

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

FERRAMENTA DE CONSTRUÇÃO DE DATA WAREHOUSE

PERCIO ALEXANDRE DE OLIVEIRA

BLUMENAU
2007

2007/1-36

PERCIO ALEXANDRE DE OLIVEIRA

FERRAMENTA DE CONSTRUÇÃO DE DATA WAREHOUSE

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 Ciências da Computação — Bacharelado.

Prof. Maurício Capobianco Lopes - Orientador

**BLUMENAU
2007**

2007/1-36

FERRAMENTA DE CONSTRUÇÃO DE DATA WAREHOUSE

Por

PERCIO ALEXANDRE DE OLIVEIRA

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

Presidente: _____
Prof. Maurício Capobianco Lopes – Orientador, FURB

Membro: _____
Prof. Oscar Dalfovo – FURB

Membro: _____
Prof. Alexander Roberto Valdameri – FURB

Blumenau, 04 de julho de 2007

À minha esposa Fernanda e ao meu filho
Thiago, pela compreensão, paciência e
incentivo durante o período deste trabalho.
Vocês são a minha maior realização.

AGRADECIMENTOS

Primeiramente a Deus, que me sustenta, dirige e me dá forças para vencer as adversidades da vida.

À minha família pelo carinho, suporte e orações em favor do meu sucesso no decorrer do meu curso.

E ao meu professor orientador Maurício Capobianco Lopes pela dedicação e comprometimento na construção do presente trabalho.

Não é preciso ter olhos abertos para ver o sol,
nem é preciso ter ouvidos afiados para ouvir o
trovão. Para ser vitorioso você precisa ver o
que não está visível.

Sun Tzu

RESUMO

O Data Warehouse é uma solução que procura de maneira flexível e eficiente tratar grandes volumes de dados e obter informações que auxiliem no processo para tomada de decisão. Em vista do processo de construção deste ambiente ser bastante trabalhoso e demandar grande quantidade de tempo e dinheiro, este trabalho apresenta uma ferramenta de fácil usabilidade com foco a usuários e projetistas de data warehouse. A ferramenta foi desenvolvida em Java garantindo a portabilidade de seu sistema que implementa as principais fases de um projeto de data warehouse: extração, transformação e carga dos dados; visualização, análise e tratamento das informações.

Palavras-chave: Data warehouse. Extração, transformação e carga de dados. Sistema de apoio a decisão.

ABSTRACT

Data Warehouse is a flexible and efficient way to handle great data amount and to get information that assists in the decision-making process. In sight of the construction process of this environment to be very laborious and it demands great amount of time and money, this work presents a simple tool with focus in users and data designers. The tool was developed in Java guaranteeing a portable system that implements the main phases of a data warehouse project: extraction, transformation and loading; consulting, analyzing and information treatment.

Key-words: Data warehouse. Extraction, transformation and load. Decision support system.

LISTA DE ILUSTRAÇÕES

Figura 1 – Estrutura interna do DW	17
Quadro 1 – Diferenças entre o nível operacional e o nível para suporte a decisão	18
Figura 2 – Característica de dados baseados em assunto	19
Figura 3 – Característica de um ambiente integrado	19
Figura 4 – Característica de não volatilidade	20
Figura 5 – Modelagem dimensional	21
Figura 6 – Camadas e componentes de um DW	23
Quadro 2 – Requisitos funcionais.....	27
Quadro 3 – Requisitos não funcionais.....	27
Figura 7 – Diagrama de caso de uso do usuário	28
Quadro 4 – Detalhamento do caso de uso UC001 – Efetuar Login	29
Quadro 5 – Detalhamento do caso de uso UC008 – Visualizar e Configurar Consultas	29
Quadro 6 – Detalhamento do caso de uso UC015 – Efetuar Logout	30
Figura 8 – Diagrama de caso de uso para o administrador.....	30
Quadro 7 – Detalhamento do caso de uso UC002 – Cadastrar Data Warehouse.....	31
Quadro 8 – Detalhamento do caso de uso UC004 – Cadastrar Dimensão	32
Quadro 9 – Detalhamento do caso de uso UC005 – Cadastrar Cubo.....	32
Quadro 10 – Detalhamento do caso de uso UC003 – Cadastrar Conexão	33
Quadro 11 – Detalhamento do caso de uso UC005 – Cadastrar Fonte de Dados	34
Quadro 12 – Detalhamento do caso de uso UC007 – Cadastrar Consulta	35
Quadro 13 – Detalhamento do caso de uso UC013 – Importar Metadados	35
Quadro 14 – Detalhamento do caso de uso UC014 – Exportar Metadados	36
Quadro 15 – Detalhamento do caso de uso UC009 – Visualizar Agendamentos	36
Quadro 16 – Detalhamento do caso de uso UC010 – Visualizar Log de Mensagens	36
Quadro 17 – Detalhamento do caso de uso UC011 – Limpar Data Warehouse	36
Quadro 18 – Detalhamento do caso de uso UC012 – Cadastrar Usuários	37
Figura 9 – Diagrama de pacotes	38
Quadro 19 – Descrição dos pacotes do sistema.....	39
Figura 10 - Diagrama de classes do pacote negócio – ETC	40
Figura 11 – Diagrama de classes do pacote negócio – PROJETO	41
Figura 12 – Diagrama de classes do pacote negócio - CONSULTA	42

Figura 13 – Diagrama de classes do pacote filtro.....	43
Figura 14 – Diagrama de classes do pacote conexão	43
Figura 15 – Diagrama de classes do pacote menu.....	43
Figura 16 – Diagrama de classes do pacote servlet.....	44
Figura 17 – Diagrama de seqüência relacionado ao processo de ETC.....	45
Figura 18 – Diagrama de seqüência relacionado ao processo de PROJETO	46
Figura 19 – Diagrama de seqüência relacionado ao processo de CONSULTA.....	47
Quadro 20 – Implementação dos <i>grids</i> Zapatec	48
Quadro 21 – Implementação do AJAX	50
Quadro 22 – Implementação da exportação e importação XML.....	51
Quadro 23 – Método de importação dos dados da classe FonteDados.....	53
Figura 20 – Tela de login.....	54
Figura 21 – Tela de cadastro de projetos de DW	54
Figura 22 – Tela do menu principal da ferramenta para usuários com perfil administrador ...	55
Figura 23 – Tela de cadastro de conexão	55
Figura 24 – Tela de cadastro de dimensão	56
Figura 25 – Tela de cadastro de atributo	56
Figura 26 – Tela de definição de chave primária da dimensão	57
Figura 27 – Tela de cadastro do cubo.....	57
Figura 28 – Tela de cadastro de indicador.....	58
Figura 29 – Tela de definição de dimensões do cubo	58
Figura 30 – Tela de cadastro de fonte de dados	59
Figura 31 – Tela de roteiro de uma fonte de dados	60
Figura 32 – Tela de importação manual de dados	60
Figura 33 – Tela de cadastro de consultas.....	61
Figura 34 – Tela de cadastro de eixos da consulta	61
Figura 35 – Tela de cadastro de eixos de indicadores da consulta	62
Figura 36 – Tela de definição da consulta com <i>drill-down</i>	62
Figura 37 – Tela de consultas para perfil de usuários	63
Figura 38 – Tela de consulta de vendas por ano com <i>drill-down</i>	63
Figura 39 – Tela de consulta de vendas por ano e mês com <i>drill-down</i>	64
Figura 40 – Tela de consulta de vendas por ano, mês e estado com <i>drill-down</i>	64
Figura 41 – Tela de cadastro de alerta para consultas	65
Figura 42 – Tela de consulta de vendas por representante com alerta	65

Figura 43 – Tela de consulta de vendas por representante e clientes com alerta	66
Figura 44 – Tela de cadastro de ranking para consultas.....	66
Figura 45 – Tela de cadastro de filtro para consultas	67
Figura 46 – Tela de consulta de vendas por produto com filtro e ranking.....	67
Quadro 24 – Quadro de resultados da consulta venda por data e estado.....	68
Quadro 25 – Quadro de resultados da consulta venda por representante e cliente	68
Quadro 26 – Quadro de resultados da consulta venda por produto.....	68
Quadro 27 – Quadro de resultados do processo de ETC.....	69
Quadro 28 – Modelo entidade relacional criado na ferramenta de DW	70
Quadro 29 – Atributos e métodos da classe Agendamento.....	75
Quadro 30 – Atributos e métodos da classe Alerta	76
Quadro 31 – Atributos e métodos da classe Atributo.....	77
Quadro 32 – Atributos e métodos da classe Conexao	78
Quadro 33 – Atributos e métodos da classe Consulta	78
Quadro 34 – Atributos e métodos da classe Cubo.....	79
Quadro 35 – Atributos e métodos da classe Dimensao	79
Quadro 36 – Atributos e métodos da classe Dw.....	80
Quadro 37 – Atributos e métodos da classe Eixo.....	81
Quadro 38 – Atributos e métodos da classe Filtro.....	81
Quadro 39 – Atributos e métodos da classe FonteDados	82
Quadro 40 – Atributos e métodos da classe Indicador	82
Quadro 41 – Atributos e métodos da classe Mensagem.....	83
Quadro 42 – Atributos e métodos da classe Ranking.....	83
Quadro 43 – Atributos e métodos da classe Usuario.....	84
Quadro 44 – Atributos e métodos da classe TipoAgendamento	84
Quadro 45 – Atributos e métodos da classe TipoBanco.....	84
Quadro 46 – Atributos e métodos da classe TipoBanco.....	85
Quadro 47 – Atributos e métodos da classe TipoEixo	85
Quadro 48 – Atributos e métodos da classe TipoIcône	85
Quadro 49 – Atributos e métodos da classe TipoRanking	85
Quadro 50 – Atributos e métodos da classe TipoUsuario	85
Quadro 51 – Tabela dw001 – Data Warehouse.....	86
Quadro 52 – Tabela dw002 - Conexão.....	86
Quadro 53 – Tabela dw003 - Dimensão	86

Quadro 54 – Tabela dw004 - Atributo.....	86
Quadro 55 – Tabela dw005 - Cubo	87
Quadro 56 – Tabela dw006 - Indicador.....	87
Quadro 57 – Tabela dw007 – Dimensão x Cubo.....	87
Quadro 58 – Tabela dw008 - Usuário	87
Quadro 59 – Tabela dw009 – Usuário x Consulta.....	87
Quadro 60 – Tabela dw010 - Consulta.....	87
Quadro 61 – Tabela dw011 – Fonte Dados	88
Quadro 62 – Tabela dw013 - Roteiro	88
Quadro 63 – Tabela dw014 - Agendamento.....	88
Quadro 64 – Tabela dw015 - Eixo.....	88
Quadro 65 – Tabela dw016 - Ranking.....	88
Quadro 66 – Tabela dw017 - Alerta	89
Quadro 67 – Tabela dw018 - Filtro	89
Quadro 68 – Tabela dw019 – Mensagem.....	89
Quadro 69 – Script SQL da ferramenta	97

LISTA DE SIGLAS

AJAX – *Asynchronous JavaScript and XML*

BI – *Business Intelligence*

DBF – *DataBase Files*

DM – *Data Mart*

DW – *Data Warehouse*

EA – *Enterprise Architect*

EIS – *Executive Information System*

ERP – *Enterprise Resources Planning*

ETC – *Extração, Transformação e Carga*

HTML – *HyperText Markup Language*

HTTP – *HyperText Transfer Protocol*

IDE – *Integrated Development Environment*

JDBC – *Java Database Connectivity*

JSP – *Java Server Pages*

MIS – *Management Information System*

MVC – *Model-View-Controller*

ODBC – *Open Database Connectivity*

OLAP – *On Line Analytical Processing*

OLTP – *On Line Transaction Processing*

SAD – *Sistema de Apoio a Decisão*

SQL – *Structure Query Language*

UML – *Unified Modeling Language*

XML - *eXtensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 OBJETIVOS DO TRABALHO	15
1.2 ESTRUTURA DO TRABALHO	16
2 FUNDAMENTAÇÃO TEÓRICA	17
2.1 DATA WAREHOUSE.....	17
2.2 CARACTERÍSTICAS DO DATA WAREHOUSE.....	18
2.3 METADADOS	20
2.4 MODELAGEM DIMENSIONAL	21
2.5 EXTRAÇÃO, TRANSFORMAÇÃO E CARGA	22
2.6 TRABALHOS CORRELATOS	23
3 DESENVOLVIMENTO DA FERRAMENTA	25
3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO.....	25
3.2 ESPECIFICAÇÃO	27
3.2.1 Diagrama de caso de uso.....	27
3.2.2 Diagrama de pacotes	37
3.2.3 Diagrama de classes	39
3.2.4 Diagrama de seqüência	44
3.3 IMPLEMENTAÇÃO	47
3.3.1 Técnicas e ferramentas utilizadas.....	48
3.3.2 Operacionalidade da implementação	53
3.4 RESULTADOS E DISCUSSÃO	68
4 CONCLUSÕES.....	71
4.1 EXTENSÕES	72
REFERÊNCIAS BIBLIOGRÁFICAS	73
APÊNDICE A – Relação das classes do pacote de negócios, seus atributos e métodos ...	75
APÊNDICE B – Relação das tabelas que compõe o modelo da ferramenta.....	86
APÊNDICE C – Script SQL da ferramenta	90

1 INTRODUÇÃO

A crescente competição em mercados cada vez mais dinâmicos está levando as empresas a tomarem decisões mais rapidamente. Sendo assim a informação tornou-se o bem mais valioso dentro das instituições. Os administradores tomam suas decisões com base na análise de dados objetivos, sintetizados e confiáveis acima de tudo, sempre com o intuito maior de melhorar e aperfeiçoar processos internos. É dentro deste cenário que hoje se torna imprescindível a utilização de recursos computacionais para levantar e analisar as informações necessárias. Uma das principais ferramentas que constitui a nova geração de Sistemas de Apoio a Decisão (SAD) é o *Data Warehouse* (DW): um banco de dados específico para propósitos gerenciais e estratégicos (DW BRASIL, 2005).

Para entender melhor o que é um DW, é relevante que se faça uma comparação com o conceito tradicional de um banco de dados transacional ou operacional, que armazena dados num determinado período onde são registradas e executadas operações pré-definidas. Os diversos sistemas de operação, cada vez mais sofisticados, registram grandes volumes de dados sobre diversas áreas da organização. As dificuldades em buscar informações e gerar conhecimento a partir destes dados, representam um enorme desafio que, se não superado, gera prejuízos devido às oportunidades perdidas e decisões errôneas. Já, em contrapartida, um banco de dados de DW armazena dados analíticos voltados à tomada de decisões podendo envolver consultas complexas que acessem um grande número de dados, exigindo assim um eficaz sistema de acesso ao dado. Neste caso existe uma grande preocupação em obter dados históricos, o que determina um grande processamento e armazenamento dos dados sempre detalhados e resumidos.

Para Inmon, Terderman e Imhoff (2001) o ambiente criado pela utilização da tecnologia de DW permite às empresas que integrem dados de diversos sistemas e módulos distintos e criem oportunidades de negócios a partir de sua utilização.

Para Kimball (1998 apud COME, 2001, p. 2), DW é o lugar onde as pessoas podem acessar seus dados. A abordagem de Ralph Kimball veio com um estilo mais simples e incremental, baseado na metodologia estrela que aponta para *Data Marts* (DM) separados, que deverão ser integrados na medida da sua evolução (BARBIERI, 2001). Já Wang (1998 apud COME, 2001, p. 2) tem uma definição um pouco mais elaborada quando diz que DW é o processo pelo qual os dados relacionados de vários sistemas operacionais são fundidos para proporcionar uma única e integrada visão de informação de negócios que abrange todas as

divisões da empresa.

Em meio a este cenário, está sendo proposto desenvolver uma ferramenta de DW que irá auxiliar seus usuários no processo de transformação de dados operacionais em informações gerenciais, viabilizando consultas em formato de tabela ou de gráficos em diversos níveis de detalhe. A ferramenta é totalmente executável em ambiente web, desta forma acessível através dos principais navegadores hoje disponíveis no mercado. Contemplará questões importantes para a área de Computação, pois se pretende não apenas construir um DW específico, mas fornecer uma ferramenta genérica para a construção e implantação de um DW, sem perder de vista sua usabilidade, dando suporte tanto ao projetista do DW quanto ao seu usuário, além de disponibilizá-la para uso didático aos alunos e professores das disciplinas de Banco de Dados do curso de Ciência da Computação e Sistemas de Informação

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é desenvolver uma ferramenta web para transformação de dados operacionais provenientes de uma única ou múltiplas fontes em informações sumarizadas e analíticas para auxiliar o usuário na tomada de decisões.

Os objetivos específicos do trabalho são:

- a) permitir a construção de projetos DW e seus modelos *Star Schema*¹ utilizando a técnica de modelagem dimensional que visa construir o cubo de decisão especificando a tabela fato e seus indicadores para a área de negócio, as dimensões com seus respectivos atributos que definem os níveis de granularidade e a fonte de dados que determina a origem das informações;
- b) separar o uso da ferramenta em usuários finais, que farão gerenciamento de consultas e administradores que terão controle total sobre todo o projeto de DW;
- c) permitir a consulta aos dados gerenciais através de tabelas, com recursos de *Drill Down*², *rankings*, alertas e filtros;
- d) exportar o metadados do modelo de DW para integração com outras ferramentas

¹ *Star Schema* ou esquema em estrela é uma metodologia de modelagem de dados, onde os dados são modelados em tabelas dimensionais ligadas a uma tabela de fatos (MOREIRA, 2006).

² *Drill Down* é uma técnica que possibilita fazer um particionamento nas informações, ou seja, partir de dados globais, consolidados, e gradativamente fazer um detalhamento dos mesmos (BISPO, 1998, p. 117).

de DW ou para uso como *templates* em novos projetos.

1.2 ESTRUTURA DO TRABALHO

O trabalho foi dividido em quatro capítulos. No segundo, é descrita a fundamentação teórica abordando assuntos relacionados ao DW, modelagem dimensional e técnicas de extração, transformação e carga de dados. Neste capítulo também são apresentados trabalhos correlatos. No terceiro, é apresentada a especificação da ferramenta assim como sua implementação e os resultados e discussões sobre a mesma. No quarto, é apresentada a conclusão do trabalho assim como extensões para futuros trabalhos.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo descreve os principais conceitos envolvidos no tema do trabalho apresentado da seguinte forma: conceito de DW comparando-o com o ambiente operacional, características do DW, metadados, modelagem dimensional, processo de ETC e trabalhos correlatos.

2.1 DATA WAREHOUSE

O termo Data Warehouse significa armazém de dados. É definido como um ambiente que provê informações de suporte à decisão que, no ambiente operacional, se tornariam difíceis de serem obtidas. Em outras palavras, pode ser comparado como um banco de dados especial, estruturado de forma a facilitar o processamento para análise dos dados.

O conceito de DW surgiu da necessidade de integrar dados corporativos espalhados em diferentes máquinas e sistemas operacionais, para torná-los acessíveis a todos os usuários dos níveis decisórios (NAVARRO, 1996). Entretanto, essa integração deve ser feita com uma seleção cuidadosa e otimizada dos dados já que a prioridade na utilização do ambiente do DW é o processamento de consultas e não o processamento de transações. A Figura 1 ilustra toda a estrutura interna que o ambiente de DW representa.

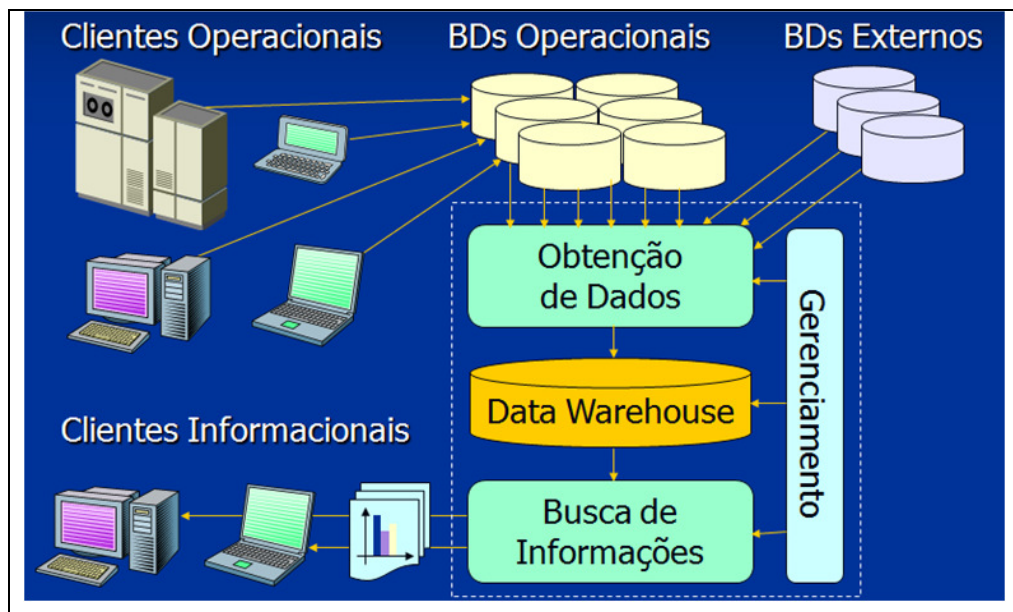


Figura 1 – Estrutura interna do DW

Geralmente, os sistemas do ambiente operacional que apóiam os usuários em suas funções do dia-a-dia são chamados *On Line Transaction Processing* (OLTP) e seu principal objetivo é executar o maior número de transações possíveis no menor tempo de processamento. Em sistemas de suporte a decisão, onde o volume de dados costuma ser muito maior e as consultas altamente complexas, são necessárias ferramentas altamente flexíveis e customizáveis. Para atender essa necessidade são adotados os sistemas *On Line Analytical Processing* (OLAP). Sistemas OLAP permitem aos usuários de alto nível, como gerentes e analistas de negócio, navegarem entre os dados da empresa com maior facilidade, proporcionando uma visão multidimensional desses dados.

São várias as diferenças entre sistemas para o nível operacional e para suporte à tomada de decisão, como se pode observar no Quadro 1.

Tópico ou função	Nível operacional	Nível suporte a decisão
Conteúdo dos dados	Valores correntes	Dados históricos, consolidados e trabalhados
Organização dos dados	Orientada à aplicação	Orientada à informação
Natureza dos dados	Dinâmica, dados normalizados	Estática Dados desnormalizados
Estrutura e formato dos dados	Complexos, desejáveis para computação operacional	Simples, desejáveis para análises de negócio
Possibilidade de acesso	Alta	Moderada e baixa
Atualização	Contínua	Periódica
Aplicação	Estruturada, processamento repetitivo	Não estruturada, processamento analítico
Tempo de resposta	Entre dois e trinta segundos	Segundos a minutos

Fonte: Navarro (1996).

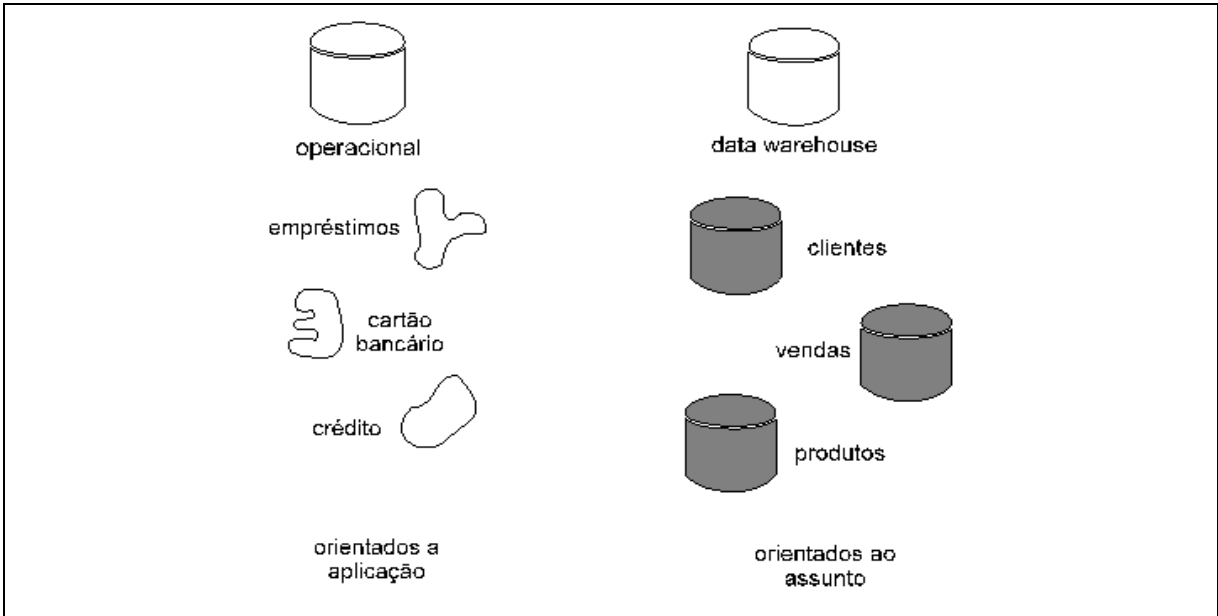
Quadro 1 – Diferenças entre o nível operacional e o nível para suporte a decisão

2.2 CARACTERÍSTICAS DO DATA WAREHOUSE

Segundo Inmon (1997, p. 33), um “data warehouse é um conjunto de dados baseado em assuntos, integrado, não-volátil, e variável em relação ao tempo, de apoio às decisões gerenciais”.

A primeira característica, que define um DW ser baseado em assuntos, quer dizer que esse conjunto de dados deve estar organizado em torno das áreas específicas em que as aplicações da empresa estão focalizando.

A Figura 2 ilustra esta característica apresentando objetos dentro de uma aplicação que são transformados em repositórios de assuntos dentro do DW.

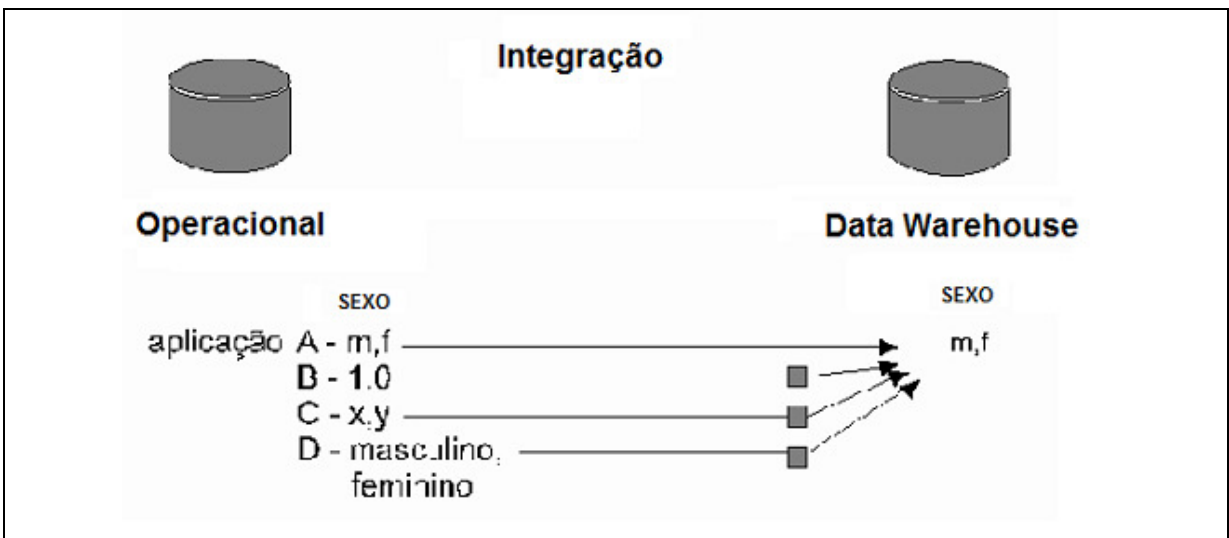


Fonte: Inmon (1997).

