

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

**DESENVOLVIMENTO DA FERRAMENTA DE ENSINO EM
DATA WAREHOUSE APLICADO NA DISCIPLINA DE
TECNOLOGIA DA INFORMAÇÃO NA GESTÃO DE
NEGÓCIOS**

JOANA SIMON

BLUMENAU
2010

2010/1-14

JOANA SIMON

**DESENVOLVIMENTO DA FERRAMENTA DE ENSINO EM
DATA WAREHOUSE APLICADO NA DISCIPLINA DE
TECNOLOGIA DA INFORMAÇÃO NA GESTÃO DE
NEGÓCIOS**

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

Prof. Oscar Dalfovo, Doutor - Orientador

**BLUMENAU
2010**

2010/1-14

**DESENVOLVIMENTO DA FERRAMENTA DE ENSINO EM
DATA WAREHOUSE APLICADO NA DISCIPLINA DE
TECNOLOGIA DA INFORMAÇÃO NA GESTÃO DE
NEGÓCIOS**

Por

JOANA SIMON

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

Presidente: _____
Prof. Oscar Dalfovo, Doutor – Orientador, FURB

Membro: _____
Prof. Cláudio Ratke, Mestre – FURB

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

Blumenau, 07 de julho de 2010.

Dedico este trabalho a todos os amigos, especialmente aqueles que me ajudaram diretamente na realização deste. Aos meus pais e ao meu irmão, que sempre me incentivaram e apoiaram nos estudos.

AGRADECIMENTOS

À minha família, pela compreensão e tolerância durante a minha ausência neste período.

Ao meu irmão, que mesmo distante se fez presente nesta etapa.

Ao meu namorado William Wachholz, pela paciência, apoio e compreensão durante a elaboração deste trabalho.

A Cristine Volpi que compartilhou comigo cada momento desses anos de curso.

Ao meu orientador, Oscar Dalfovo, pela confiança e paciência.

Agradeço a todos que, de forma direta ou indireta, contribuíram para a realização deste trabalho.

A única coisa de que devemos ter medo é do próprio medo - indefinível, irracional, um terror injustificado que paralisa os esforços para passar do retrocesso ao progresso.

Franklin D. Roosevelt

RESUMO

Com a informatização das empresas surge, a necessidade crescente de ferramentas de análise e interpretação de dados dos sistemas de informação. Este trabalho apresenta o desenvolvimento de uma ferramenta de ensino em que foi aplicada a técnica de cubo de decisão, com o intuito de promover o contato dos acadêmicos da disciplina de Tecnologia da Informação na Gestão de Negócios (TIGN) do curso de Bacharelado em Sistemas de Informação da Universidade Regional de Blumenau, com tal tecnologia. A mesma possibilita a unificação de dados, o povoamento de tabelas e a visualização dos dados através do cubo de decisão, de forma gráfica ou textual, facilitando a utilização das informações armazenadas. A metodologia utilizada baseou o seu desenvolvimento em ambiente *web*, através da linguagem de programação C Sharp (C#) e das tecnologias Asp.Net, SQLServer e Dundas OLAP Services. Como resultado, após a teoria ser apresentada em sala de aula, a partir do uso da ferramenta de ensino em *Data Warehouse*, os acadêmicos de TIGN demonstraram maior facilidade na assimilação do conteúdo. A ferramenta proporcionou também que os alunos notassem a importância de haver instrumentos de auxílio na interpretação de dados para a gestão de uma empresa.

Palavras-chave: Cubo de decisão. *Data Warehouse*. Ensino. Tecnologia da Informação.

ABSTRACT

With the computerization of business comes the increasing need for tools of analysis and interpretation of data from information systems. This paper presents the development of a teaching tool where technique was applied to the decision cube in order to promote contact between the academic discipline of Information Technology in Business Management of the course of Bachelor of Information Systems at the Regional University of Blumenau, with this technology. It enables the unification of data, table stand and display of data by using the decision cube, textual or graphical form, facilitating the use of stored information. The methodology was based on development in a web environment using the programming language C Sharp (C#) and technologies Asp.Net, SqlServer and Dundas OLAP Services. As a result, after the theory was presented in the classroom, by the use of teaching tool in Data Warehouse, the students demonstrated greater ease in the assimilation of the content. The tool also provided that the students would notice the importance of having tools to help in interpreting data for the management of a company.

Key-words: Decision Cube. Data Warehouse. Information Technology. Teaching.

LISTA DE ILUSTRAÇÕES

Figura 1 – Modelo dimensional esquema estrela	22
Quadro 1 – Requisitos funcionais	29
Quadro 2 – Requisitos não funcionais	30
Figura 2 – Diagrama de acesso a ferramenta.....	31
Figura 3 – Diagrama de caso de uso de visualização dos dados	32
Figura 4 – Diagrama de caso de uso de consulta.....	33
Figura 5 – Diagrama de atividades da disciplina.....	34
Figura 6 – Diagrama de atividades da unificação de dados	35
Figura 7 – Diagrama de atividades do povoamento das tabelas	36
Figura 8 – Diagrama de Entidade e Relacionamento.....	37
Figura 9 – Diagrama de classes	38
Figura 10 – Diagrama de classes	39
Figura 11 – Arquitetura do Dundas Olap Services	42
Figura 12 – Tela de autenticação de usuário	43
Figura 13 – Tela de opções de acesso	44
Figura 14 – Tela para manter o cadastro de salas	44
Figura 15 – Tela para realização do cadastro de sala.....	45
Figura 16 – Tela para manter o cadastro de usuários.....	45
Figura 17 – Tela para realizar o cadastro de usuário	46
Figura 18 – Tela de cadastro de usuário.....	46
Figura 19 – Tela de opções de acesso	47
Figura 20 – Controle de acesso a partes específicas da ferramenta.....	47
Figura 21 – Tela com opções da ferramenta	48
Figura 22 – Tela de consulta ao material teórico.....	48
Figura 23 – Tela para manter dados do embasamento teórico	49
Figura 24 – Tela para o cadastro de um novo assunto	49
Quadro 3 – Componente para montar a árvore de embasamento teórico	50
Figura 25 – Código para carregar dados na árvore.....	51
Quadro 4 – Código para carregar nós filhos na árvore	52
Figura 26 – Tela com as empresas cadastradas	52
Figura 27 – Tela para associar-se a uma empresa	53

Figura 28 – Tela com figura ilustrando um DW e menu	54
Figura 29 – Tela com visualização da explicação cadastrada para o dado da imagem	54
Figura 30 – Tela do cadastro de explicação da imagem	55
Figura 31 – Tela para informar o arquivo que será unificado	56
Figura 32 – Tela com os valores que serão unificados	56
Figura 33 – Tela para informar o valor que será unificado	57
Figura 34 – Tela com o <i>log</i> da unificação.....	57
Quadro 5 – Componente utilizado para montar a tabela com os valores que serão unificados	58
Figura 35 – Definindo método para retornar dados na tabela	59
Quadro 6 – Código fonte utilizado para carregar os valores que serão unificados	59
Quadro 7 – Código fonte para unificar os registros.....	60
Figura 36 – Tela exibindo os arquivos unificados disponíveis na base	61
Figura 37 – Tela exibindo o <i>log</i> após povoar a tabela com os registros	61
Quadro 8 – Código fonte para povoar as tabelas.....	64
Quadro 9 – Exemplo de arquivo para povoar as tabelas.....	65
Quadro 10 – Código fonte para processar o cubo.....	65
Figura 38 – Cubo de decisão	66
Figura 39 – Barra de ferramentas	66
Figura 40 – Controle de relatórios	67
Figura 41 – Seletor de cubo.....	67
Figura 42 – Medidas e dimensões.....	67
Figura 43 – <i>Slicer</i> e abas com gráfico e tabela.....	67
Figura 44 – Controle do cabeçalho de linhas e colunas.....	68
Quadro 11 – Componentes utilizados para implementar o cubo.....	68
Quadro 12 – Comparativo das ferramentas.....	70
Figura 45 – Gráfico de avaliação da hierarquia das informações.....	70
Figura 46 – Gráfico de avaliação da facilidade em encontrar as funcionalidades disponíveis	71
Figura 47 – Gráfico de avaliação de avisos ao tentar enviar formulário sem preencher um ou mais campo(s) obrigatório(s).....	71
Figura 48 – Gráfico de avaliação referente à ao avisar existência de campos obrigatórios em branco, manter os valores preenchidos corretamente	72
Figura 49 – Gráfico de avaliação da facilidade em consultar os dados	72
Figura 50 – Gráfico de avaliação da facilidade em consultar os dados	73

Figura 51 – Gráfico avaliando se a ferramenta cumpre de forma objetiva a tarefa de complementar a explicação teórica sobre DW apresentada em sala	73
Figura 52 – Gráfico de avaliação do nível de poluição visual do ambiente.....	74
Quadro 13 – Descrição do caso de uso Logar na ferramenta.....	79
Quadro 14 – Descrição do caso de uso Cadastrar usuários.....	79
Quadro 15 – Descrição do caso de uso Cadastrar usuários.....	80
Quadro 16 – Descrição do caso de uso Consultar <i>Data Warehouse</i>	81
Quadro 17 – Descrição do caso de uso Consultar <i>Data marts</i>	81
Quadro 18 – Descrição do caso de uso Consultar tabela de fato.....	82
Quadro 19 – Descrição do caso de uso Cadastrar dimensões	82
Quadro 20 – Descrição do caso de uso Unificar arquivos txt ou XML.....	83
Quadro 21 – Descrição do caso de uso Efetuar carga de Dados	83
Quadro 22 – Tabela de arquivos unificados	84
Quadro 23 – Tabela de Sala	84
Quadro 24 – Tabela de Usuario	85
Quadro 25 – Tabela de Empresa.....	85
Quadro 26 – Tabela de Regiao	85
Quadro 27 – Tabela de Cliente	86
Quadro 28 – Tabela de Fornecedor.....	86
Quadro 29 – Tabela de Tempo	86
Quadro 30 – Tabela de Produto.....	87
Quadro 31 – Tabela de Conceitos.....	87
Quadro 32 – Tabela de ExplicacaoFigurasDW	87
Quadro 33 – Tabela de Unificacao	88
Figura 53 – Questionário de avaliação da ferramenta	89

LISTA DE SIGLAS

BD – Banco de Dados

BI – Inteligência nos Negócios

BSI - Bacharelado em Sistemas de Informação

DM – *Data Mart*

DW – *Data Warehouse*

EA - *Enterprise Architect*

FURB - Universidade Regional de Blumenau

OLAP – *Online Analytical Processing*

SQL - *Structured Query Language*

TIGN - Tecnologia da Informação na Gestão de Negócios

TXT - Texto

UML – Linguagem de Modelagem Unificada

XML - *eXtensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO	14
1.1 OBJETIVOS DO TRABALHO.....	15
1.2 ESTRUTURA DO TRABALHO	15
2 FUNDAMENTAÇÃO TEÓRICA.....	17
2.1 BUSINESS INTELLIGENCE.....	17
2.2 DATA WAREHOUSE.....	18
2.2.1 As etapas para construir um <i>Data Warehouse Dimensional</i>	20
2.3 CUBO DE DECISÃO	21
2.3.1 Modelo Dimensional.....	21
2.4 ONLINE ANALYTICAL PROCESSING (OLAP)	24
2.5 TECNOLOGIA DA INFORMAÇÃO NA GESTÃO DE NEGÓCIOS	25
2.6 TRABALHOS CORRELATOS	26
3 DESENVOLVIMENTO DA FERRAMENTA.....	28
3.1 PRINCIPAIS REQUISITOS DE SOFTWARE	28
3.2 ESPECIFICAÇÃO.....	30
3.2.1 Diagrama de caso de uso.....	30
3.2.2 Diagrama de atividades	33
3.2.3 Modelo de Entidade e Relacionamento (MER).....	37
3.2.4 Diagrama de Classes	38
3.3 IMPLEMENTAÇÃO	40
3.3.1 Técnicas e ferramentas utilizadas	40
3.3.1.1 SQL Server Analysis Services	40
3.3.1.2 Dundas OLAP Services	41
3.3.2 Operacionalidade da implementação	42
3.3.2.1 Acesso a ferramenta.....	43
3.3.2.2 Embasamento teórico	47
3.3.2.3 Divisão em empresas	52
3.3.2.4 – Imagem mostrando um <i>Data Warehouse</i>	53
3.3.2.5 Unificar dados	55
3.3.2.6 Povoar tabelas	60
3.3.2.7 Visualizar os dados.....	66

3.4 RESULTADOS E DISCUSSÃO.....	69
4 CONCLUSÕES	75
4.1 EXTENSÕES	76
REFERÊNCIAS BIBLIOGRÁFICAS	77
APÊNDICE A – Detalhamento dos casos de uso	79
APÊNDICE B – Dicionário de dados do Modelo de Entidade e Relacionamento	84
APÊNDICE C – Questionário aplicado em sala	89

1 INTRODUÇÃO

Com o uso cada vez mais comum de sistemas de informação nas organizações, as empresas encontram a necessidade de utilizar ferramentas que facilitem a interpretação e o uso dos dados obtidos. Conforme Barbieri (2001, p. 5), primeiramente foram obtidos os dados, depois foram transformados em informação, sendo que o objetivo atual é gerar conhecimentos a partir desses dados e informações.

Turban et al. (2009, p. 27) afirma que as empresas são forçadas a captar, compreender e explorar seus dados para dar suporte à tomada de decisões, a fim de melhorar as operações de negócios.

De acordo com Barbieri (2001, p. 5), o *Business Intelligence* (BI) – expressão que, conforme tradução livre, significa Inteligência nos Negócios – permite estruturar, acessar e explorar informações, com o objetivo de desenvolver percepções, entendimentos e conhecimentos, possibilitando o uso de todo o potencial da informação armazenada, contribuindo assim para um melhor processo de tomada de decisão.

Com essa visão é apresentado na disciplina de Tecnologia da Informação na Gestão de Negócios (TIGN) do curso de Bacharelado em Sistemas de Informação da Universidade Regional de Blumenau (FURB), a ementa contendo no seu conteúdo programático, a Tecnologia da Informação, o *Enterprise Resource Planning* (ERP) e o BI.

Percebeu-se, durante as aulas da disciplina de TIGN, uma grande necessidade de abordar também a parte prática do BI. Atualmente, todo o conteúdo é transmitido por meio de trabalhos teóricos sobre o assunto e de apresentações utilizando a ferramenta PowerPoint, o que torna sua abordagem muito superficial e dificulta o aprendizado.

Esse fato, juntamente com a crescente procura no mercado por esse tipo de tecnologia, mostra que é preciso dedicar mais tempo a esse assunto na disciplina e fornecer material prático para os acadêmicos.

Como solução, foi elaborada uma das partes de um BI, utilizando-se da técnica de cubo de decisão, possibilitando assim o contato do acadêmico com essa nova tecnologia, permitindo através de um estudo de caso simular as necessidades na área de vendas de uma organização têxtil voltada à confecção de peças de vestuário. Foi disponibilizada uma ferramenta de auxílio à tomada de decisão para possibilitar aos acadêmicos, com visão de gestor, experiência na utilização desse tipo de ferramenta. Seu uso proporciona ao aluno uma noção de gestão de negócios.

A implementação foi realizada em ambiente *web*, facilitando assim o acesso do acadêmico e permitindo a ele realizar os exercícios fora da Universidade. A utilização desse trabalho na disciplina de TIGN permite acrescentar conhecimento prático aos alunos no que diz respeito ao BI tratado nesta disciplina. Esse trabalho possibilitará a outros acadêmicos realizarem a sua continuidade, a fim de juntamente a outros Trabalhos de Conclusão Curso obter-se um BI completo.

1.1 OBJETIVOS DO TRABALHO

O objetivo geral deste trabalho foi o desenvolvimento de uma ferramenta de cubo de decisão de um BI aplicado na disciplina Tecnologia da Informação na Gestão de Negócios da FURB, como material de apoio didático-pedagógico.

Os objetivos específicos do trabalho são:

- a) auxiliar o acadêmico a identificar as dimensões do cubo de decisão;
- b) permitir a importação de dados para o cubo de decisão;
- c) possibilitar ao gestor melhor utilização das informações já armazenadas para apresentar a ferramenta como material didático-pedagógico;
- d) aplicar o cubo de decisão na disciplina de Tecnologia da Informação na Gestão de Negócios do curso de Sistemas de Informação ofertada pela FURB.

1.2 ESTRUTURA DO TRABALHO

Esse trabalho é composto por quatro capítulos. No primeiro capítulo, é apresentada a origem do trabalho, justificativas, objetivos atingidos e a estrutura do trabalho.

No segundo capítulo, tem-se a fundamentação teórica sobre *Business Intelligence*, *Data Warehouse*, cubo de decisão, *Online Analytical Processing*, a disciplina de Tecnologia da Informação na Gestão de Negócios e os trabalhos correlatos.

O terceiro capítulo aborda a construção da ferramenta, mostrando seus principais requisitos, as especificações, sua implementação, as tecnologias utilizadas para o seu desenvolvimento, como utilizar a ferramenta e os resultados obtidos com este trabalho.

O quarto capítulo apresenta as conclusões a partir do desenvolvimento da ferramenta e sugestões para sua continuação em trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Esse capítulo apresenta a fundamentação teórica necessária para o entendimento da ferramenta. São abordados os conceitos referentes à *Business Intelligence* (BI) mostrando também sua origem e importância nas organizações. Na sequência é apresentado o conceito de *Data Warehouse* (DW), o embasamento teórico sobre cubo de decisão, abordando-se o modelo dimensional e o esquema estrela utilizado para a construção do cubo. Em seguida, descreve-se o conteúdo referente ao *Online Analytical Processing* (OLAP). Também são apresentados a ementa da disciplina de Tecnologia da Informação na Gestão de Negócios (TIGN) e trabalhos correlatos.

2.1 BUSINESS INTELLIGENCE

Conforme Turban et al. (2009, p. 27), o termo BI foi utilizado pelo Gartner Group na década de 1990. Mas o conceito iniciou nos anos 1970 com os sistemas de geração de relatórios, sistemas estáticos, bidimensionais e sem recursos de análise. O conceito de sistemas de informações executivas (EIS) surgiu no início da década de 80, expandindo aos gerentes e executivos de nível superior o uso de sistemas computadorizados. Esses conceitos possibilitaram a criação de sistemas de geração de relatórios dinâmicos multidimensionais, prognósticos e previsões, análise de tendências, detalhamento, acesso a status e fatores críticos de sucesso. Tais recursos apareceram em vários produtos comerciais até meados da década de 1990. Depois, estes e outros recursos novos apareceram com o nome de BI. Hoje se sabe que todas as informações necessárias aos executivos podem estar em um bom sistema de informações empresariais baseado em BI.

Turban et al. (2009, p. 21), recorda que atualmente as empresas operam em um ambiente de negócios que está se tornando cada vez mais complexo e mutante. As organizações estão, cada vez mais, submetidas a pressões, o que exige respostas rápidas a situações que estão em constante mudança, além de terem que ser inovadoras. Essas atividades exigem agilidade, tomadas de decisões com velocidade e frequência. Para que isso ocorra, é importante que as informações sejam rapidamente processadas de forma a auxiliar nas decisões necessárias, o que geralmente necessita de apoio computadorizado. Essas

necessidades não são supridas pelos sistemas antigos, esses sistemas não proporcionam rapidez na tomada de decisões e avaliação de quantidades grandes de informação que podem ser armazenadas em locais diferentes e colaborar. O BI é uma alternativa para esse tipo de problema.

Turban et al. (2009, p. 27), define que “o processo do BI baseia-se na transformação de dados em informações, depois em decisões e finalmente em ações”.

Barbieri (2001, p. 34) afirma que, de forma mais ampla, o conceito de BI pode ser entendido como a utilização de diferentes fontes de informação para se definir estratégias de competitividade nos negócios da empresa.

Conforme Barbieri (2001), as técnicas de BI definem regras e técnicas para a formatação adequada das grandes quantidades de dados, visando armazenar informações de forma estruturada, independentemente da sua origem. O objetivo das premissas de BI é a definição de estruturas modeladas dimensionalmente, armazenadas em *Data Warehouse* ou *Marts*, e interpretadas por ferramentas *Online Analytical Processing* (OLAP) ou ferramentas de *Data Mining* que auxiliam na percepção de padrões de comportamento escondidos no banco.

2.2 DATA WAREHOUSE

De acordo com Barbieri (2001, p. 49), o *Data Warehouse* (DW) é um depósito de dados atuais e históricos não volátil. É de possível interesse aos gerentes de toda a organização por fornecer dados a sistemas de apoio à decisão. Os dados são armazenados em estruturas lógicas dimensionais, possibilitando o seu processamento analítico por ferramentas como OLAP ou *Data Mining*. Ferramentas OLAP fornecem recursos para a modelagem, análise e visualização de grandes conjuntos de dados, além de sistemas de gerenciamento de banco de dados, ou ainda para sistemas de DW. Já as de *Data Mining* permitem extrair e identificar informações úteis e conhecimento dos bancos de dados através de técnicas de estatística, matemática e inteligência artificial.

O DW é um local onde são armazenadas informações confiáveis. Consiste em dados e um conjunto de ferramentas para consultar, analisar e apresentar esses dados como informações. Quanto aos dados do DW, diferentemente dos bancos de dados transacionais, onde são processadas pequenas partes do dado através de várias transações por dia, os dados

de um DW são freqüentemente obtidos com uma única transação. Transação esta que pode conter centenas ou até mesmo milhares de registros, e recebe o nome de carga de dados (KIMBALL, 1998).

Barbieri (2001) define que o objetivo do DW é armazenar os dados em vários graus de relacionamento e sumarização, de forma a facilitar e agilizar os processos de tomada de decisão por diferentes níveis gerenciais.

Kimball (1998) conceitua que há diferentes níveis de detalhe no DW. O nível de detalhe mais antigo, o nível de detalhe corrente, um nível de dados levemente resumidos e um nível de dados altamente resumidos. Os dados fluem do ambiente operacional para o nível do DW e geralmente ocorre uma quantidade significativa de transformações sobre os dados.

O DW pode ser dividido em subconjuntos, que normalmente consistem em uma única área temática. Esses subconjuntos são os *Data Marts* (DM). Os DM seriam o nível de dados levemente resumidos.

Inmon, Terdeman e Imhoff (2001) fazem uma analogia com o brinquedo Lego, descrevendo o DW como uma grande pilha de Legos não estruturados, enquanto os *Data Marts* contêm os diferentes brinquedos que são moldados a partir desses Legos.

Entre as principais características do DW, conforme Inmon (2005 apud TURBAN et al., 2009, p. 57), tem-se:

- a) orientação por assunto: proporciona visão mais abrangente da organização. Os dados são organizados por assuntos, como vendas, produtos ou clientes, e contêm apenas as informações significantes para o suporte à tomada de decisão;
- b) integrado: deve-se colocar os dados de diferentes fontes em um formato consistente. Espera-se que um DW seja totalmente integrado;
- c) variável no tempo: o tempo é uma dimensão importante à qual todo DW deve oferecer suporte. Um DW mantém dados históricos que detectam tendências, variações, relações de longo prazo para previsão e comparações, o que leva à tomada de decisões. Os dados de análise vindos de diversas fontes contêm diversos pontos de tempo (por exemplo, visualizações diárias, semanais, mensais);
- d) não-volátil: feita a carga de dados no DW, os usuários não podem alterar ou atualizá-los. Os dados obsoletos são descartados e as alterações são registradas como novos dados, o que permite ajustar o DW quase apenas para acesso a dados.

As características adicionais apresentadas por Turban et al.(2009):

- a) baseado na *web*;
- b) relacional/multidimensional: utiliza estrutura ou relacional ou multidimensional;

- c) cliente/servidor: proporciona aos usuários acesso fácil;
- d) em tempo real;
- e) inclui metadados (dados sobre dados): mostram a localização dos dados em um DW e como utilizá-los de forma eficiente.

