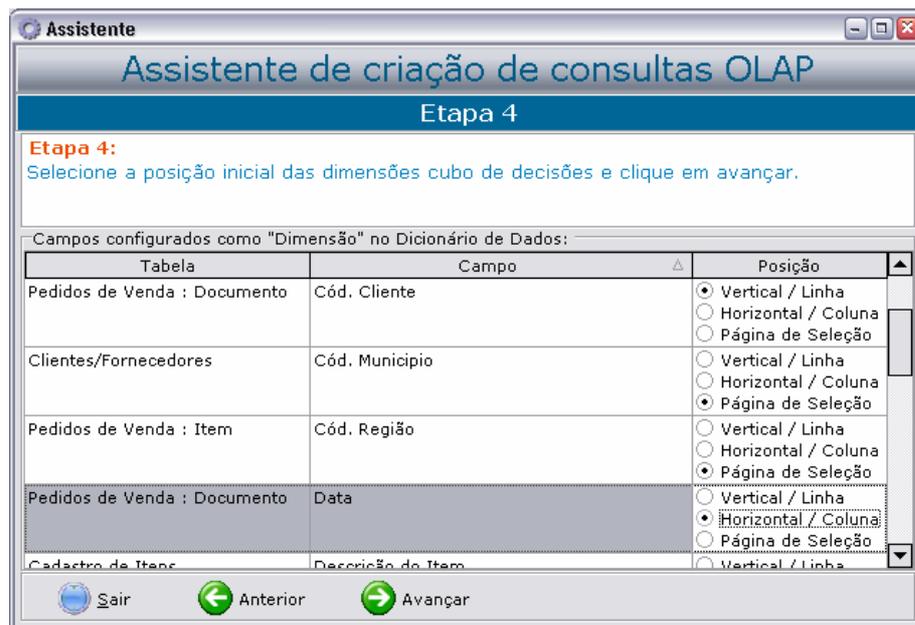


Figura 20 – Etapa 4 : Definição da posição inicial dos dados na consulta

Passando para a Etapa 5, a ferramenta permite criar agregações para cada campo

levando em consideração o parâmetro “Tipo do Dado” definido no dicionário de dados para criar as agregações possíveis. Por exemplo: campos configurados como tipo “Texto” e “Data” podem ser configurados com as agregações: Contar, Máximo e Mínimo. Já os campos configurados com o tipo “Número” podem além das agregações citadas anteriormente, ter as opções: Somar e Média. A Figura 21 representa esta etapa.

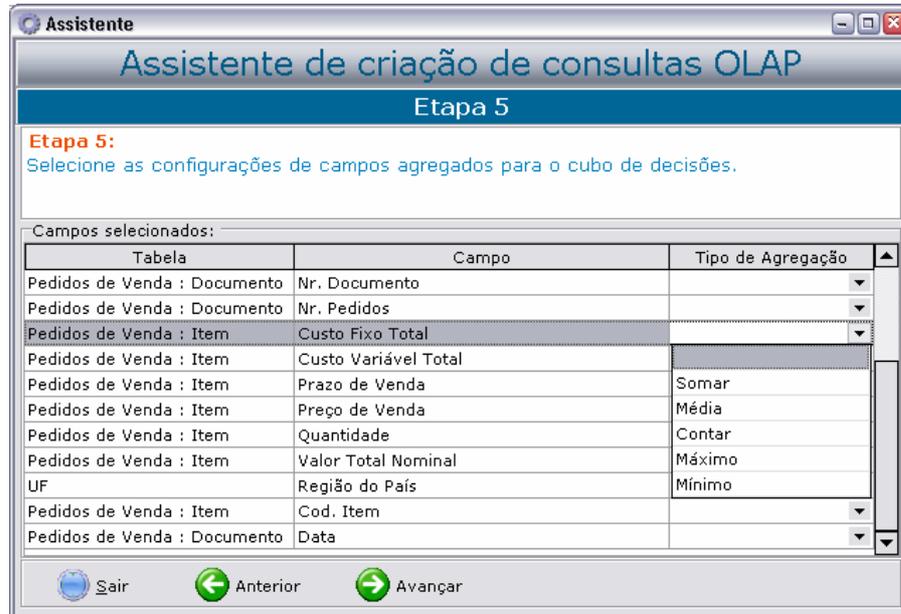


Figura 21 – Etapa 5 : Definição de Agregações

Na Etapa 6 (Figura 22), o usuário pode definir filtros para a consulta juntamente com valores iniciais e padrões se existirem.