Figura 2 – Característica de dados baseados em assunto

O fato de o DW ser um ambiente integrado faz com que os dados contidos nesse conjunto de dados, devem estar em apenas um formato de forma a se manter a consistência. Como o DW integra dados vindos de várias fontes de dados, que tratam seus dados de maneiras diferente umas das outras, é importante que esses dados estejam com uma representação única.

A Figura 3 ilustra a transformação de dados que possuem diferentes tratamentos em um ambiente operacional e que são integrados dentro do DW.



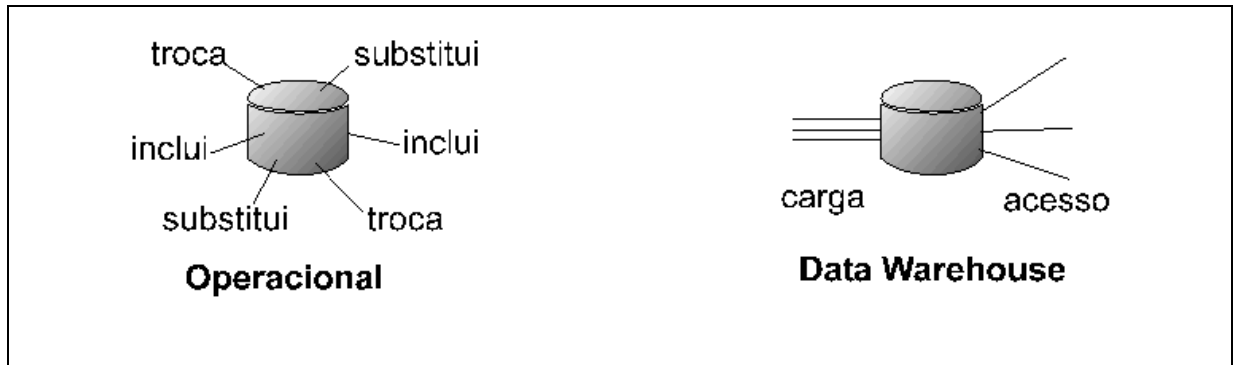
Fonte: Inmon (1997).

Figura 3 – Característica de um ambiente integrado

A terceira característica consiste na não volatilidade dos dados, ou seja, depois de trazidos para o DW, geralmente, os dados não sofrem mais modificações. Isso se dá pelo fato de que, modificações nos dados de um ambiente, geralmente, causam um trabalho grande por

requerer processos de controle de consistência e integridade, além de atividades como bloqueios e recuperação de falhas que em DW não é algo tão simples de se implementar.

A Figura 4 ilustra a diferença de operações executadas dentro de um ambiente operacional e um ambiente de DW.



Fonte: Inmon (1997).

Figura 4 – Característica de não volatilidade

Por último, a característica de o DW ser variável em relação ao tempo faz com que seus dados são relativos ao momento em que foram carregados. Normalmente a estrutura de chave do DW sempre contém um elemento de tempo relacionado. Por isso os dados do DW normalmente têm horizontes de tempo bem maiores do que os dos sistemas transacionais.

2.3 METADADOS

Pode-se definir como metadados de um DW, os dados do dado, ou seja, este deve conter muito mais do que as descrições de colunas e tabelas, devendo conter informações que adicionem valor aos dados. O metadados procura centralizar as principais informações que um projeto de DW possui, e como todo sistema, a documentação do DW através do metadados é de extrema importância para o entendimento de um projeto (VIEIRA, 2000).

De acordo com Vieira (2000) algumas informações que o metadados deve conter são:

- a) a estrutura dos dados segundo a visão do programador e dos analistas de SAD;
- b) a origem das fontes de dados que alimentam o DW;
- c) a transformação dos dados ocorrida no processo de migração para o DW;
- d) o modelo de dados e seu relacionamento com o DW;
- e) o histórico das extrações de dados;
- f) as informações sobre as consultas e relatórios;

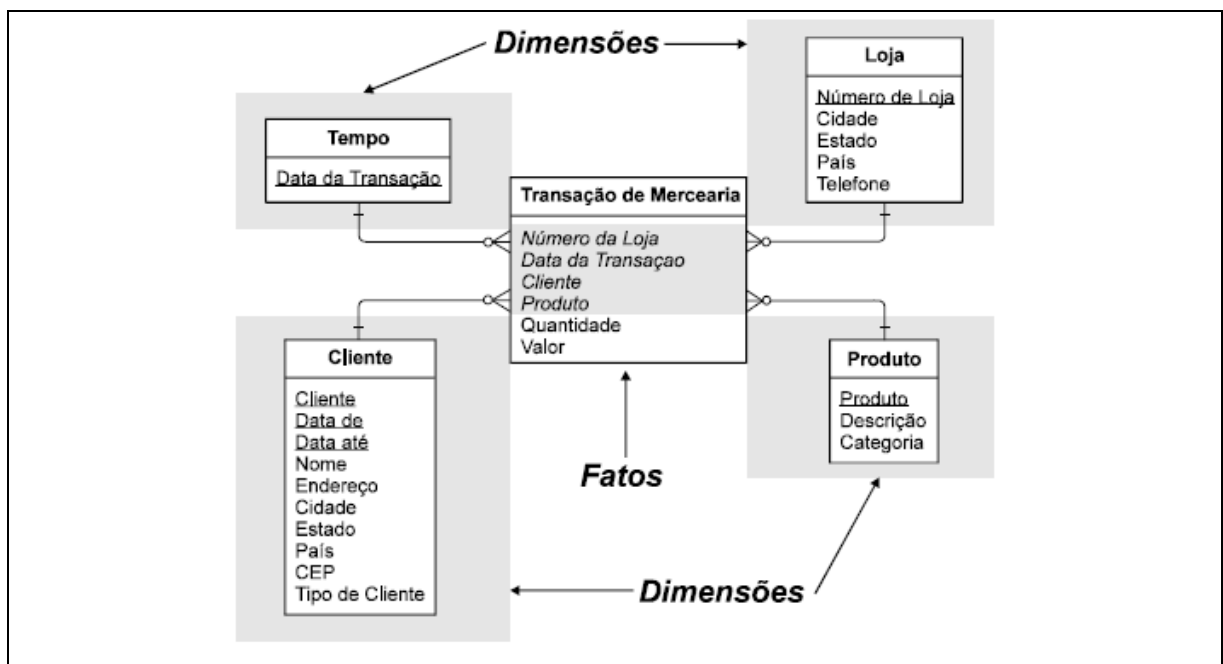
- g) acesso e segurança;
- h) os indicadores de qualidade de dados.

2.4 MODELAGEM DIMENSIONAL

Uma das técnicas utilizadas para a criação do projeto lógico de um DW é a da modelagem dimensional. Esta técnica é caracterizada pela criação do esquema estrela a partir do esquema conceitual criado na fase de análise do DW.

Para Kimball (1997 apud COME, 2001, p. 51) modelagem dimensional é uma técnica utilizada para a definição do projeto lógico de um DW. Três conceitos básicos são importantes nesta modelagem: tabelas fatos ou cubos de decisão que representam as transações de negócios, as dimensões que são os diferentes tipos de visões que os usuários irão utilizar para analisar as métricas e os indicadores ou métricas que podem ser definidos como os atributos numéricos de um fato representando o comportamento de um negócio para as dimensões.

A Figura 5 representa exatamente cada conceito dentro de uma modelagem dimensional.



Fonte: Tronchin (1998 apud COME, 2001, p. 52).

Figura 5 – Modelagem dimensional

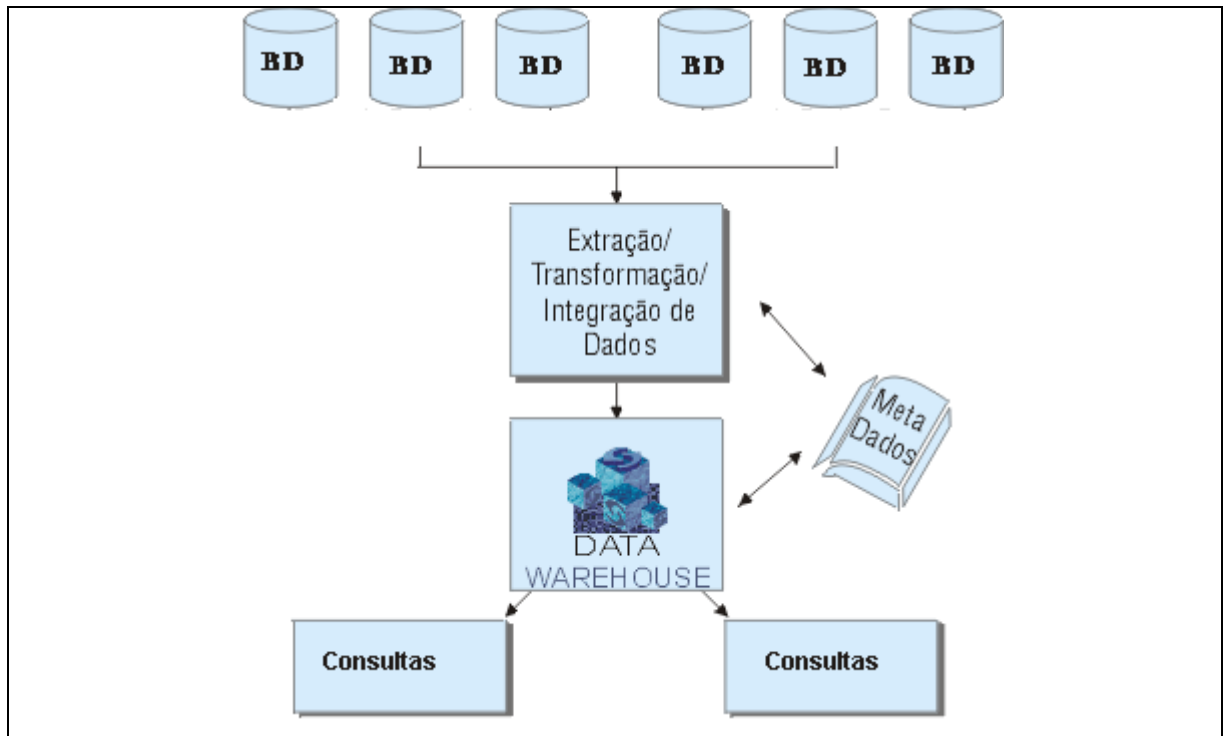
2.5 EXTRAÇÃO, TRANSFORMAÇÃO E CARGA

O processo de Extração, Transformação e Carga (ETC) é o mais importante e trabalhoso na construção de um DW. Durante essa etapa é importante ter uma eficiente integração de dados, já que os mesmos podem vir de múltiplas fontes. Sua transformação deve ser feita de forma a gerar informações consistentes e de qualidade. Essa etapa é caracterizada por ser uma das mais críticas já que uma informação carregada erroneamente trará conseqüências imprevisíveis nas fases posteriores (SILVA, 2005, p. 19).

Durante o processo de ETC, segundo Silva (2005, p. 19) cada etapa possui suas características e dificuldades:

- a) extração de dados: os dados podem vir de diferentes fontes e sistemas além de estar disponibilizados em diferentes plataformas e tecnologias, o que demanda, em alguns casos, modos de extração diferenciados. Os dados extraídos devem ser selecionados segundo o modelo de dados gerado pela etapa de modelagem do DW. Em decorrência disso, a extração dos dados torna-se muito dependente da qualidade com a qual a modelagem foi realizada. É importante também se obter dados os mais recentes possíveis para garantir que o DW contenha informações sempre atualizadas;
- b) transformação dos dados: um dos tratamentos durante esta etapa é a checagem da integridade dos dados, verificando se chaves estrangeiras realmente existem nas respectivas tabelas. Outro aspecto a ser observado é a padronização dos dados que serão incluídos no DW já que estes podem ser de diferentes sistemas, tendo, portanto, formatos diferentes. Também a limpeza dos dados tratando possíveis inconsistências e incompatibilidades, eliminação de dados inúteis e sumarização são importantes nesta etapa;
- c) carga dos dados: é nesta etapa que são realizados os tratamentos dos dados rejeitados. Campos que não possuem dados devem apresentar um valor padrão. Há duas formas de efetuar a carga dos dados: incremental, onde os dados são atualizados com novas informações sendo incluídas, e a total, onde todos os registros são excluídos e inseridos novamente.

Para ilustrar melhor todos os componentes de um DW, a Figura 6 representa todas as camadas que envolvem a integração de um ambiente operacional até o ambiente de DW.



Fonte: Navarro (1996).

Figura 6 – Camadas e componentes de um DW

2.6 TRABALHOS CORRELATOS

As principais ferramentas de DW disponíveis hoje no mercado são muitas vezes desenvolvidas pelas grandes empresas de banco de dados operacionais, como a Oracle e a própria Microsoft com o SQL. Porém, diversas empresas hoje investem em ferramentas de apoio aos seus sistemas *Enterprise Resources Planning* (ERP), como o SigaDW da Microsiga Software.

O Oracle Discoverer é uma ferramenta de *Business Intelligence* (BI) para apoio à tomada de decisão empresarial. As consultas realizadas no Oracle Discoverer podem ser exibidas em formato de tabela ou matriz, permitindo a definição de campos de seleção de página para o filtro dos dados apresentados. Várias facilidades de formatação de campos e modificação de *layout* das consultas estão disponíveis, permitindo assim, que formatações condicionais de fontes e cores sejam definidas para cada uma das colunas consultadas. Além disso, existe a possibilidade de movimentação, ordenação e agrupamento das colunas de dados além de permitir a exportação dos relatórios formatados para arquivos HTML e para o Microsoft Excel, além de outros formatos possíveis (MARQUES, 2005).

A nova versão do SQL Server 2005 da Microsoft contém dois novos componentes de apoio ao processo de DW, os quais são: SQL Server 2005 Business Intelligence Development Studio e o SQL Server 2005 Management Studio. Alguns recursos disponíveis são: análise OLAP, integração com serviços de ETC, relatórios, DW e *Data Mining*. O SQL Server 2005 inclui avanços significativos ao gerenciamento de dados corporativos nas seguintes áreas: disponibilidade, escalabilidade, segurança, gerenciamento e interoperabilidade (MICROSOFT BRASIL, 2005).

A Microsiga Software, empresa do setor de ERP, possui um recurso de construção de DW cuja principal característica é a integração de sua linguagem proprietária dentro das consultas de extração de dados, permitindo assim utilizar recursos de funções prontas para setores empresariais na montagem da carga de dados. Com o SigaDW, a Microsiga passa a oferecer um produto que segue o padrão de mercado de DW, oferecendo em uma mesma ferramenta opções de ETC e visualização das consultas. Outras características são: modelagem dimensional utilizando as técnicas de esquema estrela e cubo de decisão, totalmente executável em ambiente web, desenvolvido na linguagem padrão da Microsiga e com componentes em Java, acessa a diversos bancos de dados através de conexões *Open Database Connectivity* (ODBC) ou acesso direto a tabelas do tipo *DataBase Files* (DBF) e está totalmente incorporado ao ERP da Microsiga (MICROSIGA SOFTWARE, 2006).

Em pesquisas feitas sobre trabalhos já existentes, foram encontrados alguns trabalhos onde o objetivo era o desenvolvimento de sistemas de informação baseados em técnicas de DW.

Jark (2002) desenvolveu um *Executive Information System* (EIS) utilizando DW, mais especificamente a técnica de cubos de decisão, com o objetivo de gerenciamento de clientes, utilizando dados sobre o setor comercial e financeiro, onde apresenta todo o histórico de pedidos, devoluções dos pedidos, produtos mais vendidos, vendedores, clientes, entre outros.

Já Corradi (2002) implementou um *Management Information System* (MIS) aplicado à comercialização de planos de previdência privada, efetuando a análise da informação através da técnica de cubos de decisão.

3 DESENVOLVIMENTO DA FERRAMENTA

Este capítulo descreve o desenvolvimento da ferramenta de construção de DW, destacando como foi feita a implementação de cada fase de um projeto de DW e procurando mostrar como cada camada da arquitetura *Model-View-Controller* (MVC) foi desenvolvida. As etapas do desenvolvimento do trabalho foram:

- a) levantamento dos requisitos: foi feito um estudo sobre as principais ferramentas do mercado como o DW da Microsiga, para descobrir e obter os requisitos básicos, adicionando novos requisitos na especificação;
- b) especificação do sistema: a especificação foi feita com base no estudo da aplicação DW da Microsiga, onde foram elaborados vários diagramas desenvolvidos na análise do sistema;
- c) implementação: com o término da análise, deu-se o início do desenvolvimento em Java utilizando *Java Server Pages* (JSP), *HyperText Markup Language* (HTML), *Servlets* e *Asynchronous JavaScript and XML* (AJAX), além de outras ferramentas.

3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

O desenvolvimento deste trabalho iniciou-se fazendo um levantamento dos requisitos que as principais ferramentas de mercado possuem. A seguir são descritos os requisitos funcionais (RF) e os requisitos não-funcionais (RNF) da ferramenta. Os requisitos funcionais podem ser vistos no Quadro 2 e os não funcionais no Quadro 3.

REQUISITOS FUNCIONAIS
RF001 – O sistema deverá permitir o login de administrador e usuário.
RF002 – O sistema deverá permitir ao administrador montar o modelo <i>star schema</i> através do cadastro dos cubos de decisão, suas dimensões, atributos e indicadores de negócio.
RF003 – O sistema deverá permitir ao administrador criar conexões com banco de dados MySQL e SQL Server suportadas pelas interfaces <i>Java Database Connectivity</i> (JDBC). Esta conexão irá permitir ao sistema que realize o processo de ETC.
RF004 – O sistema deverá permitir ao administrador definir fontes de dados e <i>scripts Structure Query Language</i> (SQL) que processam a origem dos dados.
RF005 – O sistema deverá permitir ao administrador mapear a origem dos dados com os atributos e indicadores do DW, definindo o destino do dado transformado.
RF006 – O sistema deverá transformar os dados operacionais em dados gerenciais conforme definição do cubo de decisão, respeitando as chaves primárias e estrangeiras das dimensões e seus atributos.
RF007 – O sistema deverá importar os dados transformados no banco de dados do DW.
RF009 – O sistema deverá permitir ao administrador definir agendamentos para cada fonte de dados criada no projeto de DW controlando suas execuções através de logs gerados pelo sistema.
RF010 – O sistema deverá permitir ao administrador criar novas consultas do tipo tabela .
RF011 – O sistema deverá permitir ao administrador definir os eixos da tabela e os indicadores e suas funções de agregação podendo ser soma, média, contagem, mínimo ou máximo.
RF012 – O sistema deverá permitir ao administrador definir os níveis de <i>Drill Down</i> das consultas.
RF013 – O sistema deverá permitir ao administrador ou usuário criar <i>ranking</i> nas consultas (organizar indicadores em n maiores ou menores), alertas (destacar condições pré-definidas em caso de ser verdadeiro ou falso) e filtros.
RF014 – O sistema deverá permitir ao administrador exportar e importar o metadados em formato <i>eXtensible Markup Language</i> (XML).
RF015 – O sistema deverá permitir ao administrador a manutenção de usuários e seus privilégios.
RF016 – O sistema deverá implementar interfaces independentes para o usuário e para o administrador.

RF017 – O sistema deverá permitir ao administrador visualizar as mensagens de <i>log</i> geradas pelo sistema durante o processo de importação de dados.
RF018 – O sistema deverá permitir ao administrador visualizar os agendamentos em aberto do dia para todas as fontes de dados criadas no DW, podendo administrar os melhores horários de execução.

Quadro 2 – Requisitos funcionais

REQUISITOS NÃO FUNCIONAIS
RNF001 – O sistema deverá rodar em ambiente web.
RNF002 – O sistema deverá ser implementado na linguagem Java utilizando JSP e <i>Servlet</i> .
RNF003 – O sistema deverá utilizar AJAX para melhor interatividade.
RNF004 – O sistema deverá possuir uma arquitetura dividida em camadas.
RNF005 – O sistema deverá ser independente de plataforma.

Quadro 3 – Requisitos não funcionais

3.2 ESPECIFICAÇÃO

A especificação da ferramenta de construção de DW foi feita através de análise orientada a objetos utilizando diagramas UML e a ferramenta Enterprise Architect (EA) para gerar os mesmos. Nesta fase foram elaborados os diagramas de caso de uso, pacotes, classes e seqüência.

3.2.1 Diagrama de caso de uso

Os diagramas de caso de uso são utilizados para ilustrar e descrever as principais operações que um usuário ou dispositivo interagindo com o sistema poderá executar. Os principais elementos que compõem estes diagramas são: um boneco que especifica um ator, uma elipse que representa um caso de uso e linhas que conectam dois casos de uso representando uma dependência entre eles. Todo caso de uso deve possuir uma descrição do mesmo. Esta descrição deve informar todos os passos fundamentais para que a interação ocorra no caso de uso.

Na especificação de casos de uso para a ferramenta de construção de DW existem dois atores envolvidos: o usuário de consultas e o administrador projetista.

A Figura 7 ilustra o diagrama de caso de uso em um cenário em que o usuário pode efetuar o login, visualizar e configurar consultas, e efetuar o logout. O detalhamento destes casos de uso é apresentado nos Quadro 4, Quadro 5 e Quadro 6.

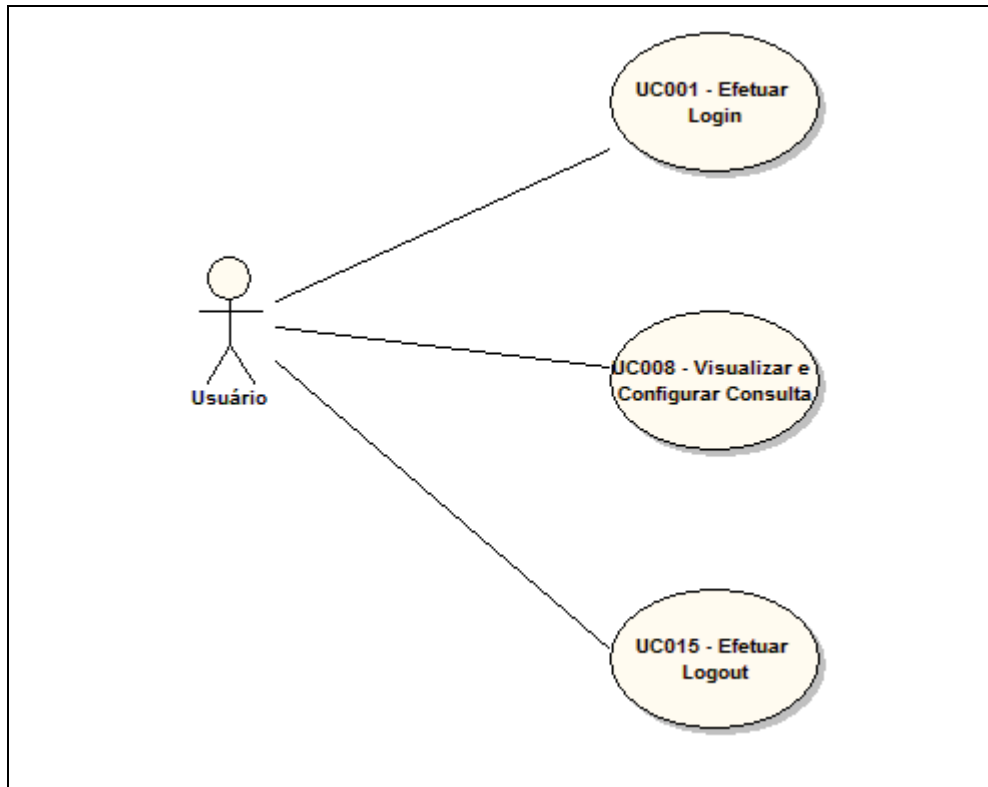


Figura 7 – Diagrama de caso de uso do usuário

UC001 – Efetuar Login

Objetivo: Permitir a um usuário ou administrador, através de uma conta e senha, acesso ao sistema.

Pré-Condição: Sistema ter conta do usuário cadastrada.

Cenários

Fluxo Principal {Principal}.

1. Sistema mostra uma tela para entrada das informações de login.
2. Usuário informa conta de login.
3. Usuário informa senha de acesso.
4. Usuário confirma acesso.
5. Sistema valida acesso e redireciona usuário de acordo com o tipo de acesso: usuário ou administrador.

Usuário Inválido {Exceção}.

1. Sistema notifica que informações de conta fornecidas pelo usuário são inválidas.

Tempo de acesso expirado {Exceção}.

1. Em caso do usuário ficar com uma sessão de acesso sem uso, o sistema entrará em timeout apresentando ao usuário uma mensagem de sessão expirada.

Pós-Condição: O sistema disponibiliza ao usuário acesso às consultas que o mesmo possui privilégio de acesso, e no caso do administrador, acesso a criar novos projetos de data warehouse.

Quadro 4 – Detalhamento do caso de uso UC001 – Efetuar Login

UC008 – Visualizar e Configurar Consulta

Objetivo: Poder visualizar as consultas agregando recursos de filtro, ranking ou alertas.

Pré-Condição: Possuir consultas cadastradas no sistema.

Cenários

Visualizar Consulta {Principal}.

1. O Usuário clica no ícone visualizar referente à consulta.
2. O Sistema apresentará a tabela de dados com ou sem recurso de drill down, filtros, alertas e ranking conforme definido na consulta.

Incluir filtros {Alternativo}.

1. O Usuário clica no ícone filtro referente à consulta.
2. O Usuário informa a expressão.
3. O Usuário confirma o cadastro.
4. O Sistema mostra uma mensagem de confirmação.
5. O Usuário poderá alterar ou excluir filtros definidos para uma consulta.

Incluir alertas {Alternativo}.

1. O Usuário clica no ícone alerta referente à consulta.
2. O Usuário informa o indicador.
3. O Usuário informa a expressão.
4. O Usuário informa a cor de Fonte e Fundo em caso da expressão ser verdadeiro.
5. O Usuário informa a cor de Fonte e Fundo em caso da expressão ser falso.
6. O Usuário confirma o cadastro.
7. O Sistema mostra uma mensagem de confirmação.
8. O Usuário poderá alterar ou excluir alertas definidos para uma consulta.

Incluir ranking {Alternativo}.

1. O Usuário clica no ícone ranking referente à consulta.
2. O Usuário informa o indicador.
3. O Usuário informa o tipo de ranking.
4. O Usuário informa a quantidade de registros que serão apresentados.
5. O Usuário confirma o cadastro.
6. O Sistema mostra uma mensagem de confirmação.
7. O Usuário poderá alterar ou excluir rankings definidos para uma consulta.

Quadro 5 – Detalhamento do caso de uso UC008 – Visualizar e Configurar Consultas

UC015 – Efetuar Logout

Objetivo: Sair do sistema

Pré-Condição: Usuário estar logado no sistema.

Cenários

Fluxo Principal {Principal}.

1. Usuário clica no link logout disponível na tela principal do sistema.

2. O Sistema limpa as sessões que estavam em aberto para aquele usuário

Quadro 6 – Detalhamento do caso de uso UC015 – Efetuar Logout

A Figura 8 ilustra o diagrama de caso de uso em um cenário em que o administrador pode efetuar as principais operações de um projeto de DW.