2.2.1 As etapas para construir um *Data Warehouse* Dimensional

Kimball (1998, p. 161) afirma que para construir um DW é preciso combinar as necessidades de informações de um grupo de usuários com os dados que realmente estão disponíveis. O projeto é fundamentado em pontos básicos de decisão de acordo com as necessidades dos usuários e os dados disponíveis.

Kimball (1996) apresenta esses pontos:

- a) escolher o processo: escolha do assunto ao qual o *Data Mart* se refere. O primeiro *Data Mart* deve ser o mais acessível para a extração de dados e responder as principais questões de negócios;
- b) determinar o nível de granularidade: definir o que um registro da tabela de fatos representa. Pode-se referir à dimensão temporal (diário, semanal, mensal), ao nível de documentos que definem transações (Ordem de Compra, Nota Fiscal), partindo sempre de modelos com a maior granularidade possível, pois dela pode-se obter todos os outros níveis de granularidade superiores. Quanto maior o grau de granularidade maior é o detalhamento;
- c) identificar e conformizar as dimensões de cada tabela de fatos: facilita o entendimento e utilização do *Data Mart*;
- d) definir os fatos a serem trabalhados, incluindo fatos pré-calculados: se estiverem de acordo com a granularidade da tabela de fato, podem ser adicionados a mesma à qualquer momento. Os fatos devem ser o mais aditivos possível;
- e) preencher tabelas de dimensão: os atributos de texto são utilizados nas interfaces do sistema e também em relatórios, sendo assim deve-se utilizar palavras reais;
- f) carga de dados: importação de dados vindos de sistemas legados;
- g) decisões de armazenamento físico;

- h) escolher a amplitude de tempo do histórico do BD;
- i) preparar as dimensões para suportar mudanças.

2.3 CUBO DE DECISÃO

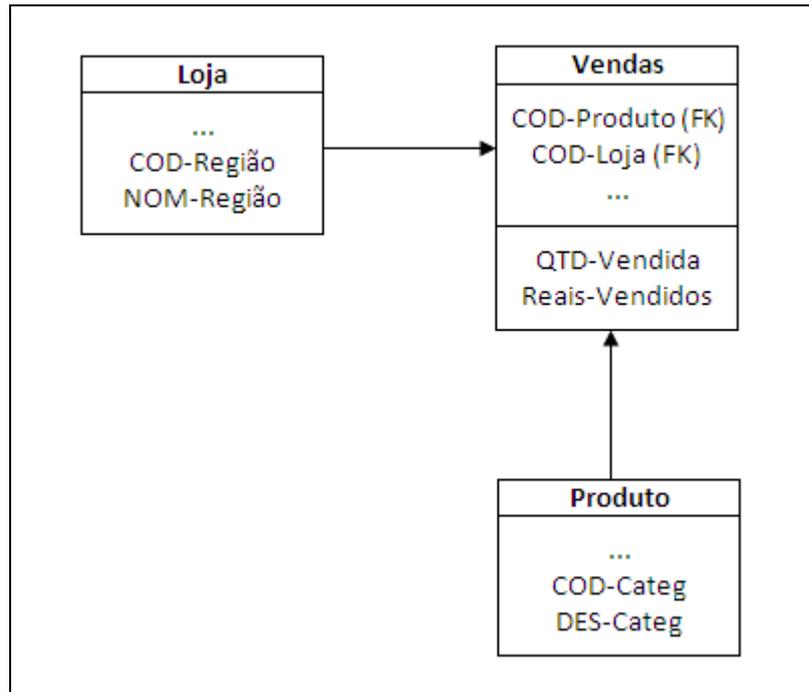
O cubo de dados representa dados em conjunto com alguma medida que se deseja analisar. Cada dimensão do cubo representa um atributo no BD, enquanto as células no cubo representam medidas que se pretende analisar. Ou seja, cada ponto interno contém as medições de acordo com as arestas. A análise do cubo através de ferramentas OLAP permite a visualização de diferentes relatórios, possibilita ao usuário obter visões parciais do cubo e analisá-lo de forma mais intuitiva como por exemplo, clicar e arrastar medidas e atributos para visualizar as informações (TURBAN et al., 2009).

2.3.1 Modelo Dimensional

Kimball (1998, p. 10), afirma que o modelo dimensional é uma estrutura de “cubo de dados”. Ao contrário do modelo entidade/relacionamento, o modelo dimensional é muito assimétrico. É formado por uma tabela principal no centro do diagrama com várias junções conectando-a as outras tabelas. Cada uma das outras tabelas possui apenas uma junção com a tabela central. Essa tabela central é denominada tabela de fatos e as demais tabelas são as tabelas de dimensão.

Turban et al. (2009, p. 79) apresenta que uma das formas para realizar a modelagem dimensional é o esquema estrela, este contém uma tabela de fatos central cercada por tabelas de dimensão.

Na figura 1, observa-se um exemplo de modelagem dimensional utilizando-se o esquema estrela, onde a tabela de fatos é a tabela de Vendas. Esta possui as medidas dos fatos (os valores numéricos) e as chaves-primárias das dimensões que com esta tabela se relacionam.



Fonte: Barbieri (2001).

Figura 1 – Modelo dimensional esquema estrela

De acordo com Barbieri (2001, p. 86), esse esquema recomenda a não normalização das tabelas de Dimensão, essa abordagem é indicada devido ao ganho de performance. A redundância conseqüente de sua utilização é compensada pela redução de comandos de junção, que seriam precisos para recompor a informação buscando-a em outra tabela. Como o maior volume de dados geralmente está nas tabelas de fatos, o que seria poupado na abordagem de Flocos de Neve através da normalização das tabelas não compensa a execução de comandos de junção necessária nessa abordagem.

Quanto a tabela de fatos, Turban et al. (2009, p. 79) afirma que “a tabela de fatos trata principalmente de *o que o data warehouse* suporta na análise de decisão”.

As tabelas Fato servem para armazenar medidas numéricas associadas a eventos de negócio. Uma tabela Fato contém vários fatos, correspondentes a cada uma de suas linhas. Cada fato pode armazenar uma ou mais medidas numéricas, que constituem os valores objetos de análise dimensional. Possuem como chave primária normalmente um campo multi-key, formado pelas chaves-primárias das dimensões que com ela se relacionam. (BARBIERI, 2001, p. 81).

Segundo Kimball (1998, p.11), “os fatos melhores e mais úteis são numéricos, continuamente valorados (diferentes a cada medida) e aditivos (podem ser adicionados às diversas dimensões)”. A adicionabilidade dessa tabela é muito importante e deve ser preservada.

Quanto às tabelas de dimensão, Turban et al. (2009, p. 80) afirma que armazenam informações de classificação e agregação sobre as linhas de fatos centrais. Essas tabelas

possuem os atributos que descrevem os dados contidos na tabela de fatos e caracterizam *como* os dados serão analisados.

Conforme Barbieri (2001, p. 81) as tabelas Dimensão representam entidades de negócios e compõem as estruturas de entrada para armazenar informações.

Essas tabelas têm relação 1:N com a tabela Fato. Apresentam sempre uma chave primária, que lhes confere unicidade, chave essa que participa da tabela Fato, como parte da sua chave múltipla. Devem ser entendida como as tabelas que realizam os filtros de valores aplicados na manipulação dos fatos e por onde as consultas entram no ambiente do DW/DM. (BARBIERI, 2001, p. 81).

Kimball (1998, p. 12), conceitua que as tabelas dimensionais armazenam as descrições textuais das dimensões do negócio. Cada descrição ajuda a definir um componente da respectiva dimensão. Por descreverem os itens de uma dimensão os atributos são mais úteis quando em forma de texto. Um exemplo de exceção a essa “regra” é o tamanho de determinado produto expresso em números, o qual é ainda considerado um atributo de dimensão útil porque se comporta mais como uma descrição textual do que como uma medição numérica.

Para diferenciar se determinado campo de dados numérico é um fato ou um atributo, Kimball (1998, p. 13) sugere perguntar “se o campo de dados numérico é uma medição que varia continuamente a cada amostragem (o que o torna um fato) ou se é uma descrição praticamente constante de um item (o que o torna um atributo de dimensão)”.

Segundo Kimball (1998, p. 13), uma das principais funções dos atributos de tabelas dimensionais é sua utilização como fonte para restrições em uma consulta ou como cabeçalhos de linha no conjunto de resposta do usuário.

O volume de dados nessas tabelas é influenciado pela granularidade definida para o DW. Turban et al. (2009) afirma que o grão define o nível mais alto de detalhe suportado pelo DW, indica se o DW inclui dados detalhados sobre as transações ou se é altamente resumido. Se o grão for muito alto, pode não admitir solicitações para examinar mais além de um valor resumido para investigar cada uma das transações detalhadas que compõem o resumo. Um baixo nível de granularidade resultará em mais dados armazenados no DW.

Para analisar os dados obtidos, uma das técnicas bastante utilizada é a de “fatiar” o cubo de decisão (*slicing and dicing*), onde *slicing* é o plano e *dicing* é o cubo, permitindo alterar a perspectiva em que se tem a visão (BARBIERI, 2001).

2.4 ONLINE ANALYTICAL PROCESSING (OLAP)

De acordo com Inmon, Welch e Glassey (1999, p. 175), o processamento analítico on-line (OLAP) visa possibilitar aos analistas, gerentes e executivos diferentes formas de visualização de informações obtidas de dados puros transformados para facilitar a sua interpretação. Essa transformação de dados em informação é feita através do OLAP.

Para Inmon, Welch e Glassey (1999, p.175), conforme a quantidade de dados armazenados no DW aumenta, faz-se necessário o uso de técnicas cada vez mais elaboradas para facilitar a utilização desses dados.

Inmon, Welch e Glassey (1999, p. 176) afirmam que o OLAP facilita a análise multidimensional, auxiliando o usuário na síntese de informações organizacionais através da visualização comparativa, personalizada e ainda por análise de dados históricos e projetados.

Codd et.al (1993 apud TURBAN et al., 2009, p. 112) definiram as características das ferramentas OLAP em 12 regras que estabelecem um padrão para a modelagem multidimensional de dados:

- a) visão conceitual multidimensional para formular consultas;
- b) transparência ao usuário;
- c) fácil acessibilidade;
- d) desempenho consistente na geração de relatórios;
- e) arquitetura cliente/servidor, possibilitando assim o uso de recursos distribuídos;
- f) dimensionalidade genérica;
- g) manipulação dinâmica de matriz com elementos nulos;
- h) suporte multiusuário;
- i) operações irrestritas com dimensões cruzadas;
- j) manipulação intuitiva dos dados;
- k) relatórios flexíveis;
- l) dimensões e níveis de agregações ilimitados.

Busca-se, assim, atender a quatro diferentes tipos de processamentos utilizados pelos analistas. Um deles é a análise categórica, baseado em dados históricos, parte da premissa de que o desempenho passado é um indicador do futuro. Outro tipo seria a análise exegética, que tem como base também dados históricos, acrescentando a capacidade de análise *drill down*, permitindo assim a consulta aos dados detalhados que determinaram um valor derivado. Um terceiro tipo é a análise contemplativa, que permite ao usuário alterar um único valor para que

se possa determinar o seu impacto. Por fim, a análise formalista possibilita alterações a múltiplas variáveis (TURBAN et al., 2009).

2.5 TECNOLOGIA DA INFORMAÇÃO NA GESTÃO DE NEGÓCIOS

De acordo com Dalfovo (2010), a disciplina Tecnologia da Informação na Gestão de Negócios (TIGN) apresenta, entre seus tópicos da ementa do plano de ensino, o conteúdo programático referente à BI. Mas esse conteúdo é abordado sem a apresentação de uma ferramenta, o que dificulta a visão de como essa técnica pode ser utilizada de forma prática.

O plano de ensino da disciplina apresenta os seguintes itens, conforme Dalfovo (2010):

- a) fundamentos e classificação de sistemas de informação: conceitos e aspectos fundamentais sobre Sistemas de Informação. Uma visão geral sobre a gestão e implantação dos modelos em sistemas de informação das organizações;
- b) custo, valor e qualidade da informação: permite que o acadêmico identifique o custo e valor da informação para a organização;
- c) vantagem competitiva e informação: baseado no conhecimento da informação possibilita identificar quais as vantagens competitivas para organização;
- d) evolução dos sistemas de informação: abordagem dos sistemas de informação existentes no mercado. Sistemas de informações gerenciais, sistemas de apoio à decisão, sistemas de apoio ao executivo;
- e) visão sistêmica nas áreas administrativa, financeira, comercial e produção;
- f) prática em softwares aplicativos de gestão empresarial;
- g) uso de tecnologia da informação e comunicação: uma visão geral das tecnologias de informação e comunicação aplicadas nas organizações como recurso na inteligência competitiva;
- h) apresentação sobre inovação tecnológica;
- i) gestão da inteligência competitiva;
- j) gestão do conhecimento: apresentação sobre gestão do conhecimento envolvendo o capital intelectual, inteligência competitiva, sociedade do conhecimento, inteligência empresarial.

A disciplina de TIGN contribui para a formação do acadêmico permitindo conhecer a modelagem e a gestão de processos organizacionais com enfoque em ferramentas analíticas e

de gerenciamento de processos. Possibilita a junção do conteúdo de várias disciplinas presentes no curso de BSI. Com essa ferramenta será permitido ao aluno da disciplina aprender de forma prática os conceitos estudados em algumas matérias ao longo do curso.

2.6 TRABALHOS CORRELATOS

Entre vários trabalhos que abordam o tema DW, alguns se destacam pela semelhança com a ferramenta elaborada neste trabalho.

Kisner (2006) apresentou como monografia no programa de Pós-Graduação em Tecnologia da Informação voltada a Gestão de Negócios conceitos, técnicas e ferramentas de *Business Intelligence*. Realizou um estudo do modelo de BI onde uniu os modelos de BI com o conhecimento gerado pelos gestores das empresas gerando assim uma nova base de dados, uma base do conhecimento. Foram comentadas algumas ferramentas disponíveis no mercado.

Zimmermann (2006) propôs, como Trabalho de Conclusão de Curso na Universidade Regional de Blumenau, um sistema de apoio à decisão baseado em BI que foi disponibilizado em uma empresa alimentícia. Para a consulta dos dados foi utilizado o componente *decision cube* da linguagem Delphi.

Strey (2002) construiu, em seu trabalho de Conclusão de Curso na Universidade Regional de Blumenau, um sistema de informação onde utilizou a metodologia de Sistema de Informação Estratégico para o Gerenciamento Operacional (SIEGO) juntamente com as técnicas de granularidade e cubo de decisão. O trabalho tinha como intuito disponibilizar informações para auxiliar a tomada de decisões de executivos de uma empresa eletromecânica de Blumenau.

Fernandes e Kantorski (2008) abordam em artigo para o curso de Sistemas de Informação da Universidade Luterana do Brasil (ULBRA), os conceitos básicos que serão importantes para o desenvolvimento desse trabalho, tais como os de BI, DW e OLAP. Descrevem uma ferramenta *web open source* que tem como principal objetivo viabilizar a criação de consultas em cubos de dados.

Franco (2001) criou em seu trabalho de Conclusão de Curso na Universidade Regional de Blumenau, um sistema direcionado para a área de gestão ambiental. Nesse trabalho, foi desenvolvido o *Data Warehouse* onde ficará disponível o *Data Mart* criado no presente trabalho. O sistema é dividido em dois módulos, o primeiro foi desenvolvido em ambiente

Delphi e acessa a base de dados localmente. O segundo módulo, desenvolvido em PHP e Macromedia Flash, permite o acesso à base de dados via internet.

Casas (2009) apresentou em seu trabalho de Conclusão de Curso, na Universidade Regional de Blumenau, um aplicativo baseado no método do *Balanced Scorecard*, onde utilizou a linguagem de programação Personal Home Pages (PHP) e JavaScript com banco de dados MySQL.

3 DESENVOLVIMENTO DA FERRAMENTA

Conforme objetivos definidos neste trabalho, foi desenvolvida uma ferramenta de ensino em DW para ser utilizada pelos alunos da disciplina de TIGN do curso de BSI da FURB. No primeiro tópico são definidos os requisitos a serem atendidos pela ferramenta. Os tópicos seguintes abordam as especificações da ferramenta, apresentando diagramas *Unified Modeling Language* (UML) com a finalidade de definir os processos realizados pelo software.

3.1 PRINCIPAIS REQUISITOS DE SOFTWARE

O DW foi criado anteriormente no trabalho de Franco (2001), já com o *Data Mart* Sistemas de Informação Aplicado ao Sistema de Gestão Ambiental (SISGA). Após este trabalho, foram criados outros *Data Marts* utilizando o mesmo *Data Warehouse*, destacando-se o mais recente, o de Casas (2009), onde criou-se o *Data Mart Balanced ScoreCard* (BSC).

Na proposta inicial constava a criação dinâmica de *Data Warehouse*, *Data Mart*, tabela fato, medidas e dimensões. Posteriormente observou-se que esses requisitos fogem do foco da disciplina de Tecnologia da Informação na Gestão de Negócios (TIGN) não sendo necessária a sua implementação. Desta forma, será utilizado o mesmo DW disponibilizado anteriormente no trabalho de Franco (2001). O *Data Mart* e suas tabelas foram pré-definidos.

Os dados para unificar e povoar as tabelas são obtidos de arquivos no formato XML criados pelos alunos. Conforme atividade desenvolvida com a turma do primeiro semestre de 2010, os alunos receberam arquivo com os dados disponíveis na base e a estrutura que deveria ser utilizada para a elaboração do arquivo com os dados. Nesta atividade, os acadêmicos ficaram responsáveis por criar registros na tabela de fato (Vendas). Desta forma, pôde-se explicar a ligação entre as tabelas em um esquema estrela.

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

Requisitos Funcionais	Caso de Uso
RF01: A ferramenta deverá permitir ao aluno realizar o seu cadastro.	UC01
RF02: A ferramenta deverá permitir ao administrador realizar consulta e manutenção do cadastro de usuários, possibilitando manter o cadastro de permissão de acesso para apresentar somente as tabelas que poderão ser utilizadas pelo usuário.	UC03
RF03: A ferramenta deverá exigir a realização do <i>login</i> que, se aceito, levará para a tela inicial.	UC02
RF04: A ferramenta deverá permitir ao usuário cadastrar e filiar-se a uma empresa.	UC07 e UC08
RF05: A ferramenta deverá permitir ao administrador manter o cadastro de empresa.	UC04
RF06: A ferramenta deverá permitir ao administrador disponibilizar fundamentação teórica sobre <i>Data Warehouse</i> para os alunos.	UC05
RF07: A ferramenta deverá permitir ao usuário consultar fundamentação teórica disponibilizada pelo docente.	UC06
RF08: A ferramenta deverá permitir ao usuário unificar os dados vindos de um sistema legado, em formato XML ou TXT.	UC10
RF09: A ferramenta deverá permitir ao usuário carregar os dados de um banco de dados relacionados à tabelas da área de vendas.	UC09
RF10: A ferramenta deverá permitir ao usuário gerar relatórios.	UC11
RF11: A ferramenta deverá permitir gravar configurações do relatório.	UC12
RF12: A ferramenta deverá permitir ao usuário visualizar o relatório.	UC014
RF13: A ferramenta deverá permitir ao usuário abrir o relatório previamente salvo.	UC12 e UC13
RF14: A ferramenta deverá permitir a visualização do relatório através de gráficos.	UC15
RF15: A ferramenta deverá permitir consultar DW.	UC16
RF16: A ferramenta deverá permitir consultar <i>Data Marts</i> .	UC17
RF17: A ferramenta deverá permitir consultar tabela de fato.	UC18
RF18: A ferramenta deverá permitir consultar medidas.	UC18
RF19: A ferramenta deverá permitir consultar dimensões.	UC19

Quadro 1 – Requisitos funcionais

O Quadro 2 lista os requisitos não funcionais previstos para a ferramenta.

Requisitos Não Funcionais
RNF01: A ferramenta deverá ser desenvolvida para o banco de dados <i>SQLServer</i> .
RNF02: A ferramenta deverá utilizar o SQL Server Analysis Services.
RNF02: A ferramenta deverá ser construída sobre a plataforma .Net.
RNF03: A ferramenta deverá ser desenvolvida na linguagem C#.
RNF04: A ferramenta deverá habilitar ao usuário o perfil a ele designado conforme usuário e senha cadastrados.
RNF05: A ferramenta deverá utilizar o DW disponível no Campeche, criado no trabalho de Franco (2001).

Quadro 2 – Requisitos não funcionais

3.2 ESPECIFICAÇÃO

A especificação foi elaborada utilizando a linguagem de modelagem *Unified Modeling Language* (UML). A UML é uma linguagem que padroniza a visualização, especificação, construção e documentação de artefatos de um software em desenvolvimento (LARMAN, 2004). Para a construção dos diagramas foi utilizada a ferramenta Enterprise Architect (EA).

3.2.1 Diagrama de caso de uso

Na figura 2 é apresentado o diagrama de casos de uso ilustrando o acesso a ferramenta.

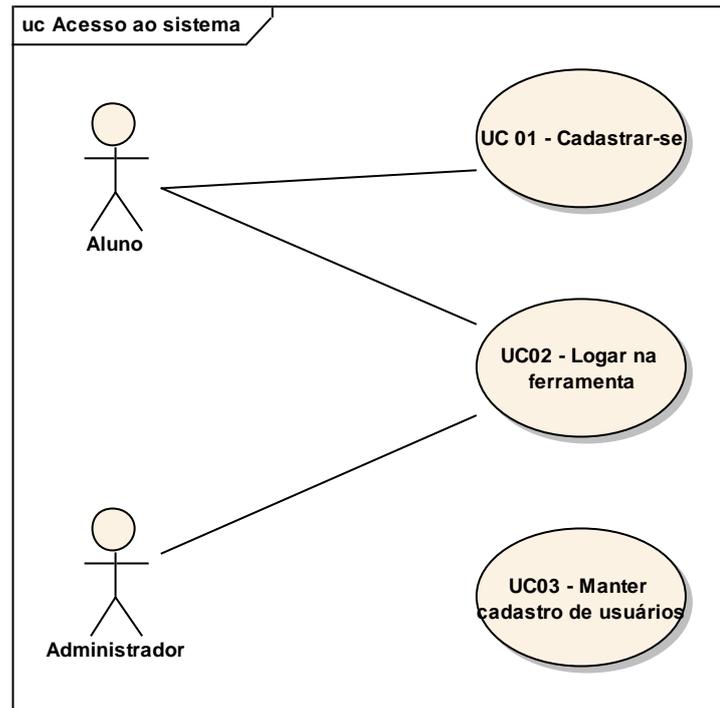


Figura 2 – Diagrama de acesso a ferramenta

A figura 3 apresenta o diagrama de casos de uso para manter os cadastros de fundamentação teórica e empresa, caso de uso para a unificação dos dados, povoamento das tabelas e a visualização desses dados.