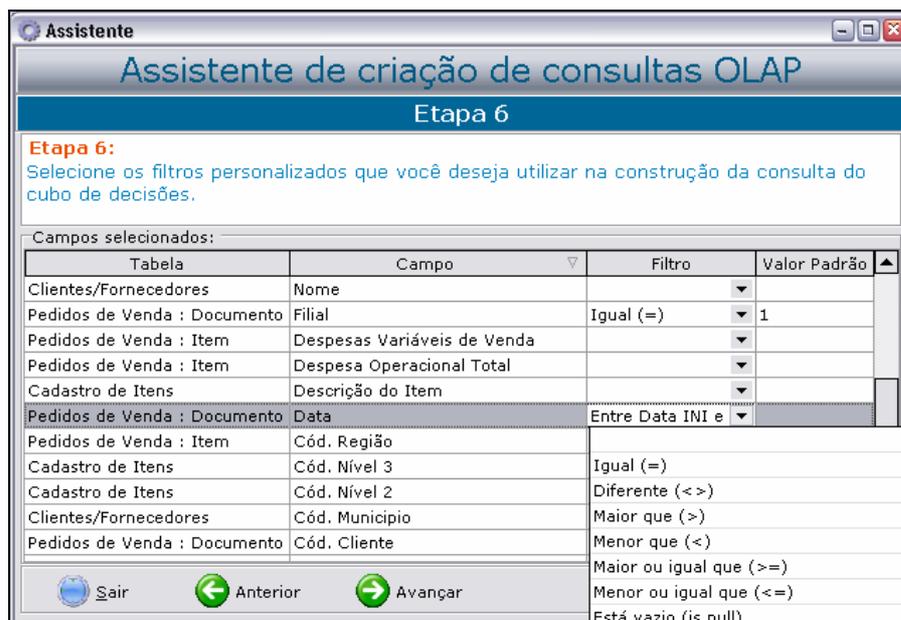


Figura 22 – Etapa 6 : Criação e filtros para a consulta

Finalizando a configuração da consulta está a Etapa 7 (Figura 23), que permite ao usuário especificar um título/nome para a nova consulta juntamente com uma descrição que

fornece mais informações sobre os dados contidos na consulta criada.

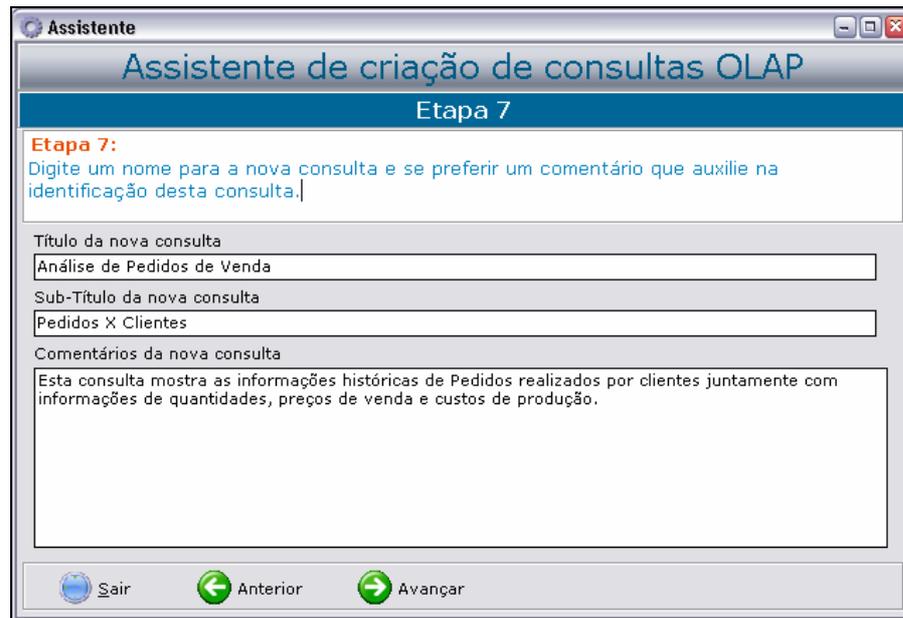


Figura 23 – Etapa 7 : Definição de nome e descrições para a consulta

Por fim a Etapa 8, visualizada na Figura 24 conclui as configurações de construção da nova consulta e fornece ao usuário opções como:

- a) testar Scripts e Dados: faz uma verificação se o script SQL gerado pelo Assistente está correto e mostra os dados em uma consulta simples para que o usuário verifique se as informações estão sendo extraídas de forma correta;
- b) executar: permite executar a consulta, digitando os parâmetros de acordo com os filtros definidos no assistente e visualizando as informações sendo aplicadas à tecnologia OLAP;
- c) salvar: permite gravar a nova consulta para utilização futura, o usuário então define a posição em que ela aparecerá na árvore de consultas mostrada no Front End Principal da ferramenta juntamente com o número identificador da consulta. A ferramenta grava no banco de dados as informações pertinentes a nova consulta de acordo com a necessidade da ferramenta;
- d) visualizar Script: permite aos administradores e usuários mais experientes analisar o *Script* SQL gerado pela ferramenta (Figura 25).