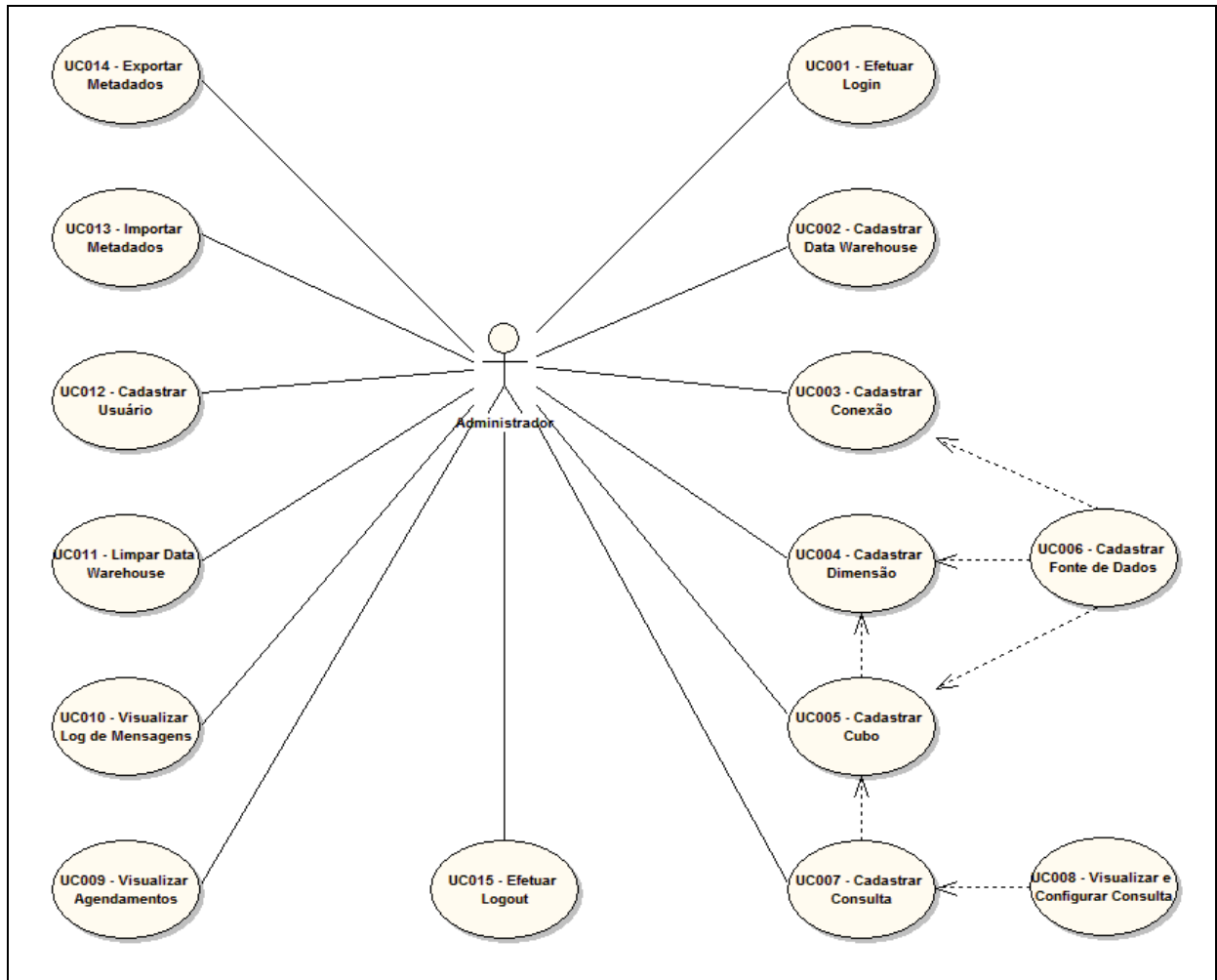


Figura 8 – Diagrama de caso de uso para o administrador

O detalhamento dos casos de uso UC001 e UC015 já estão representados anteriormente pelos Quadro 4 e Quadro 6.

As outras operações do administrador podem ser divididas nos seguintes processos dentro da ferramenta: montagem do projeto de DW, ETC, consultas, metadados e recursos adicionais da ferramenta.

Para a montagem do projeto de DW pode-se ver o detalhamento dos casos de uso UC002, UC004 e UC005 nos Quadro 7, Quadro 8 e Quadro 9.

UC002 – Cadastrar Data Warehouse

Objetivo: Criar um novo projeto de Data Warehouse baseado no modelo dimensional estrela.

Pré-Condição: Estar logado no sistema com o perfil de administrador.

Cenários

Criar novo data warehouse {Principal}.

1. Sistema mostra um ícone com a opção de criar um novo projeto de data warehouse.
2. O Administrador informa um nome para o projeto de data warehouse.
3. O Administrador informa uma descrição para o projeto.
4. O Administrador informa um tipo de ícone para o projeto.
5. O Administrador confirma o cadastro do data warehouse.
6. O Sistema cria o data warehouse e já o apresenta como opção de seleção.

Selecionar data warehouse existente {Alternativo}.

1. Sistema apresenta ícones com os data warehouses já criados.
2. Administrador poderá selecionar um data warehouse já existente para trabalhar.

Pós-Condição: O Sistema disponibiliza um novo ou já existente projeto de data warehouse para construção pelo administrador.

Quadro 7 – Detalhamento do caso de uso UC002 – Cadastrar Data Warehouse

UC004 – Cadastrar Dimensão

Objetivo: Gravar as definições referentes a uma dimensão bem como seus atributos e chave primária.

Pré-Condição: Estar logado no sistema com o perfil de administrador.

Cenários

Incluir nova dimensão {Principal}.

1. Administrador informa o nome.
2. Administrador informa a descrição.
3. Administrador confirma cadastro.

Alterar ou excluir dimensão {Alternativo}.

1. Sistema apresenta todas as dimensões cadastradas.
2. Administrador clica no ícone alterar ou excluir.
3. Sistema informa mensagem de confirmação.

Cadastrar atributos {Principal}.

1. Administrador clica no ícone atributos referente à dimensão.
2. Administrador clica no ícone novo para incluir novos atributos.
3. Administrador informa nome e descrição do atributo.
4. Administrador informa o tipo de dado do atributo.
5. Administrador informa tamanho e decimal em caso de tipo numérico.
6. Administrador poderá ainda editar ou excluir os atributos cadastrados.

Definir chave primária {Principal}.

1. Administrador clica no ícone chave primária referente à dimensão.
2. O sistema apresenta os atributos já cadastrados, disponibilizando do lado esquerdo os atributos que não pertencem a chave e do lado direito os atributos na seqüência que definem a chave primária.
3. Administrador clica nos botões de controle para arrastar os atributos de um lado para o outro.
4. Administrador confirma o cadastro da chave primária.

Pós-Condição: O sistema disponibiliza as dimensões para utilização nos cubos de decisão e define pelo menos um atributo amarrado a dimensão.

Quadro 8 – Detalhamento do caso de uso UC004 – Cadastrar Dimensão

UC005 – Cadastrar Cubo

Objetivo: Gravar as definições referentes a um cubo de decisão bem como seus indicadores e dimensões relacionadas.

Pré-Condição: Possuir dimensões cadastradas no sistema.
Estar logado no sistema com o perfil de administrador.

Cenários

Incluir novo cubo {Principal}.

1. Administrador informa o nome.
2. Administrador informa a descrição.
3. Administrador confirma cadastro.

Alterar ou excluir cubo {Alternativo}.

1. Sistema apresenta todos os cubos cadastrados.
2. Administrador clica no ícone alterar ou excluir.
3. Sistema informa mensagem de confirmação.

Cadastrar indicadores {Principal}.

1. Administrador clica no ícone indicador referente ao cubo.
2. Administrador clica no ícone novo para incluir novos indicadores.
3. Administrador informa nome e descrição do indicador.
4. Administrador informa tamanho e decimal do indicador numérico.
5. Administrador poderá ainda editar ou excluir os indicadores cadastrados.

Relacionar dimensões {Principal}.

1. Administrador clica no ícone dimensões referente ao cubo.
2. O sistema apresenta as dimensões já cadastradas, disponibilizando do lado esquerdo as que não pertencem ao cubo e do lado direito as que já estão relacionadas.
3. Administrador clica nos botões de controle para arrastar as dimensões de um lado para o outro.
4. Administrador confirma relacionamento das dimensões com o cubo

Pós-Condição: O sistema disponibiliza o cubo de decisão para utilização e geração de consultas

Quadro 9 – Detalhamento do caso de uso UC005 – Cadastrar Cubo

Para o processo de ETC pode-se ver o detalhamento dos casos de uso UC003 e UC006 nos Quadro 10 e Quadro 11.

UC003 – Cadastrar Conexão

Objetivo: Criar uma nova conexão com um banco de dados que será disponibilizado para extração de dados para as dimensões e cubos do data warehouse.

Pré-Condição: Administrador deve possuir as informações de acesso ao banco de dados.
Estar logado no sistema com o perfil de administrador.

Cenários

Incluir nova conexão {Principal}.

1. Administrador informa nome e descrição para a nova conexão.
2. Administrador informa o tipo de banco.
3. Administrador informa nome e porta do servidor.
4. Administrador informa login e senha de acesso ao banco.
5. Administrador confirma cadastro de conexão.
6. O Sistema testa a conexão.

Alterar ou Excluir Conexão {Alternativo}.

1. Sistema apresenta todas as conexões cadastradas.
2. Administrador clica no ícone alterar ou excluir.
3. Sistema informa mensagem de confirmação.

Testar Conexão {Alternativo}.

1. Sistema apresenta todas as conexões cadastradas no sistema.
2. Administrador clica no ícone testar conexão.
3. Sistema informa mensagem de conexão estabelecida ou falha na conexão.

Conexão não reconhecida {Exceção}.

1. Em caso do sistema não conseguir estabelecer uma conexão com o banco de acordo com as informações cadastradas, será exibido uma mensagem de falha na conexão.

Pós-Condição: O sistema cria uma nova conexão a um banco de dados para utilização na extração de dados.

Quadro 10 – Detalhamento do caso de uso UC003 – Cadastrar Conexão

UC006 – Cadastrar Fonte de Dados

Objetivo: Criar uma ou múltiplas fonte de dados que irá através de uma conexão extrair, transformar e carregar os dados para as dimensões ou cubos de um data warehouse.

Pré-Condição: Possuir conexões cadastradas no sistema.

Possuir dimensões ou cubos cadastrados no sistema.

Estar logado no sistema com o perfil de administrador.

Cenários*Incluir uma nova fonte de dados {Principal}.*

1. Administrador informa nome e descrição.
2. Administrador informa a conexão.
3. Administrador informa a instrução SQL.
4. Administrador confirma cadastro.

Alterar ou excluir fonte de dados {Alternativo}.

1. Sistema apresenta todas as fontes cadastradas.
2. Administrador clica no ícone alterar ou excluir.
3. Sistema informa mensagem de confirmação.

Conexão falha {Exceção}.

Em caso da conexão não ser estabelecida durante o processo de roteirização ou importação o sistema apresentará um mensagem de erro.

Instrução SQL Inválida {Exceção}.

Em caso da instrução SQL cadastrada na fonte possuir algum erro de sintaxe que invalide a execução da mesma durante o processo de roteirização ou importação o sistema apresentará um mensagem de erro.

Criar roteiro {Principal}.

1. Administrador clica no ícone roteiro referente à fonte.

2. O Sistema apresenta um quadro de caixas de seleção que trazem do lado esquerdo todos os atributos/indicadores ou chaves primárias das dimensões que precisam ser mapeados de acordo com a instrução SQL fornecidos pela fonte. Do lado direito o sistema apresenta caixas de seleção com todos os campos retornados da instrução SQL.

3. O Administrador informa para cada item o mapeamento correto.

4. O Administrador confirma o roteiro.

5. O Administrador poderá ainda retornar e editar o roteiro.

Criar agendamento {Alternativo}.

1. O Administrador clica no ícone agendamento referente à fonte.

2. O Administrador informa o tipo de agendamento.

3. O Administrador ativa ou não o agendamento.

4. O Administrador informa a data/hora de início e fim do agendamento.

5. Em caso de agendamentos semanais ou mensais, o administrador informará os dias da semana ou do mês.

6. O Administrador confirma o cadastro.

7. O Sistema retorna uma mensagem de confirmação.

Importar dados manualmente {Alternativo}.

1. O Administrador clica no ícone importar referente à fonte.

2. O Administrador confirma o início da importação.

3. O Sistema retorna uma mensagem de finalização da importação.

Pós-Condição: O sistema disponibilizará os dados carregados dentro do banco de data warehouse transformados e sumarizados conforme a definição das dimensões ou cubos de decisão.

Quadro 11 – Detalhamento do caso de uso UC005 – Cadastrar Fonte de Dados

Para as operações de consultas o administrador além de poder visualizar e configurar as consultas, conforme detalhamento do Quadro 5, o mesmo poderá cadastrar as consultas. O detalhamento do caso de uso UC007 é apresentado pelo Quadro 12.

UC007 – Cadastrar Consulta

Objetivo: Definir consultas gerenciais baseadas na modelagem dimensional do cubo de decisão.

Pré-Condição: Possuir cubos cadastrados no sistema.

Estar logado no sistema com o perfil de administrador.

Cenários

Incluir novas consultas {Principal}.

1. O Administrador informa nome da consulta.

2. O Administrador informa descrição da consulta.

3. O Administrador informa o cubo de decisão.

4. O Administrador confirma o cadastro.

5. O Sistema informa mensagem de confirmação.

Alterar ou excluir consultas {Alternativo}.

1. Sistema apresenta todas as consultas cadastradas.

2. Administrador clica no ícone alterar ou excluir.

3. Sistema informa mensagem de confirmação.

Definir eixos da consulta {Principal}.

1. O Administrador clica no ícone novo para definir os parâmetros de cada eixo.
2. O Administrador informa o eixo.
3. O Administrador informa o atributo em caso do eixo definido for X ou Y.
4. O Administrador informa o indicador e a função de agregação em caso do eixo for indicadores.
5. O Administrador confirma cadastro.
6. O Sistema atualiza o cadastro de eixos da consulta.
7. Para o eixo Y o administrador poderá clicar na coluna drill down para definir em qual nível será feito o particionamento dos dados.
8. O Administrador poderá ainda incluir ou excluir novos eixos para a consulta.

Pós-Condição: Visualizar tabelas simples ou cruzadas de indicadores gerenciais do cubo de decisão.

Quadro 12 – Detalhamento do caso de uso UC007 – Cadastrar Consulta

Para as operações referentes ao metadados pode-se ver o detalhamento dos casos de uso UC013 e UC014 nos Quadro 13 e Quadro 14.

UC013 - Importar Metadados

Objetivo: Importar para o sistema um novo projeto de data warehouse gerado em XML.

Pré-Condição: Estar logado no sistema com o perfil de administrador.
Arquivo XML a ser importado ter sido gerado pela própria ferramenta.

Cenários

Fluxo Principal {Principal}.

1. O Administrador acessa o link através do menu Ferramentas -> Importar Meta Dados
2. O Administrador seleciona um caminho local para abrir o arquivo.
3. O Administrador informa dados do projeto de data warehouse.
3. O Sistema lê o arquivo carregando todas as definições do data warehouse para o projeto informado.

Erro na leitura {Exceção}.

O Sistema apresentará uma mensagem de incompatibilidade de leitura do arquivo, em caso de qualquer problema na importação do mesmo.

Pós-Condição: Projeto de data warehouse ficará disponível para uso dentro da ferramenta

Quadro 13 – Detalhamento do caso de uso UC013 – Importar Metadados

UC014 – Exportar Metadados

Objetivo: Exportar todas as definições referentes a um projeto de data warehouse em padrão XML

Pré-Condição: Estar logado no sistema com o perfil de administrador.

Cenários

Fluxo Principal {Principal}.

1. O Administrador acessa o link através do menu Ferramentas -> Exportar Meta Dados
2. O Administrador seleciona um caminho local para salvar o arquivo.

3. O Sistema prepara o arquivo carregando todas as definições do data warehouse corrente.

Quadro 14 – Detalhamento do caso de uso UC014 – Exportar Metadados

Para outros recursos adicionais da ferramenta pode-se ver o detalhamento dos casos de uso UC009, UC010, UC011 e UC012 nos Quadro 15, Quadro 16, Quadro 17 e Quadro 18.

UC009 - Visualizar Agendamentos

Objetivo: Apresentar ao administrador todos os agendamentos de fontes do dia corrente que ainda estão em aberto para processamento.

Pré-Condição: Estar logado no sistema com o perfil de administrador.
Possuir fonte de dados com agendamentos definidos.

Cenários

Fluxo Principal {Principal}.

1. O Administrador acessa através do menu Principal -> Agendamentos.
2. O Sistema apresenta todos os agendamentos em aberto do dia.

Quadro 15 – Detalhamento do caso de uso UC009 – Visualizar Agendamentos

UC010 - Visualizar Log de Mensagens

Objetivo: Visualizar as principais ocorrências dentro do sistema como informações de importação, erro e tratamento de exceções.

Pré-Condição: Estar logado no sistema com o perfil de administrador.

Cenários

Fluxo Principal {Principal}.

1. O Administrador acessa através do menu Principal -> Logs.
2. O Sistema apresenta todas as ocorrências de logs do sistema.

Quadro 16 – Detalhamento do caso de uso UC010 – Visualizar Log de Mensagens

UC011 - Limpar Data Warehouse

Objetivo: Processar limpeza de dados e do conteúdo dos projetos de data warehouse do sistema.

Pré-Condição: Estar logado no sistema com o perfil de administrador.

Cenários

Fluxo Principal {Principal}.

1. O Administrador seleciona o tipo de limpeza: definição ou conteúdo.
2. O Administrador seleciona o tipo de entidade: data warehouse, cubo ou dimensão.
3. O Administrador seleciona as entidades que serão limpas.
4. O Administrador confirma a execução.
5. O Sistema mostra mensagem de confirmação.

Quadro 17 – Detalhamento do caso de uso UC011 – Limpar Data Warehouse

UC012 - Cadastrar Usuário

Objetivo: Criar novos usuários para acesso ao sistema.

Pré-Condição: Estar logado no sistema com o perfil de administrador.

Cenários

Incluir novos usuários {Principal}.

1. O Administrador informa o login e a senha.
2. O Administrador informa o nome e o email.
3. O Administrador informa o tipo de usuário.
4. O Sistema mostra uma mensagem de confirmação.

Alterar ou excluir usuário. {Alternativo}.

1. Sistema apresenta todos os usuários cadastrados.
2. Administrador clica no ícone alterar ou excluir.
3. Sistema informa mensagem de confirmação.

Privilégios de usuários {Alternativo}.

1. O Administrador através do menu Usuários -> Privilégios visualizará todos os usuários onde o tipo é Usuário.
2. O Administrador clica no ícone privilégios referente ao usuário.
3. O Administrador seleciona as consultas para o lado direito do quadro na qual o mesmo terá acesso.
4. O Sistema mostra uma mensagem de confirmação.

Usuários do sistema {Exceção}.

O Sistema não permite alteração e exclusão de usuários do tipo sistema

Quadro 18 – Detalhamento do caso de uso UC012 – Cadastrar Usuários

3.2.2 Diagrama de pacotes

O diagrama de pacotes demonstra uma visão mais ampla do diagrama de classes de um sistema. Um pacote é um agrupamento de classes. Assim, o diagrama consiste em descrever as dependências de cada pacote. As dependências são expressas por uma seta com uma linha tracejada. Na Figura 9 é ilustrado o diagrama de pacotes com os principais pacotes da ferramenta, onde é demonstrada a arquitetura adotada utilizando o padrão MVC. Esta arquitetura procura dividir a aplicação em camadas que são: apresentação ou visualização que se preocupa apenas em exibir as informações, controle que determina o fluxo da informação mapeando as ações geradas pela interface de visualização e lógica que é responsável por tudo que a aplicação vai fazer, modelando os dados e o comportamento por trás do processo de negócios.

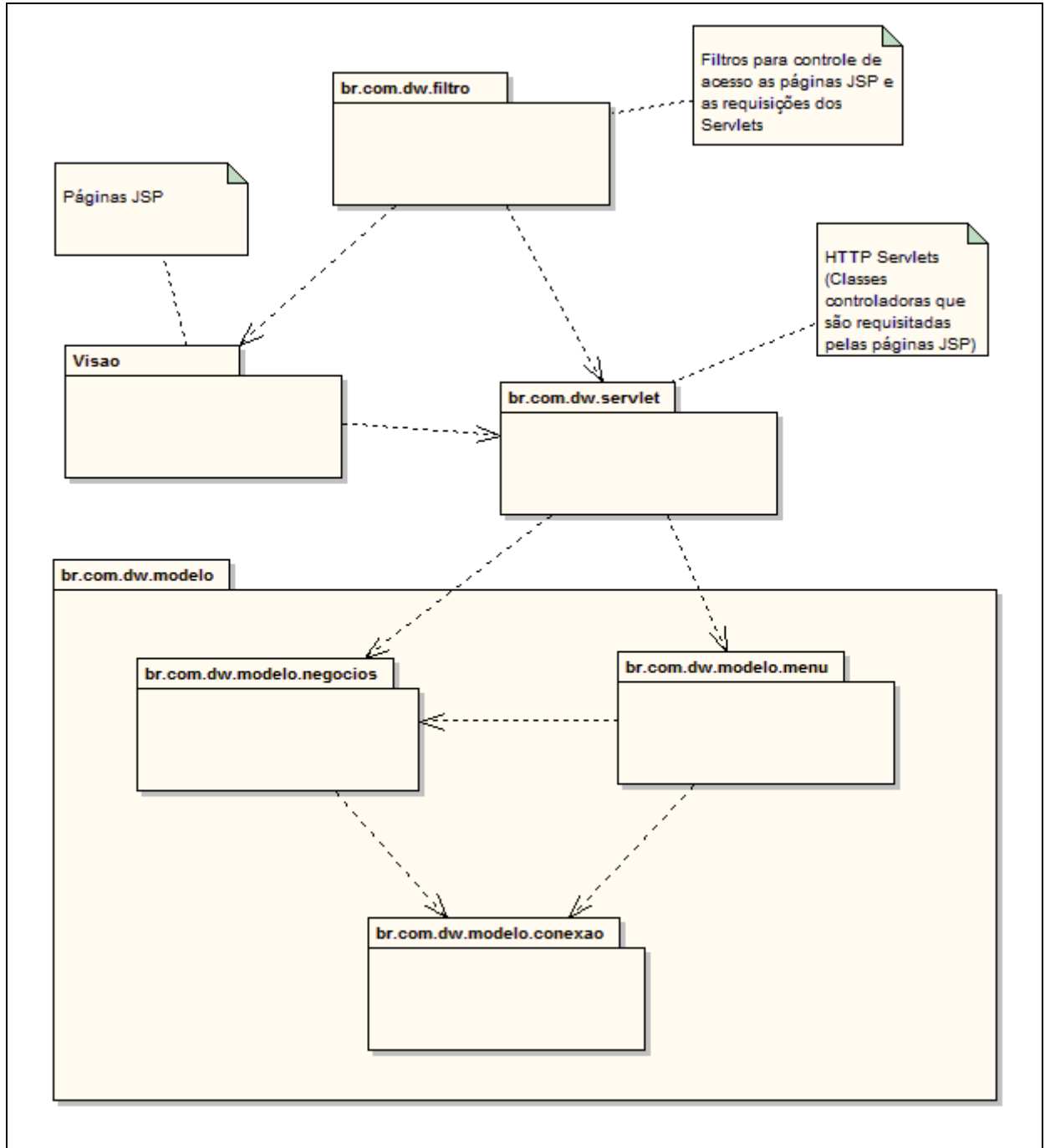


Figura 9 – Diagrama de pacotes

O Quadro 19 descreve qual a função dos principais pacotes na arquitetura do sistema.

PACOTE	DESCRIÇÃO
br.com.dw.filtro	Este pacote é composto por todas as classes que filtram dinamicamente requisições às páginas JSP e servlet.
Visão	Este pacote é composto por todas as páginas JSP do sistema. Ele é responsável em prover a camada de visão do sistema através de tags HTML. Ele pode ser visto como o V do padrão MVC.
br.com.dw.servlet	Este pacote engloba todas as classes de controles de interface, ou seja, as classes que estão ligadas a uma página JSP da camada de visão. Este pacote é responsável em disponibilizar a camada de controle do sistema. Ele pode ser visto como o C do padrão MVC.
br.com.dw.modelo.negocios	Este pacote é composto por todas as classes que formam a camada de modelo do sistema. Ele pode ser visto como o M do padrão MVC.
br.com.dw.modelo.menu	Este pacote é composto pelas classes que controlam a interação entre as interfaces através de menus dinâmicos. É nesta camada que se utiliza de implementações AJAX.
br.com.dw.modelo.conexao	Este pacote é composto pelas classes que fazem o acesso ao banco de dados através das conexões implementadas.

Quadro 19 – Descrição dos pacotes do sistema

3.2.3 Diagrama de classes

O diagrama de classes é um dos diagramas mais importantes de um projeto orientado a objetos. Cada classe representa objetos e o diagrama de classes demonstra o relacionamento entre esses objetos. O diagrama de classes modela os conceitos do sistema e define a arquitetura da aplicação a ser desenvolvida.

No pacote negócios encontram-se as classes que formam o modelo principal do

sistema. Por este motivo a apresentação de suas arquiteturas foi dividida em: ETC que modelam todo o processo de extração, transformação e carga dos dados dentro do DW representada pela Figura 10; PROJETO que modelam todo o processo de construção de um projeto de DW representada pela Figura 11 e CONSULTA que modelam todo o processo de visualização e configuração das consultas do DW representada pela Figura 12. Em cada processo existem classes comuns que são utilizadas e que possuem funcionalidades específicas.