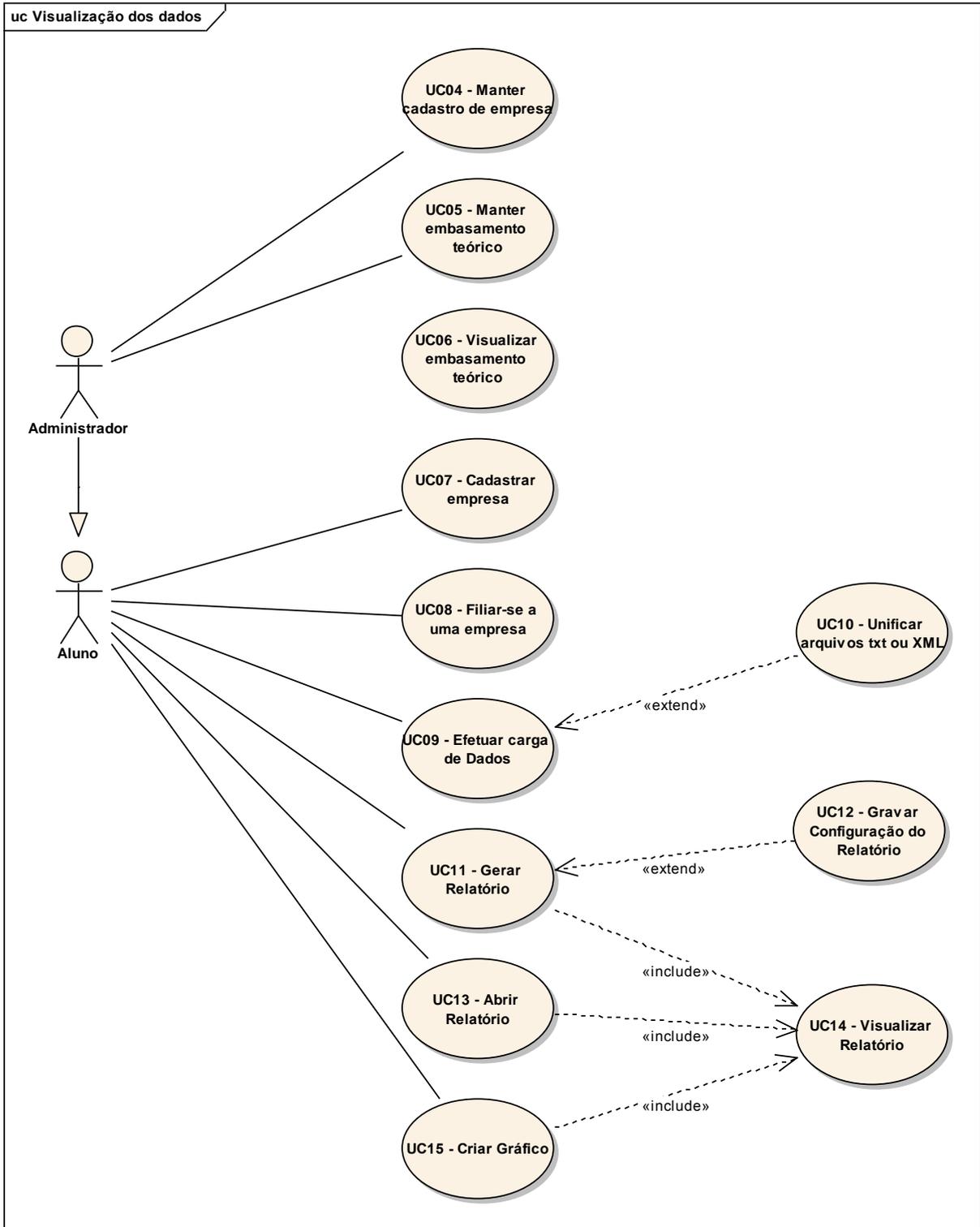


Figura 3 – Diagrama de caso de uso de visualização dos dados

Na figura 4 são ilustrados os casos de uso para consulta no *Data Warehouse*, no *Data Mart*, nas tabelas de fato e nas dimensões.

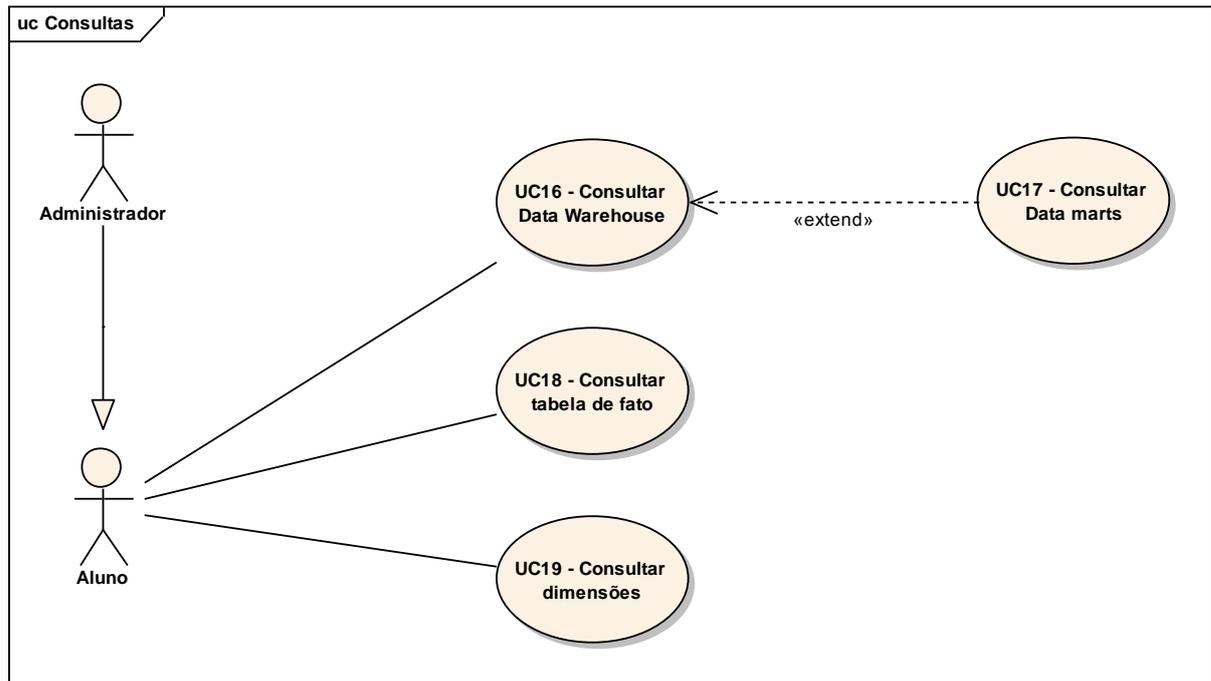


Figura 4 – Diagrama de caso de uso de consulta

3.2.2 Diagrama de atividades

Na figura 5, tem-se o diagrama de atividades que ilustra o funcionamento atual da disciplina de Tecnologia da Informação na Gestão de Negócios (TIGN), sendo que em cinza estão as alterações que esta ferramenta proporcionou na disciplina.

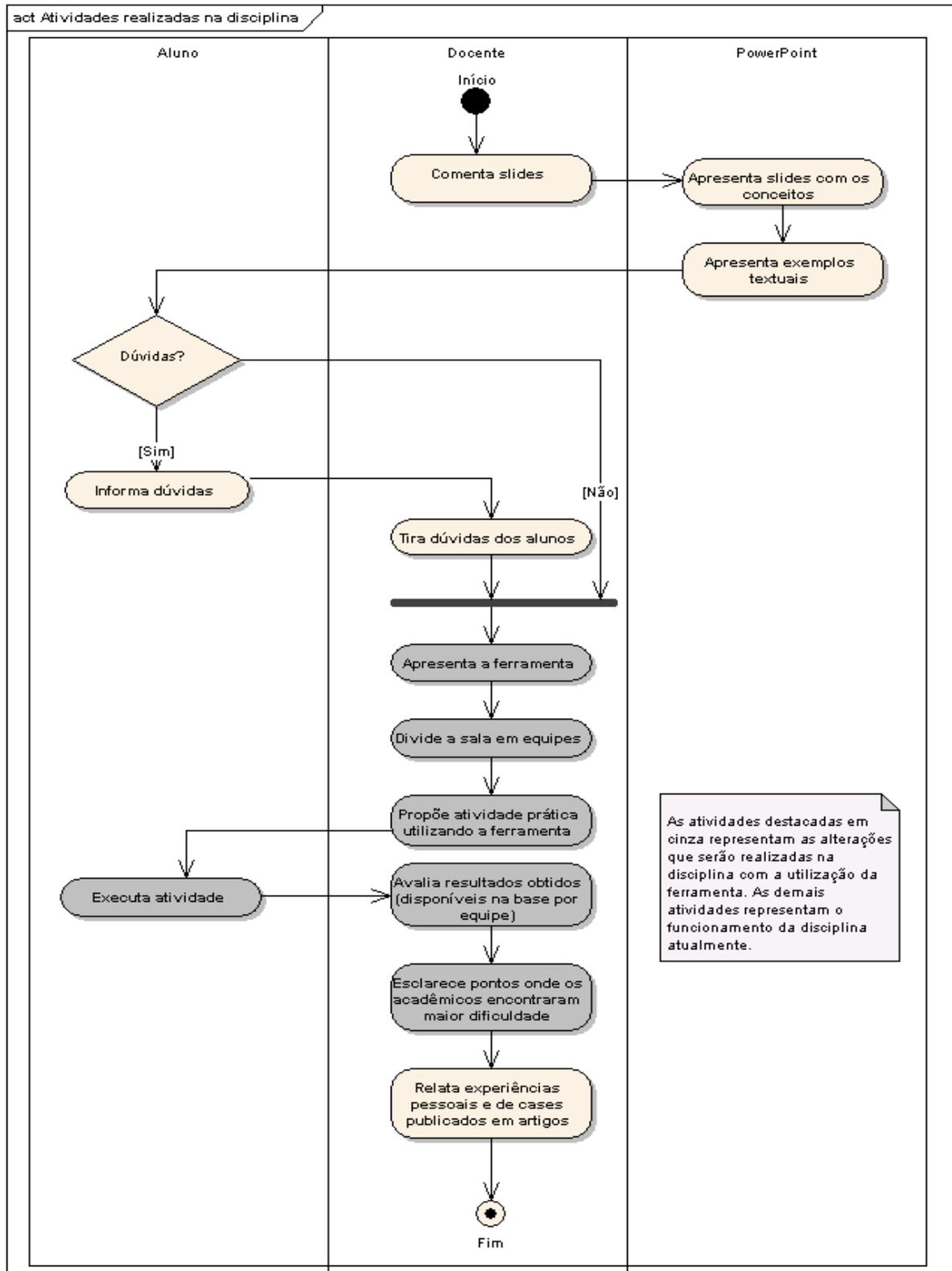


Figura 5 – Diagrama de atividades da disciplina

A figura 6 apresenta através de um diagrama de atividades o processo necessário para realizar a unificação de dados.

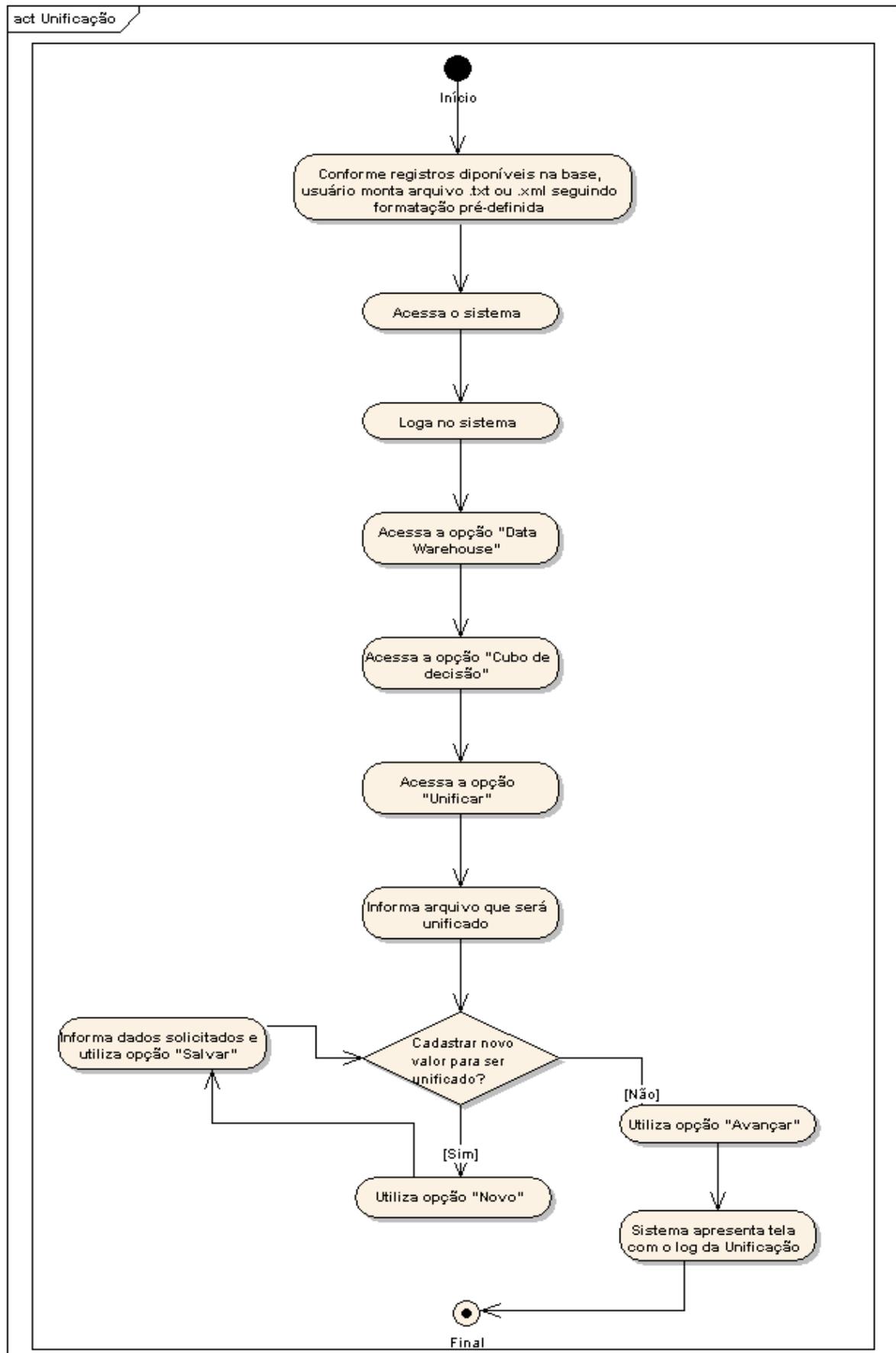


Figura 6 – Diagrama de atividades da unificação de dados

A figura 7 apresenta a seqüência de processos necessários para povoar as tabelas.

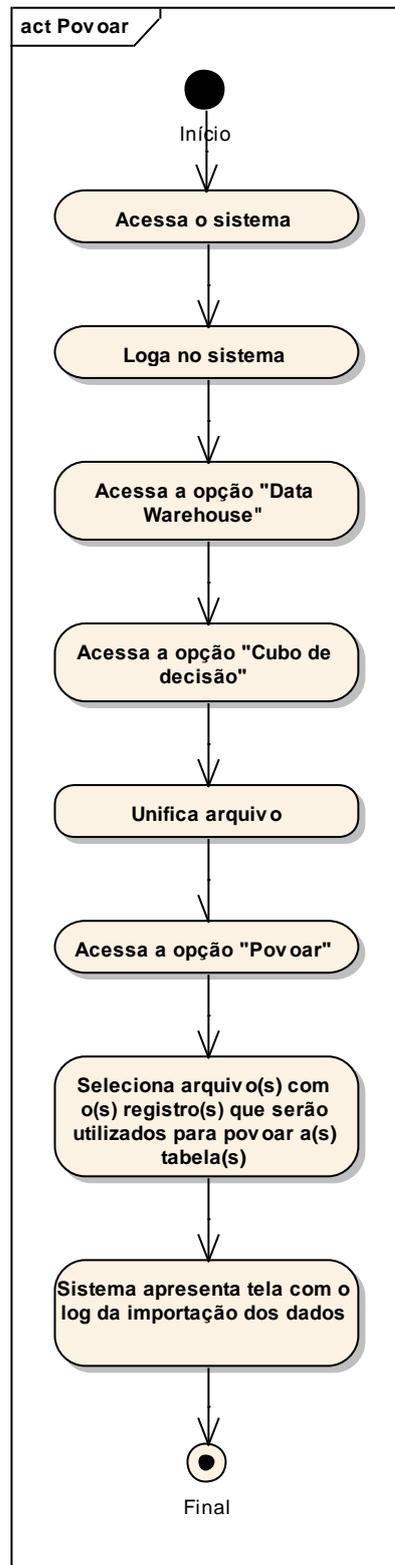


Figura 7 – Diagrama de atividades do povoamento das tabelas

3.2.3 Modelo de Entidade e Relacionamento (MER)

A figura 8 apresenta o Modelo de Entidade e Relacionamento da ferramenta, onde tem-se as entidades com suas respectivas colunas, chave primária (PK) e chaves estrangeiras (FK). O apêndice B apresenta o dicionário de dados com a descrição das tabelas e colunas. Todas as tabelas foram criadas neste trabalho, porém a criação do DW utilizado foi feita no trabalho de Franco (2001). No trabalho de Franco (2001) foram criadas outras tabelas que não tem relação com este trabalho.

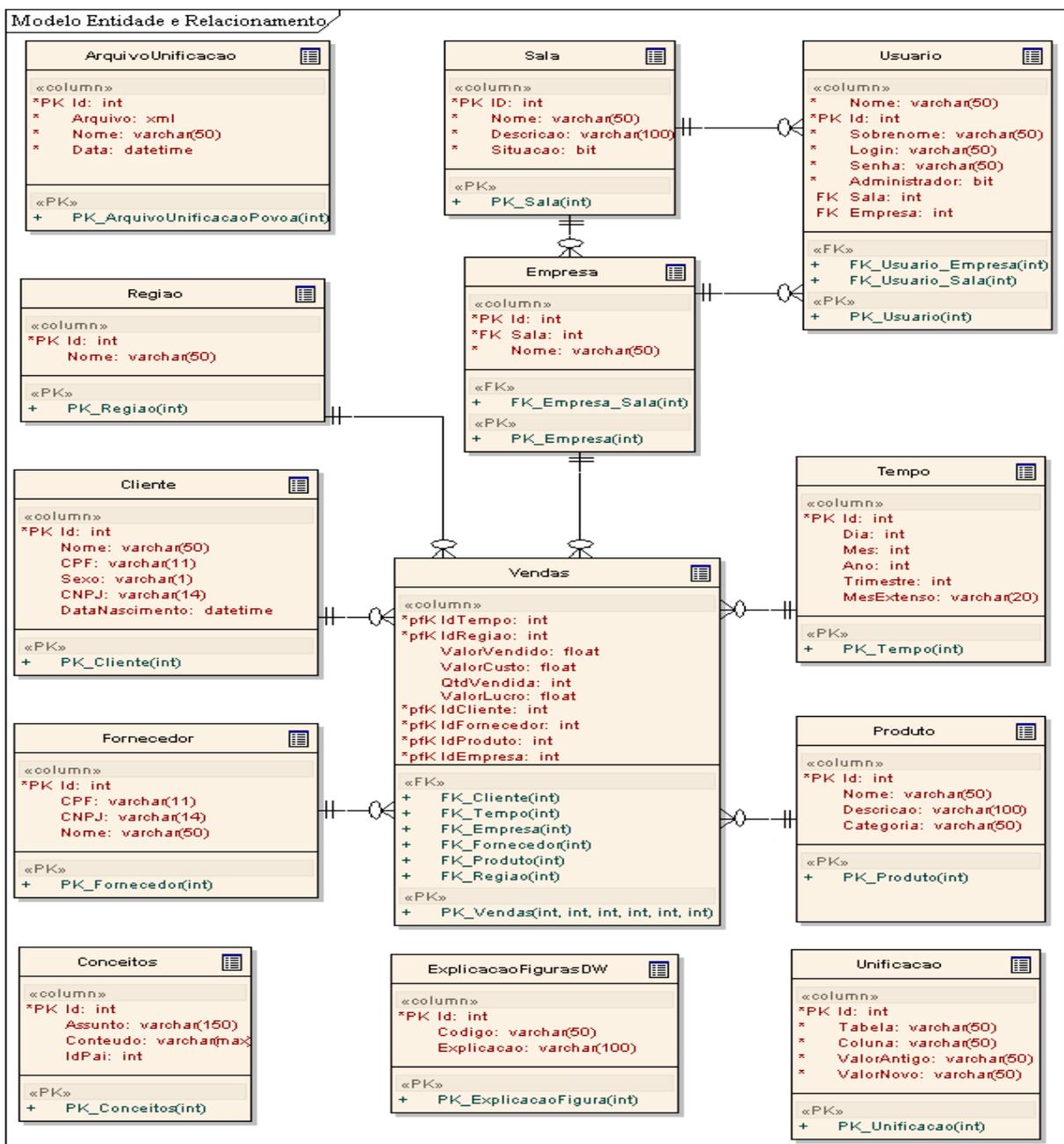


Figura 8 – Diagrama de Entidade e Relacionamento

3.2.4 Diagrama de Classes

A ferramenta é dividida em três camadas denominadas de camada de dados, camada de negócios e a camada de apresentação. Os diagramas apresentados nas figuras 9 e 10 são referentes à camada de negócios. A camada de negócios é responsável por implementar a lógica da ferramenta, disponibilizando a mesma para a camada de apresentação.

A figura 9 apresenta as classes utilizadas para realizar a unificação dos arquivos e a carga dos dados.

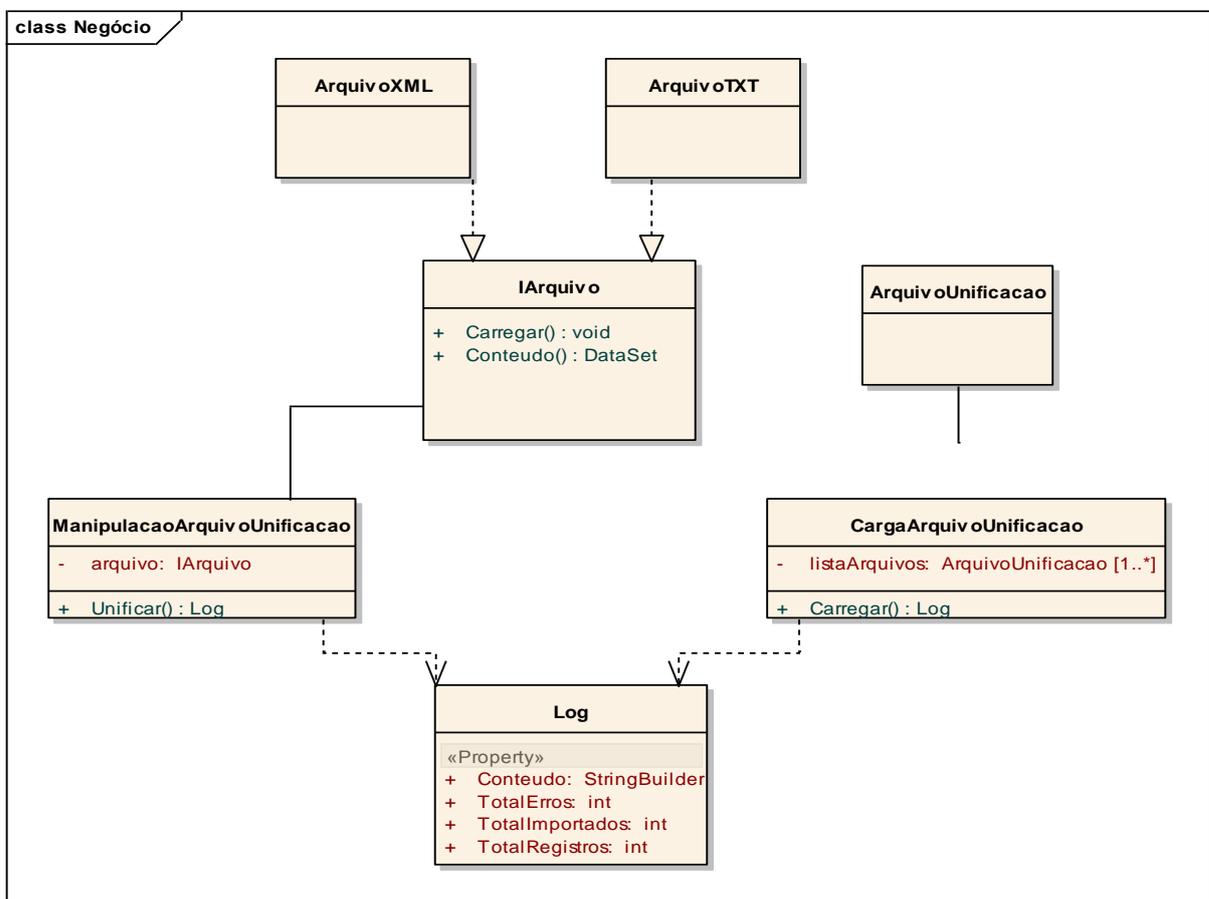


Figura 9 – Diagrama de classes

A figura 10 apresenta as demais classes utilizadas pela camada de apresentação.