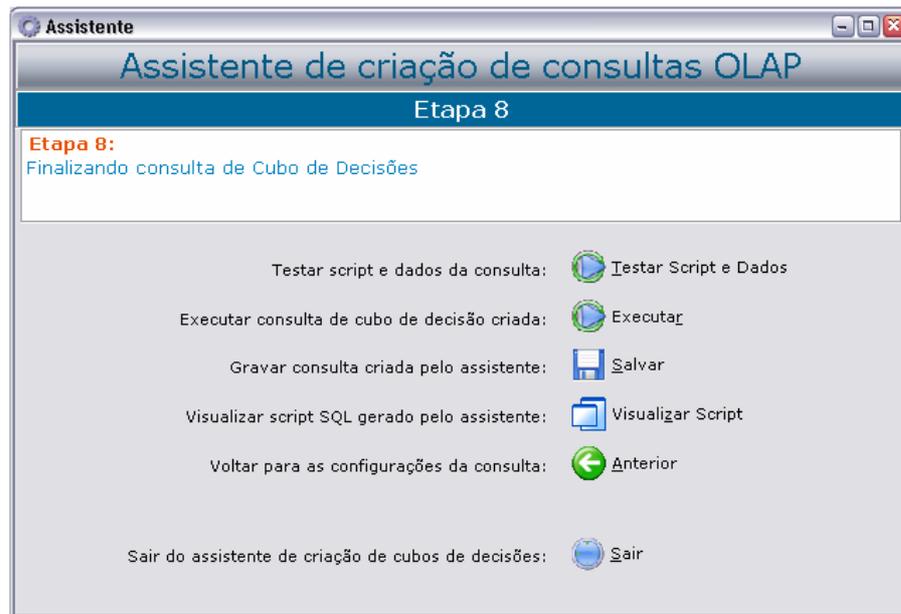


Figura 24 – Etapa 8 : Finalizando Nova Consulta

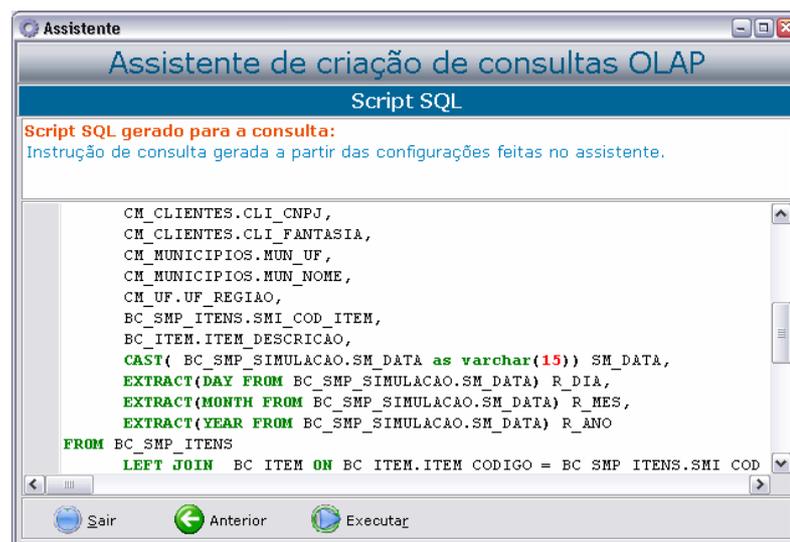


Figura 25 – Visualização de *script* SQL

3.3.2.5 Passagem de parâmetros

Ao ser executada uma consulta a ferramenta identifica se esta possui em suas configurações, definições de filtros criados pelo sistema.

Cada filtro é configurado com um tipo de objeto a ser mostrado para o usuário e com isso facilitar a digitação das informações em cada parâmetro. A ferramenta identifica os filtros e seus respectivos componentes associados e cria dinamicamente uma tela contendo todos os objetos de acordo com suas configurações.

A Figura 26, mostra os tipos de objetos criados dinamicamente de acordo com o tipo

de dados existente.