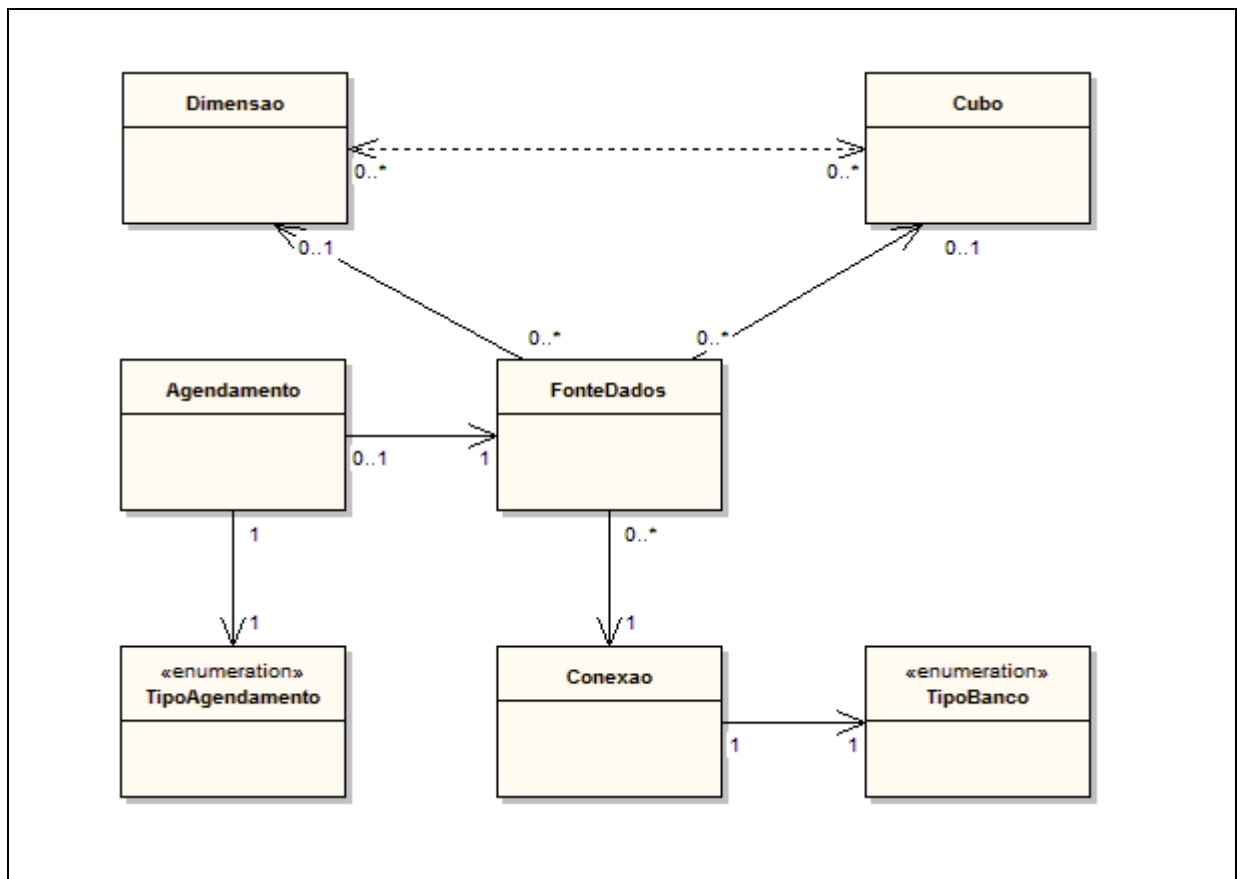


Figura 10 - Diagrama de classes do pacote negócio – ETC

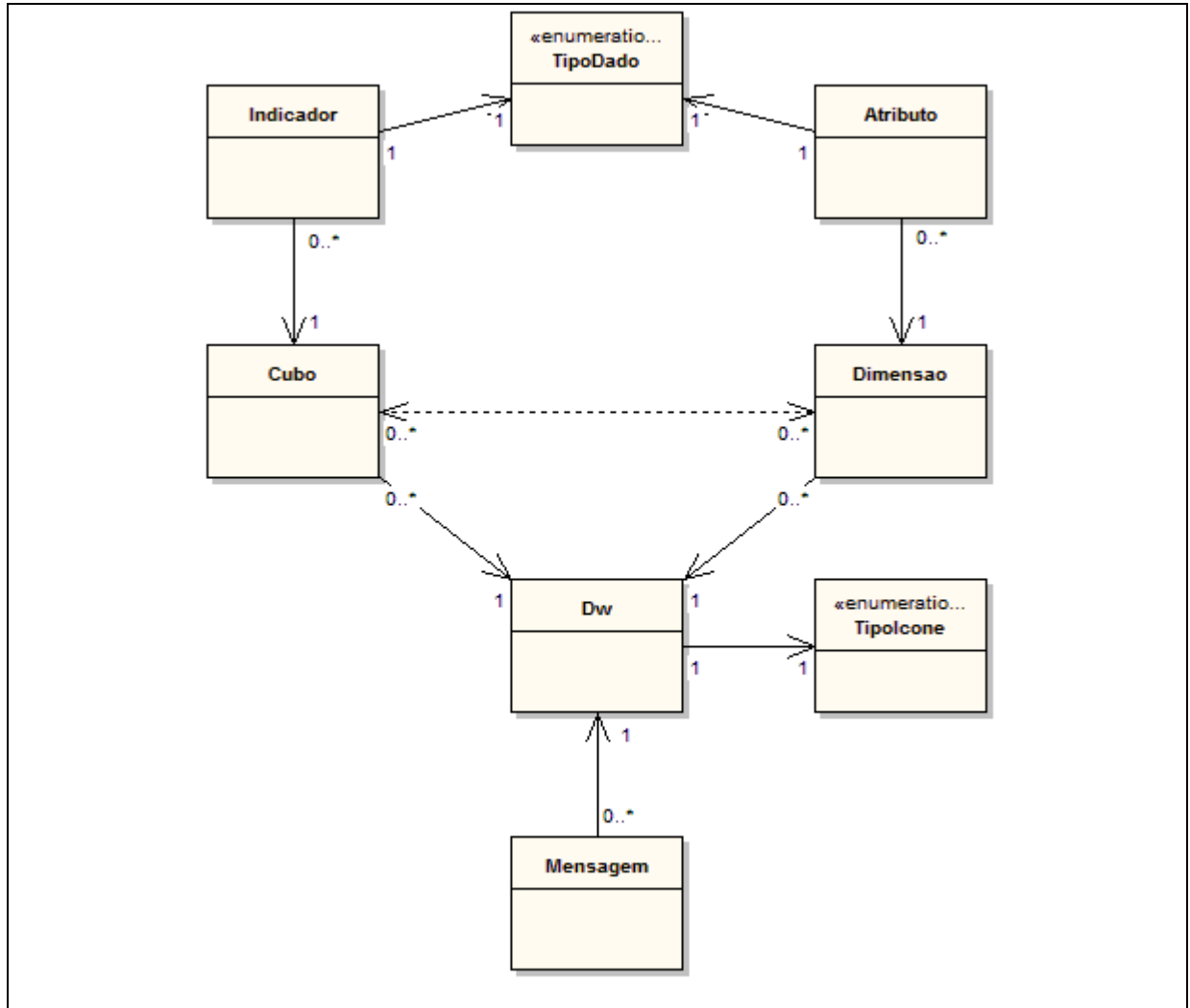


Figura 11 – Diagrama de classes do pacote negócio – PROJETO

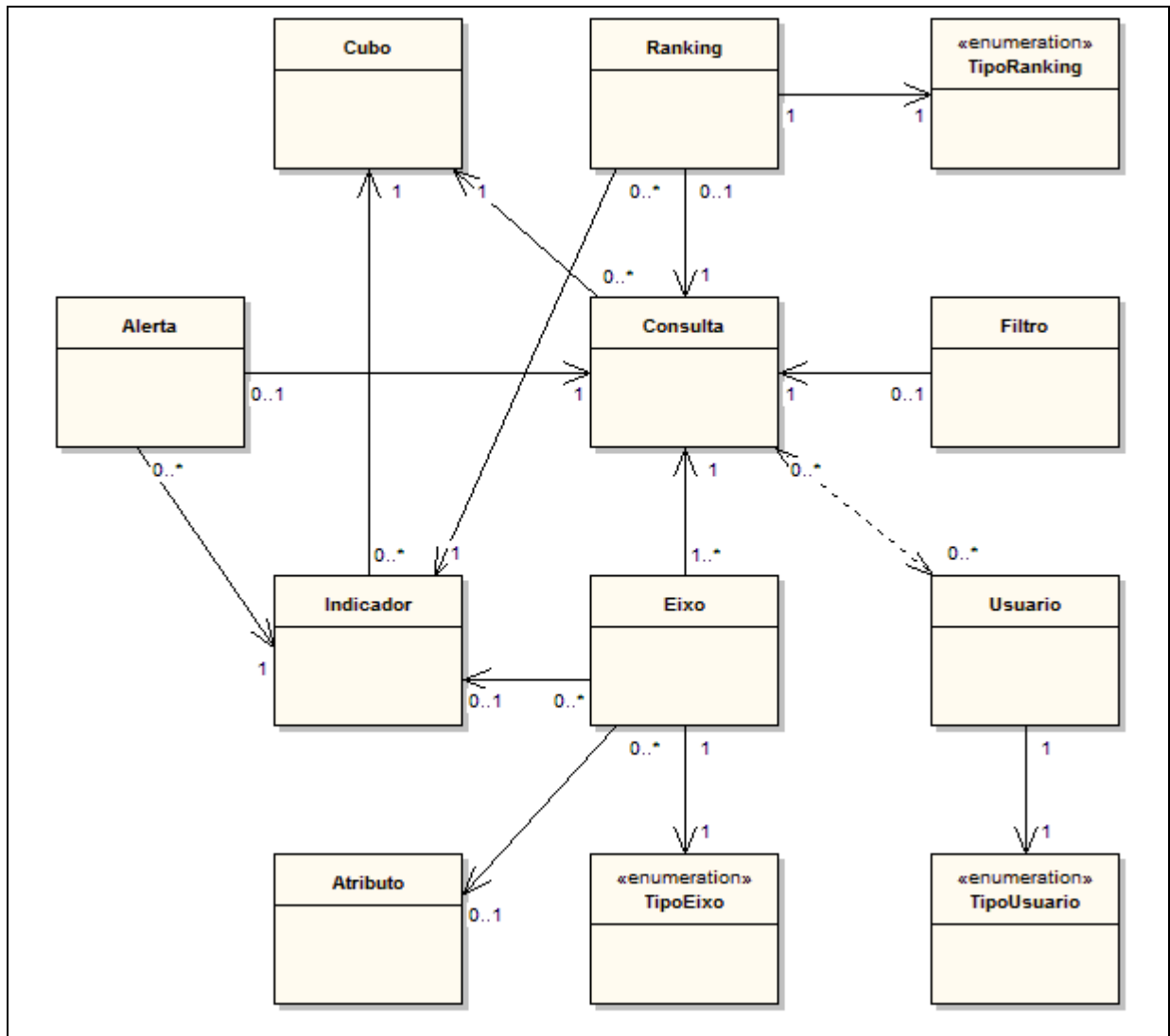


Figura 12 – Diagrama de classes do pacote negócio - CONSULTA

As classes do pacote negócio definem todas as regras e operações da ferramenta, validação e edição no banco para seus respectivos cadastros. As classes de enumeração controlam os possíveis valores dos atributos na qual possuem relacionamento. A descrição dos principais atributos e métodos de cada classe deste pacote é apresentada no Apêndice A.

O sistema ainda apresenta outros pacotes menores que apresentam diagramas mais simples e que possuem finalidades específicas dentro do sistema. A seguir são apresentados tais diagramas bem como a descrição de suas classes.

O pacote filtro é composto por uma única classe que possui a finalidade de garantir a autenticidade do acesso ao sistema e o tempo em que o usuário permanece em atividade dentro do mesmo. A Figura 13 ilustra o diagrama de classes que compõe o pacote filtro.

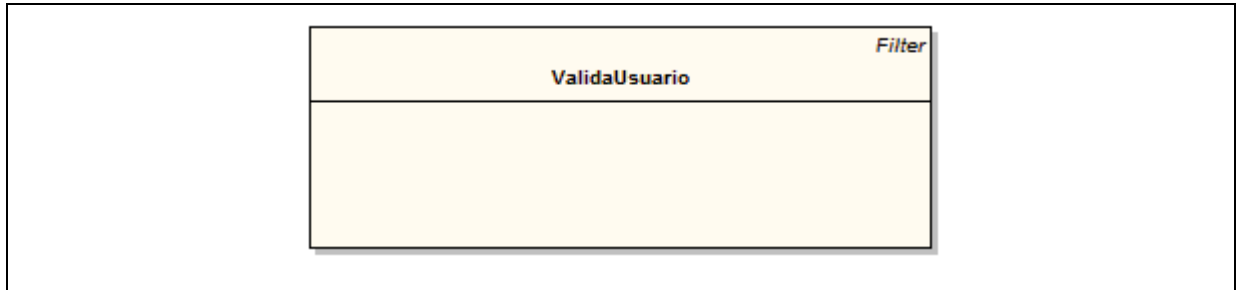


Figura 13 – Diagrama de classes do pacote filtro

O pacote conexão é composto por uma única classe que procura implementar um padrão de acesso a diferentes bancos através de um conjunto de classes JDBC. Dentre as principais funcionalidades estão conectar, desconectar, consultar e executar. A Figura 14 ilustra o diagrama de classes que compõe o pacote conexão.

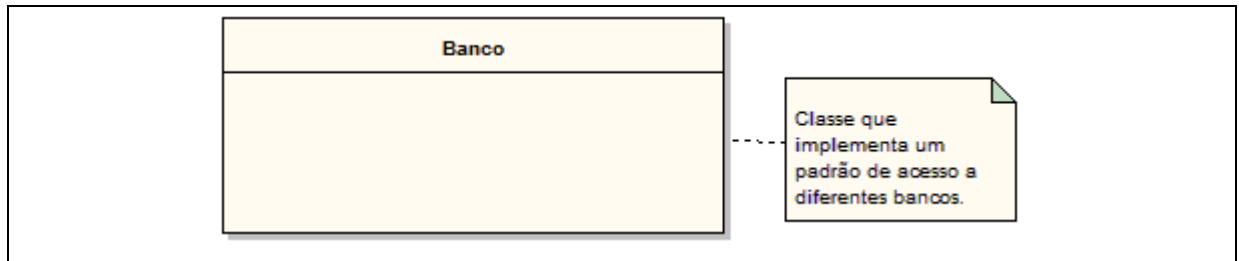


Figura 14 – Diagrama de classes do pacote conexão

As classes que compõem o pacote menu possuem a responsabilidade de disponibilizar interações dinâmicas entre as interfaces das principais funcionalidades do sistema. A Figura 15 ilustra o diagrama de classes das classes que compõe o pacote menu.

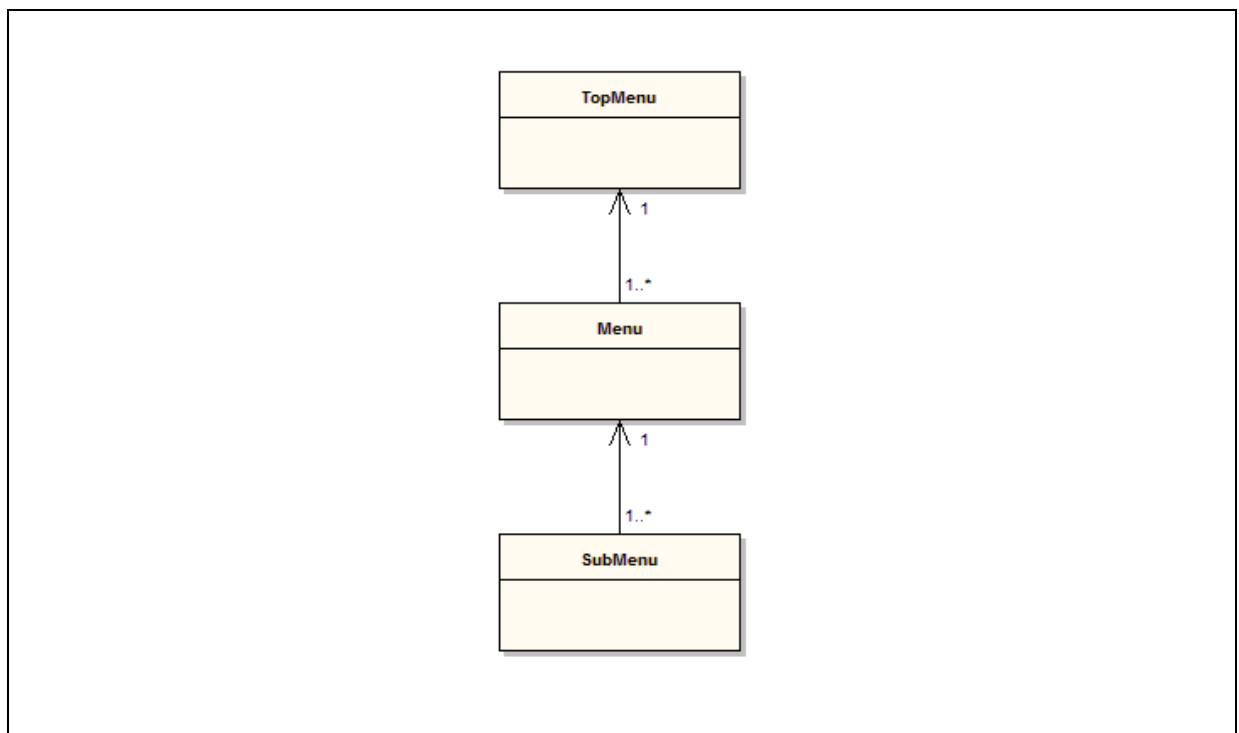


Figura 15 – Diagrama de classes do pacote menu

As classes que compõem o pacote servlet realizam todo o controle de uma requisição para ou das interfaces JSP do sistema através dos métodos *post* ou *get*. Todas as classes deste pacote são uma extensão as classes `HttpServlet` projetadas para a conexão a servidores *HyperText Transfer Protocol* (HTTP). A Figura 16 ilustra o diagrama de classes que compõem o pacote servlet.

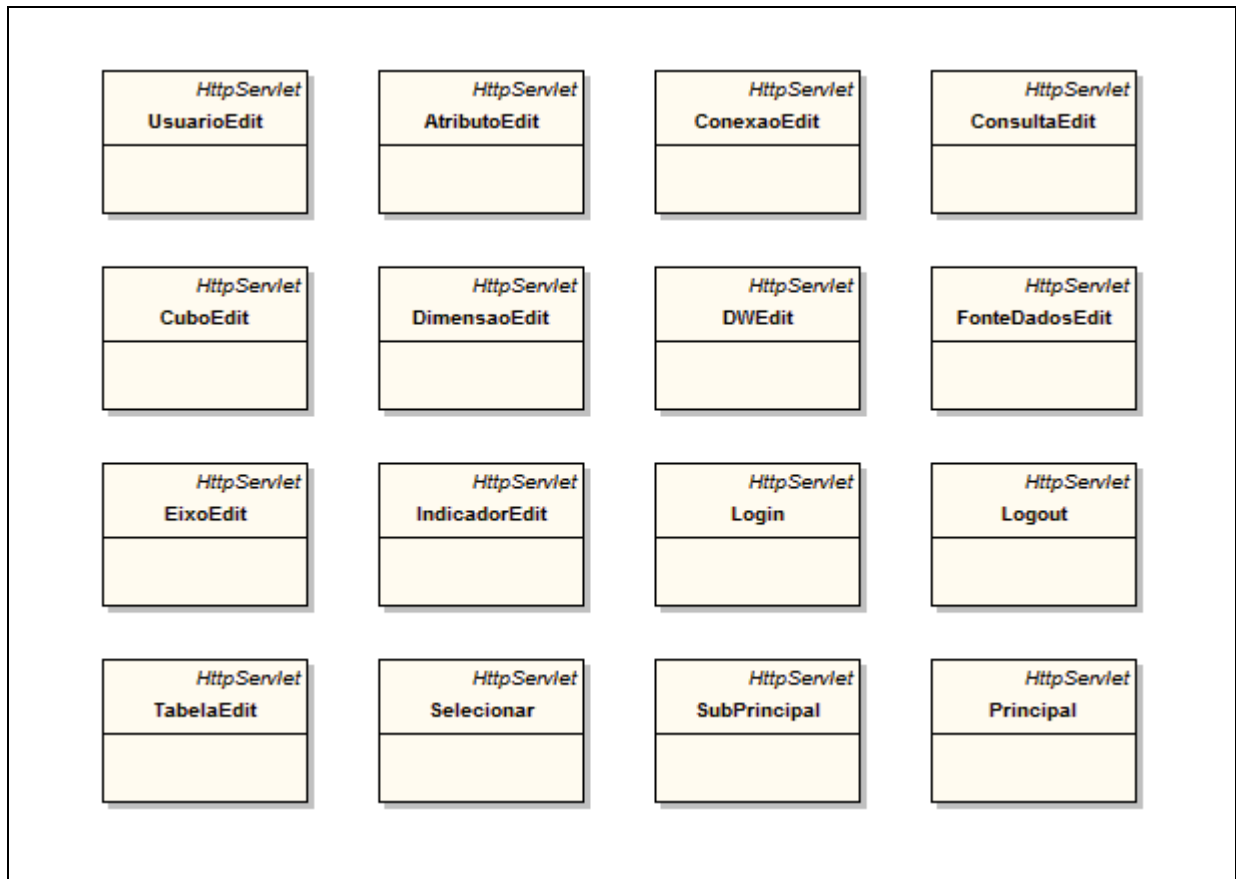


Figura 16 – Diagrama de classes do pacote servlet

3.2.4 Diagrama de seqüência

O diagrama de seqüência expressa a ordem em que são executadas as consultas e operações identificadas em um caso de uso. O diagrama é constituído por atores e por instâncias de objetos do sistema. Em seu todo, pode ser visualizada a troca de mensagens entre o autor e os objetos do sistema.

Serão ilustrados três diagramas de seqüência procurando neles abordar três grandes processos da ferramenta. No diagrama ilustrado pela Figura 17 é apresentada a fase de ETC, a qual envolve os objetos do diagrama de classes da Figura 10 e os casos de uso UC003 e UC006.

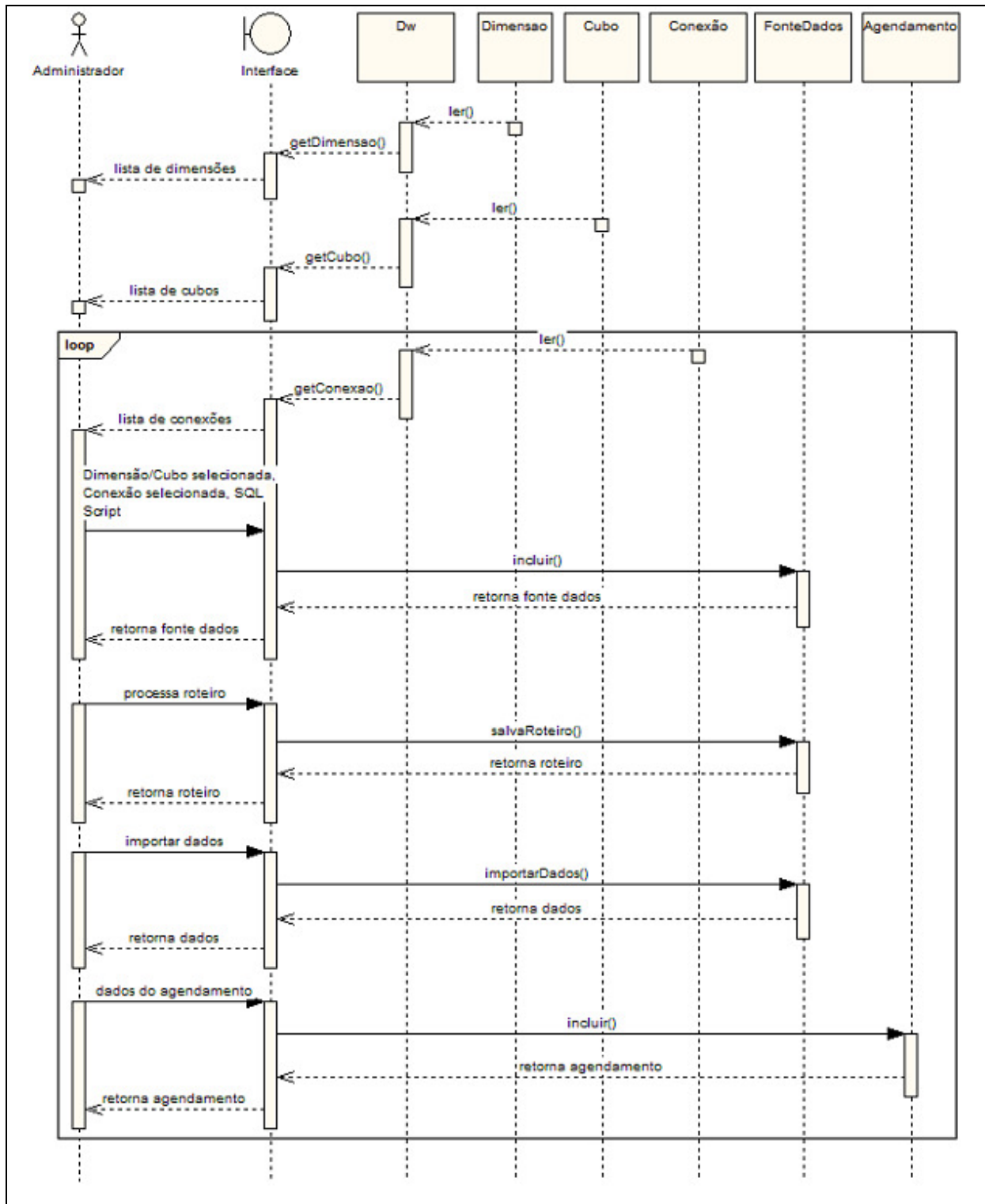


Figura 17 – Diagrama de seqüência relacionado ao processo de ETC

Em seguida na Figura 18 é ilustrado o diagrama de seqüência para os objetos do diagrama de classes da Figura 11 e os casos de uso UC002, UC004 e UC005.

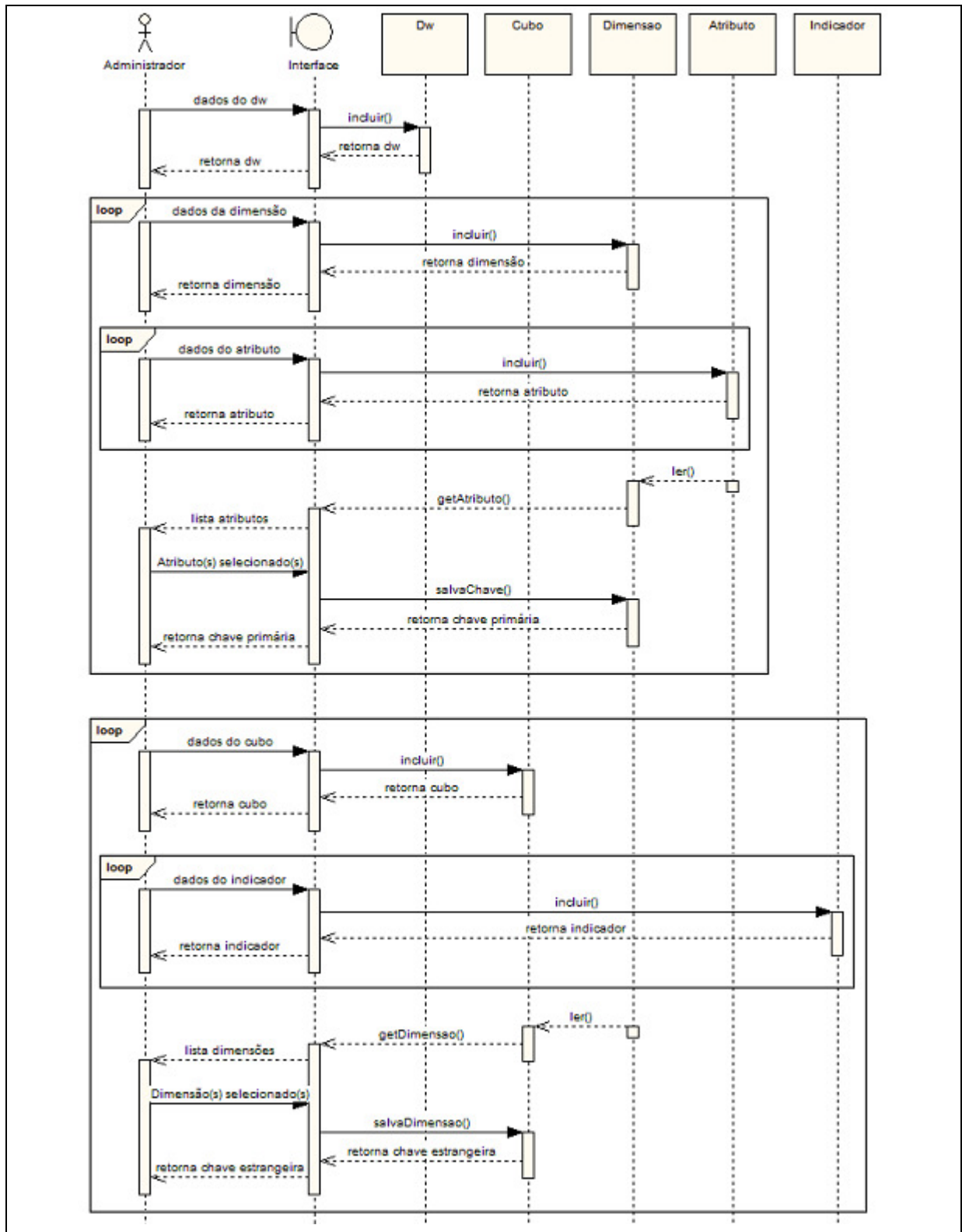


Figura 18 – Diagrama de seqüência relacionado ao processo de PROJETO

A Figura 19 ilustra o diagrama de seqüência para os objetos do diagrama de classes da Figura 12 e os casos de uso UC007 e UC008.