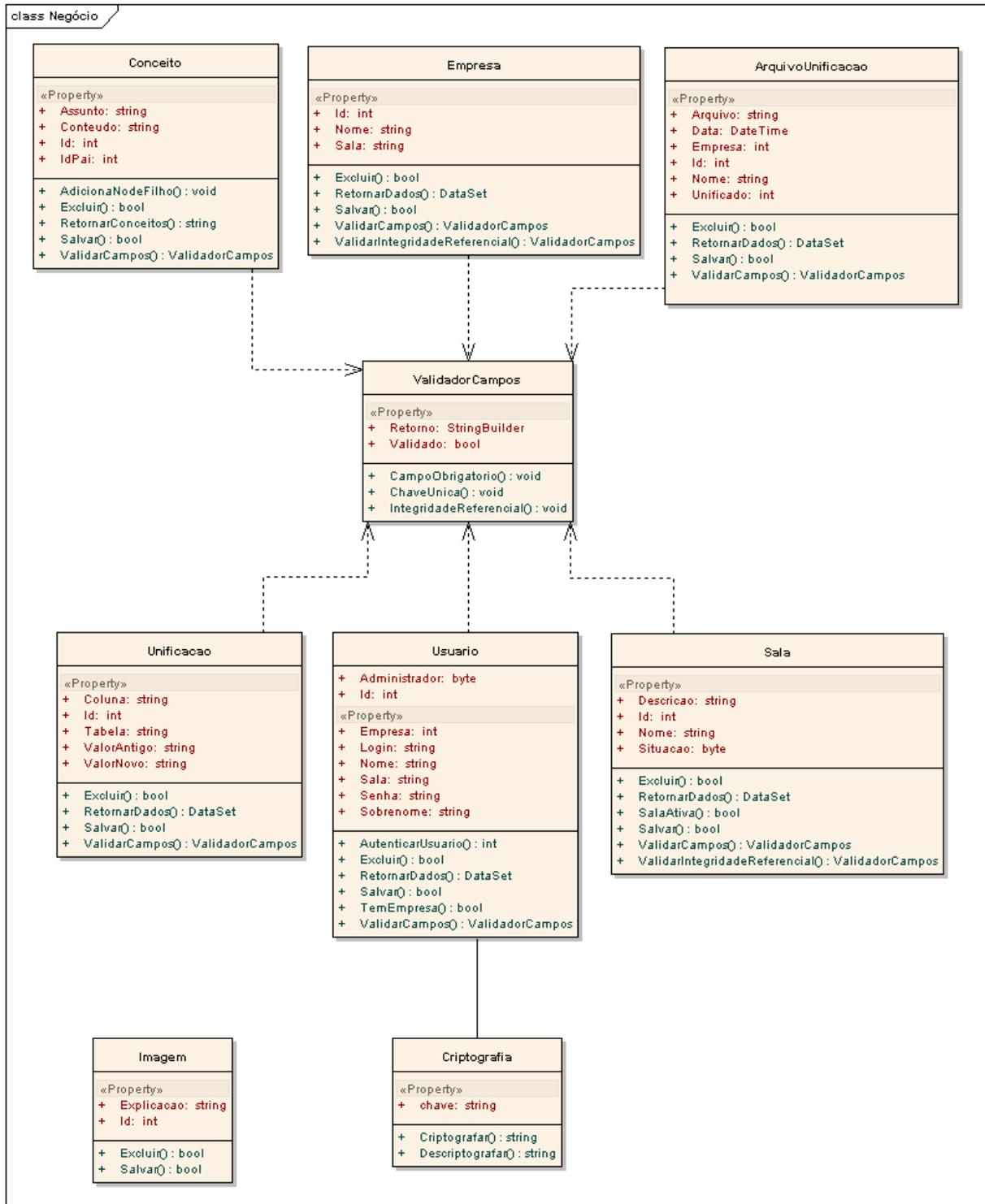


Figura 10 – Diagrama de classes

3.3 IMPLEMENTAÇÃO

A seguir são apresentadas as técnicas e ferramentas utilizadas e a operacionalidade da implementação.

3.3.1 Técnicas e ferramentas utilizadas

Para a implementação da ferramenta, foi utilizada a linguagem de programação C# com o Framework .Net 2.0, no ambiente de desenvolvimento Microsoft Visual Studio 2008. Para a persistência dos dados, foi utilizado o banco de dados SQLServer 2005. Como servidor *web*, utilizou-se o Internet Information Services (IIS).

O FckEditor foi utilizado para possibilitar editar o layout dos textos disponibilizados para os acadêmicos no “Embasamento Teórico” da ferramenta. Foi utilizada a Dynamic-link library (DLL) SQLMaker.dll para a realização do acesso e manipulação do BD.

O processamento do cubo foi feito através do SQL Server Analysis Services. Para a visualização dos dados do cubo foram utilizados componentes do Dundas Olap Web Services.

3.3.1.1 SQL Server Analysis Services

Jacobson e Stacia (2007) apresentam que o SQL Server Analysis Services (SSAS) é uma ferramenta desenvolvida pela empresa Microsoft. Esta ferramenta implementa o processamento analítico *online* (OLAP). É utilizada para a integração de dados relacionais, possibilitando a criação de soluções BI.

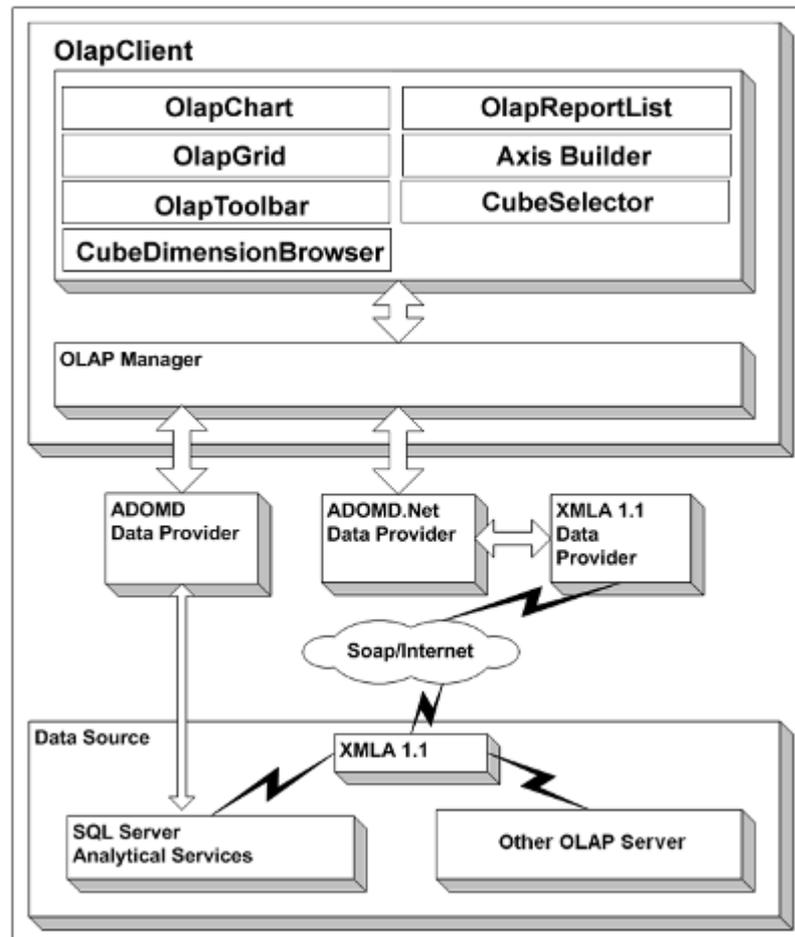
Através do SQL Server Managemet Studio 2005 (SSMS), os bancos de dados do SSAS são gerenciados. Esta ferramenta também é desenvolvida pela Microsoft e possibilita a conexão com os serviços fornecidos pela instalação do SSAS, permitindo a criação dos cubos de decisão, mapeamento do DW e definição das medidas e dimensões. O *Cube Browser* disponível nesta ferramenta permite a visualização dos dados dimensionais e a realização de operações relacionadas ao OLAP. Através dessa ferramenta pode-se também realizar o processamento do cubo de decisão, ou seja, a leitura das alterações ocorridas no DW para que

seja possível sua manipulação por ferramentas OLAP que se conectem ao SSAS (JACOBSON; STACIA, 2007).

3.3.1.2 Dundas OLAP Services

Esta ferramenta permite que, com simples operações de clicar, arrastar e soltar, sejam feitas consultas a bases de dados multidimensionais. Possibilita a visualização de dados de forma interativa.

A figura 11 apresenta a sua arquitetura, em que o *OlapClient* é o componente que foi utilizado para permitir as consultas ao cubo. É composto pelo *OlapChart*, *OlapGrid*, *OlapToolbar*, *CubeDimensionBrowser*, *OlapReportList*, *AxisBuilder* e *CubeSelector*. O *OlapChart* é o componente utilizado para a visualização dos dados em forma de gráficos. *OlapGrid* é o responsável pela visualização dos dados em forma de tabelas. Através do *OlapToolbar*, são disponibilizadas funções que permitem definir opções de *layout* e visualização dos dados, entre outras. O *CubeDimensionBrowser* permite selecionar as dimensões que serão utilizadas. Através do *OlapReportList* pode-se gerenciar os relatórios (*views*), inclusive visualizar relatórios pré-definidos por rotina. O *AxisBuilder* é utilizado para definir os valores de cada eixo do gráfico. O *CubeSelector* permite selecionar o cubo do projeto definido nas propriedades do *ADOMD .Net Data Provider* que se deseja utilizar. Os *Data Providers* são utilizados para conectar e recuperar dados de uma fonte multidimensional (DUNDAS, 2010).



Fonte: Dundas (2010).

Figura 11 – Arquitetura do Dundas Olap Services

3.3.2 Operacionalidade da implementação

Nessa seção, será apresentada uma seqüência de procedimentos que deverão ser realizados pelos alunos para a execução da unificação dos dados, carga e visualização dos mesmos.

A ferramenta apresentada é um simulador de BI para uma empresa têxtil na área de vendas. Esse simulador é utilizado na disciplina TIGN para permitir o contato do acadêmico com uma das partes do BI, o cubo de decisão. Os dados estão armazenados por região, data da venda, produto, fornecedor, cliente e empresa para a qual a venda foi feita. Na disciplina, a sala é dividida em equipes, na ferramenta cada equipe é representada por uma empresa.

A ferramenta tem como dimensões “Cliente”, “Fornecedor”, “Produto”, “Tempo”, “Região” e “Empresa”. Para normalização, será utilizado o esquema estrela.

O simulador apresenta a tabela de “Vendas” como tabela de fatos. Essa tabela possui como atributos o produto, o cliente, o fornecedor, o tempo, a região, a empresa, a quantidade, o valor vendido e o custo.

Na dimensão “Produto” estão os atributos descrição, nome e categoria. Na dimensão “Cliente” estão os atributos nome, sexo, a data de nascimento, o número de Cadastro de Pessoa Física (CPF) e número de Cadastro Nacional de Pessoa Jurídica (CNPJ). Na dimensão “Fornecedor” estão os atributos nome, CPF e CNPJ. Na dimensão “Tempo” estão os atributos ano, trimestre, mês e dia. Outra dimensão da ferramenta é a de “Região” com o nome de cada região. Esse simulador permite absorver dados de várias fontes, como sistemas legados, onde o sistema fornece arquivos txt ou XML para a importação.

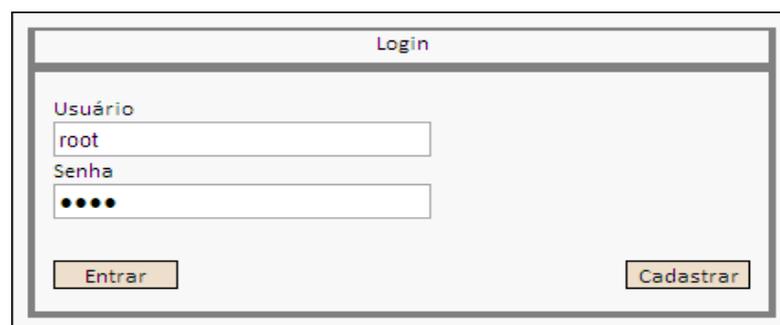
Os dados são extraídos e carregados de forma temporária em uma área de preparação chamada *Operational Data Store* (ODS) onde é feita a unificação dos registros que ficam prontos para o carregamento no DW, gerando um novo arquivo XML com os dados unificados.

Para realizar a unificação a ferramenta permitirá ao usuário informar o dado como está no arquivo para importação e como deve ficar o arquivo gerado com a unificação. Os registros são enviados para o DW. O usuário seleciona os registros que devem ser carregados em um DM, e na seqüência o usuário tem acesso aos relatórios contendo as informações relevantes ao seu negócio.

Na seqüência é apresentada a ferramenta no formato tutorial.

3.3.2.1 Acesso a ferramenta

Ao acessar a ferramenta, é mostrada a tela de *login* da ferramenta, conforme figura 12. Para a realização do *login* o usuário precisa estar cadastrado na ferramenta.



A imagem mostra uma interface de usuário para login. No topo, o título "Login" está centralizado. Abaixo dele, há dois campos de entrada: "Usuário" com o texto "root" preenchido, e "Senha" com pontos para ocultar o conteúdo. Na base da interface, há dois botões: "Entrar" à esquerda e "Cadastrar" à direita.

Figura 12 – Tela de autenticação de usuário

Ao efetuar o *login* com um usuário “Administrador”, a ferramenta apresenta tela conforme figura 13. O administrador deve cadastrar a sala que os alunos selecionarão nos seus cadastrados. Para efetuar o cadastro da sala utiliza-se a opção “Salas”.



Figura 13 – Tela de opções de acesso

Ao utilizar a opção “Salas”, a ferramenta apresenta tela para manter o cadastro de salas, conforme figura 14. O cadastro das salas é mantido apenas pelos usuários cadastrados como “Administrador”.

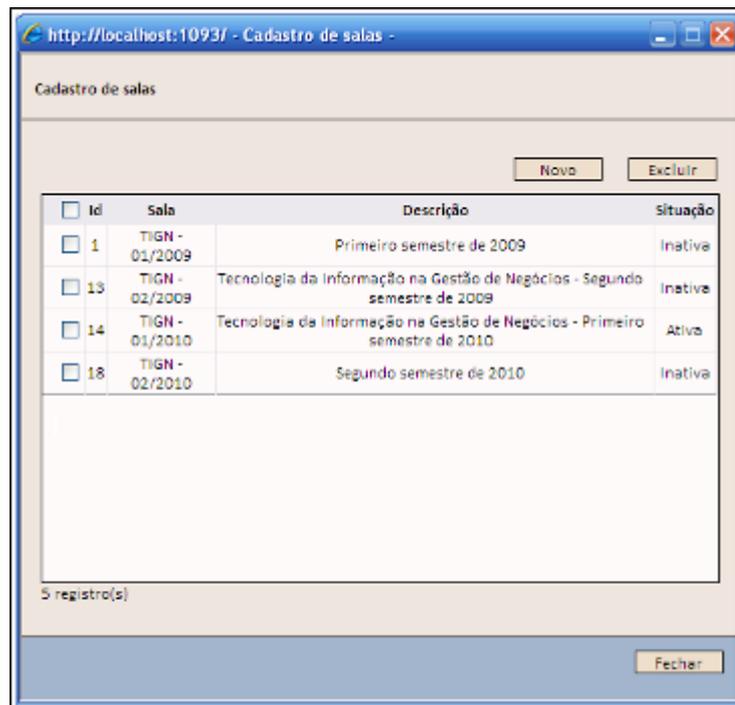


Figura 14 – Tela para manter o cadastro de salas

Clicando sobre o registro, pode-se editar, selecionando registro(s) e utilizando o botão “Excluir” pode-se excluir um ou mais registros. Ao utilizar o botão “Novo”, a ferramenta apresenta tela para o cadastro de uma sala, conforme figura 15, onde o campo em vermelho é de preenchimento obrigatório.

http://localhost:1093/?id=14 - Cadastro de salas -

Cadastro de salas

Nome: TIGN - 01/2010 Ativa:

Descrição: Tecnologia da Informação na Gestão de Negócios - Primeiro semestre de 2010

Salvar Fechar

Figura 15 – Tela para realização do cadastro de sala

A opção “Usuários”, abaixo de “Salas” nas opções de acesso, permite ao administrador manter os dados dos usuários cadastrados na ferramenta e definir outros usuários como “Administrador” da ferramenta se necessário. Ao utilizar essa opção é apresentada tela para manter os dados dos usuários, conforme figura 16.

http://localhost:1093/ - Cadastro de usuários -

Cadastro de usuários

Novo Excluir

<input type="checkbox"/>	Id	Nome	Sobrenome	Login	Sala
<input type="checkbox"/>	1	root	.	root	1
<input type="checkbox"/>	2	Joana	Simon	joana	14
<input type="checkbox"/>	3	Laura	Müller	laura	1
<input type="checkbox"/>	4	Cristine	Volpi	cris	13
<input type="checkbox"/>	5	Dunga	Simon	dunga	14
<input type="checkbox"/>	6	William	Wachholz	william	1
<input type="checkbox"/>	8	Marlon	da Silva	marlonchad	14
<input type="checkbox"/>	9	Erickson	Marqueti	erickson	14
<input type="checkbox"/>	10	Airto	Irineu Furlan	afurlan	14
<input type="checkbox"/>	12	Charles	Moro	cmoro	14

1 2

10 registro(s)

Fechar

Figura 16 – Tela para manter o cadastro de usuários

O cadastro de usuário funciona de forma semelhante ao cadastro de salas. Utiliza-se o botão “Novo” para realizar o cadastro de um usuário, conforme figura 17.

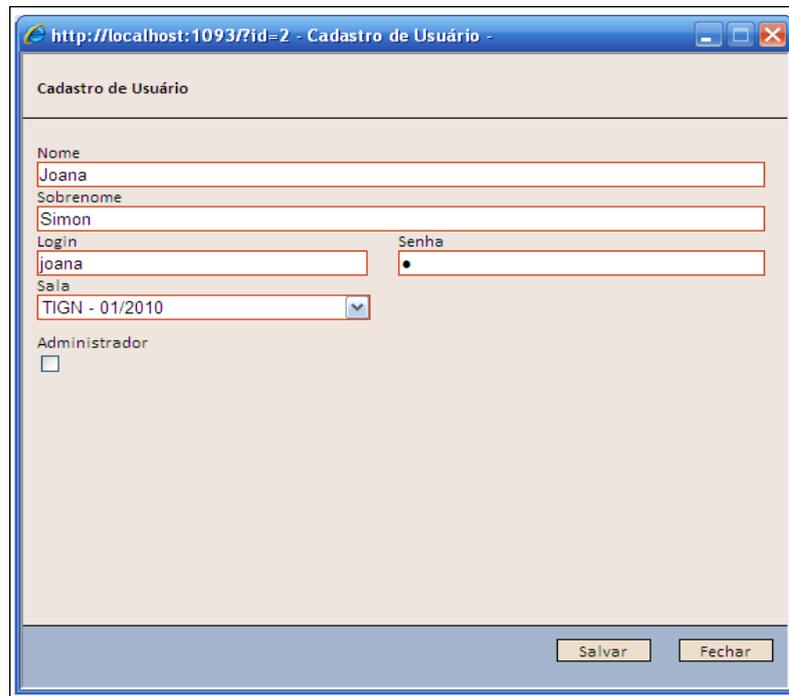


Figura 17 – Tela para realizar o cadastro de usuário

O cadastro de usuários é mantido apenas pelo administrador da ferramenta, mas o próprio aluno, utilizando a função “Cadastrar” disponível na tela de *login*, pode realizar o seu cadastro. A figura 18 apresenta a tela de cadastro de usuário, em que a sala é a sala da disciplina - no exemplo, seria a sala do primeiro semestre de 2010 da disciplina de Tecnologia da Informação na Gestão de Negócios (TIGN).

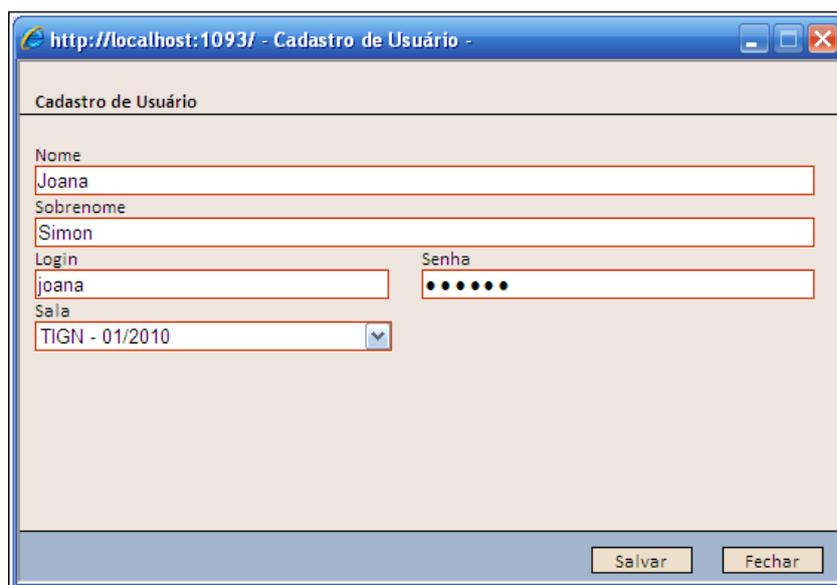


Figura 18 – Tela de cadastro de usuário

Ao efetuar o *login*, para os usuários que não são administradores da ferramenta,

apresenta-se tela com as opções de acesso disponíveis, conforme figura 19. Em “Data Warehouse” estão as principais funcionalidades definidas neste trabalho, em *Data Mining* e Redes Neurais disponibilizar-se-á trabalhos que serão desenvolvidos por outros acadêmicos e integrados neste.



Figura 19 – Tela de opções de acesso

Quanto ao controle de acesso à ferramenta, como se observa na figura 20, separou-se as partes da ferramenta com diferentes níveis de acesso (Nenhum, usuário cadastrado ou usuário cadastrado como administrador) em diretórios diferentes. Em cada um desses diretórios criou-se um arquivo de configuração onde especificou-se quem pode ter acesso a esta parte da ferramenta (este diretório). O controle de usuário é feito utilizando o recurso *Forms Authentication* do Framework .Net.