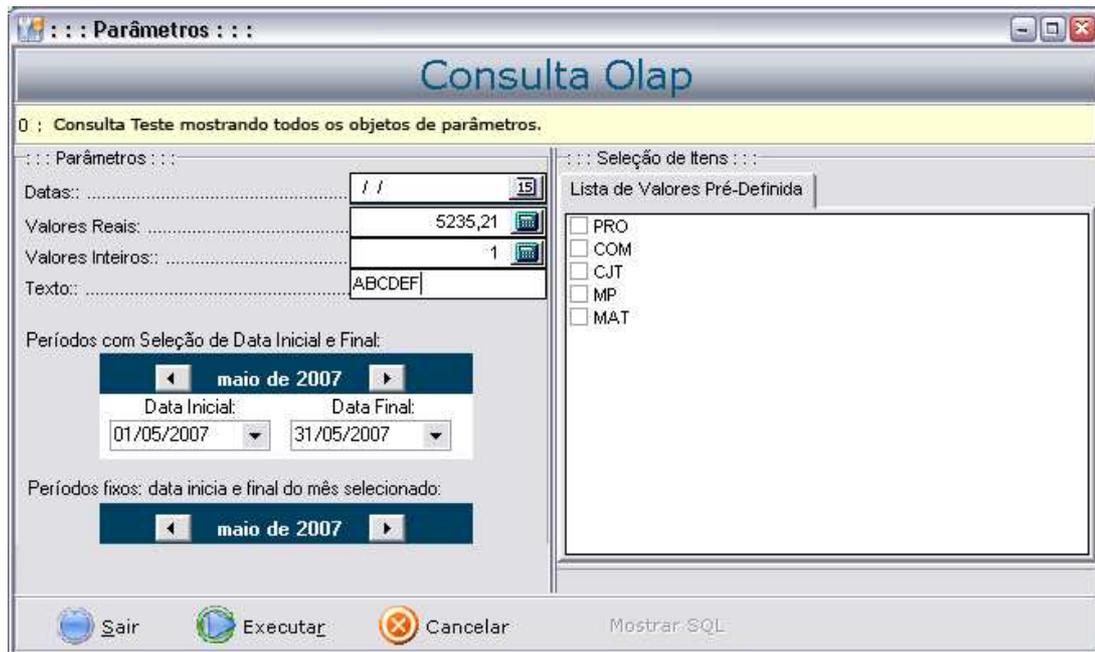


Figura 26 – Objetos aplicados às configurações de filtro das consultas

Dentre os objetos disponíveis podemos citar:

- a) datas: objetos que tratam a formatação de data para facilitar a digitação da informação pelo usuário além de mostrar um calendário para seleção de datas se necessário;
- b) valores Reais e Inteiros: objetos criados para formatação dos valores que mostra uma calculadora para auxiliar o usuário na entrada da informação;
- c) texto: caixa de texto padrão para tratamento de valores alfanuméricos;
- d) períodos com Seleção de Data Inicial e Final: objeto utilizado quando os parâmetros da consulta referem-se a um período de tempo como meses e anos, possibilitando o usuário informar a data inicial e final para o filtro selecionado;
- e) períodos fixos: data inicial e final do mês selecionado: objeto que mostra uma caixa de diálogo com fácil seleção de meses e anos. Utilizado para consultas que exigem o período completo do mês, ou seja informações entre a data inicial e final do mês selecionado;
- f) lista de Valores Pré-definidos: utilizado quando existem valores padrões e fixos que o usuário pode selecionar sem ter que digitar um a um, melhorando a interação com o usuário.

Quando a consulta for executada a ferramenta automaticamente aplica os valores digitados nos objetos ao seu filtro correspondente relacionado na instrução SQL da consulta.

Estes valores podem ser modificados a qualquer momento na visualização da consulta

tornando flexível a passagem de parâmetros da consulta.

Parte do código de criação dinâmica dos objetos pode ser visualizada no Quadro 24. Nesta função a ferramenta lê as configurações de filtros da consulta e cria componentes visuais na tela - mostrados anteriormente na Figura 26 - de acordo com o tipo de cada informação definido no cadastro de dicionário de dados da ferramenta. Define a ordem em que os componentes serão mostrados, posiciona os componentes corretamente na tela, define a descrição de cada parâmetro e interliga os componentes criados aos seus respectivos parâmetros e filtros definidos na consulta.

```

Procedure TFormParametros.fsCriarComponentes;
{Cria componentes} // Criar uma funcao depois
Var
  iGPI, i, iNumTPanelCriados: integer;
  vIbDataSet: TIBDataSet;

  vDataSource: TDataSource;
  vParentAtual: TWinControl;

  vIndiceComponente: integer;

Function fsCriarPainel( vName: String; vParent: TWinControl ): TWinControl;
Begin
  ComPanel[ i + 1 ] := TPanel.Create( self );
  With ComPanel[ i + 1 ] Do
  Begin
    Align := alTop;
    BevelOuter := bvNone;
    AutoSize := true;
    Height := 20;
    parent := vParent;
    name := cPrefixoComPanel + vName;
    Caption := '';
    vListaComponentes.AddObject( name, ComPanel[ i + 1 ] );
    result := ComPanel[ i + 1 ];
  End;
End;

Function fsCriarRxCalcEdit( vName: String; vParent: TWinControl ): boolean;
Begin
  ComRxCalcEdit[ i + 1 ] := TRxCalcEdit.Create( self );
  With ComRxCalcEdit[ i + 1 ] Do
  Begin
    Align := cAlinhamentoCamposEdit;
    Hint := 'Digite um valor para este campo';
    AutoSelect := true;
    CheckOnExit := true;

    ZeroEmpty := false;
    width := cWidthCamposEdit;
    parent := vParent;
    Ctl3D := false;
    name := cPrefixoComRxCalcEdit + vName;
    vListaComponentes.AddObject( name, ComRxCalcEdit[ i + 1 ] );

  End;
End;
.
.
Begin
  fsListaComponentesFree;

  screen.cursor := crHourGlass;
  LockWindowUpdate( Application.Handle );

  iNumTPanelCriados := 0;
  vListaComponentes := TStringList.Create; // Cria Lista de Componentes Virtual

  DD_CONSULTA_PARAMETRO.Close;
  DD_CONSULTA_PARAMETRO.Open;

  DD_CONSULTA_PARAMETRO.Last;

  GBSelecaoItens.Visible := false;
  GBParametros.Visible := false;