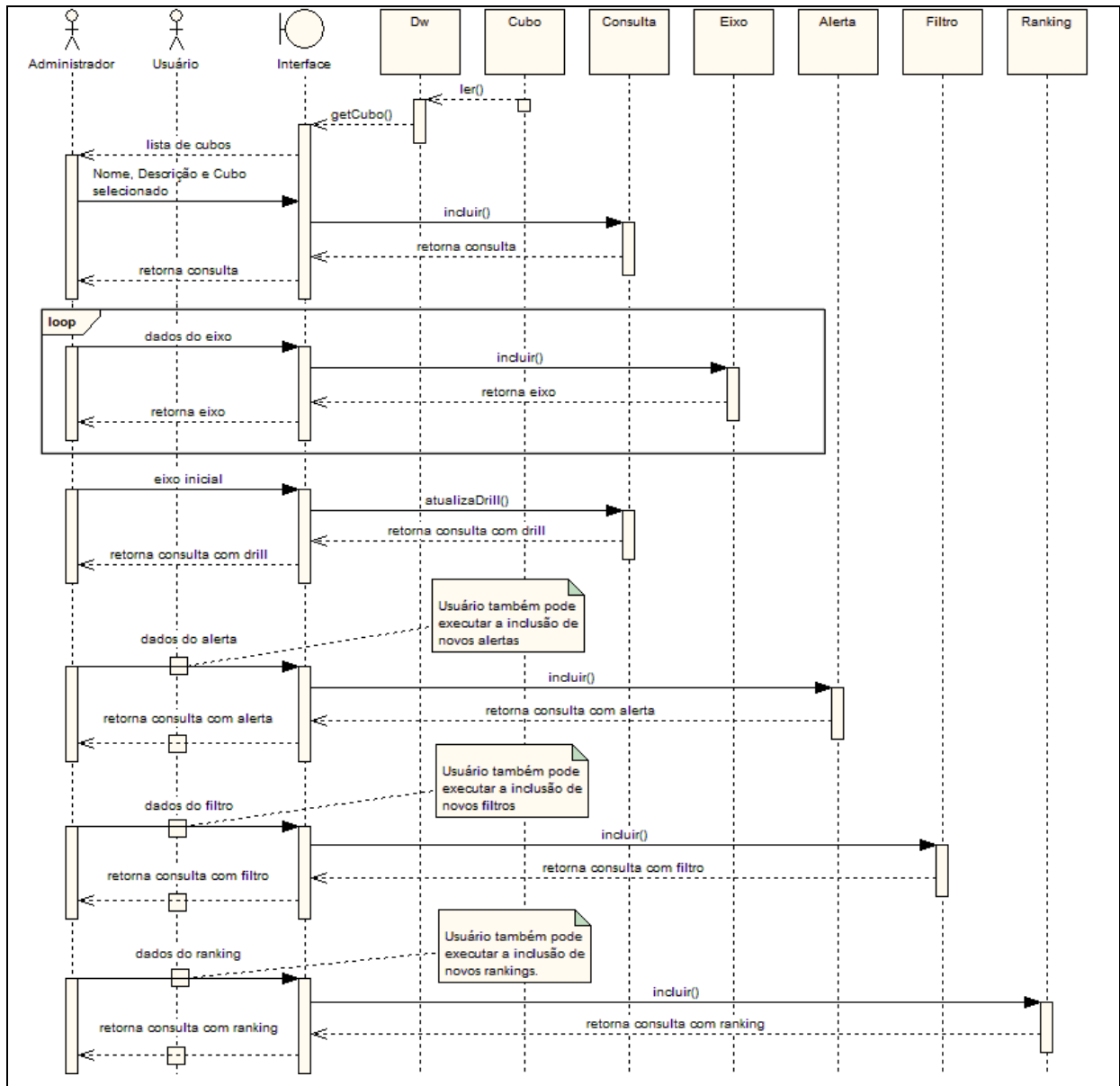


Figura 19 – Diagrama de seqüência relacionado ao processo de CONSULTA

3.3 IMPLEMENTAÇÃO

Nesta seção é descrita a implementação do trabalho assim como as técnicas, ferramentas e padrões utilizados.

3.3.1 Técnicas e ferramentas utilizadas

A ferramenta foi toda desenvolvida para a plataforma Java. O sistema foi compilado utilizando o J2SE 1.5 e roda em um servidor que implementa a especificação J2EE 1.4 ou superior. Para o desenvolvimento de aplicações web foi utilizado o *Integrated Development Environment* (IDE) Eclipse 3.2.1 acrescido do *plugin* MyEclipse 5.1.1 que já implementa projetos utilizando *servlets* com interfaces JSP. O servidor de aplicações utilizado no desenvolvimento foi o Apache Tomcat 5.5.23 que já vem junto com o IDE por padrão. Para a implantação do AJAX foram utilizadas implementações javascripts com *grids*. Para manipulação dos *grids* utilizou-se uma biblioteca de *scripts* da Zapatec, que já possui diversas modelagens para tabela de dados. O banco de dados escolhido para o desenvolvimento do metadados foi o MySQL 5.0 com interfaces de conexão JDBC. As tabelas do modelo objeto relacional estão descritas pelo Apêndice B e o *script* que cria o dicionário de dados da ferramenta é apresentado no Apêndice C.

A seguir são apresentadas alguns trechos de códigos que demonstram como foram implementadas algumas das técnicas citadas. O Quadro 20 ilustra a função javascript que carrega as formatações e controles de um *grid* dentro da interface.

```
function iniGridX() {
var objGrid = new Zapatec.Grid({
    source: 'gridSourceX',
    sourceType: 'html',
    theme: 'lightblue',
    container: 'divMiniX',
    callbackCellOnClick: onCellClick,
    rowsPerPage: 4
});
}
```

Quadro 20 – Implementação dos *grids* Zapatec

O Quadro 21 apresenta as principais funções que implementam o recurso AJAX que executa sempre que uma operação de processamento pelo lado do servidor é solicitada pelo lado do cliente.

```
var objRequest;
var action = "";
var method = "";
var conteudoRequisicao = "";

// Função para instanciar o objeto HttpRequest.
function ajaxInit() {
    try {
        objRequest = new XMLHttpRequest();
    } catch(e) {
        try {
```

```

    objRequest = new XMLHttpRequest("Msxml2.XMLHTTP");
} catch(ex) {
    try {
        objRequest = new XMLHttpRequest();
    } catch(exc) {
        alert("Esse browser não tem recursos para uso do Ajax.");
        objRequest = false;
    }
}
}
}

// Pega os campos e seus valores do formulário para concatenar na
variável(conteudoRequisicao) que será enviada para o servidor.
function getDataForm(myForm) {
    if(myForm != "" && myForm.action != undefined){
        action = myForm.action;
        method = myForm.method;
        conteudoRequisicao = "";
        for(i = 0; i < myForm.elements.length; i++){
            elementos = myForm.elements[i];
            if(elementos.name != ""){
                conteudoRequisicao += elementos.name + "=" +
escape(elementos.value) + "&";
            }
        }
        conteudoRequisicao = conteudoRequisicao.substring(0,
conteudoRequisicao.length -1);
    } else {
        alert("Form Invalido.");
    }
}

// Processa uma requisição pro Servidor através de um formulário.
function ajaxProcess(myForm, fuctionProcess) {
    if(objRequest){
        getDataForm(myForm);
        objRequest.open(method, action, true);
        objRequest.onreadystatechange = fuctionProcess;
        if(method == "post"){
            objRequest.setRequestHeader("Content-Type", "application/x-
www-form-urlencoded");
        }
        conteudoRequisicao = conteudoRequisicao != "" ? conteudoRequisicao :
null;
        objRequest.send(conteudoRequisicao);
    }
}

// Processa uma requisição pro Servidor via GET.
function ajaxProcessUrl(url, fuctionProcess) {
    if(objRequest && url != ""){
        method = "GET";
        action = url;
        objRequest.open(method, action, true);
        objRequest.onreadystatechange = fuctionProcess;
        objRequest.send(null);
    }
}

// Verifica se o objeto XMLHttpRequest contém um retorno válido.

```

```

function isAjax() {
    if(objRequest.readyState == 4){
        if(objRequest.status == 200){
            return true;
        } else {
            alert("Problem: "+objRequest.statusText);
        }
    }
    return false;
}

```

Quadro 21 – Implementação do AJAX

Para implementar a operação de exportação e importação do metadados para arquivos XML é utilizado a biblioteca XStream 1.2.2. e o Quadro 22 ilustra a implementação.

```

// Funcao que exporta o metadados
public void exportar(String arquivo) {
    XStream sxml = new XStream(new DomDriver());
    Metadados md = new Metadados(this);
    sxml.alias("metadados", Metadados.class);
    String oxml = sxml.toXML(md);
    try {
        PrintWriter saida = new PrintWriter(new
FileOutputStream(arquivo));
        saida.println(oxml);
        saida.close();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
}
// Funcao que importa o metadados
public void importar(String arquivo) {
    String linha = null;
    String oxml = "";
    try{
        FileReader reader = new FileReader(arquivo);
        BufferedReader leitor = new BufferedReader(reader);
        while((linha = leitor.readLine()) != null){
            oxml += linha;
        }
        leitor.close();
        reader.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
    XStream sxml = new XStream(new DomDriver());
    sxml.alias("metadados", Metadados.class);
    Metadados md = (Metadados)sxml.fromXML(oxml);
    //Cria dw
    Dw d = md.getDw();
    d.setDataCriado();
    d.incluir();
    //Cria conexoes
    ArrayList<Conexao> c = md.getCnx();
    for (int i=0;i<c.size();i++) {
        c.get(i).setDw(d);
        c.get(i).incluir();
    }
    //Cria dimensoes
    ArrayList<Dimensao> di = md.getDimensao();

```

```

    for (int i=0;i<di.size();i++) {
        di.get(i).setDw(d);
        di.get(i).setDataCriado();
        di.get(i).incluir();
    }
    //Cria cubos
    ArrayList<Cubo> cb = md.getCubo();
    for (int i=0;i<cb.size();i++) {
        cb.get(i).setDw(d);
        cb.get(i).setDataCriado();
        cb.get(i).incluir();
    }
}

```

Quadro 22 – Implementação da exportação e importação XML

No processo de ETC o Quadro 23 ilustra a implementação do método de importação dos dados da classe FonteDados.

```

// Importar fonte de dados
public boolean importarDados() {
    String valsql = "";
    String sql = "";
    String intosql = "";
    String updsql = "";
    String wheresql = "";
    boolean result = true;
    this.cnx.ler();
    //...
    ResultSet rs = null;
    ResultSet rsAdo = null;
    if (bc.conectar()) {
        rs = bc.consultar(this.sqldados);
        if (rs == null) result = false;
    } else result = false;
    Dw dw = null;
    String mensagem = "";
    int totIncl = 0;
    int totAlt = 0;
    int totDesc = 0;
    //Dimensao
    if (this.dim.getId() != 0) {
        this.dim.ler();
        dw = this.dim.getDw();
        Banco bci = new Banco();
        bci.conectar();
        ArrayList<Atributo> atb = this.dim.getAtributo();
        long inicio = System.currentTimeMillis();
        while (rs.next()) {
            intosql = "";
            valsql = "";
            updsql = "";
            wheresql = "";
            //...
            intosql = intosql.substring(0, intosql.length()-2);
            valsql = valsql.substring(0, valsql.length()-2);
            intosql = "INSERT INTO "+this.dim.getTblDim()+"
("+intosql+") ";
            valsql = "VALUES ("+valsql+)";
            updsql = updsql.substring(0, updsql.length()-2);
            if (wheresql.length() > 0) {

```

```

                    wheresql = wheresql.substring(0, wheresql.length()-
5);
                    wheresql = " WHERE "+wheresql;
                }
                updsql = "UPDATE "+this.dim.getTblDim()+" SET
"+updsql+" "+wheresql;
                sql = intosql+valsql;
                int r = bci.executar(sql);
                if (r < 0) {
                    r = bci.executar(updsql);
                    totAlt++;
                    if (r < 0) totDesc++;
                } else totIncl++;
            }
            long fim = System.currentTimeMillis();
            long tempo = (fim - inicio)/1000;
            bci.desconectar();
        }
        //Cubo
        if (this.cub.getId() != 0) {
            this.cub.ler();
            dw = this.cub.getDw();
            Banco bci = new Banco();
            bci.conectar();
            bci.executar("TRUNCATE TABLE "+this.cub.getTblCub());
            ArrayList<Dimensao> dim = this.cub.getDimensao();
            ArrayList<ArrayList> datb = new ArrayList<ArrayList>();
            for (int i=0;i<dim.size();i++) {
                datb.add(dim.get(i).getChavePrimaria());
            }
            ArrayList<Indicador> ind = this.cub.getIndicadores();
            long inicio = System.currentTimeMillis();
            while (rs.next()) {
                intosql = "";
                valsql = "";
                updsql = "";
                wheresql = "";
                // Saber quais sao as dimensoes envolvidas e suas chaves
primarias
                for (int i=0;i<dim.size();i++) {
                    intosql += dim.get(i).getTblDim()+", ";
                    wheresql += dim.get(i).getTblDim()+" = ";
                    sql = "";
                    //...
                    sql = sql.substring(0, sql.length()-5);
                    sql = "SELECT id FROM "+dim.get(i).getTblDim()+"
WHERE "+sql;
                    rsAdo = bci.consultar(sql);
                    if (rsAdo.next()) {
                        valsql += "+rsAdo.getInt(1)+", ";
                        wheresql += "+rsAdo.getInt(1)+" AND ";
                    }
                }
                // Varre indicadores
                for (int i=0;i<ind.size();i++) {
                    //...
                }
                intosql = intosql.substring(0, intosql.length()-2);
                valsql = valsql.substring(0, valsql.length()-2);

```

```

        updsql = updsql.substring(0, updsql.length()-2);
        wheresql = wheresql.substring(0, wheresql.length()-5);
        intosql = "INSERT INTO "+this.cub.getTblCub()+"
("+intosql+" ) ";
        valsq = "VALUES ("+valsq+")";
        updsql = "UPDATE "+this.cub.getTblCub()+" SET "+updsql+"
WHERE "+wheresql;
        sql = intosql+valsq;
        int r = bci.executar(sql);
        if (r < 0) {
            r = bci.executar(updsql);
            totAlt++;
            if (r < 0) totDesc++;
        } else totIncl++;
    }
    long fim = System.currentTimeMillis();
    long tempo = (fim - inicio)/1000;
    bci.desconectar();
}
if (result) dw.gerarLog(mensagem);
return result;
}

```

Quadro 23 – Método de importação dos dados da classe FonteDados

3.3.2 Operacionalidade da implementação

Neste item será apresentada a operacionalidade do sistema assim como suas principais interfaces e operações. Para ilustração do funcionamento de todo o processo de construção de um DW tomou-se como estudo de caso o faturamento de uma empresa fictícia FILLPRINT com as seguintes definições de projeto:

- a) manter o histórico de vendas da empresa;
- b) consultar a soma das vendas em valor total da nota fiscal por data (ano e mês) e clientes (estado);
- c) consultar a média de vendas em valor total da nota fiscal por representante (nome) e clientes (nome), alertando o usuário onde a média das vendas foram abaixo de duzentos reais na cor vermelha e acima na cor verde;
- d) consultar os vinte produtos mais vendidos em quantidade no ano de 2007.

Para o estudo de caso da FILLPRINT será criado dentro da ferramenta um projeto de DW utilizando o usuário padrão DWADMIN. A Figura 20 ilustra a tela de login de usuário.



Figura 20 – Tela de login

A Figura 21 mostra a tela onde o usuário que possui perfil de administrador pode cadastrar um novo projeto de DW ou já selecionar projetos existentes.

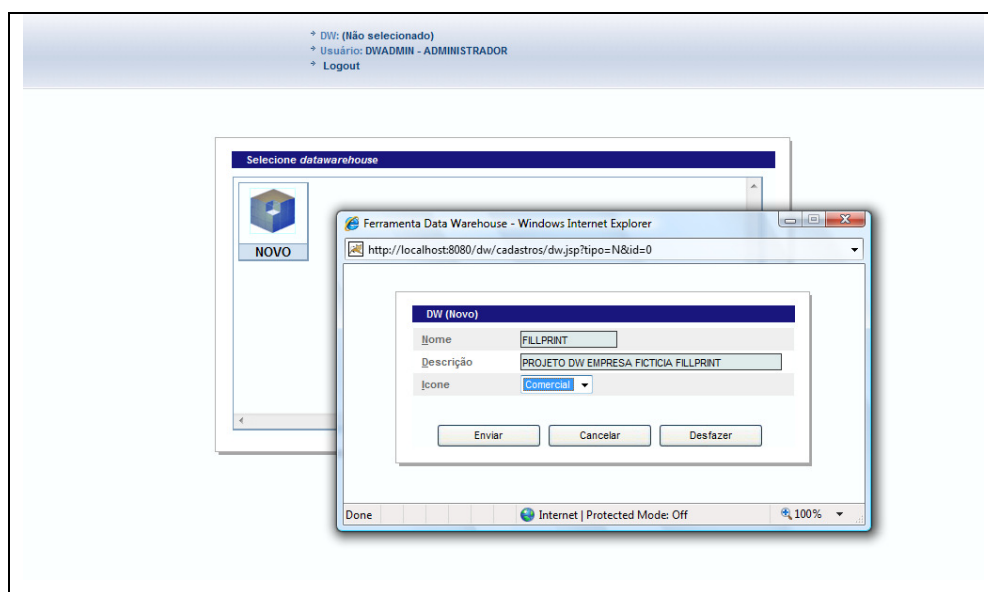


Figura 21 – Tela de cadastro de projetos de DW

Com o projeto criado, o administrador poderá selecionar o mesmo e conseqüentemente ter uma lista de opções referente a todas as funcionalidades que um administrador pode executar dentro da ferramenta. A Figura 22 ilustra a lista de opções que tem como página padrão o *log* de mensagens. Estas mensagens são todas as ocorrências que são criadas em decorrência das operações que o administrador executa para um projeto de DW. Desta forma o administrador pode consultar a data e hora de tarefas e movimentações realizadas.



Figura 22 – Tela do menu principal da ferramenta para usuários com perfil administrador

Para dar seqüência na montagem do projeto é necessário que seja cadastrada uma conexão com o banco de origem dos dados de vendas da empresa FILLPRINT. A Figura 23 ilustra as informações obrigatórias para efetuar o cadastro. O administrador poderá ainda validar a conexão através de uma das opções disponíveis para cada conexão cadastrada.

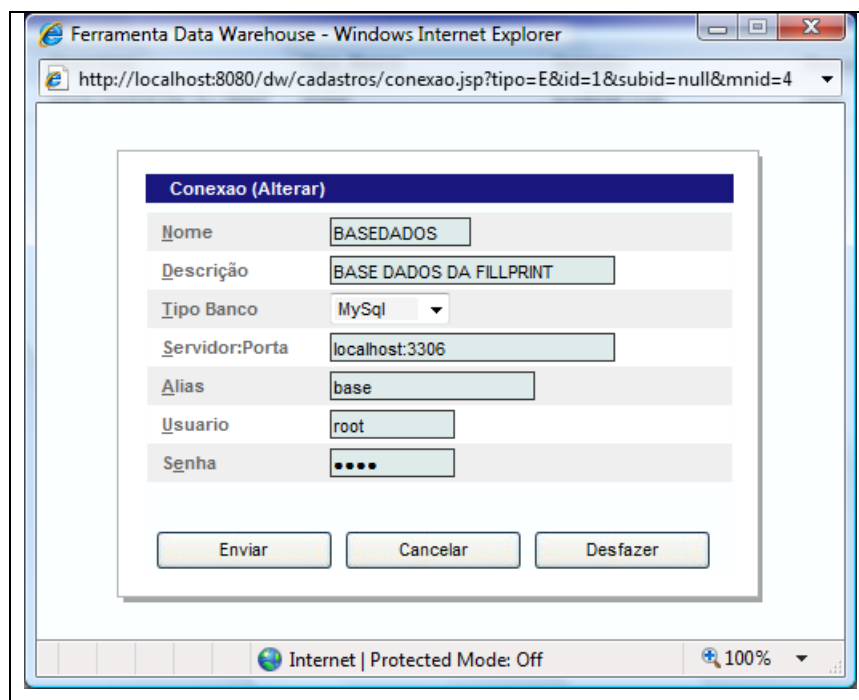


Figura 23 – Tela de cadastro de conexão

Tendo uma conexão estabelecida com uma base de dados, o projeto de DW pode começar a ser definido através do cadastro das dimensões e atributos, cubos e indicadores e suas fontes de dados respectivamente. Para o estudo de caso da FILLPRINT será necessário o cadastro das seguintes definições:

- a) dimensão cliente: atributos nome e estado;
- b) dimensão data: atributos ano e mês;
- c) dimensão representante: atributo nome;
- d) dimensão produto: atributos nome e tipo;
- e) cubo venda: indicadores valor total e quantidade.

A Figura 24 ilustra o cadastro de uma dimensão e a Figura 25 ilustra o cadastro de um atributo. Após todos os atributos estarem definidos para a dimensão, é realizada a definição da chave primária da dimensão ilustrada pela Figura 26.

The screenshot shows a web browser window titled 'Ferramenta Data Warehouse - Windows Internet Explorer'. The address bar contains the URL: <http://localhost:8080/dw/cadastros/dimensao.jsp?tipo=E&id=4&subid=null&mnid=5>. The main content area displays a form titled 'Dimensão (Alterar)'. The form contains two input fields: 'Nome' with the value 'PRODUTO' and 'Descrição' with the value 'DIMENSAO PRODUTO'. Below the input fields are three buttons: 'Enviar', 'Cancelar', and 'Desfazer'. The browser's status bar at the bottom shows 'Internet | Protected Mode: Off' and a zoom level of '100%'.

Figura 24 – Tela de cadastro de dimensão

The screenshot shows a web browser window titled 'Ferramenta Data Warehouse - Windows Internet Explorer'. The address bar contains the URL: <http://localhost:8080/dw/cadastros/atributo.jsp?tipo=E&id=1&subid=4&mnid=15>. The main content area displays a form titled 'Atributo (Alterar)'. The form contains several input fields: 'DIMENSAO' with the value 'PRODUTO', 'Nome' with 'NOME DO PRODUTO', 'Descrição' with 'NOME DO PRODUTO', 'Tipo' with a dropdown menu set to 'Caracter', 'Tamanho' with '30', and 'Decimais' with '0'. Below the input fields are three buttons: 'Enviar', 'Cancelar', and 'Desfazer'. The browser's status bar at the bottom shows 'Internet | Protected Mode: Off' and a zoom level of '100%'.

Figura 25 – Tela de cadastro de atributo

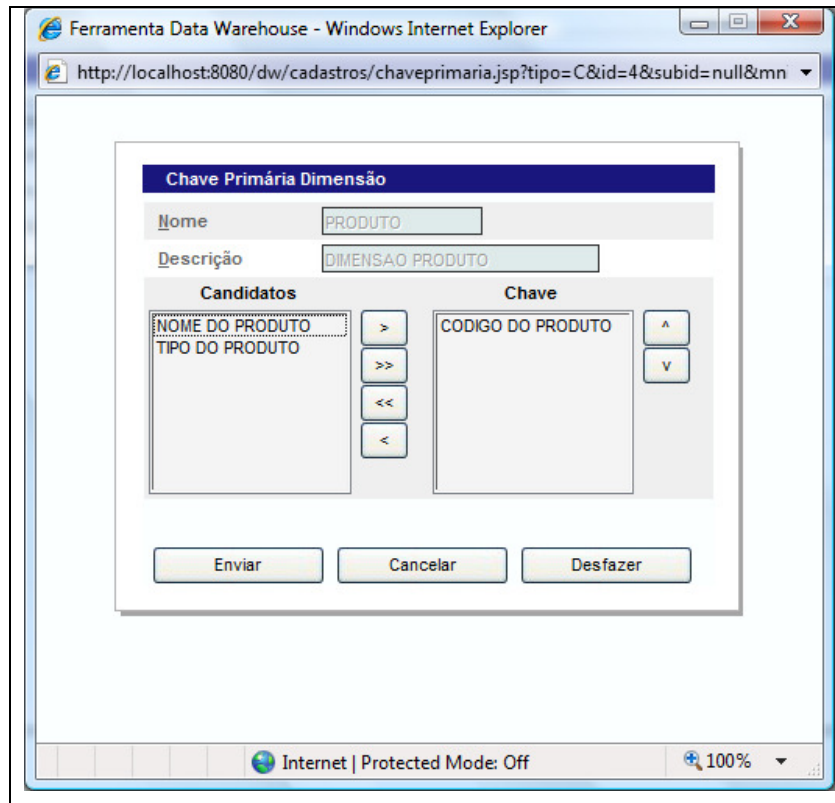


Figura 26 – Tela de definição de chave primária da dimensão

Após o administrador cadastrar todas as dimensões que serão utilizadas no cubo de decisão, é feito o cadastro do mesmo e definidos seus indicadores. A Figura 27 ilustra o cadastro de um cubo e a Figura 28 ilustra o cadastro de um indicador. Para o cubo é necessário a definição das dimensões que serão utilizadas, ilustrada pela Figura 29.

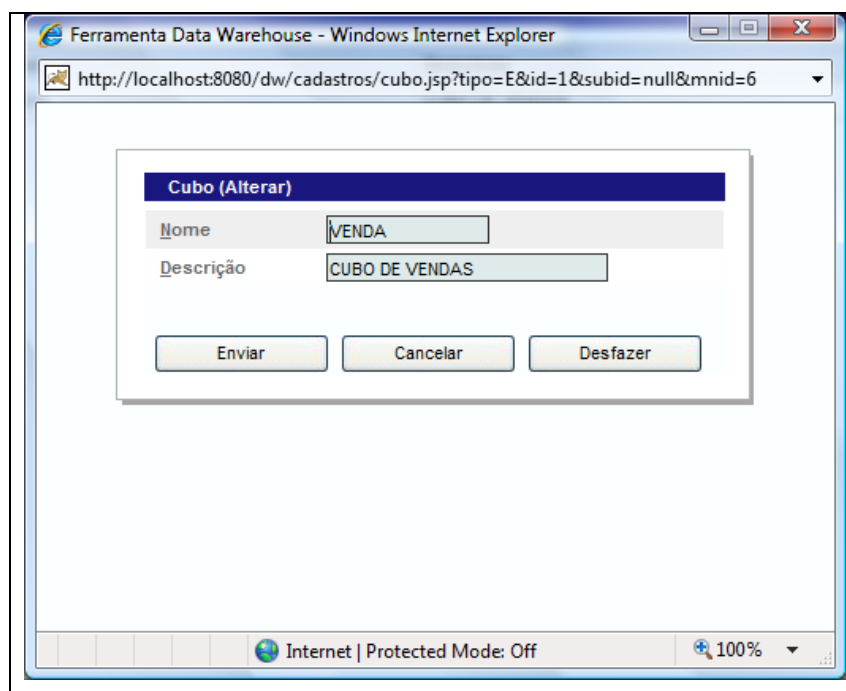


Figura 27 – Tela de cadastro do cubo

Ferramenta Data Warehouse - Windows Internet Explorer

http://localhost:8080/dw/cadastros/indicador.jsp?tipo=N&id=0&subid=1&mnid=16

Indicador (Novo)

CUBO: VENDA

Nome: QUANTIDADE

Descrição: QUANTIDADE DO ITEM DA NOTA

Tipo: Numerico

Tamanho: 8

Decimais: 2

Enviar Cancelar Desfazer

Internet | Protected Mode: Off 100%

Figura 28 – Tela de cadastro de indicador

Ferramenta Data Warehouse - Windows Internet Explorer

http://localhost:8080/dw/cadastros/dimensaoocubo.jsp?tipo=C&id=1&subid=null&mr

Cubo X Dimensão

Nome: VENDA

Descrição: CUBO DE VENDAS

Disponíveis

Associadas ao Cubo: CLIENTE, DATA, REPRESENTANTE, PRODUTO

Enviar Cancelar Desfazer

Internet | Protected Mode: Off 100%

Figura 29 – Tela de definição de dimensões do cubo

O cadastro do cubo de decisão determina que o modelo projetado para o DW esteja pronto para ser carregado com os dados. Desta forma, o processo de ETC é realizado através das fontes de dados que cada dimensão e cubo possuem, pelo roteiro desta fonte de dados que realiza o mapeamento da origem do dado com o projeto definido no DW e, por último, pela

importação dos dados que podem ainda ser agendados e processados periodicamente. A Figura 30 ilustra o cadastro de uma fonte de dados, neste caso para o cubo de decisão VENDA. A Figura 31 ilustra o roteiro desta mesma fonte de dados. Neste caso, por ser um cubo, é necessário mapear tanto os indicadores como as chaves primárias de cada dimensão relacionada. Para as dimensões apenas os atributos necessitam ser roteirizados. A Figura 32 ilustra o momento em que a fonte de dados é importada manualmente.