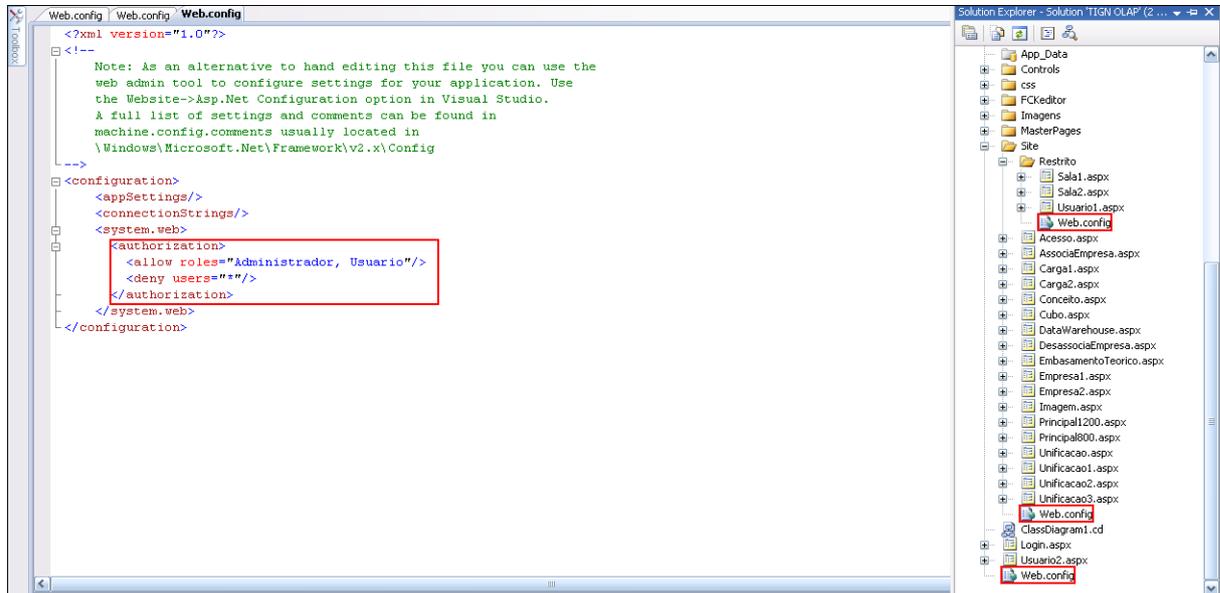


Figura 20 – Controle de acesso a partes específicas da ferramenta

3.3.2.2 Embasamento teórico

Utilizando a opção “Data Warehouse”, a ferramenta apresenta tela, conforme figura

21, com opção para cadastrar empresa, associar-se a uma empresa, acesso ao embasamento teórico sobre BI e acesso ao cubo de decisão incluindo a unificação e carga dos dados que serão visualizados pelo mesmo.

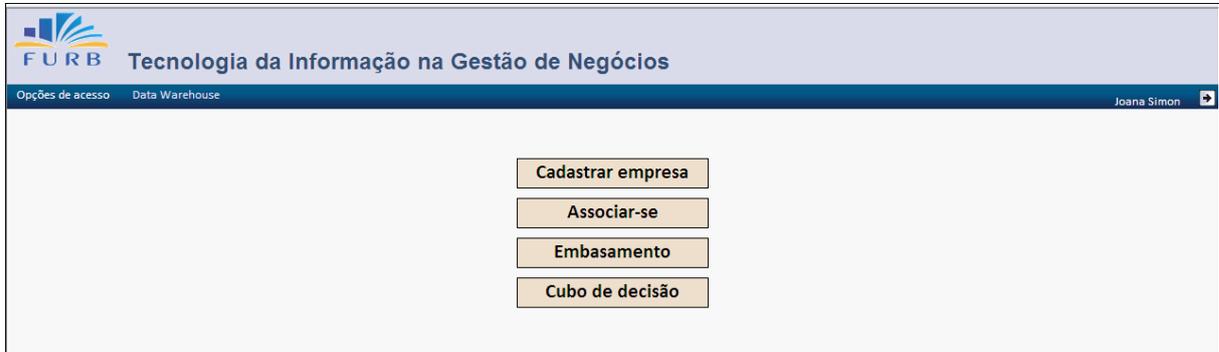


Figura 21 – Tela com opções da ferramenta

Na opção Embasamento está disponível a teoria sobre BI, DW, Cubo de decisão e OLAP, conforme figura 22. Os acadêmicos poderão utilizar para consulta, mas apenas usuários cadastrados como “Administrador” poderão manter esses dados.

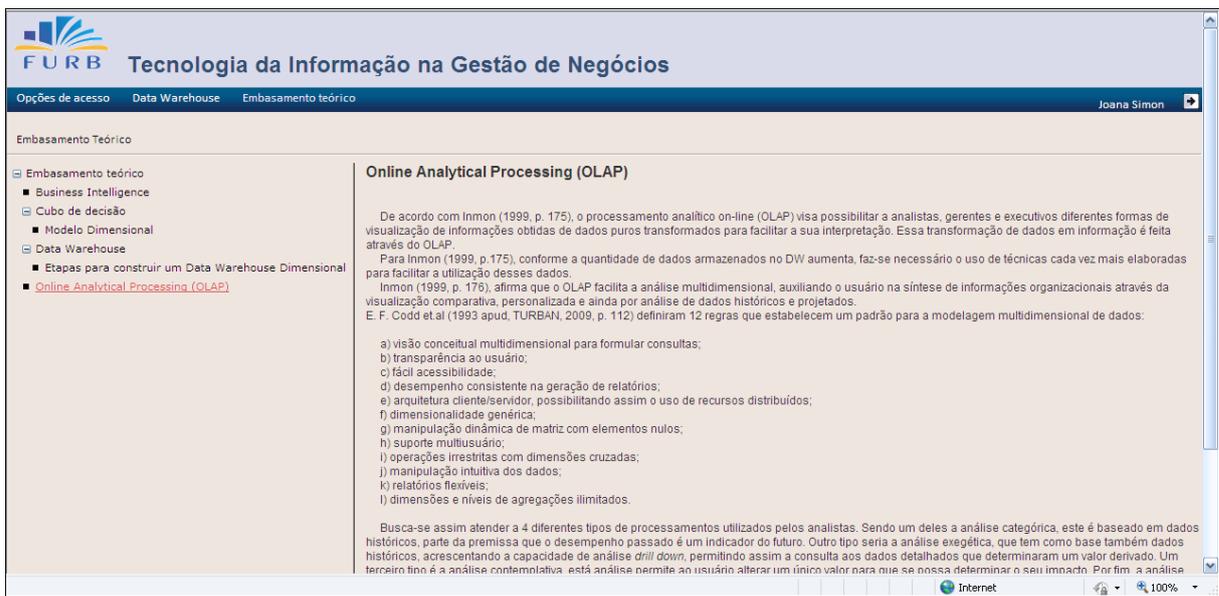


Figura 22 – Tela de consulta ao material teórico

Apenas o administrador poderá adicionar, alterar ou excluir esses dados, o que é feito através dos botões disponíveis na parte inferior da tela, conforme figura 23.

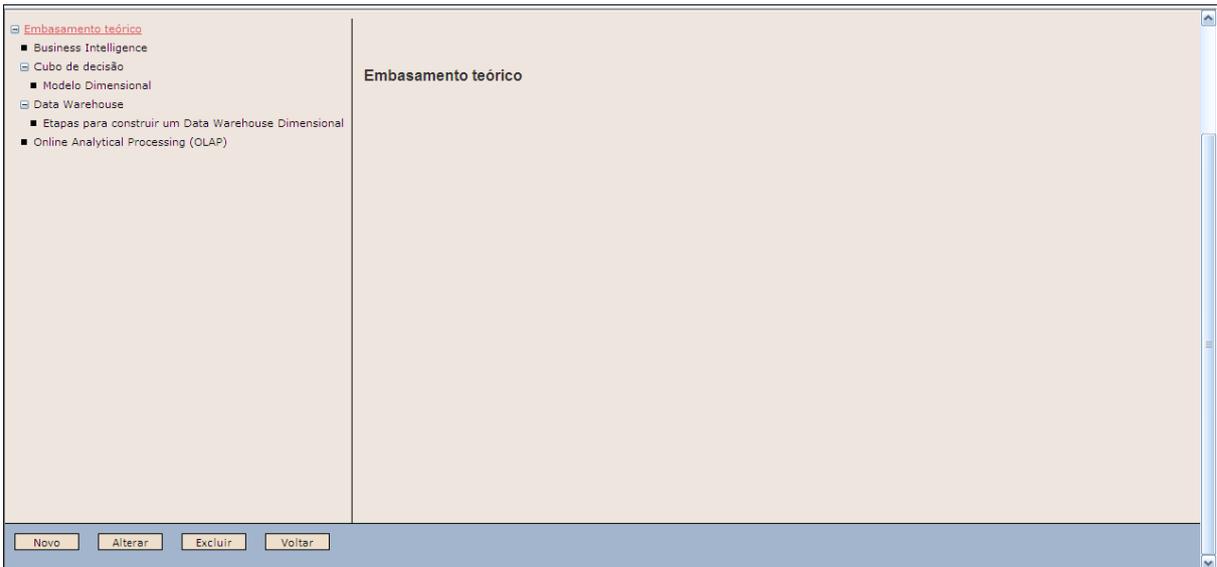


Figura 23 – Tela para manter dados do embasamento teórico

Para inserir um novo assunto no embasamento teórico deve-se selecionar o registro pai, onde será criado o filho. Por exemplo, para adicionar o material referente a BI foi selecionado “Embasamento Teórico” e utilizado a opção “Novo”, preencheu-se o “Assunto” e o “Conteúdo”, conforme figura 24, criando-se assim o registro abaixo de “Embasamento teórico”.

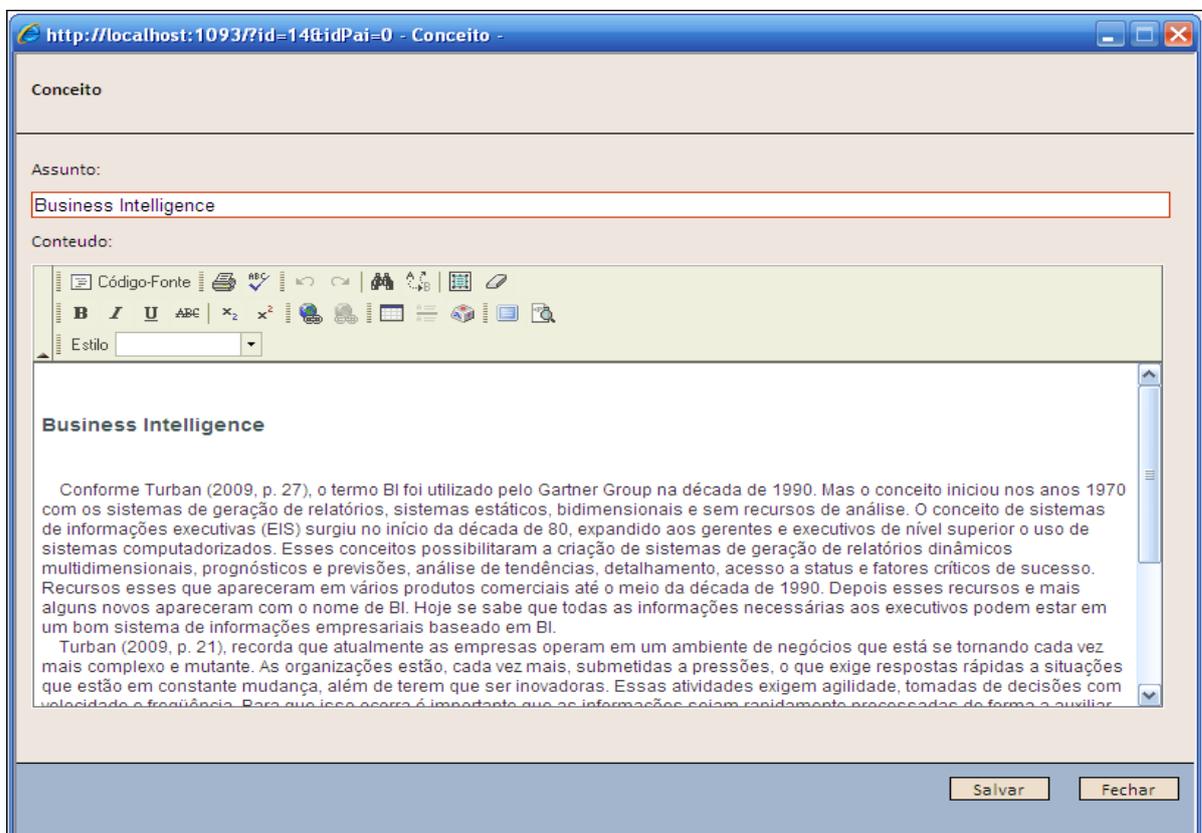


Figura 24 – Tela para o cadastro de um novo assunto

Para alterar ou excluir um assunto, deve-se clicar no assunto e utilizar a opção “Alterar” ou “Excluir”.

O quadro 3 mostra o componente “TreeView” utilizado para montar a árvore de embasamento teórico.

```
<asp:TreeView ID="treeViewConceitos" runat="server" ImageSet="Simple"
  NodeIndent="10" DataSourceID="XmlDataSource1"
  onselectednodechanged="treeViewConceitos_SelectedNodeChanged">
  <ParentNodeStyle Font-Bold="False" />
  <HoverNodeStyle Font-Underline="True" ForeColor="#DD5555" />
  <SelectedNodeStyle Font-Underline="True" ForeColor="#DD5555"
    HorizontalPadding="0px" VerticalPadding="0px" />
  <DataBindings>
    <asp:TreeNodeBinding DataMember="Conceito" ValueField="Id" TextField="Assunto"></asp:TreeNodeBinding>
  </DataBindings>
  <NodeStyle Font-Names="Verdana" Font-Size="8pt" ForeColor="Black"
    HorizontalPadding="0px" NodeSpacing="0px" VerticalPadding="0px" />
</asp:TreeView>
```

Quadro 3 – Componente para montar a árvore de embasamento teórico

Como se pode observar na figura 25, os botões para inserir, alterar, ou excluir são visíveis apenas para os usuários cadastrados como “Administrador”. Ao utilizar a opção “Embasamento teórico”, a rotina cria um arquivo XML, tendo como primeiro nó a declaração, o cabeçalho com a versão e codificação do arquivo. O segundo nó é de “Conceito”, com o identificador único (ID) e o assunto do registro raiz (com ID 0, “Embasamento teórico”).

```

if (!User.IsInRole("Administrador"))
{
    buttonNovo.Visible = false;
    buttonAlterar.Visible = false;
    buttonExcluir.Visible = false;
}

if (!IsPostBack)
{
    XmlDataSource1.Data = Negocio.Conceito.RetornarConceitos();
    XmlDataSource1.DataBind();

    Negocio.Conceito conceito = new Negocio.Conceito(0);
    conteudo.InnerHtml = conceito.Conteudo;
}

public static string RetornarConceitos()
{
    XmlDocument xmldoc = new XmlDocument();

    XmlNode xmlnode = xmldoc.CreateXmlDeclaration("1.0", "UTF-8", null);
    xmldoc.AppendChild(xmlnode);

    SQLMaker.SelectMaker selectNodePrincipal = new SQLMaker.SelectMaker("Conceitos");
    selectNodePrincipal.AddCampo("Id");
    selectNodePrincipal.AddCampo("Assunto");
    selectNodePrincipal.AddFiltroWhere("Id", SQLMaker.SelectMaker.OperadorFiltro.Igual, "0");
    selectNodePrincipal.Executa("Metadados");

    if (selectNodePrincipal.Next())
    {
        xmlnode = xmldoc.CreateElement("Conceito");

        XmlAttribute attrId = xmldoc.CreateAttribute("Id");
        attrId.Value = selectNodePrincipal.GetCampoString("Id");
        xmlnode.Attributes.Append(attrId);
        XmlAttribute attrAssunto = xmldoc.CreateAttribute("Assunto");
        attrAssunto.Value = selectNodePrincipal.GetCampoString("Assunto");
        xmlnode.Attributes.Append(attrAssunto);

        xmldoc.AppendChild(xmlnode);

        adicionaNodeFilho(ref xmldoc, ref xmlnode);
    }

    return xmldoc.InnerXml;
}

```

Figura 25 – Código para carregar dados na árvore

Caso tenha registro(s) filho(s), a rotina adiciona o nó filho e verifica se este também tem filho, conforme quadro 4. Após adicionar todos os registros é retornada uma *string* que é carregada formando a árvore com os conceitos.

```

private static void adicionaNodeFilho(ref XmlDocument xmldoc, ref XmlNode xmlNodePai)
{
    SQLMaker.SelectMaker selectNode = new SQLMaker.SelectMaker("Conceitos");
    selectNode.AddCampo("Id");
    selectNode.AddCampo("Assunto");
    selectNode.AddFiltroWhere("IdPai", SQLMaker.SelectMaker.OperadorFiltro.Igual, xmlNodePai.Attributes["Id"].Value);
    selectNode.AddOrdenacao("Assunto", SQLMaker.SelectMaker.TipoOrdenacao.Ascendente);
    selectNode.Executa("Metadados");

    while (selectNode.Next())
    {
        XmlNode xmlnode = xmldoc.CreateElement("Conceito");

        XmlAttribute attrId = xmldoc.CreateAttribute("Id");
        attrId.Value = selectNode.GetCampoString("Id");
        xmlnode.Attributes.Append(attrId);
        XmlAttribute attrAssunto = xmldoc.CreateAttribute("Assunto");
        attrAssunto.Value = selectNode.GetCampoString("Assunto"); ;
        xmlnode.Attributes.Append(attrAssunto);

        xmlNodePai.AppendChild(xmlnode);

        adicionaNodeFilho(ref xmldoc, ref xmlnode);
    }
}

```

Quadro 4 – Código para carregar nós filhos na árvore

3.3.2.3 Divisão em empresas

Nas aulas, a sala será dividida em equipes, sendo que cada equipe cadastrará uma empresa. Para efetuar o cadastro utiliza-se o botão “Empresa”, a ferramenta apresenta tela com as empresas já cadastradas, conforme figura 26, e opção “Novo” para criar uma empresa.

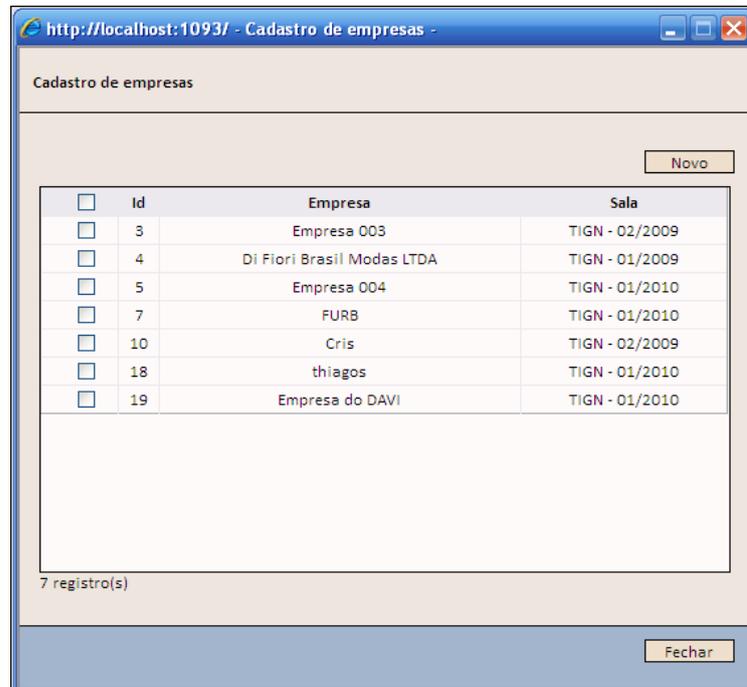


Figura 26 – Tela com as empresas cadastradas

Ao utilizar o mesmo botão “Empresa” para um usuário cadastrado como “Administrador”, a ferramenta apresenta tela para manter os dados das empresas, com funcionamento semelhante ao cadastro de usuários e salas.

Cada membro da equipe deve associar-se a empresa criada anteriormente. Utilizando o botão “Associar-se”, a ferramenta apresenta tabela com as empresas cadastradas para a sala definida no usuário corrente, conforme figura 27. Deve-se selecionar a empresa e salvar.

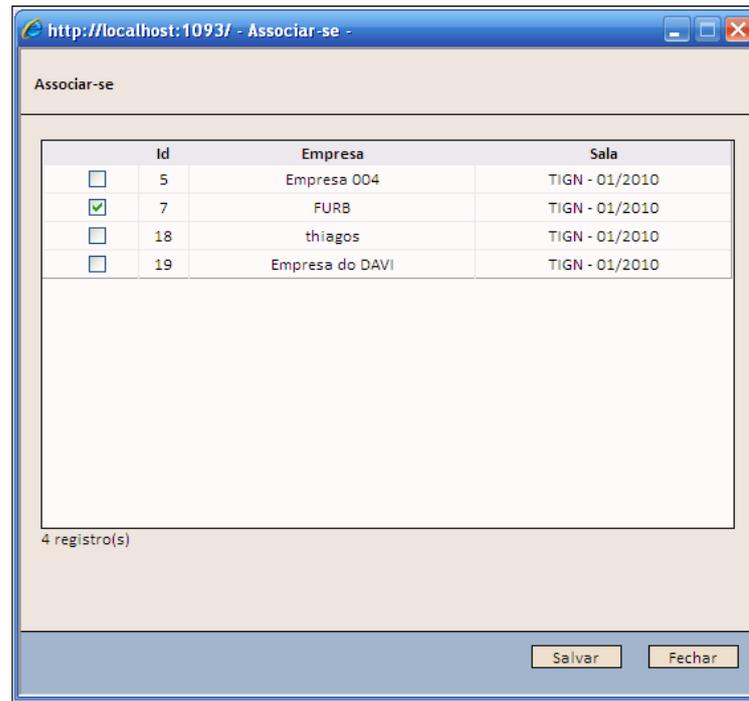


Figura 27 – Tela para associar-se a uma empresa

3.3.2.4 – Imagem mostrando um *Data Warehouse*

Ao utilizar a opção “Cubo de decisão”, a ferramenta apresenta tela com figura ilustrando o funcionamento de um DW e menu para “Unificar” os dados, “Povoar” as tabelas ou “Visualizar” os dados processados no cubo, conforme figura 28.



Figura 28 – Tela com figura ilustrando um DW e menu

Ao passar o *mouse* sobre uma parte da imagem, é informada uma explicação previamente cadastrada pelo administrador da ferramenta, conforme figura 29.

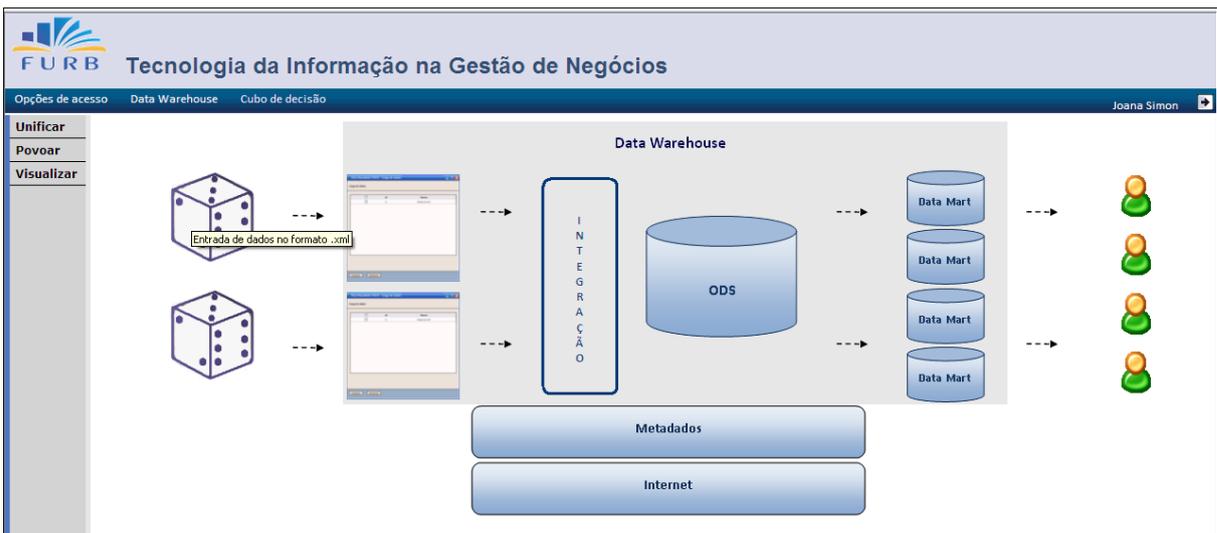


Figura 29 – Tela com visualização da explicação cadastrada para o dado da imagem

Para realizar o cadastro da explicação, um administrador da ferramenta deve clicar sobre a parte da imagem em que se deseja cadastrar a explicação e informar a explicação, conforme figura 30.