```

```

i := 0;

While Not DD_CONSULTA_PARAMETRO.Bof Do
Begin
  If uppercase( DD_CONSULTA_PARAMETRO.CDDTIPO.AsString ) = vTipoFloat Then
  Begin
    GBParametros.Visible := true;
    vParentAtual := fsCriarPainel( DD_CONSULTA_PARAMETRO.CDDPARAMETRO.value, GBParametros );
    fsCriarRxCalcEdit( DD_CONSULTA_PARAMETRO.CDDPARAMETRO.value, vParentAtual );
    fsCriarLabel( DD_CONSULTA_PARAMETRO.CDDPARAMETRO.value, vParentAtual );

    vIndiceComponente := vListaComponentes.IndexOf( cPrefixoComRxCalcEdit +
DD_CONSULTA_PARAMETRO.CDDPARAMETRO.value );
    TRxCalcEdit( vListaComponentes.Objects[ vIndiceComponente ] ).DisplayFormat :=
vgFormatoNumeroReal; // Define o Formato do numero float padrao
    inc( iNumTPanelCriados );
  End;

  If uppercase( DD_CONSULTA_PARAMETRO.CDDTIPO.AsString ) = vTipoDate Then
  Begin
    GBParametros.Visible := true;
    vParentAtual := fsCriarPainel( DD_CONSULTA_PARAMETRO.CDDPARAMETRO.value, GBParametros );
    fsCriarDateEdit( DD_CONSULTA_PARAMETRO.CDDPARAMETRO.value, vParentAtual );
    fsCriarLabel( DD_CONSULTA_PARAMETRO.CDDPARAMETRO.value, vParentAtual );

    If DD_CONSULTA_PARAMETRO.CDDVALOR_DEFAULT.IsNull = false Then
    Begin
      vIndiceComponente := vListaComponentes.IndexOf( cPrefixoComDateEdit +
DD_CONSULTA_PARAMETRO.CDDPARAMETRO.value );

      If DD_CONSULTA_PARAMETRO.CDDVALOR_DEFAULT.AsString = vDefaultHoje Then
        TDateEdit( vListaComponentes.Objects[ vIndiceComponente ] ).text := DateToStr( date )
      Else
        TDateEdit( vListaComponentes.Objects[ vIndiceComponente ] ).text :=
DD_CONSULTA_PARAMETRO.CDDVALOR_DEFAULT.AsString;
      End;

      inc( iNumTPanelCriados );
    End;
    .
    .
    .

    DD_CONSULTA_PARAMETRO.Prior;
  End;
  .
  .
  .
End;

```

Quadro 24 – Função de criação dinâmica dos objetos e parâmetros da consulta

3.3.2.6 Visualização da consulta OLAP (Cubo de Decisões)

Neste momento a ferramenta utiliza as informações da consulta criada pelo usuário com todas as definições de dimensões, descrições iniciais e aplica ao Cubo de Decisões criando dinamicamente suas dimensões, descrição, formatações e campos calculados.

Um exemplo do código de criação dinâmica das dimensões do cubo de decisão pode ser visualizado no Quadro 25.

```

Procedure TFormPreview.acCriaCuboExecute( Sender: TObject );
Var
  si: TSummaryInfo;
  DimSum: TDimSum;
Begin
  DD_CONSULTA_DIMENSAO.close;
  DD_CONSULTA_DIMENSAO.Open;

  HC.Active := false;

  { Inicializa a montagem do Cubo de Decisoes }
  HC.BuildDimensionMap;

  { Define as dimensoes }
  Begin
    DD_CONSULTA_DIMENSAO.Close;
    DD_CONSULTA_DIMENSAO.Open;
    DD_CONSULTA_DIMENSAO.First;

    While Not DD_CONSULTA_DIMENSAO.Eof Do
      Begin
        If DD_CONSULTA_DIMENSAOCDDPOSICAO.value = 'C' Then
          HC.DefineFieldAsDimension( DD_CONSULTA_DIMENSAOCDDCAMPO.value,
          DD_CONSULTA_DIMENSAOCDDDESCRICAO.value, dgCol );

          If DD_CONSULTA_DIMENSAOCDDPOSICAO.value = 'L' Then
            HC.DefineFieldAsDimension( DD_CONSULTA_DIMENSAOCDDCAMPO.value,
            DD_CONSULTA_DIMENSAOCDDDESCRICAO.value, dgRow );

          If DD_CONSULTA_DIMENSAOCDDPOSICAO.value = 'P' Then
            HC.DefineFieldAsDimension( DD_CONSULTA_DIMENSAOCDDCAMPO.value,
            DD_CONSULTA_DIMENSAOCDDDESCRICAO.value, dgPage );

          If DD_CONSULTA_DIMENSAOCDDPOSICAO.value = 'S' Then
            Begin
              HC.DefineFieldAsDimension( DD_CONSULTA_DIMENSAOCDDCAMPO.value,
              DD_CONSULTA_DIMENSAOCDDDESCRICAO.value, dgPage );

              HC.DefineFieldAsSummary( DD_CONSULTA_DIMENSAOCDDCAMPO.value );

              {
                si := G.FindSummaryFieldName( DD_CONSULTA_DIMENSAOCDDCAMPO.value );
                If si <> Nil Then
                  Begin
                    si.FunctionCount.Visible := True;
                    si.FunctionSum.Visible := False;
                  End;
              }
            End;
            DD_CONSULTA_DIMENSAO.next;
          End;
        End;
      HC.DimensionMapChanged;
      HC.Active := True;
    End;
  End;