The image shows a web browser window titled "Ferramenta Data Warehouse - Windows Internet Explorer". The address bar shows the URL: `http://localhost:8080/dw/cadastros/fontedados.jsp?tipo=N&id=0&subid=1&mnid=14`. The main content area displays a form titled "Fonte de Dados (Novo)". The form has the following fields and values:

CUBO	VENDA
Nome	ETC VENDA
Descrição	FONTE DE DADOS CUBO VENDA
Conexão	BASEDADOS
SQL (dados)	SELECT * FROM NOTASAIIDA

At the bottom of the form are three buttons: "Enviar", "Cancelar", and "Desfazer". The browser's status bar at the bottom indicates "Internet | Protected Mode: Off" and a zoom level of "100%".

Figura 30 – Tela de cadastro de fonte de dados

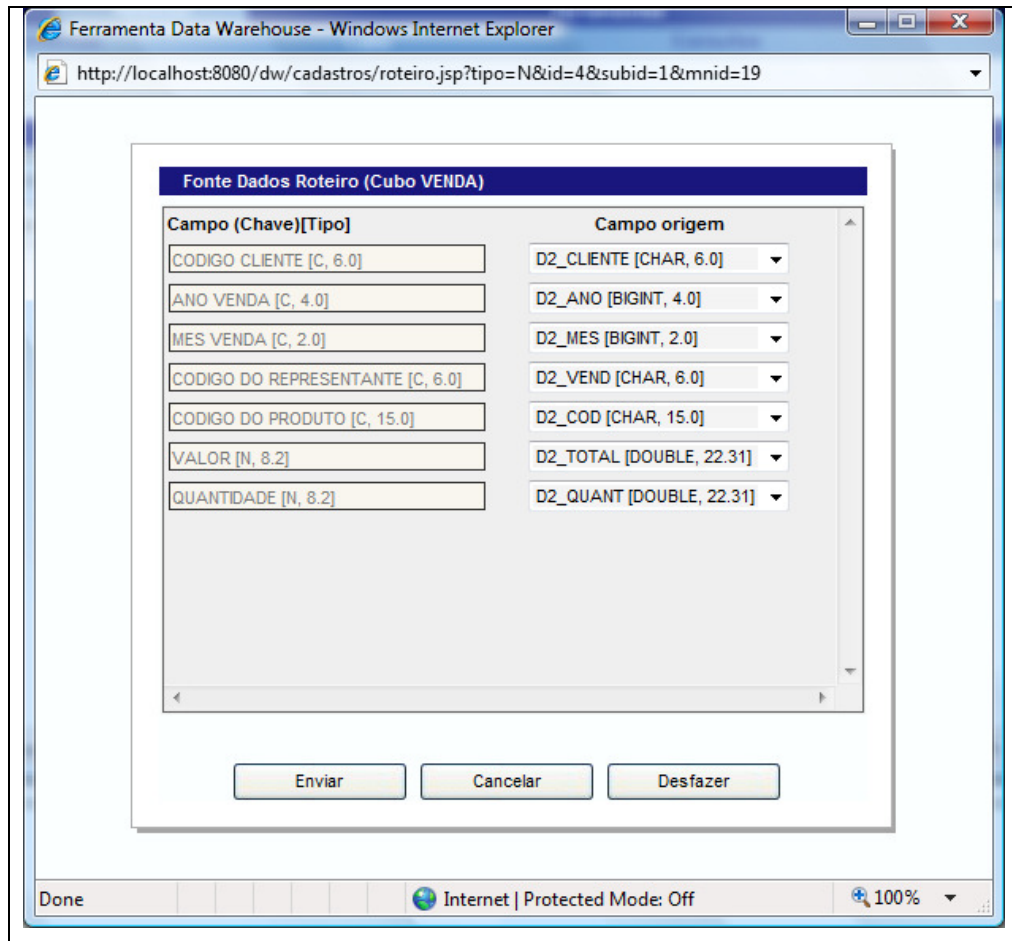


Figura 31 – Tela de roteiro de uma fonte de dados

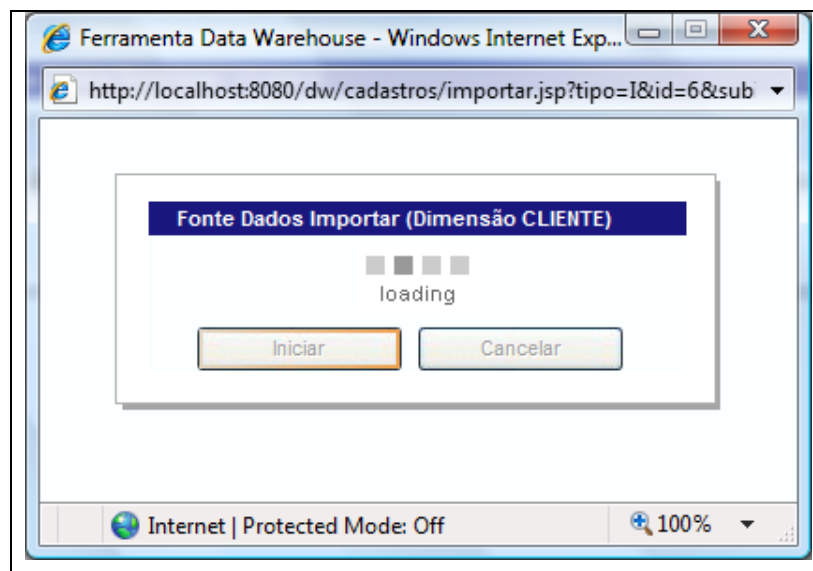


Figura 32 – Tela de importação manual de dados

O próximo passo é a montagem das consultas. O administrador deverá cadastrar a consulta e definir os seus eixos. A Figura 33 ilustra o cadastro de uma consulta e a Figura 34 ilustra a definição de um novo eixo para esta consulta. O eixo de indicadores possui a definição das funções de agregação representada na Figura 35. Para a consulta de vendas por

estado, o administrador irá utilizar o recurso de *drill-down* que é representado pela Figura 36.

Ferramenta Data Warehouse - Windows Internet Explorer
 http://localhost:8080/dw/cadastros/consulta.jsp?tipo=N&id=0&subid=null&mnid=7

Consulta (Novo)

Nome: VENDA ESTADO

Descrição: VENDA ANO, MES E ESTADO

Cubo: VENDA

Enviar Cancelar Desfazer

Internet | Protected Mode: Off 100%

Figura 33 – Tela de cadastro de consultas

Ferramenta Data Warehouse - Windows Internet Explorer
 http://localhost:8080/dw/cadastros/eixo.jsp?tipo=N&exid=0&id=1

Eixos (Consulta VENDA ESTADO)

Eixo: Eixo Y

Atributo: ANO VENDA

Enviar Cancelar

CLIENTE
 NOME DO CLIENTE
 ESTADO DO CLIENTE
 CODIGO CLIENTE

DATA
 ANO VENDA
 MES VENDA

REPRESENTANTE
 NOME DO REPRESENTANTE
 CODIGO DO REPRESENTANTE

PRODUTO

Inter 100%

Figura 34 – Tela de cadastro de eixos da consulta

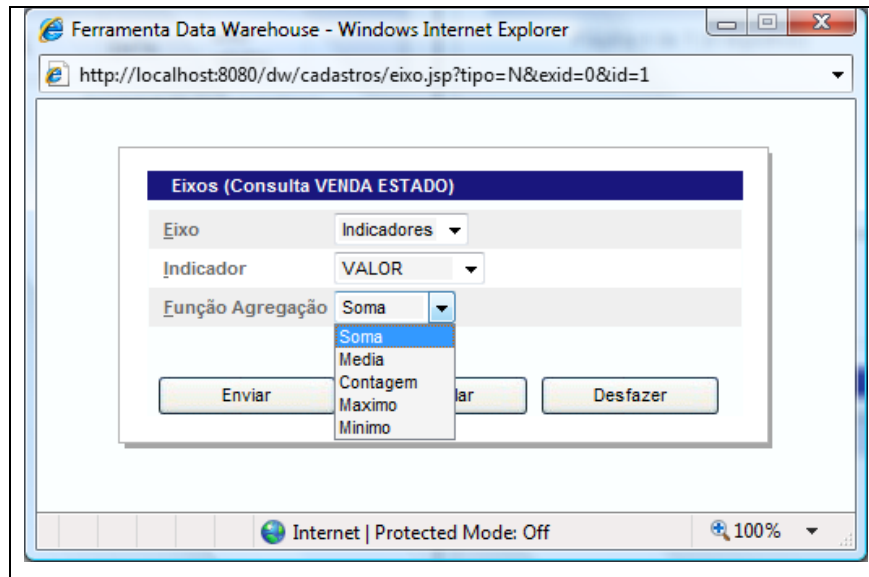


Figura 35 – Tela de cadastro de eixos de indicadores da consulta

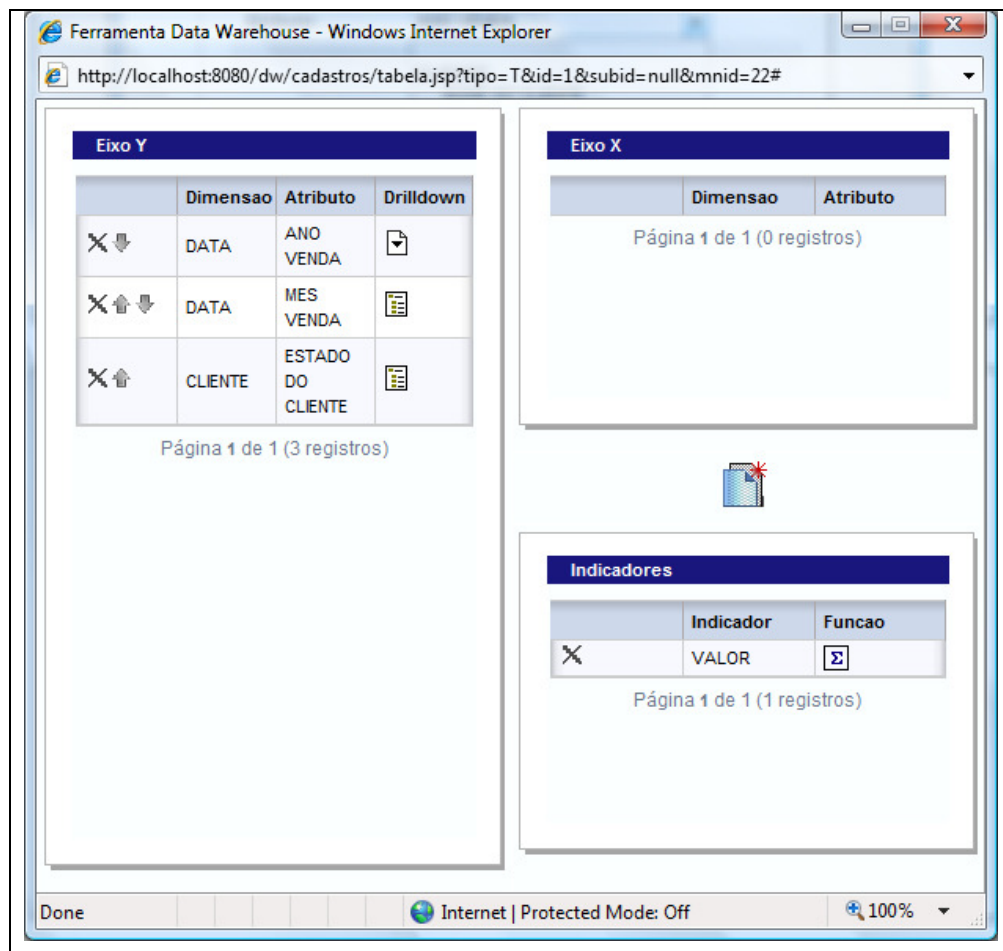


Figura 36 – Tela de definição da consulta com *drill-down*

Para visualizar o resultado das consultas será utilizado um perfil de usuário onde apenas a visualização das consultas, a edição de filtro, alerta e ranking estarão disponíveis. A Figura 37 ilustra a interface de usuário.

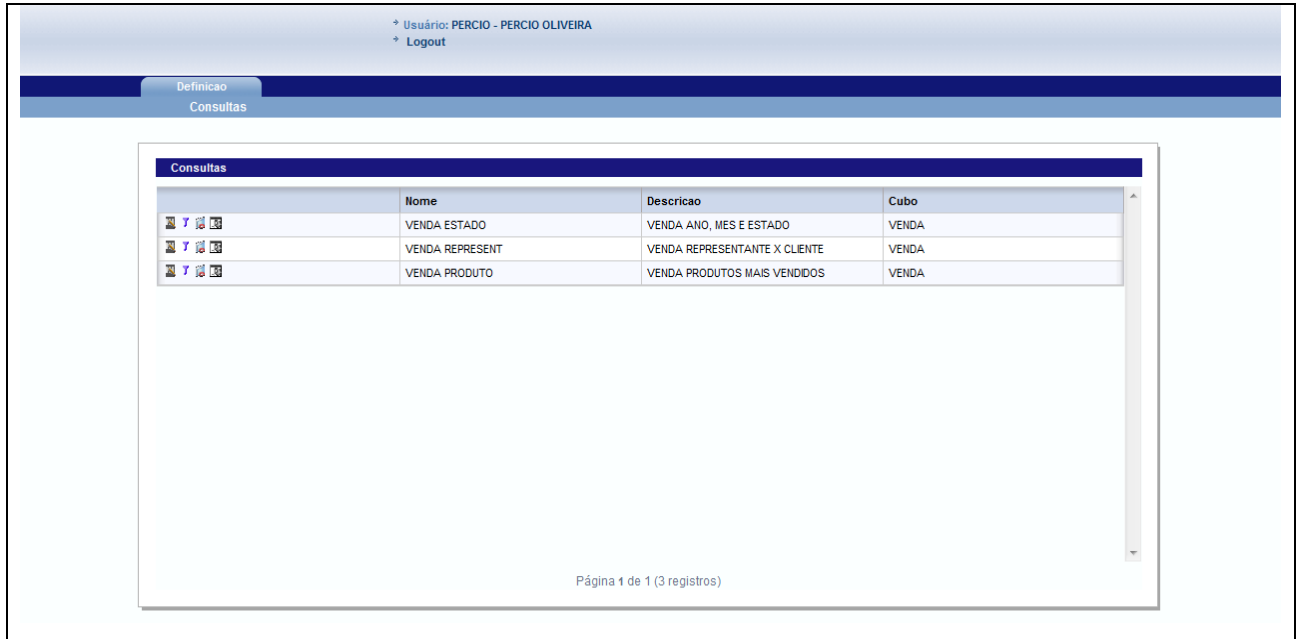


Figura 37 – Tela de consultas para perfil de usuários

As Figura 38, Figura 39 e Figura 40 mostram um exemplo do resultado da consulta de vendas por ano, mês e estado do cliente.

ANO VENDA	VALOR TOTAL DA NOTA FISCAL
2004	3573.57
2005	16122.11
2006	134675.83
2007	784410.36

Page 1 of 1 (4 rows)

Figura 38 – Tela de consulta de vendas por ano com *drill-down*

ANO VENDA	MES VENDA	VALOR TOTAL DA NOTA FISCAL
2007	1	220440.95
	2	198277.83
	3	179166.66
	4	160096.65
	5	26428.27

Page 1 of 1 (5 rows)

Figura 39 – Tela de consulta de vendas por ano e mês com *drill-down*

ANO VENDA	MES VENDA	ESTADO DO CLIENTE	VALOR TOTAL DA NOTA FISCAL
2007	3	PR	2893.62
		RJ	248.00
		RS	5335.00
		SC	170553.66
		SP	136.38

Page 1 of 1 (5 rows)

Figura 40 – Tela de consulta de vendas por ano, mês e estado com *drill-down*

Para montar a consulta de vendas por representantes e clientes é utilizado o recurso de alerta para destacar os resultados obtidos. A Figura 41 ilustra o cadastro de um alerta.

Ferramenta Data Warehouse - Windows Internet Explorer

http://localhost:8080/dw/cadastros/alerta.jsp?tipo=L&id=2&subid=null&mnid=25

Alerta (VENDA REPRESENT)

Indicador: VALOR

Operador: Menor

Quantidade: 200.0

Em condição verdadeira:

Cor da Fonte: #FFFFFF

Cor do Fundo: #FF3333

Em condição não verdadeira:

Cor da Fonte: #FFFFFF

Cor do Fundo: #009900

Enviar Cancelar Desfazer

Internet | Protected Mode: Off 100%

Figura 41 – Tela de cadastro de alerta para consultas

As Figura 42 e Figura 43 mostram um exemplo do resultado da consulta de vendas por representante e clientes.

Ferramenta Data Warehouse - Windows Internet Explorer

http://localhost:8080/dw/cadastros/vertabela.jsp?pos=-1&id=2&valor=

NOME DO REPRESENTANTE	VALOR TOTAL DA NOTA FISCAL
EDUARDO VIEIRA	787.73
FRANCISCO MARTINS	784.75
JOSE LIMA DA SILVA	413.42

Page 1 of 1 (3 rows)

Done Internet | Protected Mode: Off 100%

Figura 42 – Tela de consulta de vendas por representante com alerta

NOME DO REPRESENTANTE	NOME DO CLIENTE	VALOR TOTAL DA NOTA FISCAL
JOSE LIMA DA SILVA	8A. SUPERINTEND. DE POLICIA RO	58.22
	A4 PRINT SERVICE COM E SERVICO	264.61
	ACIB - ASSOCIACAO EMPRESARIAL	145.24
	ADALBERTO SIDNEI DE MENEZES	129.22
	ADELITA NIEHUES DE AVIS ME	135.73
	ADEMAR DESPACHANTE LTDA	181.55
	ALBANY INTERNATIONAL TECIDOS T	1202.32
	ALBERTO SCHUTZ NETO	470.38
	ALCINO MIRANDA	130.12
	ALFA PAPELARIA LTDA ME	138.93
	ALFA PRINT EDITORA E GRAFICA L	284.28
	ALTENBURG INDUSTRIA TEXTIL LTD	546.81
	AMC TEXTIL LTDA	1068.00
	ANA DA GLORIA DA SILVA MAIA	57.40
	ANA HELENA RIBAS DE ALMEIDA	302.20
	ANA MARIA SCHWERENDT	59.29
	ANA PAULA HOESELMANN	180.60
	ANSELMO VON ZESCHAU	367.68
	APP E.E.B. ERWIN RADTKE	520.00
	APP ESC.MUN.ENS.FUND. ERICH BL	271.46
	APP ESCOLA BASICA MUN. HERMANN	390.44
	APP ESCOLA BASICA MUNIC.PAULIN	124.00
	APP ESCOLA DE ED.BASICA PREF.L	113.48
	AQUARELA DIGITAL LTDA ME	320.98
	ARTELIER DA CASA	258.44

Page 1 2 3 4 5 > >> of 7 (311 rows)

Figura 43 – Tela de consulta de vendas por representante e clientes com alerta

Para montar a consulta de vendas por produto são utilizados os recursos de filtro e ranking. A Figura 44 ilustra o cadastro do ranking e a Figura 45 ilustra o cadastro de filtro.

Ranking (VENDA PRODUTO)

Indicador: QUANTIDADE

Processar: Maiores

Quantidade: 20

Enviar Cancelar Desfazer

Figura 44 – Tela de cadastro de ranking para consultas

Ferramenta Data Warehouse - Windows Internet Explorer

http://localhost:8080/dw/cadastros/filtro.jsp?tipo=F&id=3&subid=null&mnid=26

Filtro (VENDA PRODUTO)

Indicador/Atributo: ANO VENDA

Adicionar

Expressao: A.0005 = 2007

Enviar Cancelar Limpar Filtro

Internet | Protected Mode: Off 100%

Figura 45 – Tela de cadastro de filtro para consultas

A Figura 46 mostra um exemplo do resultado da consulta de produtos.

Ferramenta Data Warehouse - Windows Internet Explorer

http://localhost:8080/dw/cadastros/vertabela.jsp?pos=-1&id=3&valor=

ANO VENDA	NOME DO PRODUTO	QUANTIDADE DO ITEM DA NOTA
2007	PAP. COUCHE FOSCO 170GR A3	1000.00
	PAP. COUCHE FOSCO 170GR A4	1000.00
	FITA 12MM PRETO/BRANCO M231	209.00
	CAIXA PAPEL A4 75GR	203.00
	TONER GPR-2- IR 400	174.00
	TONER MAGENTA CLC1120/1100	158.00
	TONER YELLOW CLC 1120/1100	156.00
	TONER BLACK CLC 1120/1100	141.00
	TONER IR 1310/1630/1670	137.00
	TONER CYAN CLC 1120/1100	133.00
	TONER NP6412/7130/6012	118.00
	TONER GPR-19 IR7105	92.00
	TONER GPR-18 2016/2020	86.00
	TONER NPG-1 2120/6115/6221	61.00
	TONER GPR-7 IR85/105	60.00
	GARRA DE SEPARACAO SUPERIOR	53.00
	TONER B6300	51.00
	CAPA PRETA A4	50.00
	TONER GPR-8 -IR1600/2000	49.00
	TONER TN250	46.00

Page 1 of 1 (20 rows)

Done Internet | Protected Mode: Off 100%

Figura 46 – Tela de consulta de vendas por produto com filtro e ranking

3.4 RESULTADOS E DISCUSSÃO

A ferramenta de DW apresentada neste trabalho foi desenvolvida com o propósito final de garantir desempenho e usabilidade funcional nas consultas gerenciais que se utilizam de grande volume de dados gravados em bases históricas e banco de dados transacionais.

Para avaliar o resultado obtido é utilizado o estudo de caso citado neste trabalho para apresentação da operacionalidade do sistema e a ferramenta MySQL Query Browser. É feita uma comparação entre as consultas processadas diretamente no banco transacional e um banco de dados gerado pela ferramenta de DW. Os pontos avaliados são: tempo de processamento de um consulta, total de tabelas consultadas e quantidade de registros processados.

O Quadro 24 mostra os resultados obtidos para a consulta de vendas por ano, mês e estado do cliente em cada nível de detalhamento dos dados.

Venda ano, mês e estado						
Nível	Tempo processamento (s)		Total de tabelas		Total de registros	
	BD	DW	BD	DW	BD	DW
1	0,0185	0,0064	1	2	3518	2323 x 1
2	0,0192	0,0066	1	2	3518	2323 x 1
3	1,2249	0,0164	2	3	982 x 3518	27 x 145 x 1

Quadro 24 – Quadro de resultados da consulta venda por data e estado

O Quadro 25 mostra os resultados obtidos para a consulta de vendas por representante e clientes.

Venda representante e cliente						
Nível	Tempo processamento (s)		Total de tabelas		Total de registros	
	BD	DW	BD	DW	BD	DW
1	0,0414	0,0119	2	2	3 x 3518	4 x 232
2	4,3536	0,0145	3	3	3 x 3518 x 982	4 x 232 x 1

Quadro 25 – Quadro de resultados da consulta venda por representante e cliente

O Quadro 26 mostra os resultados obtidos para a consulta de venda por ano e produto.

Venda produto						
Nível	Tempo processamento (s)		Total de tabelas		Total de registros	
	BD	DW	BD	DW	BD	DW
1	18,1847	0,0195	2	3	3518 x 15969	27 x 145 x 1

Quadro 26 – Quadro de resultados da consulta venda por produto

O processo de ETC das principais fontes de dados envolvidas neste projeto é que é responsável por este desempenho favorável ao DW, tornando as consultas mais eficientes e rápidas. O Quadro 27 mostra o tempo exigido pelo processamento de ETC. A otimização

deste processo através de consultas SQL, bancos de dados indexados e relacionamentos corretos é que vão garantir um processo de ETC cada vez mais eficiente.

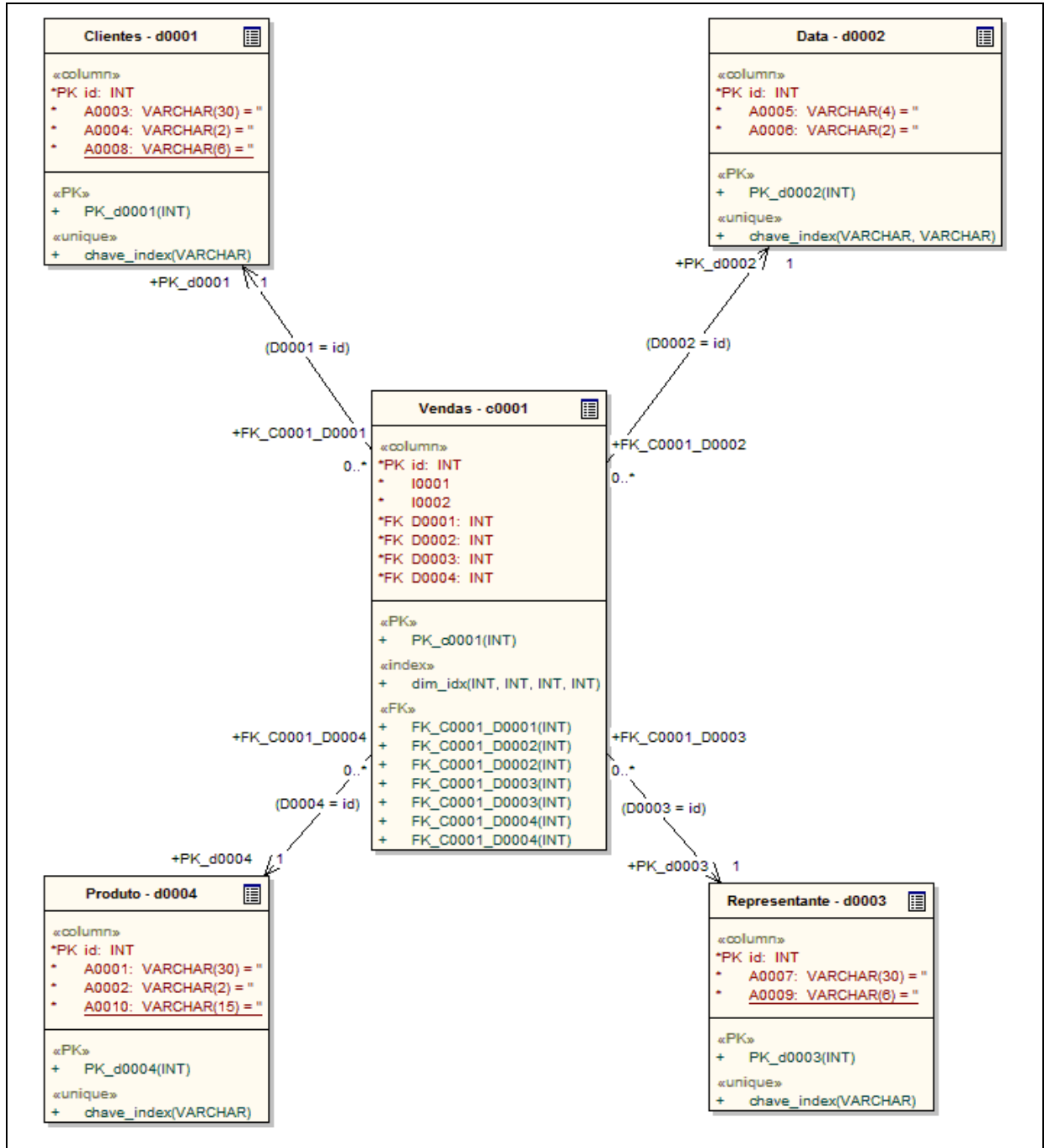
	Tempo processamento (min.)	Total de registros
Dimensão Cliente	1m 51s	970
Dimensão Produto	26min 12s	11594
Dimensão Data	5min 23s	3496
Dimensão Representante	< 1s	3
Cubo Venda	6min 45s	4359

Quadro 27 – Quadro de resultados do processo de ETC

Uma das principais vantagens de migrar os dados transacionais para um banco de dados DW é a organização dos dados garantindo a integridade e qualidade com que os dados são gravados. É no processo de ETC que as informações passam a ser distribuídas e modeladas seguindo as definições do projeto criado na ferramenta. Para obter qualidade dos dados a ferramenta implementa conceitos de chave primária dentro das dimensões e chave estrangeira dos cubos em relação às dimensões. Cada chave primária definida na dimensão garante a unicidade dos registros, não permitindo ocorrências duplicadas. Servem ainda como referência na montagem das chaves primárias dos cubos de decisão. A ferramenta utiliza-se destas referências para executar a limpeza dos dados garantindo que registros que não possuem integridade referencial válidas sejam descartados.