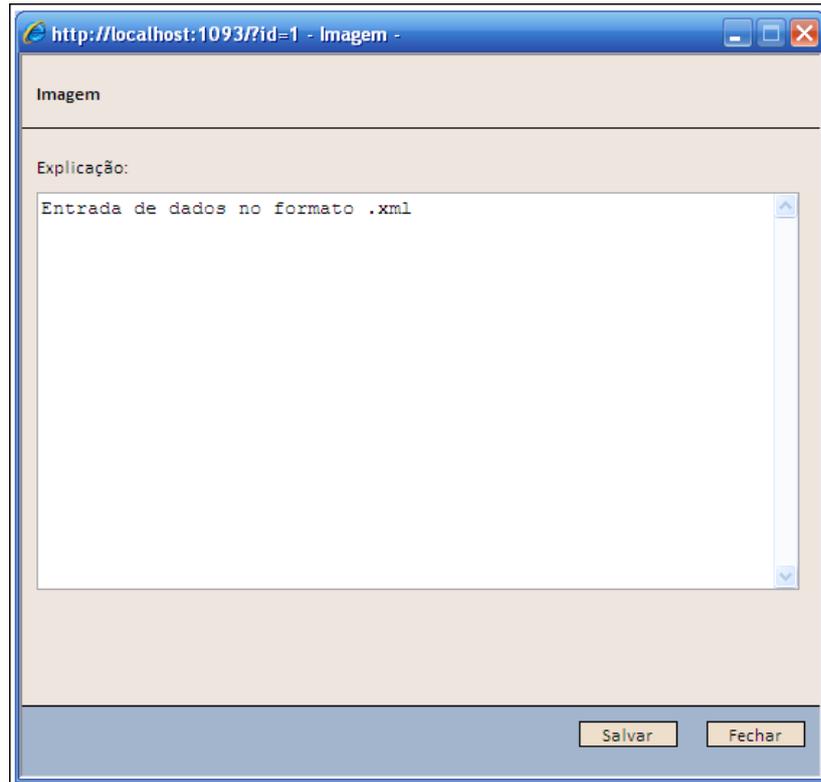


Figura 30 – Tela do cadastro de explicação da imagem

3.3.2.5 Unificar dados

Conforme atividade já desenvolvida em sala com os acadêmicos do primeiro semestre de 2010 na disciplina de TIGN, após explicar o funcionamento do esquema estrela, foram informados os registros existentes no banco e passada a estrutura de arquivo XML que as equipes deveriam seguir para a criação de registros. Solicitou-se que as mesmas criassem registros para que pudessem unificar e povoar as tabelas.

Ao utilizar a opção “Unificar”, a ferramenta apresenta tela onde se deve informar o arquivo que será unificado, conforme figura 31.

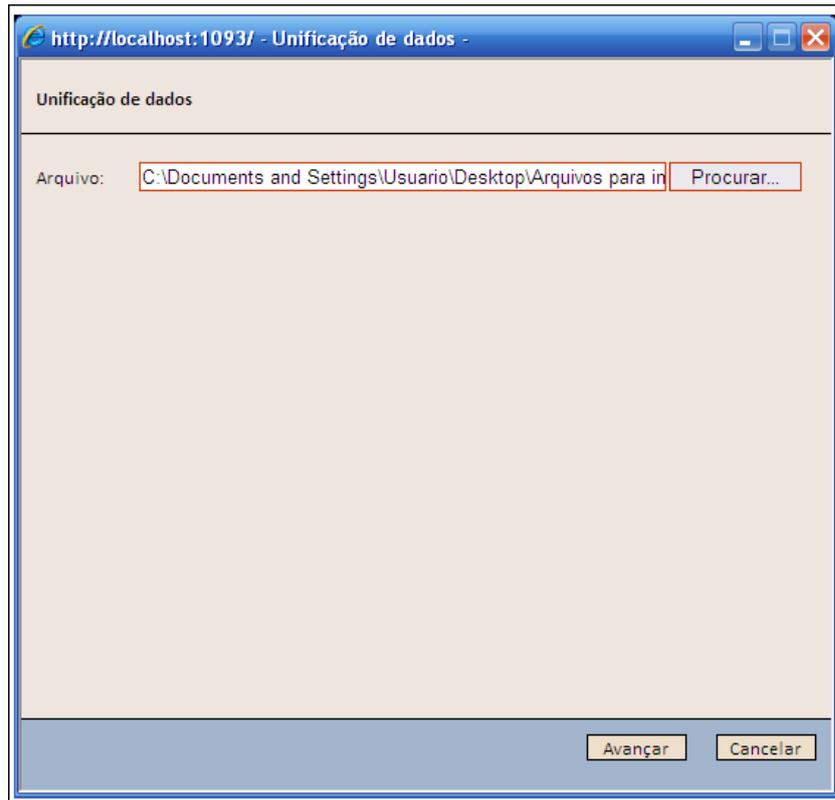


Figura 31 – Tela para informar o arquivo que será unificado

Informado o arquivo que será unificado, utiliza-se a opção “Avançar”. É apresentada a tabela com as informações que serão unificadas no arquivo, conforme figura 32.

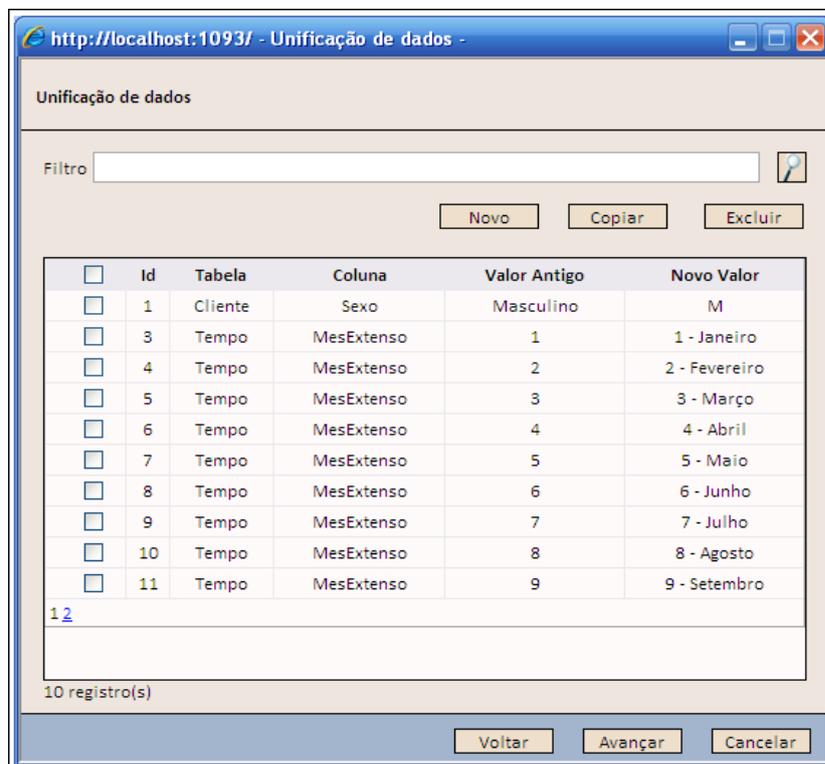
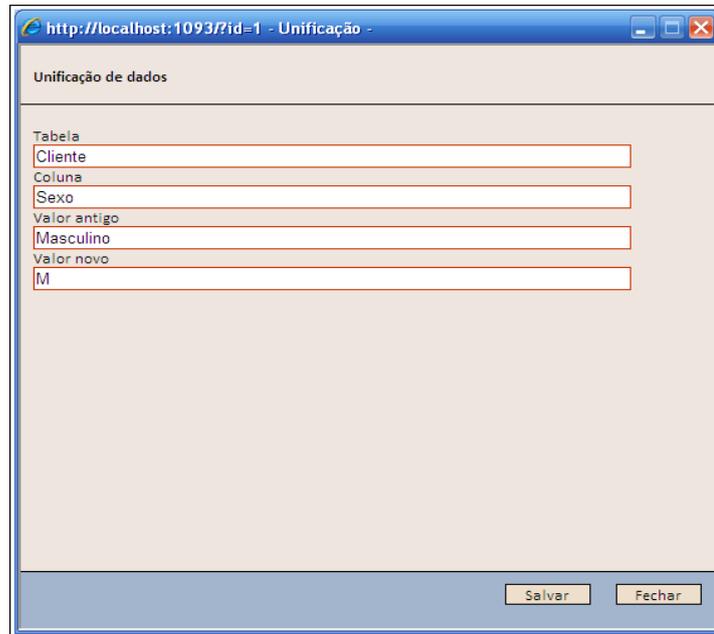


Figura 32 – Tela com os valores que serão unificados

Selecionando um registro, pode-se, através do botão “Copiar”, gerar no banco um registro semelhante ao selecionado, para ser editado. Através do botão “Excluir”, pode-se deletar do banco o registro selecionado. A opção “Novo” permite cadastrar um valor para ser unificado. Como se observa na figura 33, é preciso informar a “Tabela” e a “Coluna” que se deseja alterar, o valor do arquivo que deve ser alterado e o valor para o qual será alterado.



Unificação de dados

Tabela
Cliente

Coluna
Sexo

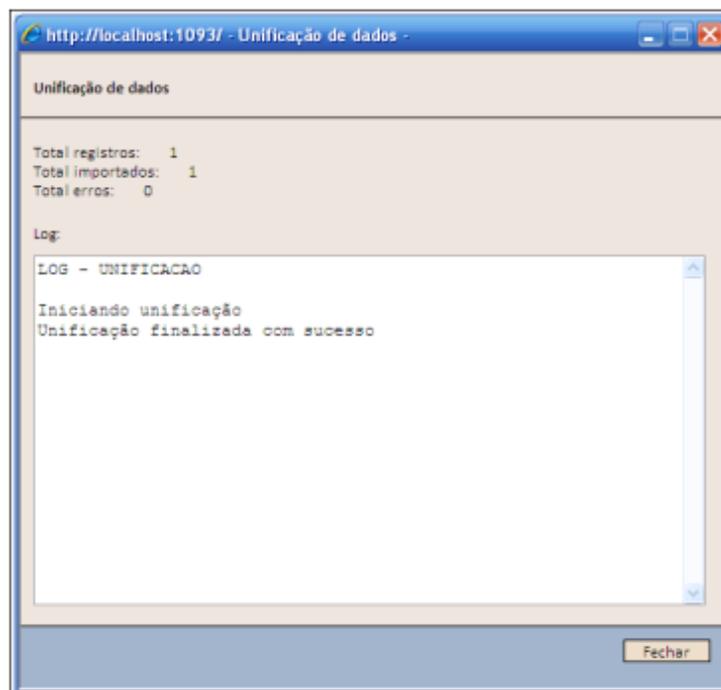
Valor antigo
Masculino

Valor novo
M

Salvar Fechar

Figura 33 – Tela para informar o valor que será unificado

Realizada a unificação, a ferramenta armazena o arquivo unificado na base de dados e apresenta *log* da unificação, conforme figura 34.



Unificação de dados

Total registros: 1
Total importados: 1
Total erros: 0

Log:

LOG - UNIFICACAO

Iniciando unificação
Unificação finalizada com sucesso

Fechar

Figura 34 – Tela com o *log* da unificação

Para montar a tabela da figura 32 foi utilizado o componente asp “GridView”, em que foi definido *layout*, os métodos utilizados para realizar a manutenção dos registros da tabela e os campos que serão listados em cada coluna, conforme mostra o quadro 5.

```

<asp:GridView CssClass="grid" ID="gridViewUnificacao" runat="server"
  Font-Names="calibri,verdana"
  Font-Size="10pt"
  AutoGenerateColumns="False"
  onrowdatabound="gridViewUnificacao_RowDataBound" AllowPaging="True"
  DataSourceID="ObjectDataSource1" EnableViewState="False"
  ondatabound="gridViewUnificacao_DataBound" >
  <PagerSettings Mode="NumericFirstLast" />
  <RowStyle HorizontalAlign="Center" />
  <Columns>
    <asp:TemplateField>
      <HeaderTemplate>
        <input id="chkAllItems" type="checkbox"
          onclick="CheckAllDataGridCheckBoxes(document.forms[0].chkAllItems.checked) " />
      </HeaderTemplate>
      <ItemTemplate>
        <asp:CheckBox id="chkItemChecked" runat="server"></asp:CheckBox>
      </ItemTemplate>
    </asp:TemplateField>
    <asp:BoundField DataField="Id" HeaderText="Id" InsertVisible="False"
      ReadOnly="True" SortExpression="Id" Visible="True" />
    <asp:BoundField DataField="Tabela" HeaderText="Tabela"
      SortExpression="Tabela" />
    <asp:BoundField DataField="Coluna" HeaderText="Coluna"
      SortExpression="Coluna" />
    <asp:BoundField DataField="ValorAntigo" HeaderText="Valor Antigo"
      SortExpression="ValorAntigo" />
    <asp:BoundField DataField="ValorNovo" HeaderText="Novo Valor"
      SortExpression="ValorNovo" />
  </Columns>
  <HeaderStyle BackColor="#EBE9ED" HorizontalAlign="Center" />
</asp:GridView>

```

Quadro 5 – Componente utilizado para montar a tabela com os valores que serão unificados

A figura 35 mostra onde é definido o método que retorna os dados que serão carregados na *grid*.

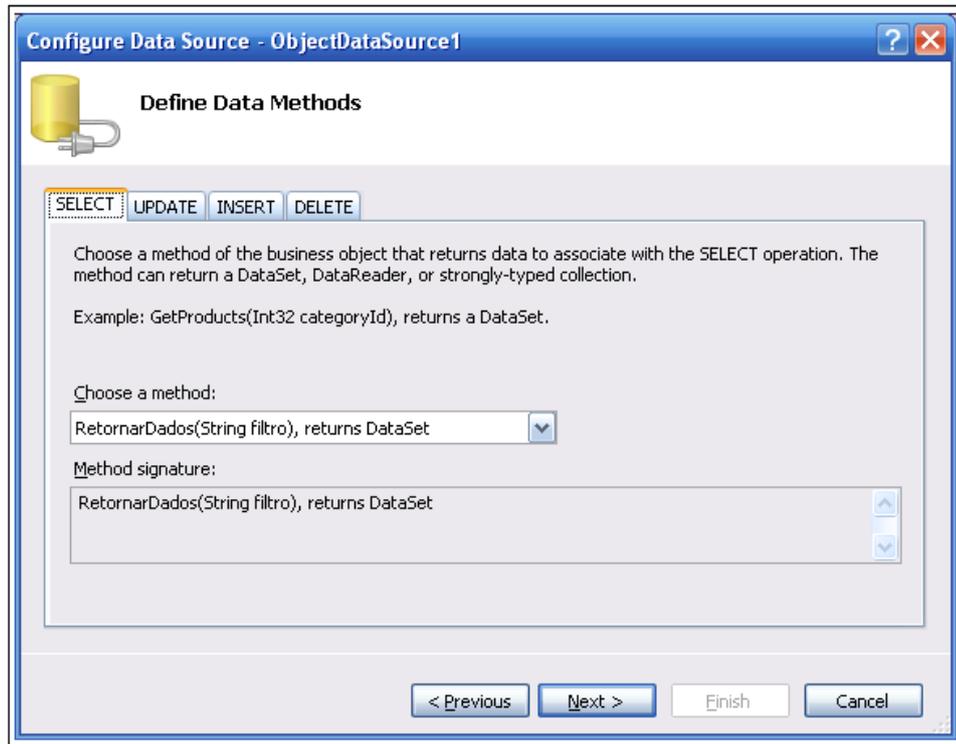


Figura 35 – Definindo método para retornar dados na tabela

No quadro 6, pode-se observar o código fonte utilizado para retornar os valores que serão carregados na *grid*. É feito um *select* na tabela de “Unificacao” pegando os campos que serão mostrados em cada coluna e, caso o usuário tenha realizado um filtro, filtrando pelo valor informado, se uma das colunas contiver o valor informado o registro é retornado na *grid* dos valores que serão unificados.

```

public static DataSet RetornarDados(string filtro)
{
    SQLMaker.SelectMaker selectUnificacao = new SQLMaker.SelectMaker("UNIFICACAO");
    selectUnificacao.AddCampo("Id");
    selectUnificacao.AddCampo("Tabela");
    selectUnificacao.AddCampo("Coluna");
    selectUnificacao.AddCampo("ValorAntigo");
    selectUnificacao.AddCampo("ValorNovo");
    selectUnificacao.AddCampo("Selecacao");
    if (filtro != null)
    {
        selectUnificacao.AddFiltroWhere("Id", SQLMaker.SelectMaker.OperadorFiltro.Like, filtro);
        selectUnificacao.AddFiltroOr("Tabela", SQLMaker.SelectMaker.OperadorFiltro.Like, filtro);
        selectUnificacao.AddFiltroOr("Coluna", SQLMaker.SelectMaker.OperadorFiltro.Like, filtro);
        selectUnificacao.AddFiltroOr("ValorAntigo", SQLMaker.SelectMaker.OperadorFiltro.Like, filtro);
        selectUnificacao.AddFiltroOr("ValorNovo", SQLMaker.SelectMaker.OperadorFiltro.Like, filtro);
    }
    return selectUnificacao.ExecutaDataSet("ODS");
}

```

Quadro 6 – Código fonte utilizado para carregar os valores que serão unificados

No quadro 7, pode-se observar que o arquivo que será unificado é percorrido. É feito um *select* na tabela de “Unificacao”, onde ficam armazenadas as tabelas com as respectivas

colunas que precisam ter o valor unificado, o valor antigo e o valor para o qual deve ser alterado. Caso, na tabela de “Unificacao”, esteja definido que o valor informado no arquivo para determinada coluna da tabela deve ser alterado, é feita a unificação desse valor. Isso ocorre quando nessa tabela há um registro com a mesma “Tabela”, “Campo” e “Valor antigo” presentes no arquivo que será unificado.

Ao finalizar a unificação um arquivo XML unificado é salvo no banco, ficando disponível para povoar as tabelas.

```

public Log Unificar()
{
    Log log = new Log("UNIFICACAO");

    log.Conteudo.AppendLine("Iniciando unificação");
    log.TotalRegistros = 1;
    log.TotalImportados = 0;
    log.TotalErros = 0;
    try
    {
        foreach (DataTable tabela in arquivo.Conteudo().Tables)
        {
            foreach (DataColumn coluna in tabela.Columns)
            {
                SQLMaker.SelectMaker selectUnificacao = new SQLMaker.SelectMaker("UNIFICACAO");
                selectUnificacao.AddCampo("Tabela");
                selectUnificacao.AddCampo("Coluna");
                selectUnificacao.AddCampo("ValorAntigo");
                selectUnificacao.AddCampo("ValorNovo");
                selectUnificacao.AddFiltroWhere("Tabela", SQLMaker.SelectMaker.OperadorFiltro.Igual, tabela.TableName);
                selectUnificacao.AddFiltroAnd("Coluna", SQLMaker.SelectMaker.OperadorFiltro.Igual, coluna.ColumnName);
                selectUnificacao.Executa("ODS");

                while (selectUnificacao.Next())
                {
                    foreach (DataRow linha in tabela.Rows)
                    {
                        if (linha.Field<string>(coluna.ColumnName) == selectUnificacao.GetCampoString("ValorAntigo"))
                            linha.SetField<string>(coluna.ColumnName, selectUnificacao.GetCampoString("ValorNovo"));
                    }
                }
            }
        }
        ArquivoUnificacao arquivoUnificacao = new ArquivoUnificacao(-1);
        arquivoUnificacao.Arquivo = arquivo.Conteudo().GetXml();
        arquivoUnificacao.Nome = arquivo.GetNome();
        arquivoUnificacao.Unificado = 1;
        arquivoUnificacao.Data = DateTime.Now;
        arquivoUnificacao.Salvar();

        log.TotalImportados++;
        log.Conteudo.AppendLine("Unificação finalizada com sucesso");
    }
    catch (Exception e)
    {
        log.TotalErros++;
        log.Conteudo.AppendLine("Erro ao unificar arquivo. Mensagem: " + e.Message);
    }

    return log;
}

```

Quadro 7 – Código fonte para unificar os registros

3.3.2.6 Povoar tabelas

Ao utilizar a opção “Povoar” disponível no menu, a ferramenta apresenta tela com os

registros já unificados disponíveis na base, conforme figura 36. Para povoar as tabelas o usuário logado na ferramenta precisa estar vinculado a uma empresa.

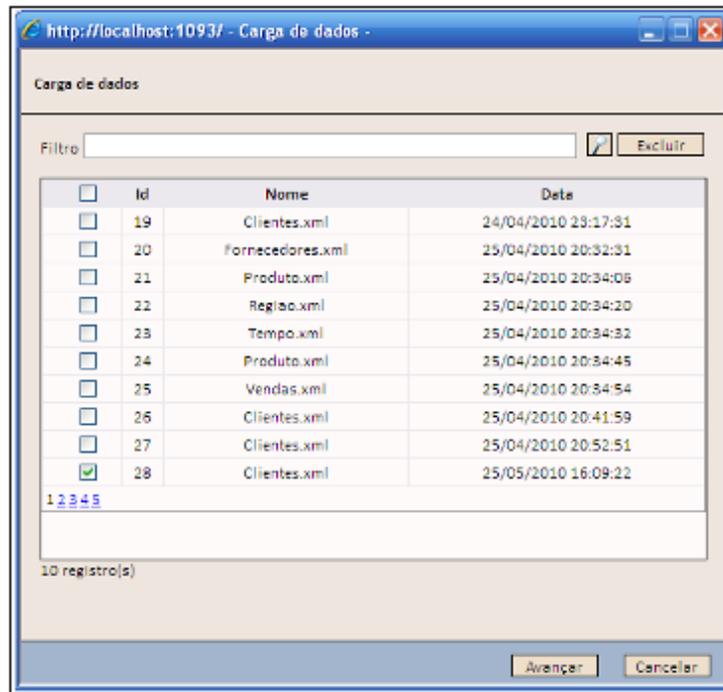


Figura 36 – Tela exibindo os arquivos unificados disponíveis na base

O usuário deve selecionar o(s) registro(s) com os dados que deseja povoar, conforme figura 36, e utilizar a opção “Avançar”. A ferramenta irá povoar as tabelas com o(s) registro(s) do(s) arquivo(s) selecionado(s) e é apresentado o *log* do processo para o usuário, conforme figura 37.

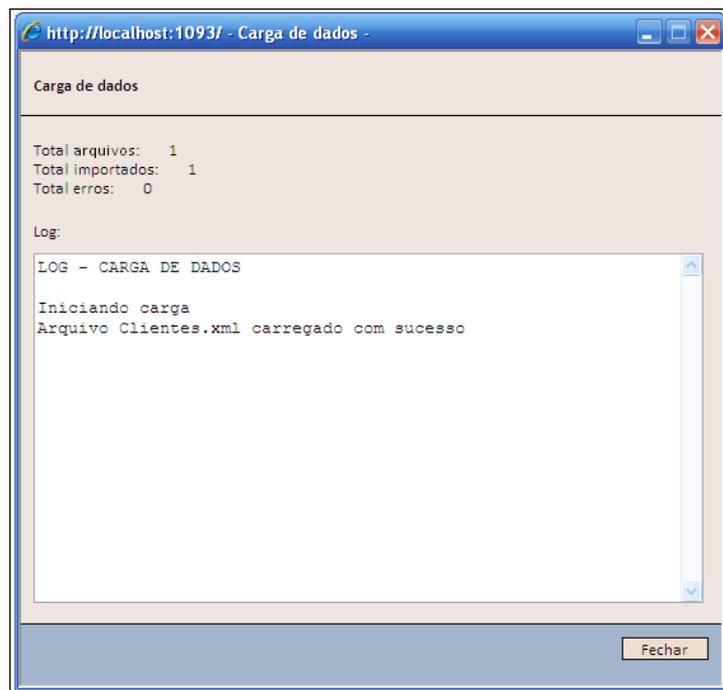


Figura 37 – Tela exibindo o *log* após povoar a tabela com os registros

Através da unificação e do povoamento de tabelas, o objetivo específico, que era permitir a importação de dados, foi atendido.