```

Quadro 25 – Função de que cria dinamicamente as dimensões e campos calculados no Cubo de Decisão

A Figura 27, mostra as informações sendo aplicadas à tecnologia OLAP utilizando os componentes visuais HierCube descritos no item 3.3.1.2 deste trabalho.

Com esta tecnologia o usuário pode a partir de agora, exercitar seu poder de análise e “brincar” com a informação como se estivesse com um cubo em suas mãos. Pode-se visualizar a informação de diferentes ângulos movendo as dimensões presentes na consulta tanto para a linha horizontal ou vertical do cubo resultando numa interação diferente.

dados para outros formatos de arquivo para que possam ser utilizados em outras finalidades (Figura 29).



Figura 29 – Impressão e Exportação de Dados

A Figura 30, mostra os recursos que permitem salvar e recuperar cenários de dados que servem como modelos pré-formatados de visualização evitando que o usuário tenha que reorganizar as dimensões do cubo de decisão a cada consulta e padronizando as análises.



Figura 30 – Opções do Cubo de Decisão

É disponibilizado também a visualização das informações em forma de Gráficos (Figura 31). Lembrando que neste serão apresentados as informações representadas pela primeira dimensão da coluna versus a primeira dimensão da linha.

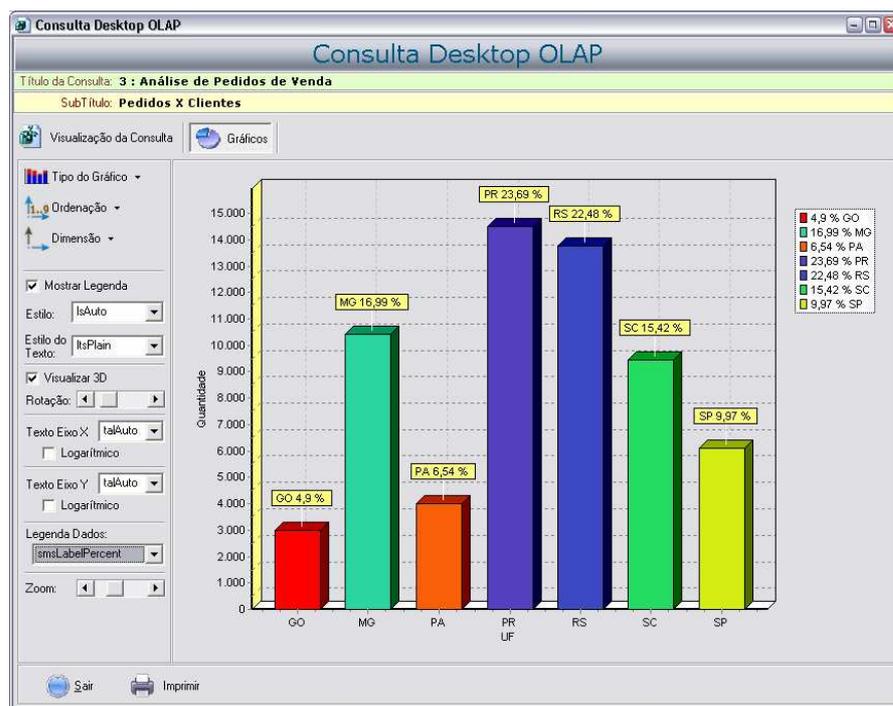


Figura 31 – Gráficos dos dados consultados.

3.4 RESULTADOS E DISCUSSÃO

Com base nos trabalhos correlatos, observou-se que a ferramenta Desktop OLAP se assemelha às ferramentas comerciais desenvolvidas para bancos de dados proprietários e pode somar ao portfólio de produtos e soluções da empresa Baptistetti Consultoria se tornando um atrativo tecnológico para seus atuais e futuros clientes.