Pode-se afirmar ainda em relação aos trabalhos correlatos, que a ferramenta possui características bem parecidas no quesito técnicas e conceitos de implementação. Porém, como a mesma é desenvolvida em ambiente web, tornando-se acessível através de qualquer navegador de internet, o usuário poderá controlar e analisar suas consultas gerenciais de qualquer computador com acesso a rede internet, tendo assim um grande diferencial de usabilidade.

Para o projeto do estudo de caso o Quadro 28 apresenta as tabelas do banco DW, suas chaves primárias e chaves estrangeiras criadas pela ferramenta. A estrutura criada pela ferramenta e somente acessada por ela, segue a regra de utilizar o caracter “D” para as dimensões, “C” para os cubos, “I” para os indicadores e “A” para os atributos, todos acrescidos do identificador único criado na inclusão do registro nas tabelas do metadados.



Quadro 28 – Modelo entidade relacional criado na ferramenta de DW

4 CONCLUSÕES

Com o propósito principal de obter informações gerenciais detalhadas e resumidas provenientes de banco de dados históricos e transacionais, a ferramenta desenvolvida atingiu os objetivos propostos no início deste trabalho e demonstrou ser bastante eficiente, uma vez que aplica os conceitos e técnicas de um sistema de apoio à decisão, neste caso o DW, auxiliando no processo de extração de dados, transformando-os em informações e apresentando-os de forma a obter indicações da evolução e histórico dos dados. Assim, ela cumpre sua função original que é fornecer informações para o planejamento operacional, tático e até mesmo estratégico de uma organização e principalmente garantir aos administradores e usuários uma interface amigável para consultar e realizar suas análises.

O desenvolvimento desta ferramenta apresentou técnicas de modelagem de dados que mostram cada vez mais o quanto é importante organizar, referenciar e principalmente garantir a integridade dos dados, transformando-os em informações de grande valor para as organizações. Tenho como aprendizado conceitos de organização de dados, que anteriormente eram vistos apenas como registros lançados dentro de uma base transacional, sem uso e principalmente sem valor agregado.

A ferramenta através dos cubos de decisão orienta as informações por assunto, permitindo montar consultas para cada característica em comum que os dados possam apresentar. Ainda, de forma integrada, preocupa-se em trazer dados que possuam informações idênticas, porém de diferentes fontes unificando o estado do dado.

O metadados do projeto de DW pode ser exportado em formato XML, incluindo toda a estrutura de objetos referente ao modelo de dados e servindo como uma documentação e referência futura a novos projetos.

As ferramentas escolhidas para a construção deste sistema proporcionaram a produção desejada. A tecnologia AJAX junto com as técnicas de *grid* forneceram ao sistema um ambiente web dinâmico e visualmente agradável.

Todas as ferramentas proprietárias relacionadas nos trabalhos correlatos são de alto custo, o que leva os usuários sem recursos financeiros a manter planilhas gerenciais como forma de controlar e centralizar seus dados operacionais, sem uma técnica específica e segurança confiável.

4.1 EXTENSÕES

Como extensão do presente trabalho sugere-se a abertura da ferramenta para novos bancos de dados, e a utilização de um *framework* de persistência de dados como o Hibernate utilizando mapeamentos objeto relacional que garantem uma aplicação independente do banco de dados.

Pode-se também implementar alguns outros recursos como: campo virtual nas consultas que permitem ao usuário utilizando os indicadores já existentes determinar novos indicadores durante o processamento de uma consulta; novas funções de agregação além das já existentes como percentual de participação total, média interna e acumulado; processar ranking por percentual de pareto e curva ABC; controle de usuários por projetos, cubos e edição de consultas; algoritmos de mineração de dados dentro da ferramenta para viabilizar a aplicação em outros conceitos de inteligência de negócios.

A apresentação visual da ferramenta pode-se agregar novos recursos principalmente na visualização das consultas em modelo tabela com opções de seleção e totalizadores e gráficos com montagem de painéis padronizados de indicação.

A integração com ferramentas de relatórios ou planilhas como Crystal Reports e Excel também podem permitir opções para disposição das informações, além de serem ferramentas já de grande aceitação no mercado.

Para a modelagem dimensional dos projetos de DW pode-se viabilizar o uso de outras técnicas além do modelo estrela, como o *snow flake* que permite o relacionamento entre dimensões.

Sugere-se também a abertura dos métodos de atualização das fontes de dados, permitindo ao usuário definir o tipo de método: atualização, inserção, aceitar registros inválidos, rejeitar registros inválidos ou ainda ignorar os registros inválidos.

REFERÊNCIAS BIBLIOGRÁFICAS

BARBIERI, Carlos. **BI – Business Intelligence – Modelagem & Tecnologia**. Rio de Janeiro: Editora Axel Books, 2001.

BISPO, Carlos. **Uma análise da nova geração de sistemas de apoio a decisão**. 1998. 174 f. Dissertação (Mestrado em Engenharia da Produção) – Curso de Pós-graduação em Engenharia da Produção, Universidade de São Paulo, São Carlos. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/18/18140/tde-04042004-152849/>>. Acesso em: 19 set. 2006.

COME, Gilberto. **Contribuição ao estudo da implementação de data warehousing: um caso no setor de telecomunicações**. 2001. 132 f. Dissertação (Mestrado em Administração) – Curso de Pós-graduação em Administração, Universidade de São Paulo, São Paulo. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/12/12139/tde-19042002-183028/>>. Acesso em: 19 set. 2006.

CORRADI, Lauricio. **Sistema de informação gerencial aplicado a comercialização de planos de previdência privada utilizando a técnica de cubo de decisão**. 2002. 73 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

DW BRASIL. **Características de um data warehouse**. Brasília, 2005. Disponível em: <http://www.dwbrasil.com.br/html/artdw_carac.html>. Acesso em: 11 set. 2006.

INMON, William H. **Como construir o data warehouse**. Tradução Ana Maria Netto Guz. Rio de Janeiro: Campus, 1997.

INMON, William H.; TERDERMAN R.; IMHOFF C. **Data warehousing: como transformar informações em oportunidades de negócios**. São Paulo: Berkeley, 2001.

JARK, Gustavo. **Sistema de informações executivas baseado em data warehouse aplicado a gerenciamento de clientes**. 2002. 54 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

MARQUES, Sandro. **Oracle Discoverer: uma ferramenta de suporte à decisão para o Oracle E-Business Suite (relatórios para o Oracle E-Business Suite)**. São Paulo, 2005. Disponível em: <<http://www.albaspectrum.com/PressReleases/POR-Oracle-Discoverer-Press-Release.htm>>. Acesso em: 12 set. 2006.

MICROSIGA SOFTWARE. **Protheus: help**. Version 8.11. [S.l.], 2006. Documento eletrônico disponibilizado com o Ambiente Protheus 8.11.

MICROSOFT BRASIL. **Gerenciamento de dados corporativos**. São Paulo, 2005. Disponível em: <<http://www.microsoft.com/brasil/sql/2005/overview2.mspx>>. Acesso em: 19 set. 2006.

MOREIRA, Eduardo. **Modelo dimensional para data warehouse**. São Paulo, 2006. Disponível em: <http://www.scielo.br/scielo.php?pid=S0104-530X2005000100006&script=sci_arttext>. Acesso em: 19 set. 2006.

NAVARRO, Maria C. **O que é Data Warehouse?** Brasília, 1996. Disponível em: <<http://www.serpro.gov.br/publicacao/tematec/1996/ttec27>>. Acesso em: 13 mai. 2007.

SILVA, Diogo. **SITC: uma ferramenta de transformação e carga para um data warehouse**. 2005. 31 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Instituto de Matemática, Universidade da Bahia, Salvador. Disponível em: <http://twiki.im.ufba.br/bin/view/MAT057/TrabalhosSemestre20052#SITC_Uma_ferramenta_de_Transform>. Acesso em: 19 set. 2006.

VIEIRA, Fernando. **Alguns conceitos sobre DW**. São Paulo, 2000. Disponível em: <<http://www.datawarehouse.inf.br/>>. Acesso em: 19 set. 2006.

APÊNDICE A – Relação das classes do pacote de negócios, seus atributos e métodos

Classe Agendamento	
Atributos	Tipo
id	private: <i>int</i>
ftd	private: <i>FonteDados</i>
tipo	private: <i>TipoAgendamento</i>
ativo	private: <i>boolean</i>
dtinicio	private: <i>Date</i>
dtfim	private: <i>Date</i>
semana	private: <i>String</i>
mensal	private: <i>String</i>
Métodos	Tipo
Agendamento ()	public:
getAtivo ()	public: <i>boolean</i>
getDtfim ()	public: <i>Date</i>
getDtinicio ()	public: <i>Date</i>
getFtd ()	public: <i>FonteDados</i>
getId ()	public: <i>int</i>
getMensal ()	public: <i>String</i>
getSemana ()	public: <i>String</i>
getTipo ()	public: <i>TipoAgendamento</i>
setAtivo (<i>boolean</i>)	public: <i>void</i>
setDtfim (<i>Date</i>)	public: <i>void</i>
setDtinicio (<i>Date</i>)	public: <i>void</i>
setFtd (<i>FonteDados</i>)	public: <i>void</i>
setId (<i>int</i>)	public: <i>void</i>
setMensal (<i>String</i>)	public: <i>void</i>
setSemana (<i>String</i>)	public: <i>void</i>
setTipo (<i>TipoAgendamento</i>)	public: <i>void</i>
getDiasSemana ()	public: <i>String</i>
getDiasMes ()	public: <i>String</i>
getDia (<i>int</i>)	private: <i>String</i>
ler ()	public: <i>void</i>
remover ()	public: <i>void</i>
incluir ()	public: <i>void</i>
alterar ()	public: <i>void</i>

Quadro 29 – Atributos e métodos da classe Agendamento

Classe Alerta	
Atributos	Tipo
id	private: <i>int</i>
con	private: <i>Consulta</i>
ind	private: <i>Indicador</i>
expressao	private: <i>String</i>
corFonteV	private: <i>String</i>
corFonteF	private: <i>String</i>

corFundoV	private: <i>String</i>
corFundoF	private: <i>String</i>
Métodos	Tipo
Alerta ()	public:
getCon ()	public: <i>Consulta</i>
getCorFonteF ()	public: <i>String</i>
getCorFonteV ()	public: <i>String</i>
getCorFundoF ()	public: <i>String</i>
getCorFundoV ()	public: <i>String</i>
getExpressao ()	public: <i>String</i>
getId ()	public: <i>int</i>
getInd ()	public: <i>Indicador</i>
setCon (<i>Consulta</i>)	public: <i>void</i>
setCorFonteF (<i>String</i>)	public: <i>void</i>
setCorFonteV (<i>String</i>)	public: <i>void</i>
setCorFundoF (<i>String</i>)	public: <i>void</i>
setCorFundoV (<i>String</i>)	public: <i>void</i>
setExpressao (<i>String</i>)	public: <i>void</i>
setId (<i>int</i>)	public: <i>void</i>
setInd (<i>Indicador</i>)	public: <i>void</i>
ler ()	public: <i>void</i>
remover ()	public: <i>void</i>
incluir ()	public: <i>void</i>
alterar ()	public: <i>void</i>

Quadro 30 – Atributos e métodos da classe Alerta

Classe Atributo	
Atributo	Tipo
id	private: <i>int</i>
dim	private: <i>Dimensao</i>
nome	private: <i>String</i>
descricao	private: <i>String</i>
chaveseq	private: <i>int</i>
tamanho	private: <i>int</i>
decimais	private: <i>int</i>
tipo	private: <i>TipoDado</i>
Métodos	Tipo
Atributo ()	public:
getDimensao ()	public: <i>Dimensao</i>
getDescricao ()	public: <i>String</i>
getId ()	public: <i>int</i>
getNome ()	public: <i>String</i>
getTipo ()	public: <i>TipoDado</i>
getChaveSeq ()	public: <i>int</i>
getTamanho ()	public: <i>int</i>
getDecimais ()	public: <i>int</i>
setDimensao (<i>Dimensao</i>)	public: <i>void</i>
setDescricao (<i>String</i>)	public: <i>void</i>
setId (<i>int</i>)	public: <i>void</i>

setNome (<i>String</i>)	public: <i>void</i>
setTipo (<i>TipoDado</i>)	public: <i>void</i>
setTamanho (<i>int</i>)	public: <i>void</i>
setDecimais (<i>int</i>)	public: <i>void</i>
setChaveSeq (<i>int</i>)	public: <i>void</i>
getColDim ()	public: <i>String</i>
getColDef ()	private: <i>String</i>
ler ()	public: <i>void</i>
remover ()	public: <i>void</i>
incluir ()	public: <i>void</i>
alterar ()	public: <i>void</i>

Quadro 31 – Atributos e métodos da classe Atributo

Classe Conexao	
Atributos	Tipo
id	private: <i>int</i>
dw	Private: <i>Dw</i>
nome	private: <i>String</i>
descricao	private: <i>String</i>
tipobanco	private: <i>TipoBanco</i>
servidor	private: <i>String</i>
banco	private: <i>String</i>
usuario	private: <i>String</i>
senha	private: <i>String</i>
Métodos	Tipo
Conexao ()	public:
getUsuario ()	public: <i>String</i>
getNome ()	public: <i>String</i>
getTipoBanco ()	public: <i>TipoBanco</i>
getId ()	public: <i>int</i>
getSenha ()	public: <i>String</i>
getDescricao ()	public: <i>String</i>
getBanco ()	public: <i>String</i>
getServidor ()	public: <i>String</i>
getDw ()	public: <i>Dw</i>
setSenha (<i>String</i>)	public: <i>void</i>
setDescricao (<i>String</i>)	public: <i>void</i>
setId (<i>int</i>)	public: <i>void</i>
setUsuario (<i>String</i>)	public: <i>void</i>
setNome (<i>String</i>)	public: <i>void</i>
setTipoBanco (<i>TipoBanco</i>)	public: <i>void</i>
setBanco (<i>String</i>)	public: <i>void</i>
setServidor (<i>String</i>)	public: <i>void</i>
setDw (<i>Dw</i>)	public: <i>void</i>
testarConexao ()	public: <i>boolean</i>
ler ()	public: <i>void</i>
remover ()	public: <i>void</i>
incluir ()	public: <i>void</i>

alterar ()	public: <i>void</i>
------------	---------------------

Quadro 32 – Atributos e métodos da classe Conexao

Classe Consulta	
Atributos	Tipo
id	private: <i>int</i>
nome	private: <i>String</i>
descricao	private: <i>String</i>
cubo	private: <i>Cubo</i>
Métodos	Tipo
Consulta ()	public:
getDescricao ()	public: <i>String</i>
getId ()	public: <i>int</i>
getNome ()	public: <i>String</i>
getCubo ()	public: <i>Cubo</i>
setDescricao (<i>String</i>)	public: <i>void</i>
setId (<i>int</i>)	public: <i>void</i>
setNome (<i>String</i>)	public: <i>void</i>
setCubo (<i>Cubo</i>)	public: <i>void</i>
getEixoYHTML ()	public: <i>String</i>
getEixoXHTML ()	public: <i>String</i>
getEixoIHTML ()	public: <i>String</i>
getIndicadores ()	public: <i>ArrayList<Indicador></i>
getAtributos ()	public: <i>ArrayList<Atributo></i>
atualizaDrill (<i>Eixo</i>)	public: <i>void</i>
sobe (<i>int, int</i>)	public: <i>void</i>
desce (<i>int, int</i>)	public: <i>void</i>
getPos (<i>int</i>)	private: <i>int</i>
getPosD (<i>int</i>)	private: <i>int</i>
getPosI (<i>int</i>)	private: <i>int</i>
atbExiste (<i>int</i>)	public: <i>boolean</i>
indExiste (<i>int</i>)	public: <i>boolean</i>
getProxPos (<i>TipoEixo</i>)	public: <i>int</i>
refazerPos ()	public: <i>void</i>
getTabelaHTML ()	public: <i>String</i>
ler ()	public: <i>void</i>
remover ()	public: <i>void</i>
incluir ()	public: <i>void</i>
alterar ()	public: <i>void</i>

Quadro 33 – Atributos e métodos da classe Consulta

Classe Cubo	
Atributos	Tipo
id	private: <i>int</i>
nome	private: <i>String</i>
descricao	private: <i>String</i>
dataCriado	private: <i>Date</i>
dw	private: <i>Dw</i>
Métodos	Tipo

Cubo ()	public:
getDataCriado ()	public: <i>Date</i>
getDescricao ()	public: <i>String</i>
getId ()	public: <i>int</i>
getNome ()	public: <i>String</i>
getDw ()	public: <i>Dw</i>
setDescricao (<i>String</i>)	public: <i>void</i>
setId (<i>int</i>)	public: <i>void</i>
setNome (<i>String</i>)	public: <i>void</i>
setDw (<i>Dw</i>)	public: <i>void</i>
getTblCub ()	public: <i>String</i>
getDimensao ()	public: <i>ArrayList<Dimensao></i>
getIndicadores ()	public: <i>ArrayList<Indicador></i>
salvaDimensao (<i>String[]</i> , <i>String[]</i>)	public: <i>void</i>
ler ()	public: <i>void</i>
remover ()	public: <i>void</i>
incluir ()	public: <i>void</i>
alterar ()	public: <i>void</i>

Quadro 34 – Atributos e métodos da classe Cubo

Classe Dimensao	
Atributos	Tipo
id	private: <i>int</i>
nome	private: <i>String</i>
descricao	private: <i>String</i>
dataCriado	private: <i>Date</i>
dw	private: <i>Dw</i>
Métodos	Tipo
Dimensao ()	public:
getDataCriado ()	public: <i>Date</i>
getDescricao ()	public: <i>String</i>
getId ()	public: <i>int</i>
getNome ()	public: <i>String</i>
getDw ()	public: <i>Dw</i>
setDescricao (<i>String</i>)	public: <i>void</i>
setId (<i>int</i>)	public: <i>void</i>
setNome (<i>String</i>)	public: <i>void</i>
setDw (<i>Dw</i>)	public: <i>void</i>
getTblDim ()	public: <i>String</i>
getAtributo ()	public: <i>ArrayList<Atributo></i>
getChavePrimaria ()	public: <i>ArrayList<Atributo></i>
salvaChave (<i>String[]</i>)	public: <i>void</i>
isCubo (<i>int</i>)	public: <i>boolean</i>
ler ()	public: <i>void</i>
remover ()	public: <i>void</i>
incluir ()	public: <i>void</i>
alterar ()	public: <i>void</i>

Quadro 35 – Atributos e métodos da classe Dimensao

Classe Dw	
Atributos	Tipo
id	private: <i>int</i>
nome	private: <i>String</i>
descricao	private: <i>String</i>
icone	private: <i>TipoIcone</i>
dataCriado	private: <i>Date</i>
Métodos	Tipo
Dw ()	public:
getDescricao ()	public: <i>String</i>
getIcone ()	public: <i>TipoIcone</i>
getId ()	public: <i>int</i>
getNome ()	public: <i>String</i>
getDataCriado ()	public: <i>Date</i>
setDescricao (<i>String</i>)	public: <i>void</i>
setIcone (<i>TipoIcone</i>)	public: <i>void</i>
setId (<i>int</i>)	public: <i>void</i>
setNome (<i>String</i>)	public: <i>void</i>
gerarLog (<i>String</i>)	public: <i>void</i>
limparLog ()	public: <i>void</i>
getDimensao ()	public: <i>ArrayList<Dimensao></i>
getCubo ()	public: <i>ArrayList<Cubo></i>
getConexao ()	public: <i>ArrayList<Conexao></i>
incluir ()	public: <i>int</i>
ler ()	public: <i>void</i>
remover ()	public: <i>void</i>
alterar ()	public: <i>void</i>

Quadro 36 – Atributos e métodos da classe Dw

Classe Eixo	
Atributos	Tipo
id	private: <i>int</i>
con	private: <i>Consulta</i>
tipo	private: <i>TipoEixo</i>
atb	private: <i>Atributo</i>
ind	private: <i>Indicador</i>
funcao	private: <i>String</i>
Métodos	Tipo
Eixo ()	public:
getId ()	public: <i>int</i>
getTipo ()	public: <i>TipoEixo</i>
getConsulta ()	public: <i>Consulta</i>
getAtributo ()	public: <i>Atributo</i>
getIndicador ()	public: <i>Indicador</i>
getFuncao ()	public: <i>String</i>
setId (<i>int</i>)	public: <i>void</i>
setTipo (<i>TipoEixo</i>)	public: <i>void</i>
setConsulta (<i>Consulta</i>)	public: <i>void</i>
setAtributo (<i>Atributo</i>)	public: <i>void</i>

setIndicador (<i>Indicador</i>)	public: <i>void</i>
setFuncao (<i>String</i>)	public: <i>void</i>
remover ()	public: <i>void</i>
incluir ()	public: <i>void</i>

Quadro 37 – Atributos e métodos da classe Eixo

Classe Filtro	
Atributos	Tipo
id	private: <i>int</i>
con	private: <i>Consulta</i>
expressao	private: <i>String</i>
Métodos	Tipo
Filtro ()	public:
getCon ()	public: <i>Consulta</i>
getExpressao ()	public: <i>String</i>
getId ()	public: <i>int</i>
setCon (<i>Consulta</i>)	public: <i>void</i>
setExpressao (<i>String</i>)	public: <i>void</i>
setId (<i>int</i>)	public: <i>void</i>
ler ()	public: <i>void</i>
remover ()	public: <i>void</i>
incluir ()	public: <i>void</i>
alterar ()	public: <i>void</i>

Quadro 38 – Atributos e métodos da classe Filtro

Classe FonteDados	
Atributo	Tipo
id	private: <i>int</i>
dim	private: <i>Dimensao</i>
cub	private: <i>Cubo</i>
cnx	private: <i>Conexao</i>
nome	private: <i>String</i>
descricao	private: <i>String</i>
sqldados	private: <i>String</i>
Métodos	Tipo
FonteDados ()	public:
getDimensao ()	public: <i>Dimensao</i>
getDescricao ()	public: <i>String</i>
getId ()	public: <i>int</i>
getNome ()	public: <i>String</i>
getSqlDados ()	public: <i>String</i>
getCubo ()	public: <i>Cubo</i>
getConexao ()	public: <i>Conexao</i>
setDescricao (<i>String</i>)	public: <i>void</i>
setId (<i>int</i>)	public: <i>void</i>
setNome (<i>String</i>)	public: <i>void</i>
setSqlDados (<i>String</i>)	public: <i>void</i>
setDimensao (<i>Dimensao</i>)	public: <i>void</i>
setCubo (<i>Cubo</i>)	public: <i>void</i>

setConexao (<i>Conexao</i>)	public: <i>void</i>
getCampos ()	public: <i>ArrayList<String[]></i>
getCamposOrigem ()	public: <i>ArrayList<String[]></i>
getNomeDimCub ()	public: <i>String</i>
limpaRoteiro ()	public: <i>void</i>
salvaRoteiro (<i>String, int</i>)	public: <i>void</i>
isRoteiro (<i>String, int</i>)	public: <i>boolean</i>
getIdTabela (<i>String</i>)	public: <i>int</i>
cpRoteiro (<i>int</i>)	public: <i>String</i>
importarDados ()	public: <i>boolean</i>
ler ()	public: <i>void</i>
remover ()	public: <i>void</i>
incluir ()	public: <i>void</i>
alterar ()	public: <i>void</i>

Quadro 39 – Atributos e métodos da classe FonteDados

Classe Indicador	
Atributos	Tipo
id	private: <i>int</i>
cub	private: <i>Cubo</i>
nome	private: <i>String</i>
descricao	private: <i>String</i>
tamanho	private: <i>int</i>
decimais	private: <i>int</i>
tipo	private: <i>TipoDado</i>
Métodos	Tipo
Indicador ()	public:
getCubo ()	public: <i>Cubo</i>
getDescricao ()	public: <i>String</i>
getId ()	public: <i>int</i>
getNome ()	public: <i>String</i>
getTipo ()	public: <i>TipoDado</i>
getTamanho ()	public: <i>int</i>
getDecimais ()	public: <i>int</i>
setCubo (<i>Cubo</i>)	public: <i>void</i>
setDescricao (<i>String</i>)	public: <i>void</i>
setId (<i>int</i>)	public: <i>void</i>
setNome (<i>String</i>)	public: <i>void</i>
setTipo (<i>TipoDado</i>)	public: <i>void</i>
setTamanho (<i>int</i>)	public: <i>void</i>
setDecimais (<i>int</i>)	public: <i>void</i>
getColCub ()	public: <i>String</i>
getColDef ()	private: <i>String</i>
ler ()	public: <i>void</i>
remover ()	public: <i>void</i>
incluir ()	public: <i>void</i>
alterar ()	public: <i>void</i>

Quadro 40 – Atributos e métodos da classe Indicador

Classe Mensagem	
Atributos	Tipo
id	private: <i>int</i>
dataMsg	private: <i>Date</i>
dw	private: <i>Dw</i>
mensagem	private: <i>String</i>
Métodos	Tipo
Mensagem ()	public:
Mensagem (<i>Dw, String</i>)	public:
getDataMsg ()	public: <i>Date</i>
getDw ()	public: <i>Dw</i>
getId ()	public: <i>int</i>
getMensagem ()	public: <i>String</i>
setDataMsg (<i>Date</i>)	public: <i>void</i>
setDw (<i>Dw</i>)	public: <i>void</i>
setId (<i>int</i>)	public: <i>void</i>
setMensagem (<i>String</i>)	public: <i>void</i>
removerAll ()	public: <i>void</i>
incluir ()	public: <i>void</i>

Quadro 41 – Atributos e métodos da classe Mensagem

Classe Ranking	
Atributos	Tipo
id	private: <i>int</i>
con	private: <i>Consulta</i>
ind	private: <i>Indicador</i>
tipo	private: <i>TipoRanking</i>
quantidade	private: <i>int</i>
Métodos	Tipo
Ranking ()	public:
getCon ()	public: <i>Consulta</i>
getId ()	public: <i>int</i>
getInd ()	public: <i>Indicador</i>
getQuantidade ()	public: <i>int</i>
getTipo ()	public: <i>TipoRanking</i>
setCon (<i>Consulta</i>)	public: <i>void</i>
setId (<i>int</i>)	public: <i>void</i>
setInd (<i>Indicador</i>)	public: <i>void</i>
setQuantidade (<i>int</i>)	public: <i>void</i>
setTipo (<i>TipoRanking</i>)	public: <i>void</i>
ler ()	public: <i>void</i>
remover ()	public: <i>void</i>
incluir ()	public: <i>void</i>
alterar ()	public: <i>void</i>

Quadro 42 – Atributos e métodos da classe Ranking

Classe Usuario	
Atributos	Tipo
id	private: <i>int</i>

nome	private: <i>String</i>
tipo	private: <i>TipoUsuario</i>
login	private: <i>String</i>
senha	private: <i>String</i>
email	private: <i>String</i>
Métodos	Tipo
Usuario ()	public:
getLogin ()	public: <i>String</i>
getNome ()	public: <i>String</i>
getTipo ()	public: <i>TipoUsuario</i>
getId ()	public: <i>int</i>
getSenha ()	public: <i>String</i>
getEmail ()	public: <i>String</i>
setSenha (<i>String</i>)	public: <i>void</i>
setEmail (<i>String</i>)	public: <i>void</i>
setId (<i>int</i>)	public: <i>void</i>
setLogin (<i>String</i>)	public: <i>void</i>
setNome (<i>String</i>)	public: <i>void</i>
setTipo (<i>TipoUsuario</i>)	public: <i>void</i>
validarLogin ()	public: <i>boolean</i>
getDw ()	public: <i>ArrayList<Dw></i>
ler ()	public: <i>void</i>
remover (<i>int</i>)	public: <i>void</i>
incluir ()	public: <i>void</i>
alterar ()	public: <i>void</i>