O quadro 8 mostra que para povoar as tabelas, a rotina percorre os arquivos já unificados selecionados pelo usuário. Para cada arquivo, a ferramenta percorre as tabelas do mesmo e insere os registros correspondentes.

```

public Log Carregar(int idUsuario)
{
    Log log = new Log("CARGA DE DADOS");

    log.Conteudo.AppendLine("Iniciando carga");
    log.TotalRegistros = listaArquivos.Count;
    log.TotalImportados = 0;
    log.TotalErros = 0;

    foreach (ArquivoUnificacao arquivo in listaArquivos)
    {
        try
        {
            DataSet dsArquivo = new DataSet();

            MemoryStream streamArquivo = new MemoryStream(System.Text.Encoding.ASCII.GetBytes(arquivo.Arquivo));

            dsArquivo.ReadXml(streamArquivo);
            foreach (DataTable tabela in dsArquivo.Tables)
            {
                if (tabela.TableName != "Tabelas")
                {
                    foreach (DataRow linha in tabela.Rows)
                    {
                        if (tabela.TableName == "Vendas")
                        {
                            ProcessarCubo();

                            Negocio.Usuario usuario = new Negocio.Usuario(idUsuario);

                            SQLMaker.SelectMaker selectVendas = new SQLMaker.SelectMaker("VENDAS");
                            selectVendas.AddCampo("ValorCusto");
                            selectVendas.AddCampo("QtdVendida");
                            selectVendas.AddCampo("ValorVendido");
                            selectVendas.AddCampo("ValorLucro");

                            selectVendas.AddFiltroWhere("IdRegiao", SQLMaker.SelectMaker.OperadorFiltro.Igual,
                                linha["IdRegiao"].ToString());

                            selectVendas.AddFiltroAnd("IdCliente", SQLMaker.SelectMaker.OperadorFiltro.Igual,
                                linha["IdCliente"].ToString());

                            selectVendas.AddFiltroAnd("IdFornecedor", SQLMaker.SelectMaker.OperadorFiltro.Igual,
                                linha["IdFornecedor"].ToString());

                            selectVendas.AddFiltroAnd("IdProduto", SQLMaker.SelectMaker.OperadorFiltro.Igual,
                                linha["IdProduto"].ToString());

                            selectVendas.AddFiltroAnd("IdTempo", SQLMaker.SelectMaker.OperadorFiltro.Igual,
                                linha["IdTempo"].ToString());

                            selectVendas.AddFiltroAnd("IdEmpresa", SQLMaker.SelectMaker.OperadorFiltro.Igual,
                                usuario.Empresa.ToString());

                            selectVendas.Executa();
                            if (selectVendas.Next())
                            {
                                SQLMaker.UpdateMaker updateVendas = new SQLMaker.UpdateMaker("VENDAS");

                                updateVendas.AddCampo("ValorCusto");
                                updateVendas.AddValor((selectVendas.GetCampoDouble("ValorCusto") +
                                    Convert.ToDouble(linha["ValorCusto"].ToString())).ToString());

                                updateVendas.AddCampo("QtdVendida");
                                updateVendas.AddValor((selectVendas.GetCampoDouble("QtdVendida") +
                                    Convert.ToDouble(linha["QtdVendida"].ToString())).ToString());

                                updateVendas.AddCampo("ValorVendido");
                                updateVendas.AddValor((selectVendas.GetCampoDouble("ValorVendido") +
                                    Convert.ToDouble(linha["ValorVendido"].ToString())).ToString());

                                updateVendas.AddCampo("ValorLucro");
                                updateVendas.AddValor((selectVendas.GetCampoDouble("ValorLucro") +
                                    (Convert.ToDouble(linha["ValorVendido"].ToString()) -
                                    Convert.ToDouble(linha["ValorCusto"].ToString()))).ToString());

                                updateVendas.AddFiltroWhere("IdRegiao", SQLMaker.UpdateMaker.OperadorFiltro.Igual,
                                    linha["IdRegiao"].ToString());
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

updateVendas.AddFiltroAnd("IdCliente", SQLMaker.UpdateMaker.OperadorFiltro.Igual,
    linha["IdCliente"].ToString());

updateVendas.AddFiltroAnd("IdFornecedor", SQLMaker.UpdateMaker.OperadorFiltro.Igual,
    linha["IdFornecedor"].ToString());

updateVendas.AddFiltroAnd("IdProduto", SQLMaker.UpdateMaker.OperadorFiltro.Igual,
    linha["IdProduto"].ToString());

updateVendas.AddFiltroAnd("IdTempo", SQLMaker.UpdateMaker.OperadorFiltro.Igual,
    linha["IdTempo"].ToString());

updateVendas.AddFiltroAnd("IdEmpresa", SQLMaker.UpdateMaker.OperadorFiltro.Igual,
    usuario.Empresa.ToString());

updateVendas.Executa();
}
else
{
    SQLMaker.InsertMaker insertVendas = new SQLMaker.InsertMaker("VENDAS");
    foreach (DataColumn coluna in tabela.Columns)
    {
        insertVendas.AddCampo(coluna.ColumnName);
        insertVendas.AddValor(linha[coluna].ToString());
    }

    insertVendas.AddCampo("ValorLucro");
    insertVendas.AddValor((Convert.ToDouble(linha["ValorVendido"].ToString()) -
        Convert.ToDouble(linha["ValorCusto"].ToString())));

    insertVendas.AddCampo("IdEmpresa");
    insertVendas.AddValor(usuario.Empresa.ToString());
    insertVendas.Executa();
}
}
else
{
    SQLMaker.InsertMaker insertTabela = new SQLMaker.InsertMaker(tabela.TableName);
    foreach (DataColumn coluna in tabela.Columns)
    {
        insertTabela.AddCampo(coluna.ColumnName);
        insertTabela.AddValor(linha[coluna].ToString());
    }
    insertTabela.Executa();
}
}
}
}
log.TotalImportados++;
log.Conteudo.AppendLine("Arquivo " + arquivo.Nome + " carregado com sucesso");
}
catch (Exception e)
{
    log.TotalErros++;
    log.Conteudo.AppendLine("Erro ao carregar arquivo " + arquivo.Nome +
        ". Mensagem: " + e.Message + "CargaArquivoUnificacao - Carregar");
}
}
ProcessarCubo();

return log;

```

Quadro 8 – Código fonte para povoar as tabelas

Seguindo o exemplo do quadro 9, para a tabela “Cliente” a ferramenta irá inserir três registros e para a tabela “Fornecedor”, um registro, utilizando para atribuição dos valores de cada registro os valores das *tags* correspondentes às colunas da tabela no BD.

```

<?xml version="1.0" encoding="utf-8" ?>
<Tabelas>
  <Cliente>
    <Id>14</Id>
    <Nome>Colcci</Nome>
    <CNPJ>93343218778222</CNPJ>
    <Sexo></Sexo>
    <DataNascimento></DataNascimento>
  </Cliente>
  <Cliente>
    <Id>15</Id>
    <Nome>Renner</Nome>
    <CNPJ>93343218778211</CNPJ>
    <Sexo></Sexo>
    <DataNascimento></DataNascimento>
  </Cliente>
  <Cliente>
    <Id>16</Id>
    <Nome>Marisa</Nome>
    <CNPJ>61334543218778</CNPJ>
    <Sexo></Sexo>
    <DataNascimento></DataNascimento>
  </Cliente>
  <Fornecedor>
    <Id>2</Id>
    <Nome>willi aviamentos</Nome>
    <CNPJ>83646984002244</CNPJ>
  </Fornecedor>
</Tabelas>

```

Quadro 9 – Exemplo de arquivo para povoar as tabelas

Ao realizar a carga de uma venda, a ferramenta verifica se há uma venda no BD cadastrada com o mesmo cliente, fornecedor, produto, tempo, região e empresa. Caso já tenha, acrescenta os valores dessa venda na venda já cadastrada. Caso não tenha, cria um registro na base para esta venda.

Ao realizar a carga já é feito também o processamento do cubo de dados conforme quadro 10.

```

private void ProcessarCubo()
{
    AdomdConnection cn = new AdomdConnection("Provider=MSOLAP;Data Source=localhost;Initial Catalog=OLAP");
    cn.Open();

    AdomdCommand cmd = cn.CreateCommand();

    cmd.CommandText = "<Batch xmlns='http://schemas.microsoft.com/analysisservices/2003/engine'><Parallel><Process> <Object>" +
        "<DatabaseID>Olap</DatabaseID>" +
        "<CubeID>Data Source View</CubeID> " +
        "</Object><Type>ProcessFull</Type><WriteBackTableCreation>UseExisting</WriteBackTableCreation> </Process> </Parallel> </Batch>";
    try
    {
        cmd.ExecuteNonQuery();
    }
    finally
    {
        cn.Close();
    }
}

```

Quadro 10 – Código fonte para processar o cubo

3.3.2.7 Visualizar os dados

A opção “Visualizar” do menu permite ao usuário visualizar os dados através do cubo de decisão, conforme figura 38. Através dessa tela o objetivo específico, que era possibilitar ao gestor melhor utilização das informações já armazenadas, foi atendido.

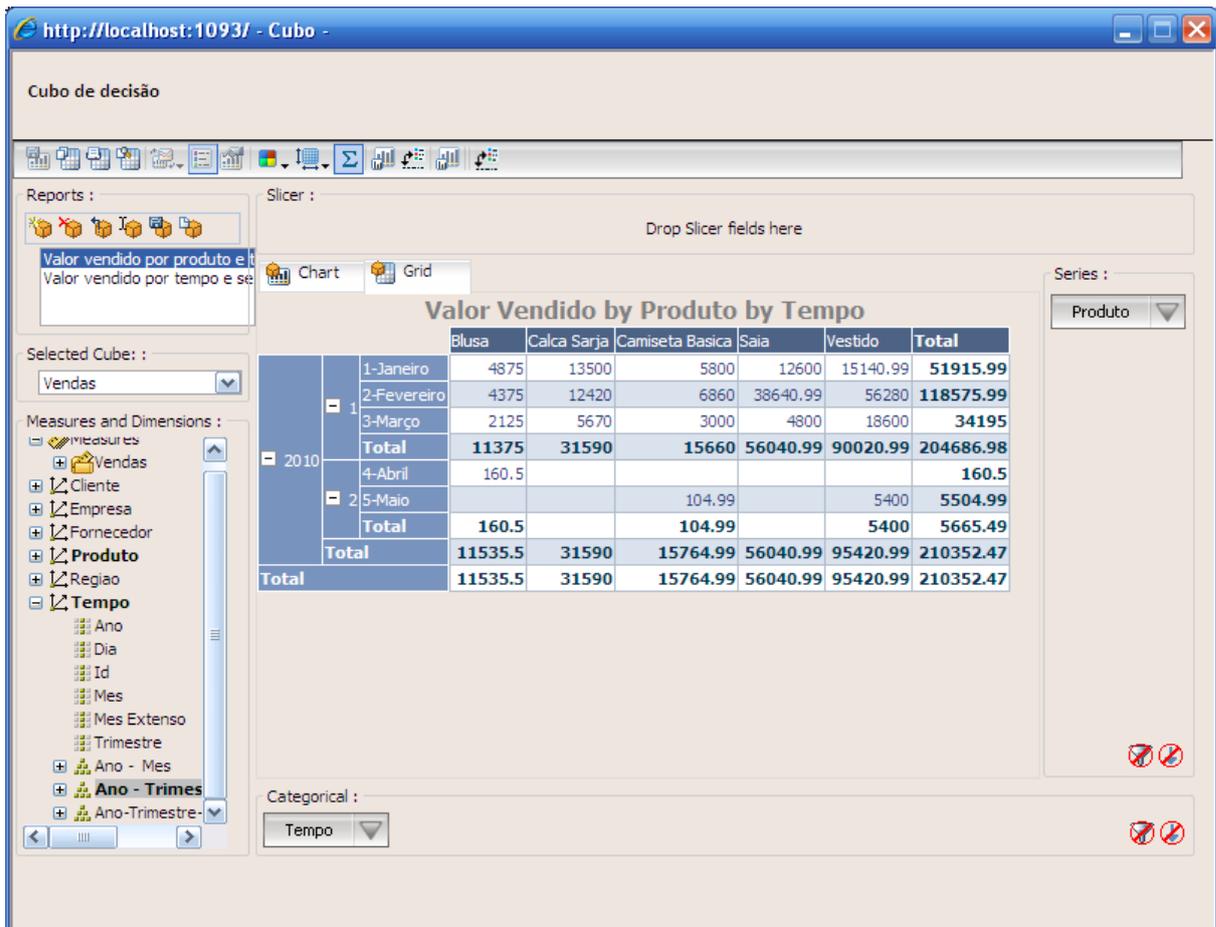


Figura 38 – Cubo de decisão

A figura 39 apresenta a barra de ferramentas, com funções como salvar ou copiar a imagem do gráfico, visualizar impressão ou imprimir o gráfico, selecionar o tipo de gráfico, alterar propriedades do gráfico, alterar *layout* do gráfico e da tabela, alterar orientação da tabela, mostrar total na tabela, visualizar apenas a barra de ferramentas, o gráfico e a tabela e inverter posição dos dados.



Figura 39 – Barra de ferramentas

A figura 40 mostra o controle de relatórios, permite organizar e gerir as *views* dos dados. Pode-se adicionar, remover, renomear, salvar, carregar um relatório previamente salvo ou visualizar relatórios pré-definidos por rotina.

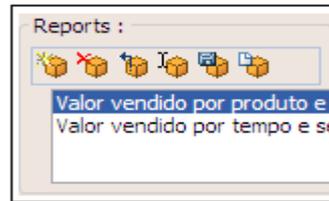


Figura 40 – Controle de relatórios

A figura 41 apresenta o seletor de cubo, caso em trabalhos futuros sejam criados outros cubos seus dados também poderão ser visualizados nesta ferramenta.

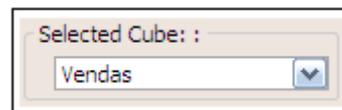


Figura 41 – Seletor de cubo

A figura 42 apresenta as medidas e dimensões disponíveis, para visualizar os dados basta clicar nos atributos da tabela fato e das dimensões que se deseja visualizar e arrastar para onde se pretende visualizar. Esta tela atendeu ao objetivo específico que era identificar as dimensões.



Figura 42 – Medidas e dimensões

Através do “Slicer”, é possível fatiar o cubo, por exemplo, visualizando apenas os dados de determinada empresa, conforme figura 43. As abas de gráfico e tabela permitem visualizar o gráfico ou a tabela.

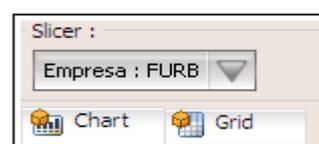


Figura 43 – Slicer e abas com gráfico e tabela

A figura 44 apresenta o controle do cabeçalho das linhas e colunas. Arrastando medidas, dimensões ou atributos nesses espaços, a informação é mostrada, pode-se realizar filtros e definir ordenação dos valores. Caso tenha-se mais de um valor, os mesmos são empilhados conforme ordem em que aparecem em “Categorical” e “Series”, para alterar a ordem basta clicar e arrastar para a ordem desejada.

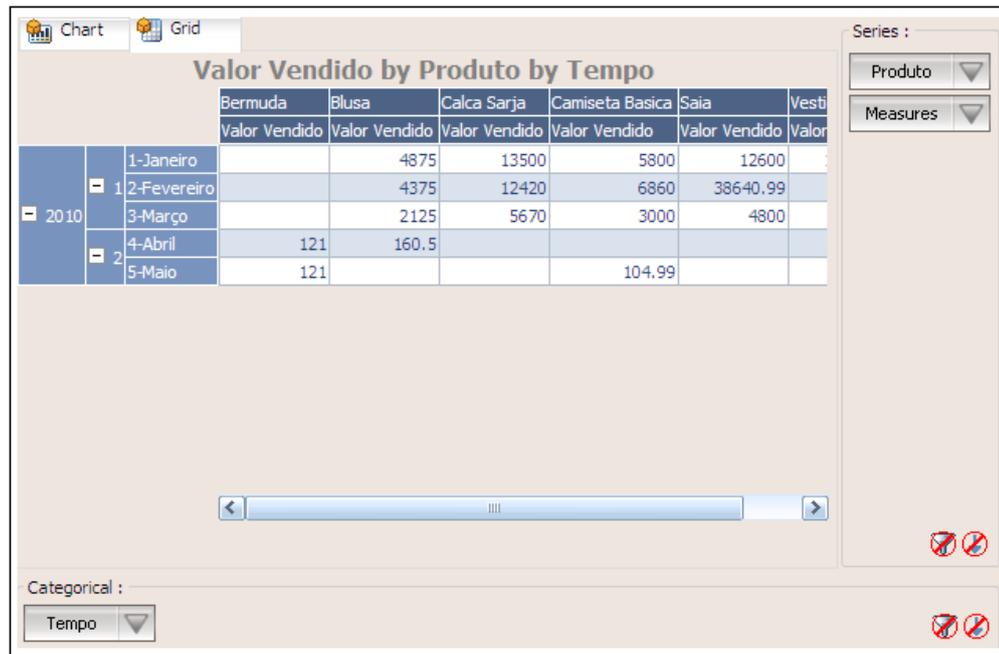


Figura 44 – Controle do cabeçalho de linhas e colunas

No quadro 11 apresenta-se o uso do componente “AdomdNetDataProvider” para fazer a ligação dos componentes de interface com o projeto do SSAS, e também o uso do componente “OlapClient” que disponibiliza os componentes de interface.

```
<DODPN:AdomdNetDataProvider ID="AdomdNetDataProvider1" runat="server"
  CommandTimeout="0" ConnectionString="Data Source=localhost; Provider=MSOLAP; Initial Catalog=OLAP;">
</DODPN:AdomdNetDataProvider>

<DOCWC:OlapClient ID="OlapClient1" runat="server" border="0" cellpadding="0"
  cellspacing="0" DataProviderID="AdomdNetDataProvider1"
  DimensionBrowserWidth="150px" Font-Names="Tahoma" Font-Size="8pt"
  height="492px" rules="none" SeriesBuilderWidth="100px"
  style="text-align: left;" width="780px">
  <ToolBarSettings>
    <Maximize Description="Switch to Simple View"></Maximize>
  </ToolBarSettings>
  <ToolBarStyle BackColor="White" BorderColor="" BorderWidth="" PageColor="" ToolbarStyle="Flat1"> </ToolBarStyle>
</DOCWC:OlapClient>
```

Quadro 11 – Componentes utilizados para implementar o cubo

3.4 RESULTADOS E DISCUSSÃO

O diferencial do presente trabalho em relação aos trabalhos correlatos é a possibilidade de utilização da ferramenta como material de apoio à disciplina de TIGN, o aplicativo desenvolvido em Casas (2009) é para apoio didático-pedagógico, mas apresentava *Balanced Scorecard* e não o cubo de decisão. Por ser uma ferramenta desenvolvida em plataforma *web*, também é de fácil acesso para os acadêmicos da disciplina, permitindo a realização de atividades à distância.

Em Zimmermann (2006), o foco era o desenvolvimento de uma ferramenta comercial com o intuito de auxiliar os executivos a extrair informações em um período de tempo menor e focado na tomada de decisão.

Kisner (2006) demonstrou os objetivos de um BI, as ferramentas que compõem um BI, os fatores críticos de sucesso de um BI, os pontos relevantes para obter sucesso em um projeto de um BI e os benefícios trazidos pelo mesmo. Strey (2002) implementou uma ferramenta comercial para realizar demonstrações dos valores de compras efetuadas pelos clientes, dos valores que a empresa tem a receber, informações sobre as vendas e demonstrar as informações para a possível melhoria dos processos.

Franco (2001) disponibilizou em seu sistema de gestão ambiental, consultas gráficas por meio de ferramentas através de relatórios e telas, mas não permitia acesso ao cubo de decisão através da internet. Casas (2009), apresentou um aplicativo utilizado como material de apoio didático-pedagógico, baseado no método do *Balanced Scorecard*, esse aplicativo permite a criação de perspectivas, de objetivos estratégicos, indicadores de desempenho e a visualização gráfica das informações através do mapa estratégico.

O quadro 12 apresenta as tecnologias utilizadas no presente trabalho e nos trabalhos correlatos que abordaram cubo de decisão.

	Presente trabalho	Fernandes e Kantorski	Zimmermann	Strey	Franco
Layout de páginas	CSS	Sitemesh	Delphi	Delphi	Delphi
BD	SQLServer	PostgreSQL	Firebird (SGBD Interbase)	Interbase	Oracle

Manipulação do BD	SQLMaker.dll	Hibernate	Biblioteca Delphi	Biblioteca Delphi	Biblioteca Delphi
Linguagem	C#	Java	Delphi	Delphi	Delphi
Servidor OLAP	SQL Server Analysis Services	Mondrian	Decision Cube	Decision Cube	Decision Cube
Consultas OLAP	Dundas	JPivot	Decision Cube	Decision Cube	Decision Cube

Quadro 12 – Comparativo das ferramentas

A ferramenta de ensino em DW foi apresentada para a turma de Tecnologia de Informação na Gestão de Negócios do primeiro semestre de 2010, com vinte alunos matriculados. Foi desenvolvida atividade em sala com os acadêmicos, utilizando-se a ferramenta. Aplicou-se um questionário de avaliação da ferramenta com esses alunos, conforme apresentado no apêndice C, dos quais onze foram os respondentes. A maior parte dos acadêmicos que realizaram a avaliação está no sexto (36,36%) ou oitavo (36,36%) semestre, demonstrando assim, que os alunos provavelmente possuem uma boa percepção do assunto ao qual foi questionado.

Na figura 45, apresenta-se o gráfico mostrando que na avaliação referente à hierarquia das informações, dos onze alunos respondentes, oito consideraram que a ferramenta possui as informações dispostas seguindo uma seqüência lógica, respeitando a hierarquia das informações. Para dois deles atendeu além do esperado. Apenas um afirmou atender de forma parcial.



Figura 45 – Gráfico de avaliação da hierarquia das informações

A figura 46 mostra que ao procurar uma funcionalidade, dez dos onze alunos encontraram a mesma com a facilidade esperada. Apenas um acadêmico informou que a

ferramenta atendeu de forma parcial a este critério.

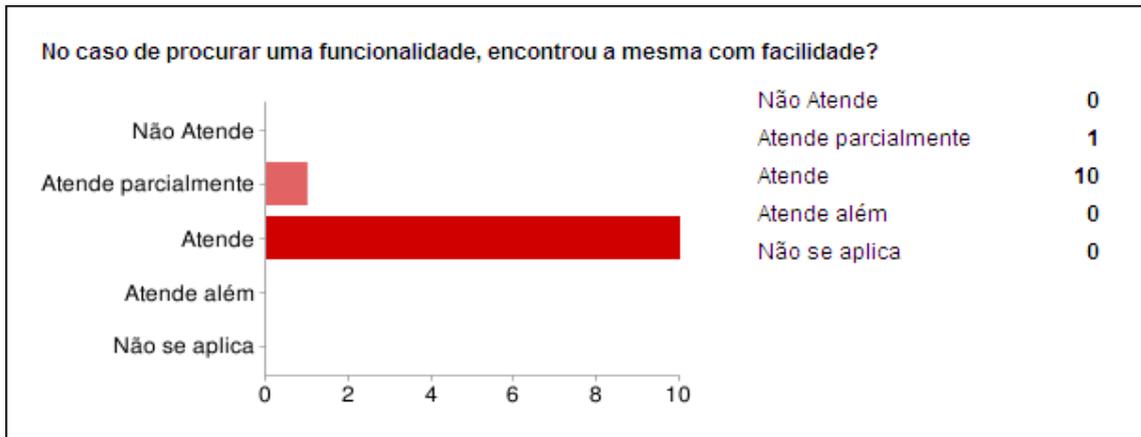


Figura 46 – Gráfico de avaliação da facilidade em encontrar as funcionalidades disponíveis

A figura 47 apresenta que dez alunos dos onze questionados consideraram que, ao não preencher um campo obrigatório, a ferramenta apresenta mensagem alertando o usuário, atendendo ao esperado. Apenas um acadêmico informou que a ferramenta atendeu de forma parcial ao esperado.