Observou-se também que a ferramenta desenvolvida pode ser utilizada em inúmeros trabalhos auxiliando na extração e manipulação multidimensional de informações. Atuando como facilitador e como uma real ferramenta de desenvolvimento de consultas, ela deixa para o desenvolvedor o trabalho de projetar e criar a estrutura Data Warehouse pertinente ao seu projeto enquanto o desenvolvimento da tecnologia, desenvolvimento de *interface*, funcionalidades, e tratamento da informação de forma multidimensional fica por conta da ferramenta Desktop OLAP proposta aqui.

Com os conhecimentos adquiridos no estudo da estrutura e funcionamento da tecnologia OLAP, aliado ao desenvolvimento utilizando os componentes HierCube para tratamento multidimensional de informações, foram adquiridas diversas técnicas de criação de aplicações bem como estruturação de bancos de dados, que resultam em performance e organização no tratamento da informação.

4 CONCLUSÕES

Neste item mencionam-se as considerações gerais, as dificuldades encontradas na realização deste trabalho e as extensões para trabalhos futuros.

4.1 CONSIDERAÇÕES

No que diz respeito aos objetivos apresentados como proposta inicial para este trabalho, pode-se concluir que foram alcançados com sucesso.

Com o desenvolvimento do assistente de criação passo a passo, a ferramenta mostrou-se flexível na criação de consultas, pois ela deixa a geração de código e interligações do banco de dados muito transparente para o usuário, isto aliado ao conceito de divisão das informações por áreas de negócio, torna o processo de criação de consultas não estruturadas de fácil manipulação por qualquer usuário que tenha o conhecimento das ações do seu negócio, facilitando a localização da informação no contexto geral do banco de dados.

O processo de verificação e criação da estrutura funcional no banco de dados quando a ferramenta é inicializada pela primeira vez, torna ela genérica ao ponto de ser utilizada em qualquer outra base de dados de aplicações que tenham necessidades similares as apresentadas neste trabalho.

A implementação de uma ferramenta genérica como a aqui proposta, traz diversos benefícios a uma equipe de desenvolvimento de sistemas, pois o foco não está na resolução de uma necessidade ou problema em específico, mas sim, em desenvolver o mecanismo que atenderá a todas as necessidades similares.

No caso do objeto deste estudo, qualquer nova implementação ou funcionalidade adicionada ao aplicativo, servirá automaticamente para todas as situações em que a ferramenta se aplica. Na maioria dos casos isso evita o re-trabalho e desenvolvimento desnecessários.

A ferramenta, após realizada a fase de testes, mostrou-se bastante útil aos programadores diminuindo o tempo de desenvolvimento de consultas, sem a necessidade de criar para cada nova consulta, módulos específicos para o tratamento da informação no formato de cubo de decisões. Outro benefício trazido com a ferramenta foi uma maior organização e documentação da estrutura do banco de dados através do dicionário de dados

proposto.

Para os consultores de custo da empresa a ferramenta é um atrativo que possibilita inicialmente conhecer e interagir cada vez mais com os sistemas desenvolvidos, flexibilizando sua ação e tomada de decisão junto aos clientes.

Mediante o treinamento e familiarização da ferramenta por parte dos consultores de custo, esta será disponibilizada aos clientes finais através de seus gerentes de negócio e pessoas responsáveis pela análise das informações da empresa.

O desenvolvimento deste trabalho trouxe uma visão mais detalhada das tecnologias e técnicas de processamento, tratamento e visualização de informações, objetivando realmente resultado.

O estudo da estrutura OLAP bem como DW e DM revela tendências e conceitos que podem ser aplicados diretamente em meu desenvolvimento profissional ajudando na criação de aplicações e implementações de alto padrão tecnológico, com menos esforço e obtendo melhores resultados tanto para o desenvolvimento próprio como para os usuários que estarão utilizando os projetos.

4.2 EXTENSÕES

Como sugestões de continuidade deste trabalho estão:

- a) evoluir a ferramenta tornando-a compatível com outros bancos de dados utilizados atualmente;
- b) desenvolver um assistente para alteração das consultas existentes facilitando sua manutenção pelo usuário final;
- c) desenvolver um editor de consulta simplificado para utilização dos desenvolvedores em manutenções simples de consultas.

4.3 DIFICULDADES

Uma das dificuldades encontradas foi desenvolver as técnicas de geração de código SQL com base nas configurações do dicionário de dados, já que componentes proprietários

pesquisados que disponibilizam as mesmas funcionalidades, não atendiam aos quesitos de integração com a *interface* proposta pela ferramenta.

Com este desenvolvimento a ferramenta se torna mais flexível para a evolução e desenvolvimento de interfaces amigáveis e também novas funcionalidades, não se limitando às implementadas em aplicativos comerciais existentes.

REFERÊNCIAS BIBLIOGRÁFICAS

BAPTISTETTI, Dirceu. [entrevista do proprietário/gerente da empresa Baptistetti Consultoria S/C Ltda.]. 2006.

BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. Rio de Janeiro: Campus, 2002.