Quadro 43 – Atributos e métodos da classe Usuario

Classe TipoAgendamento	
Atributos	Tipo
Diario «enum»	public:
Semanal «enum»	public:
Mensal «enum»	public:

Quadro 44 – Atributos e métodos da classe TipoAgendamento

Classe TipoBanco		
Atributos	Tipo	Parâmetro
MySql «enum»	public:	""com.mysql.jdbc.Driver""
SqlServer «enum»	public:	""com.microsoft.jdbc.sqlserver.SQLServerDriver""
valor	private: <i>String</i>	
Métodos	Tipo	
TipoBanco (<i>String</i>)	package:	
getValor ()	public: <i>String</i>	

Quadro 45 – Atributos e métodos da classe TipoBanco

Classe TipoDado		
Atributos	Tipo	Parâmetro
Caracter «enum»	public:	""VARCHAR""
Numerico «enum»	public:	""FLOAT""
Data «enum»	public:	""DATETIME""

valor	private: <i>String</i>	
Métodos	Tipo	
TipoDado (<i>String</i>)	package:	
getValor ()	public: <i>String</i>	

Quadro 46 – Atributos e métodos da classe TipoBanco

Classe TipoEixo		
Atributos	Tipo	Parâmetro
X «enum»	public:	""Eixo X""
Y «enum»	public:	""Eixo Y""
I «enum»	public:	""Indicadores""
valor	private: <i>String</i>	
Métodos	Tipo	
TipoEixo (<i>String</i>)	package:	
getValor ()	public: <i>String</i>	

Quadro 47 – Atributos e métodos da classe TipoEixo

Classe TipoIcone		
Atributos	Tipo	Parâmetro
Padrão «enum»	public:	""dw_new.gif""
Produção «enum»	public:	""dw_fab.gif""
Financeiro «enum»	public:	""dw_fin.gif""
RH «enum»	public:	""dw_rh.gif""
Comercial «enum»	public:	""dw_ven.gif""
valor	private: <i>String</i>	
Métodos	Tipo	
TipoIcone (<i>String</i>)	package:	
getValor ()	public: <i>String</i>	

Quadro 48 – Atributos e métodos da classe TipoIcone

Classe TipoRanking		
Atributos	Tipo	Parâmetro
Maiores «enum»	public:	""DESC""
Menores «enum»	public:	""ASC""
valor	private: <i>String</i>	
Métodos	Tipo	
TipoRanking (<i>String</i>)	package:	
getValor ()	public: <i>String</i>	

Quadro 49 – Atributos e métodos da classe TipoRanking

Classe TipoUsuario	
Atributos	Tipo
Usuario «enum»	public:
Administrador «enum»	public:
UsuarioSistema «enum»	public:

Quadro 50 – Atributos e métodos da classe TipoUsuario

APÊNDICE B – Relação das tabelas que compõe o modelo da ferramenta

Tabela dw001 – Data Warehouse	
Coluna	Tipo
id	<i>INT (Primary Key)</i>
nome	<i>VARCHAR(15)</i>
descricao	<i>VARCHAR(50)</i>
dataCriado	<i>DATETIME</i>
icone	<i>VARCHAR(20)</i>

Quadro 51 – Tabela dw001 – Data Warehouse

Tabela dw002 – Conexão	
Coluna	Tipo
id	<i>INT (Primary Key)</i>
id_dw	<i>INT (Foreign Key)</i>
nome	<i>VARCHAR(12)</i>
descricao	<i>VARCHAR(30)</i>
tipo_banco	<i>VARCHAR(50)</i>
servidor	<i>VARCHAR(30)</i>
banco	<i>VARCHAR(20)</i>
usuario	<i>VARCHAR(10)</i>
senha	<i>VARCHAR(10)</i>

Quadro 52 – Tabela dw002 - Conexão

Tabela dw003 – Dimensão	
Coluna	Tipo
id	<i>INT (Primary Key)</i>
id_dw	<i>INT (Foreign Key)</i>
nome	<i>VARCHAR(15)</i>
descricao	<i>VARCHAR(30)</i>
dataCriado	<i>DATETIME</i>

Quadro 53 – Tabela dw003 - Dimensão

Tabela dw004 - Atributo	
Coluna	Tipo
id	<i>INT (Primary Key)</i>
id_dim	<i>INT (Foreign Key)</i>
chave_seq	<i>INT</i>
nome	<i>VARCHAR(30)</i>
descricao	<i>VARCHAR(30)</i>
tipo	<i>VARCHAR(10)</i>
tamanho	<i>INT</i>
decimais	<i>INT</i>

Quadro 54 – Tabela dw004 - Atributo

Tabela dw005 – Cubo	
Coluna	Tipo
id	<i>INT (Primary Key)</i>
id_dw	<i>INT (Foreign Key)</i>

nome	<i>VARCHAR(15)</i>
descricao	<i>VARCHAR(30)</i>
dataCriado	<i>DATETIME</i>

Quadro 55 – Tabela dw005 - Cubo

Tabela dw006 - Indicador	
Coluna	Tipo
id	<i>INT (Primary Key)</i>
id_cubo	<i>INT (Foreign Key)</i>
nome	<i>VARCHAR(30)</i>
descricao	<i>VARCHAR(30)</i>
tipo	<i>VARCHAR(10)</i>
tamanho	<i>INT</i>
decimais	<i>INT</i>

Quadro 56 – Tabela dw006 - Indicador

Tabela dw007 – Dimensão x Cubo	
Coluna	Tipo
id	<i>INT (Primary Key)</i>
id_cubo	<i>INT (Foreign Key)</i>
id_dim	<i>INT (Foreign Key)</i>

Quadro 57 – Tabela dw007 – Dimensão x Cubo

Tabela dw008 - Usuário	
Coluna	Tipo
id	<i>INT (Primary Key)</i>
login	<i>VARCHAR(15)</i>
senha	<i>VARCHAR(20)</i>
nome	<i>VARCHAR(30)</i>
email	<i>VARCHAR(40)</i>
tipo	<i>VARCHAR(20)</i>

Quadro 58 – Tabela dw008 - Usuário

Tabela dw009 – Usuário x Consulta	
Coluna	Tipo
id	<i>INT (Primary Key)</i>
id_usuario	<i>INT (Foreign Key)</i>
id_consulta	<i>INT (Foreign Key)</i>

Quadro 59 – Tabela dw009 – Usuário x Consulta

Tabela dw010 - Consulta	
Coluna	Tipo
id	<i>INT (Primary Key)</i>
id_cubo	<i>INT (Foreign Key)</i>
nome	<i>VARCHAR(15)</i>
descricao	<i>VARCHAR(30)</i>

Quadro 60 – Tabela dw010 - Consulta

Tabela dw011 – Fonte Dados	
Coluna	Tipo
id	<i>INT (Primary Key)</i>
id_dim	<i>INT (Foreign Key)</i>

id_cubo	<i>INT</i> (Foreign Key)
id_conexao	<i>INT</i> (Foreign Key)
nome	<i>VARCHAR</i> (15)
descricao	<i>VARCHAR</i> (50)
sqldados	<i>VARCHAR</i> (255)

Quadro 61 – Tabela dw011 – Fonte Dados

Tabela dw013 - Roteiro	
Coluna	Tipo
id	<i>INT</i> (Primary Key)
id_fonte	<i>INT</i> (Foreign Key)
cpoDest	<i>VARCHAR</i> (20)
id_tabela	<i>INT</i>

Quadro 62 – Tabela dw013 - Roteiro

Tabela dw014 - Agendamento	
Coluna	Tipo
id	<i>INT</i> (Primary Key)
id_fonte	<i>INT</i> (Foreign Key)
tipo	<i>VARCHAR</i> (10)
ativo	<i>TINYINT</i>
inicio	<i>DATETIME</i>
fim	<i>DATETIME</i>
semana	<i>VARCHAR</i> (7)
mensal	<i>VARCHAR</i> (31)

Quadro 63 – Tabela dw014 - Agendamento

Tabela dw015 - Eixo	
Coluna	Tipo
id	<i>INT</i> (Primary Key)
id_consulta	<i>INT</i> (Foreign Key)
id_x	<i>INT</i> (Foreign Key)
id_y	<i>INT</i> (Foreign Key)
id_i	<i>INT</i> (Foreign Key)
posX	<i>INT</i>
posY	<i>INT</i>
posI	<i>INT</i>
posD	<i>INT</i>
func	<i>VARCHAR</i> (10)

Quadro 64 – Tabela dw015 - Eixo

Tabela dw016 - Ranking	
Coluna	Tipo
id	<i>INT</i> (Primary Key)
id_consulta	<i>INT</i> (Foreign Key)
id_indicador	<i>INT</i> (Foreign Key)
tipo	<i>VARCHAR</i> (20)
quantidade	<i>INT</i>

Quadro 65 – Tabela dw016 - Ranking

Tabela dw017 - Alerta	
------------------------------	--

Coluna	Tipo
id	<i>INT</i> (Primary Key)
id_consulta	<i>INT</i> (Foreign Key)
id_indicador	<i>INT</i> (Foreign Key)
operador	<i>VARCHAR</i> (15)
quantidade	<i>FLOAT</i>
corFonteV	<i>VARCHAR</i> (7)
corFonteF	<i>VARCHAR</i> (7)
corFundoV	<i>VARCHAR</i> (7)
corFundoF	<i>VARCHAR</i> (7)

Quadro 66 – Tabela dw017 - Alerta

Tabela dw018 - Filtro	
Coluna	Tipo
id	<i>INT</i> (Primary Key)
id_consulta	<i>INT</i> (Foreign Key)
expressao	<i>VARCHAR</i> (100)

Quadro 67 – Tabela dw018 - Filtro

Tabela dw019 - Mensagem	
Coluna	Tipo
id	<i>INT</i> (Primary Key)
id_dw	<i>INT</i> (Foreign Key)
dataMsg	<i>DATETIME</i>
mensagem	<i>VARCHAR</i> (500)

Quadro 68 – Tabela dw019 – Mensagem

APÊNDICE C – Script SQL da ferramenta

```
CREATE DATABASE DW;

USE DW;

CREATE TABLE MN001
(
    id INTEGER NOT NULL AUTO_INCREMENT,
    nome VARCHAR(100) NOT NULL,
    PRIMARY KEY (id)
)
;

CREATE TABLE MN002
(
    id INTEGER NOT NULL AUTO_INCREMENT,
    id_menu INTEGER NULL,
    nome VARCHAR(100) NOT NULL,
    sqltext VARCHAR(950) NULL,
    objNome VARCHAR(20) NULL,
    filtradw VARCHAR(1) NOT NULL,
    filtraobj VARCHAR(10) NULL,
    PRIMARY KEY (id),
    KEY (id_menu)
)
;

CREATE TABLE DW013
(
    id INTEGER NOT NULL AUTO_INCREMENT,
    id_fonte INTEGER NOT NULL,
    cpoDest VARCHAR(20) NOT NULL,
    id_tabela INTEGER NOT NULL,
    PRIMARY KEY (id),
    KEY (id_fonte)
)
;

CREATE TABLE DW014
(
    id INTEGER NOT NULL AUTO_INCREMENT,
    id_fonte INTEGER NOT NULL,
    tipo VARCHAR(10) NOT NULL,
    ativo BOOLEAN NOT NULL,
    inicio DATETIME NOT NULL,
    fim DATETIME NOT NULL,
    semana VARCHAR(7) NULL,
    mensal VARCHAR(31) NULL,
    PRIMARY KEY (id),
    KEY (id_fonte)
)
;

CREATE TABLE DW006
(
    id INTEGER NOT NULL AUTO_INCREMENT,
    id_cubo INTEGER NOT NULL,
```

```
nome VARCHAR(30) NOT NULL,  
descricao VARCHAR(30) NOT NULL,  
tipo VARCHAR(10) NOT NULL,  
tamanho INTEGER NOT NULL,  
decimais INTEGER NOT NULL,  
PRIMARY KEY (id),  
KEY (id_cubo)  
)  
;  
  
CREATE TABLE DW011  
(  
    id INTEGER NOT NULL AUTO_INCREMENT,  
    id_dim INTEGER NULL,  
    id_cubo INTEGER NULL,  
    id_conexao INTEGER NOT NULL,  
    nome VARCHAR(15) NOT NULL,  
    descricao VARCHAR(50) NOT NULL,  
    sqldados VARCHAR(255) NOT NULL,  
    PRIMARY KEY (id),  
    KEY (id_conexao),  
    KEY (id_dim),  
    KEY (id_cubo)  
)  
;  
  
CREATE TABLE DW007  
(  
    id INTEGER NOT NULL AUTO_INCREMENT,  
    id_cubo INTEGER NOT NULL,  
    id_dim INTEGER NOT NULL,  
    PRIMARY KEY (id),  
    KEY (id_dim),  
    KEY (id_cubo)  
)  
;  
  
CREATE TABLE DW004  
(  
    id INTEGER NOT NULL AUTO_INCREMENT,  
    id_dim INTEGER NOT NULL,  
    chave_seq INTEGER NOT NULL DEFAULT 0,  
    nome VARCHAR(30) NOT NULL,  
    descricao VARCHAR(30) NOT NULL,  
    tipo VARCHAR(10) NOT NULL,  
    tamanho INTEGER NOT NULL,  
    decimais INTEGER NOT NULL,  
    PRIMARY KEY (id),  
    KEY (id_dim)  
)  
;  
  
CREATE TABLE DW009  
(  
    id INTEGER NOT NULL AUTO_INCREMENT,  
    id_usuario INTEGER NOT NULL,  
    id_consulta INTEGER NULL,  
    PRIMARY KEY (id),  
    KEY (id_usuario),  
    KEY (id_consulta)  
)  
;
```

```
;  
CREATE TABLE DW003  
(  
    id INTEGER NOT NULL AUTO_INCREMENT,  
    id_dw INTEGER NOT NULL,  
    nome VARCHAR(15) NOT NULL,  
    descricao VARCHAR(30) NOT NULL,  
    dataCriado DATETIME NOT NULL,  
    PRIMARY KEY (id),  
    KEY (id_dw)  
);  
;  
CREATE TABLE DW005  
(  
    id INTEGER NOT NULL AUTO_INCREMENT,  
    id_dw INTEGER NOT NULL,  
    nome VARCHAR(15) NOT NULL,  
    descricao VARCHAR(30) NOT NULL,  
    dataCriado DATETIME NOT NULL,  
    PRIMARY KEY (id),  
    KEY (id_dw)  
);  
;  
CREATE TABLE DW010  
(  
    id INTEGER NOT NULL AUTO_INCREMENT,  
    id_cubo INTEGER NOT NULL,  
    nome VARCHAR(15) NOT NULL,  
    descricao VARCHAR(30) NOT NULL,  
    PRIMARY KEY (id),  
    KEY (id_cubo)  
);  
;  
CREATE TABLE DW002  
(  
    id INTEGER NOT NULL AUTO_INCREMENT,  
    id_dw INTEGER NOT NULL,  
    nome VARCHAR(12) NOT NULL,  
    descricao VARCHAR(30) NOT NULL,  
    tipo_banco VARCHAR(50) NOT NULL,  
    servidor VARCHAR(30) NOT NULL,  
    banco VARCHAR(20) NOT NULL,  
    usuario VARCHAR(10) NOT NULL,  
    senha VARCHAR(10) NOT NULL,  
    PRIMARY KEY (id),  
    KEY (id_dw)  
);  
;  
CREATE TABLE DW008  
(  
    id INTEGER NOT NULL AUTO_INCREMENT,  
    login VARCHAR(15) NOT NULL,  
    senha VARCHAR(20) NOT NULL,  
    nome VARCHAR(30) NOT NULL,  
    email VARCHAR(40) NULL,  
    tipo VARCHAR(20) NOT NULL,
```

```
        PRIMARY KEY (id)
    )
;

CREATE TABLE DW001
(
    id INTEGER NOT NULL AUTO_INCREMENT,
    nome VARCHAR(15) NOT NULL,
    descricao VARCHAR(50) NOT NULL,
    dataCriado DATETIME NOT NULL,
    icone VARCHAR(20) NOT NULL,
    PRIMARY KEY (id)
)
;

CREATE TABLE DW015
(
    id INTEGER NOT NULL AUTO_INCREMENT,
    id_consulta INTEGER NOT NULL,
    id_x INTEGER NULL,
    id_y INTEGER NULL,
    id_i INTEGER NULL,
    posX INTEGER NULL,
    posY INTEGER NULL,
    posI INTEGER NULL,
    posD INTEGER NULL,
    func VARCHAR(10) NULL,
    PRIMARY KEY (id),
    KEY (id_consulta),
    KEY (id_x),
    KEY (id_y),
    KEY (id_i)
)
;

CREATE TABLE DW016
(
    id INTEGER NOT NULL AUTO_INCREMENT,
    id_consulta INTEGER NOT NULL,
    id_indicador INTEGER NOT NULL,
    tipo VARCHAR(20) NOT NULL,
    quantidade INTEGER NOT NULL,
    PRIMARY KEY (id),
    KEY (id_consulta),
    KEY (id_indicador)
)
;

CREATE TABLE DW017
(
    id INTEGER NOT NULL AUTO_INCREMENT,
    id_consulta INTEGER NOT NULL,
    id_indicador INTEGER NOT NULL,
    operador VARCHAR(15) NOT NULL,
    quantidade FLOAT NULL,
    corFonteV VARCHAR(7) NULL,
    corFonteF VARCHAR(7) NULL,
    corFundoV VARCHAR(7) NULL,
    corFundoF VARCHAR(7) NULL,
    PRIMARY KEY (id),
    KEY (id_consulta),
```

```

        KEY (id_indicador)
    )
;

CREATE TABLE DW018
(
    id INTEGER NOT NULL AUTO_INCREMENT,
    id_consulta INTEGER NOT NULL,
    expressao VARCHAR(100) NOT NULL,
    PRIMARY KEY (id),
    KEY (id_consulta)
)
;

CREATE TABLE DW019
(
    id INTEGER NOT NULL AUTO_INCREMENT,
    id_dw INTEGER NOT NULL,
    dataMsg DATETIME NOT NULL,
    mensagem VARCHAR(500) NULL,
    PRIMARY KEY (id),
    KEY (id_dw)
)
;

ALTER TABLE MN002 ADD CONSTRAINT FK_MN002_MN001 FOREIGN KEY (id_menu)
REFERENCES MN001 (id) ON DELETE CASCADE;

ALTER TABLE DW013 ADD CONSTRAINT FK_DW013_DW011 FOREIGN KEY (id_fonte)
REFERENCES DW011 (id) ON DELETE CASCADE;

ALTER TABLE DW014 ADD CONSTRAINT FK_DW014_DW011 FOREIGN KEY (id_fonte)
REFERENCES DW011 (id) ON DELETE CASCADE;

ALTER TABLE DW006 ADD CONSTRAINT FK_DW006_DW005 FOREIGN KEY (id_cubo)
REFERENCES DW005 (id) ON DELETE CASCADE;

ALTER TABLE DW011 ADD CONSTRAINT FK_DW011_DW002 FOREIGN KEY (id_conexao)
REFERENCES DW002 (id) ON DELETE CASCADE;

ALTER TABLE DW011 ADD CONSTRAINT FK_DW011_DW003 FOREIGN KEY (id_dim)
REFERENCES DW003 (id) ON DELETE CASCADE;

ALTER TABLE DW011 ADD CONSTRAINT FK_DW011_DW005 FOREIGN KEY (id_cubo)
REFERENCES DW005 (id) ON DELETE CASCADE;

ALTER TABLE DW007 ADD CONSTRAINT FK_DW007_DW003 FOREIGN KEY (id_dim)
REFERENCES DW003 (id) ON DELETE CASCADE;

ALTER TABLE DW007 ADD CONSTRAINT FK_DW007_DW005 FOREIGN KEY (id_cubo)
REFERENCES DW005 (id) ON DELETE CASCADE;

ALTER TABLE DW004 ADD CONSTRAINT FK_DW004_DW003 FOREIGN KEY (id_dim)
REFERENCES DW003 (id) ON DELETE CASCADE;

ALTER TABLE DW009 ADD CONSTRAINT FK_DW009_DW008 FOREIGN KEY (id_usuario)
REFERENCES DW008 (id) ON DELETE CASCADE;

ALTER TABLE DW009 ADD CONSTRAINT FK_DW009_DW010 FOREIGN KEY (id_consulta)
REFERENCES DW010 (id) ON DELETE CASCADE;

```

```

ALTER TABLE DW003 ADD CONSTRAINT FK_DW003_DW001 FOREIGN KEY (id_dw)
REFERENCES DW001 (id) ON DELETE CASCADE;

ALTER TABLE DW003 ADD CONSTRAINT FK_DW003_DW0011 FOREIGN KEY (id_dw)
REFERENCES DW001 (id) ON DELETE CASCADE;

ALTER TABLE DW005 ADD CONSTRAINT FK_DW005_DW001 FOREIGN KEY (id_dw)
REFERENCES DW001 (id) ON DELETE CASCADE;

ALTER TABLE DW010 ADD CONSTRAINT FK_DW010_DW005 FOREIGN KEY (id_cubo)
REFERENCES DW005 (id) ON DELETE CASCADE;

ALTER TABLE DW002 ADD CONSTRAINT FK_DW002_DW001 FOREIGN KEY (id_dw)
REFERENCES DW001 (id) ON DELETE CASCADE;

ALTER TABLE DW015 ADD CONSTRAINT FK_DW015_DW010 FOREIGN KEY (id_consulta)
REFERENCES DW010 (id) ON DELETE CASCADE;

ALTER TABLE DW015 ADD CONSTRAINT FK_DW015X_DW004 FOREIGN KEY (id_x)
REFERENCES DW004 (id) ON DELETE CASCADE;

ALTER TABLE DW015 ADD CONSTRAINT FK_DW015Y_DW004 FOREIGN KEY (id_y)
REFERENCES DW004 (id) ON DELETE CASCADE;

ALTER TABLE DW015 ADD CONSTRAINT FK_DW015_DW006 FOREIGN KEY (id_i)
REFERENCES DW006 (id) ON DELETE CASCADE;

ALTER TABLE DW016 ADD CONSTRAINT FK_DW016_DW011 FOREIGN KEY (id_consulta)
REFERENCES DW010 (id) ON DELETE CASCADE;

ALTER TABLE DW016 ADD CONSTRAINT FK_DW016_DW006 FOREIGN KEY (id_indicador)
REFERENCES DW006 (id) ON DELETE CASCADE;

ALTER TABLE DW017 ADD CONSTRAINT FK_DW017_DW011 FOREIGN KEY (id_consulta)
REFERENCES DW010 (id) ON DELETE CASCADE;

ALTER TABLE DW017 ADD CONSTRAINT FK_DW017_DW006 FOREIGN KEY (id_indicador)
REFERENCES DW006 (id) ON DELETE CASCADE;

ALTER TABLE DW018 ADD CONSTRAINT FK_DW018_DW011 FOREIGN KEY (id_consulta)
REFERENCES DW010 (id) ON DELETE CASCADE;

ALTER TABLE DW019 ADD CONSTRAINT FK_DW019_DW001 FOREIGN KEY (id_dw)
REFERENCES DW001 (id) ON DELETE CASCADE;

INSERT INTO DW008 (login, senha, nome, email, tipo)
VALUES ('DWADMIN', 'A', 'Administrador', ' ', 'UsuarioSistema');

INSERT INTO MN001(nome) VALUES ('Principal');
INSERT INTO MN001(nome) VALUES ('Definicao');
INSERT INTO MN001(nome) VALUES ('Ferramentas');
INSERT INTO MN001(nome) VALUES ('Usuarios');

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (1, 'Log', 'SELECT id, Date(dataMsg) Data, Time(dataMsg) Hora,
Mensagem FROM DW019', null, 'S', null);

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (1, 'Agendamento', 'SELECT id, (SELECT Descricao FROM DW011 WHERE id
= id_fonte) Fonte, Tipo, case when ativo = 1 then ''Sim'' else ''Nao'' end
Ativo, time(inicio) ''Hora Execucao'', Semana, Mensal FROM DW014 WHERE

```



```

now() between inicio and fim and time(now()) < time(inicio) and tipo =
'Diario' union SELECT id, (SELECT Descricao FROM DW011 WHERE id =
id_fonte) Fonte, Tipo, case when ativo = 1 then 'Sim' else 'Nao' end
Ativo, time(inicio) 'Hora Execucao', Semana, Mensal FROM DW014 WHERE
now() between inicio and fim and time(now()) < time(inicio) and
substring(semana,dayofweek(now()),1) = 1 and tipo = 'Semanal' union
SELECT id, (SELECT Descricao FROM DW011 WHERE id = id_fonte) Fonte, Tipo,
case when ativo = 1 then 'Sim' else 'Nao' end Ativo, time(inicio)
'Hora Execucao', Semana, Mensal FROM DW014 WHERE now() between inicio and
fim and time(now()) < time(inicio) and
substring(mensal,dayofmonth(now()),1) = 1 and tipo = 'Mensal'', null,
'N', null);

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (1, 'Configuracao', 'SELECT Id, Nome, Descricao, DataCriado, Icone
FROM DW001', 'dw', 'N', null);

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (2, 'Conexao', 'SELECT Id, Nome, Descricao, Tipo_Banco, Servidor,
Banco, Usuario FROM DW002', 'conexao', 'S', null);

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (2, 'Dimensao', 'SELECT Id, Nome, Descricao, DataCriado FROM DW003',
'dimensao', 'S', null);

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (2, 'Cubo', 'SELECT Id, Nome, Descricao, DataCriado FROM DW005',
'cubo', 'S', null);

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (2, 'Consultas', 'SELECT D10.Id, D10.Nome, D10.Descricao, D5.Nome
Cubo FROM DW010 D10 INNER JOIN DW005 D5 ON D5.id = id_cubo', 'consulta',
'S', null);

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (3, 'Exportar&nbsp;Metadados', null, 'exportar', 'N', null);

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (3, 'Importar&nbsp;Metadados', null, 'importar', 'N', null);

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (3, 'Limpeza', null, 'limpeza', 'N', null);

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (4, 'Manutencao', 'SELECT Id, Login, Nome, Email, Tipo FROM DW008',
'usuario', 'N', null);

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (4, 'Previdios', 'SELECT Id, Login, Nome, Email FROM DW008 WHERE
Tipo = 'Usuario'', null, 'N', null);

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (null, 'Fonte&nbsp;de&nbsp;Dados&nbsp;Dimensao', 'SELECT Id, Nome,
Descricao FROM DW011', 'fontedados', 'N', 'id_dim');

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (null, 'Fonte&nbsp;de&nbsp;Dados&nbsp;Cubo', 'SELECT Id, Nome,
Descricao FROM DW011', 'fontedados', 'N', 'id_cubo');

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (null, 'Atributos&nbsp;da&nbsp;Dimensao', 'SELECT Id, CASE WHEN

```

```

Chave_Seq = 0 THEN NULL ELSE Chave_Seq END Chave, Nome, Descricao, Tipo,
Tamanho, Decimais FROM DW004', 'atributo', 'N', 'id_dim');

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (null, 'Indicadores&nbsp;do&nbsp;Cubo', 'SELECT Id, Nome, Descricao,
Tipo, Tamanho, Decimais FROM DW006', 'indicador', 'N', 'id_cubo');

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (null, 'Chave&nbsp;Primaria', null, 'chaveprimaria', 'N', 'id_dim');

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (null, 'Dimensao&nbsp;Cubo', null, 'dimensaocubo', 'N', 'id_cubo');

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (null, 'Roteiro', null, 'roteiro', 'N', null);

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (null, 'Importar', null, 'importar', 'N', null);

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (null, 'Agendamento', null, 'agendamento', 'N', null);

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (null, 'Tabela', null, 'tabela', 'N', null);

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (null, 'Previdios', null, 'previdio', 'N', null);

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (null, 'Ranking', null, 'ranking', 'N', null);

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (null, 'Alerta', null, 'alerta', 'N', null);

INSERT INTO MN002(id_menu, nome, sqltext, objNome, filtradw, filtraobj)
VALUES (null, 'Filtro', null, 'filtro', 'N', null);

```

Quadro 69 – Script SQL da ferramenta