BINDER, Fabio Vinicius. **Sistemas de apoio a decisão**. São Paulo: Érica, 1994.

CANTU, Carlos Henrique. **Firebird essencial**. Rio de Janeiro : Ciência Moderna, 2005. xxii,308 p, il.

CIELO, Ivã. **Um pouco sobre OLAP**. Brasília, jan. 2000. Disponível em: <<http://www.datawarehouse.inf.br/>>. Acesso em: 02 abril. 2007.

CORRADI, Lauricio; AZAMBUJA, Ricardo Alencar de; **Sistema de informação gerencial aplicado a comercialização de planos de previdência privada utilizando a técnica de cubo de decisão**. , 2002. xiii, 73p, il. Universidade Regional de Blumenau, Curso de Ciências da Computação, Blumenau, 2002.

DATE, C.J. **Introdução a sistemas de banco de dados**. Rio de Janeiro: Campus, 2000.

DWBRASIL. **Olap**. Brasília, 2005. Disponível em: <<http://www.dwbrasil.com.br/html/olap.html>>. Acesso em: 20 mai. 2007.

Firebirdsql. **About Firebird**. [S.l.], 2005. Disponível em: <<http://www.firebirdsql.org/>>. Acesso em: 10 nov. 2006.

FURLAN, Jose Davi. **Modelagem de objetos através da UML-The Unified Modeling Language**. Sao Paulo : Makron Books do Brasil, 1998. xiv, 329p, il.

GSI. **Mineração de Dados**. Grupo de Sistemas Inteligentes. Maringá, 2004. Disponível em: <http://www.din.uem.br/ia/a_multid/mineracao/>. Acesso em: 25 mai. 2007.

INMON, W. H. **Como construir o data warehouse**. Rio de Janeiro: Campus, 1997.

INMON, William H.; WELCH, J. D.; GLASSEY, Katharine L. **Gerenciando Data Warehouse: técnicas práticas para monitorar operações e performances; administrar dados e ferramentas; gerenciar alterações e crescimento**. São Paulo, Makron Books, 1999.

INMON, William H; TERDEMAN, Robert H.; IMHOFF, Claudia. **Data warehousing: como transformar informações em oportunidades de negócios**. São Paulo: Berkeley Brasil, 2001.

ITALIANO, Isabel C.; ESTEVES, Luiz A. **Modelagem de Data Warehouses e Data Marts**. SQL Magazine, Grajaú, RJ. Ed. 13, 2003

LEITÃO, Claudia Nolla. **Construção de Aplicações com o Uso de Ferramentas OLAP**. 2000.72 f, il. Projeto Final do Curso de Graduação de Bacharelado em Informática - Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2000. Disponível em <<http://dataware.nce.ufrj.br:8080/dataware/publicacoes/dataware/fisico/projetosFinais/datawarehousing/LEITAO-2000.pdf>>. Acesso em: 18 fev. 2007.

MARTINS, Roseléia. **Ferramenta para auxílio à administração de materiais em indústrias gráficas utilizando a tecnologia OLAP**. 2005.72 f, il. Trabalho de conclusão de curso - Universidade Regional de Blumenau, Curso de Sistemas de Informação, Blumenau, 2005.

MEDEIROS, Ernani. **Desenvolvendo software com UML 2.0: definitivo**. São Paulo : Pearson Makron Books, 2004. xxi, 264 p, il.

Microsoft. **Business Intelligence**. [S.l.], 2005. Disponível em: <<http://www.microsoft.com/sql/solutions/bi/default.msp>>. Acesso em: 22 out 2005.

O'BRIEN, James A. **Sistemas de informação: e as decisões gerenciais na era da internet**. 2. ed. São Paulo : Saraiva, 2004. 436 p.

OLIVEIRA, Adelize Generini de. **Data warehouse: conceitos e soluções**. [S.l.] : Advanced, c1998.

Oracle. **Business Intelligence Discoverer**. [S.l.], 2005. Disponível em: <<http://www.oracle.com/technology/products/discoverer/index.html>>. Acesso em: 01 set. 2005.

RadarSoft. **HierCube**. [S.l.], 2007. Disponível em: <<http://www.radar-soft.com/pt/products/vcl.aspx>>. Acesso em: 15 mar. 2007.

Reyes, A.; Amaro, B.; Patrício, M.; **Manual de usuário para levantamentos com correção diferencial com pós-processamento**. São Paulo, 2000. Disponível em: <<http://gps.ciagri.usp.br/apostila.pdf>>. Acesso em: 18 maio. 2007.

SCOTT, Kendall et al. **UML essencial: um breve guia para a linguagem-padrao de modelagem de objetos**. 2. ed. Porto Alegre : Bookman, c2000. 169p, il. Tradução de: UML distilled.

YOURDON, Edward. **Análise estruturada moderna**. Rio de Janeiro : Campus, 1990. 836p, il, 23cm. (Serie Yourdon press). Tradução de: Modern structured analysis (3. ed.).