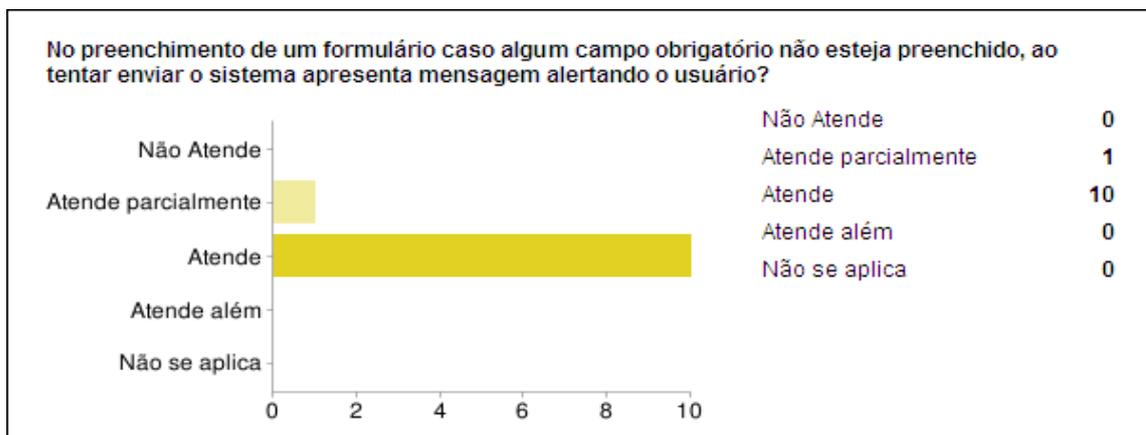


Figura 47 – Gráfico de avaliação de avisos ao tentar enviar formulário sem preencher um ou mais campo(s) obrigatório(s)

A figura 48 mostra que ao apresentar mensagem solicitando preenchimento de campos obrigatórios, para nove acadêmicos os campos preenchidos corretamente tiveram seus valores mantidos atendendo ao esperado. Para um dos onze respondentes, a ferramenta atendeu de forma parcial e para outro, não atendeu ao esperado.

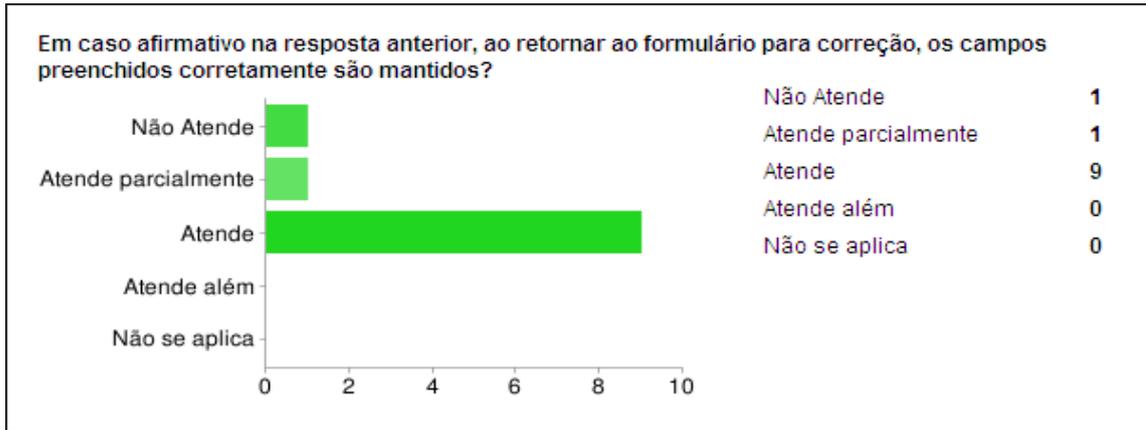


Figura 48 – Gráfico de avaliação referente à ao avisar existência de campos obrigatórios em branco, manter os valores preenchidos corretamente

Para seis dos onze respondentes, é fácil realizar a consulta de dados na ferramenta, atendendo além do esperado. Para dois deles, atende ao esperado e para três, atende de forma parcial, conforme figura 49.

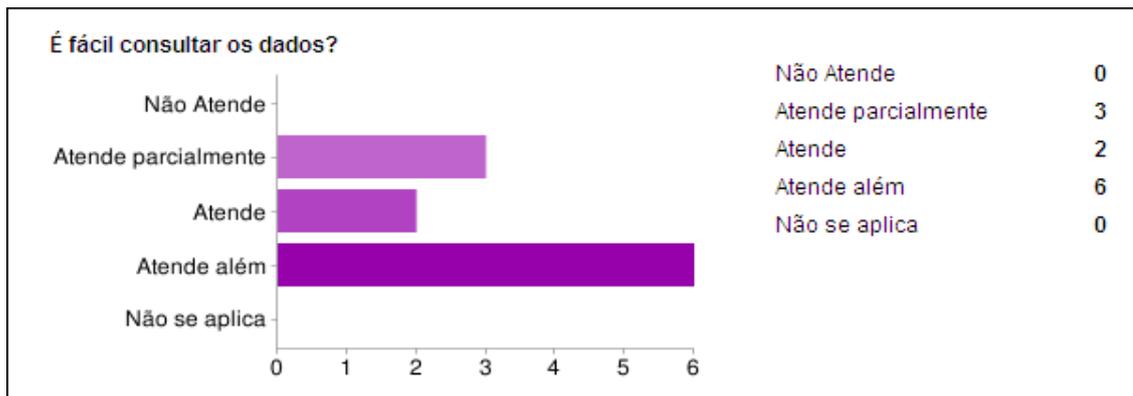


Figura 49 – Gráfico de avaliação da facilidade em consultar os dados

Para oito dos onze respondentes, ao ocorrer algum evento em que o usuário necessite ser informado, a mensagem apresentada é coerente, atendendo ao esperado. Para dois deles atendeu de forma parcial, conforme figura 50.



Figura 50 – Gráfico de avaliação da facilidade em consultar os dados

A figura 51 apresenta que para seis dos onze respondentes a ferramenta cumpre de forma objetiva a tarefa de complementar a explicação teórica sobre DW apresentada em sala, atendendo ao esperado. Para cinco acadêmicos atendeu além do esperado.

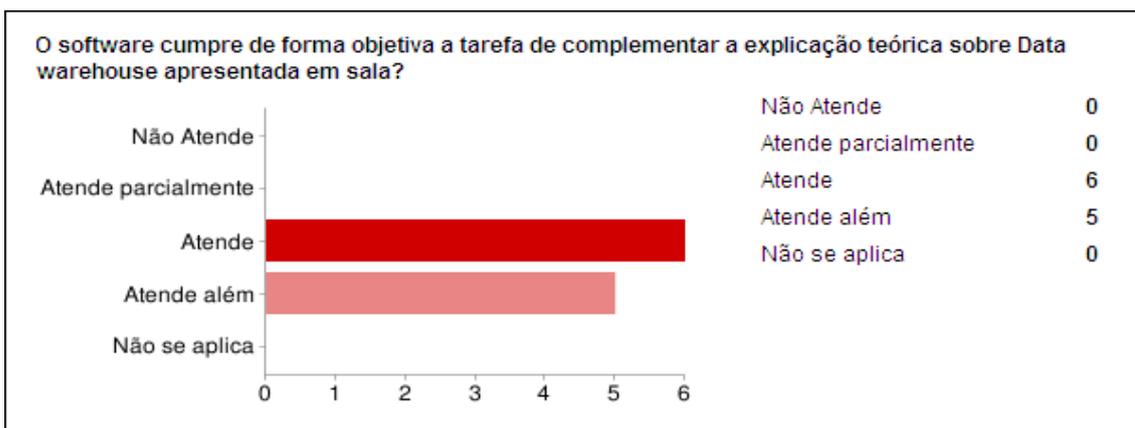


Figura 51 – Gráfico avaliando se a ferramenta cumpre de forma objetiva a tarefa de complementar a explicação teórica sobre DW apresentada em sala

A figura 52 mostra que de um a cinco, sendo um “nenhum” e cinco “altamente poluído”, para seis dos onze respondentes o nível de poluição visual do ambiente é um, para quatro deles é três e para apenas um dos respondentes o nível de poluição visual é quatro.

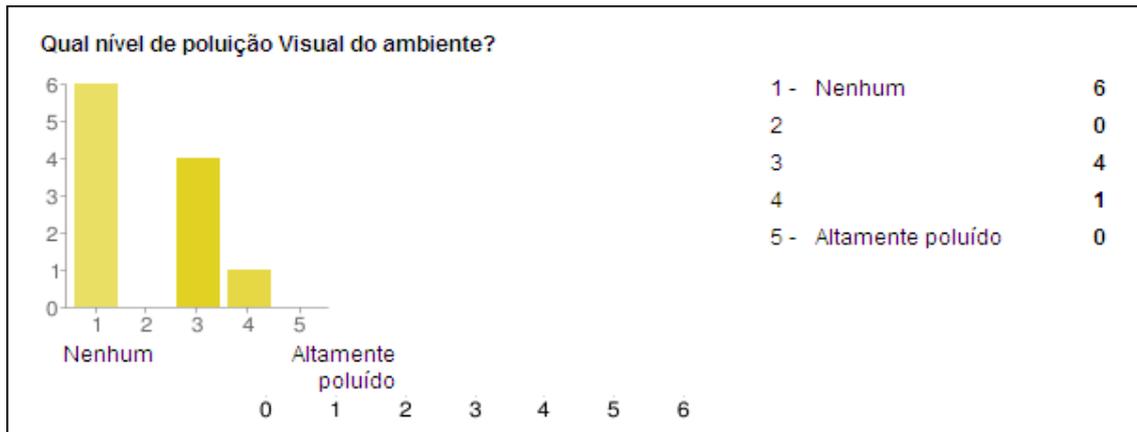


Figura 52 – Gráfico de avaliação do nível de poluição visual do ambiente

4 CONCLUSÕES

A utilização da ferramenta de ensino em *Data Warehouse* proporcionou aos acadêmicos de TIGN, após a teoria passada em sala de aula, maior facilidade em assimilar o conteúdo.

A ferramenta desenvolvida neste trabalho possibilitou também uma melhor percepção da importância, em uma organização, de uma ferramenta para o auxílio na interpretação dos dados armazenados para facilitar uma melhor gestão da empresa.

Os objetivos propostos foram atingidos com sucesso. Em relação aos objetivos específicos, foi desenvolvida uma ferramenta de Cubo de decisão de um BI. Através da ferramenta, podem-se identificar as dimensões. Os processos de unificação de dados e povoamento das tabelas permitem a importação de dados, que era um dos objetivos específicos. A ferramenta possibilitou ao gestor melhor utilização das informações, como material didático-pedagógico, para auxiliá-lo na disciplina. Também foi aplicada a ferramenta na disciplina de TIGN do curso de BSI ofertada pela FURB, em que os alunos criaram os arquivos, unificaram, povoaram as tabelas com o arquivo unificado e, utilizando o cubo de decisão, puderam visualizar esses dados através de tabelas e gráficos.

Entre as principais limitações da ferramenta está o fato de o cubo ser pré-definido. O usuário não cria de forma dinâmica a estrutura de tabelas fatos, medidas e dimensões. Os arquivos para a unificação de dados necessitam ser TXT ou XML seguindo estrutura previamente definida.

Este trabalho proporcionou aprendizado na área de gestão de negócios, em que os gestores necessitam de ferramentas para auxílio na tomada de decisão, ferramentas que facilitem a utilização do grande volume de dados armazenados. Com este trabalho houve aprendizado também de como utilizar o banco de dados SQLServer, o Framework .Net, o SSAS e a linguagem de programação C# não vistos na grade curricular atual. Com o auxílio dessas ferramentas obteve-se um cubo de decisão que pode ser disponibilizado na *web*, facilitando o acesso ao mesmo, o que não ocorre nos sistemas desenvolvidos por Zimmermann (2006) ou Strey (2002).

4.1 EXTENSÕES

Como extensão para este trabalho sugere-se:

- a) permitir a criação de cubos de forma dinâmica, permitindo apresentar outros exemplos além da área de vendas. O usuário poderia criar tabelas fato, medidas, dimensões e atributos para as dimensões, a fim de obter outros cubos;
- b) possibilitar a leitura de outros documentos como planilhas do Excel;
- c) implementar a análise contemplativa para permitir ao usuário alterar um único valor para que se possa determinar o seu impacto, implementar também a análise formalista para possibilitar alterações a múltiplas variáveis;
- d) implementar *Balanced Scorecard* para demonstrar na análise de valores, quais indicadores de desempenho estão melhores e quais as ações que poderão ser tomadas;
- e) incluir uma ferramenta de *Data Mining*. Esta ferramenta faria uma simulação a partir dos dados deste projeto apresentando as técnicas de associação, cluster e árvore de decisão;
- f) implementar ferramenta para realizar a análise de dados utilizando redes neurais;
- g) pode-se ainda acrescentar a geração de Metadados. Permitindo assim identificar a localização dos dados no DW, sua origem, seu significado e as transformações que ocorreram no processo *Extract, Transform and Load* (ETL).

REFERÊNCIAS BIBLIOGRÁFICAS

BARBIERI, C. **BI-Business Intelligence: modelagem & tecnologia**. Rio de Janeiro: Axcel Books, 2001.

CASAS, Jerusa C. D. **Desenvolvimento de um aplicativo de apoio didático pedagógico baseado em Balanced Scorecard**. 2009. 70 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

DALFOVO, O. **Plano de ensino-aprendizagem integral: ementa disciplina tecnologia da informação na gestão de negócio**. Blumenau, 2010. Disponível em: <http://www.inf.furb.br/~dalfovo/Bsi/planos_de_ensino-aprendizagem_si.html>. Acesso em: 10 jun. 2010.

DUNDAS. **Data Visualization**. Disponível em: <<http://www.dundas.com>>. Acesso em: 10 jun. 2010.

FERNANDES, M. S.; KANTORSKI, G. Z. **Neuro BI: Uma Ferramenta Web open source para apoio à Tomada de Decisão**. Santa Maria, 2008. Disponível em: <<http://revistaseletronicas.pucrs.br/ojs/index.php/hifen/article/view/4597/3484>>. Acesso em: 10 jun. 2010.

FRANCO, Cristiano R. **Sistemas de informação aplicado ao sistema de gestão ambiental**. 2001. 71f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

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

INMON, W. H.; TERDEMAN, R. H.; IMHOFF, C. **Data Warehousing: como transformar informações em oportunidades de negócios**. São Paulo: Editora Barkeley, 2001.

JACOBSON, R; STACIA, M. **Microsoft SQL Server Analysis Services**. Porto Alegre: Editora Bookman.2007.

KIMBALL, R. **Data warehouse toolkit**. 26. ed. São Paulo: Makron Books, 1998.

KIMBALL, R. **The data warehouse toolkit: practical techniques for building dimensional data warehouses**. 26. ed. New York : John Wiley E Sons, 1996.

KISNER, Gilvani B. **Business intelligence como diferencial competitivo na gestão de negócios.** 2006. 74 f. Monografia (Especialização em Tecnologia da Informação) - Programa de Pós-Graduação em Tecnologia da Informação Integrada na Gestão de Negócios, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

LARMAN G. **Utilizando UML e padrões:** uma introdução à análise e ao projeto orientados a objetos. Porto Alegre: Bookman, 2004.

STREY, Rodrigo R. **Sistema de informação aplicado ao setor financeiro de uma empresa baseado no Siego utilizando *Data Warehouse*.** 2002. 76 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

TURBAN, E. et al. **Business intelligence:** um enfoque gerencial para a inteligência do negócio. Porto Alegre: Bookman, 2009.

ZIMMERMANN, Thiago R. **Desenvolvimento de um sistema de apoio à decisão baseado em Business Intelligence.** 2006. 77 f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

APÊNDICE A – Detalhamento dos casos de uso

No Quadro 13 apresenta-se o caso de uso "Logar na ferramenta".

Nome do Caso de Uso	UC02 – Logar na ferramenta
Descrição	Usuário acessa a ferramenta via navegador Internet e informa os dados para <i>login</i> e senha armazenados no cadastro do colaborador.
Ator	Administrador/Aluno
Pré-condição	Ferramenta deve estar hospedado no servidor <i>web</i> . Usuário deve estar cadastrado no banco de dados.
Fluxo principal	<ol style="list-style-type: none"> 1. Usuário preenche seu <i>login</i> e sua senha; 2. Ferramenta valida os dados de <i>login</i> e senha do usuário; 3. Ferramenta direciona o Usuário para a página de menu do protótipo conforme sua permissão de acesso.
Fluxo alternativo (a)	<ul style="list-style-type: none"> • nome de usuário e/ou senha inválido(s); • alerta com mensagem “Usuário e/ou senha inválido(s)” é mostrado.
Fluxo alternativo (a)	<ul style="list-style-type: none"> • sala do aluno inativa; • alerta com mensagem “A sala definida para este usuário está inativa. Para ativar a sala contate o docente da disciplina.”.
Pós-condição	Usuário entra conectado à ferramenta.

Quadro 13 – Descrição do caso de uso Logar na ferramenta

No Quadro 14, apresenta-se o caso de uso "Cadastrar-se".

Nome do Caso de Uso	UC01 – Cadastrar-se
Descrição	Usuário acessa a ferramenta para realizar o cadastro necessário para fazer o <i>login</i> na mesma.
Ator	Aluno
Pré-condição	Aluno deve acessar a ferramenta.
Fluxo principal	<ol style="list-style-type: none"> 1. Usuário acessa a opção “Cadastrar” na tela inicial da ferramenta; 2. Aluno dados solicitados (nome, sobrenome, login, senha e sala); 3. Aluno clica no botão “Salvar”; 4. Ferramenta valida as informações digitadas pelo usuário; 5. Ferramenta persiste os dados no banco de dados; 6. Ferramenta apresenta mensagem de “Usuário cadastrado com sucesso!”.
Fluxo alternativo (a)	<ul style="list-style-type: none"> • campo obrigatório em branco; • alerta com mensagem “Campo obrigatório[nome do campo] não preenchido” é mostrado.
Pós-condição	Aluno efetuou o seu cadastro na ferramenta.

Quadro 14 – Descrição do caso de uso Cadastrar usuários

No Quadro 15, apresenta-se o caso de uso "Cadastrar usuários".

Nome do Caso de Uso	UC03 – Manter cadastro de usuários
Descrição	Administrador acessa o Cadastro de usuário para manter o cadastro dos usuários que terão acesso a ferramenta.
Ator	Administrador
Pré-condição	Administrador deve fazer <i>login</i> na ferramenta.
Fluxo principal	<ol style="list-style-type: none"> 1. Administrador acessa opção “Usuários” na tela inicial da ferramenta; 2. Ferramenta apresenta tela com os usuários cadastrados; 3. Administrador opta por incluir, alterar ou excluir usuário; 4. Ferramenta persiste os dados no banco de dados; 5. Ferramenta apresenta tela com os usuários cadastrados, já com o registro incluído, alterado ou excluído.
Cenário – Visualização	Ferramenta mostra usuários atualmente cadastrados na mesma.
Cenário – Edição	<ol style="list-style-type: none"> 1. Administrador (Usuário) clica em um registro para edição; 2. Ferramenta mostra o nome, sobrenome, nome do usuário, senha, sala e se o usuário é administrador da ferramenta, para edição; 3. Administrador altera registro e seleciona opção para atualizar os dados (nome, sobrenome, nome do usuário, senha, sala e se o usuário é administrador da ferramenta); 4. Ferramenta valida os dados.
Cenário – Inclusão	<ol style="list-style-type: none"> 1. Administrador utiliza opção “Novo” para incluir um novo registro; 2. Ferramenta apresenta tela solicitando dados; 3. Administrador informa os dados solicitados; 4. Ferramenta valida os dados.
Cenário – Exclusão	<ol style="list-style-type: none"> 1. Administrador seleciona um registro para exclusão e utiliza opção “Excluir”; 2. Ferramenta exclui o registro.
Fluxo alternativo (a)	<ul style="list-style-type: none"> • campo obrigatório em branco; • alerta com mensagem “Campo obrigatório[nome do campo] não preenchido” é mostrado.
Pós-condição	Administrador visualizou, editou, apagou, cadastrou um usuário ou o usuário realizou o seu cadastro.

Quadro 15 – Descrição do caso de uso Cadastrar usuários

No Quadro 16 apresenta-se o caso de uso "Consultar *Data Warehouse*".

Nome do Caso de Uso	UC16 – Consultar <i>Data Warehouse</i>
Descrição	A ferramenta acessa o <i>Data Warehouse</i> para consultar os dados do <i>Data Warehouse</i> que será controlado pela mesma.
Ator	Usuário
Pré-condição	O usuário deve fazer <i>login</i> na ferramenta.
Fluxo principal	a) a ferramenta recebe uma string fixa de conexão com o banco de dados para acessar o <i>Data Warehouse</i> ;

	b) ferramenta valida a <i>String</i> de conexão; c) ferramenta persiste os dados no banco.
Cenário – Edição	1. A ferramenta acessa o <i>Data Warehouse</i> pré-definido; 2. Ferramenta persiste os dados no banco.
Pós-condição	A ferramenta está pré-direcionada para o <i>Data Warehouse</i> existente.

Quadro 16 – Descrição do caso de uso Consultar *Data Warehouse*

No Quadro 17 apresenta-se o caso de uso "Consultar *Data marts*".

Nome do Caso de Uso	UC17 - Consultar <i>Data Marts</i>
Descrição	A ferramenta acessa o <i>Data Mart</i> para consultar dados do <i>Data Mart</i> que será controlado por ela.
Ator	Usuário
Pré-condição	Usuário deve fazer <i>login</i> na ferramenta. Ter no mínimo um <i>Data Warehouse</i> cadastrado. Ter no mínimo um <i>Data Mart</i> cadastrado.
Fluxo principal	a) a ferramenta recebe uma string fixa de conexão com o banco de dados para acessar o <i>Data Mart</i> ; b) ferramenta valida a <i>String</i> de conexão; c) ferramenta persiste os dados no banco.
Cenário – Inclusão	1. Ferramenta acessa o <i>Data Mart</i> pré-definido; 2. Ferramenta persiste os dados no banco.
Pós-condição	A ferramenta está pré-direcionado para o <i>Data Mart</i> existente.

Quadro 17 – Descrição do caso de uso Consultar *Data marts*

No Quadro 18 apresenta-se o caso de uso "Consultar tabela de fato".

Nome do Caso de Uso	UC18 - Consultar tabela de fato
Descrição	Usuário acessa opção Visualizar para consultar a tabela através do cubo de decisão.
Ator	Administrador/Aluno
Pré-condição	Usuário deve fazer <i>login</i> na ferramenta. Ferramenta deve ter ao menos um <i>Data Mart</i> cadastrado. Ferramenta deve ter pelo menos uma tabela fato cadastrada. Os dados já devem ter sido unificados e as tabelas povoadas. Ferramenta deve ter pelo menos um cubo processado.
Fluxo principal	a) usuário opta por visualizar os dados; b) ferramenta apresenta tela com a tabela de fato e suas medidas; c) usuário clica na medida que deseja visualizar e arrasta para a tabela de visualização de dados; d) ferramenta apresenta dados cadastrados na tabela fato.

Cenário – Visualização	A ferramenta mostra os registros de tabela de fato cadastrados para a empresa do Usuário.
Pós-condição	Tabela de fato visualizada.

Quadro 18 – Descrição do caso de uso Consultar tabela de fato

No Quadro 19 apresenta-se o caso de uso "Consultar dimensões".

Nome do Caso de Uso	UC19 – Consultar dimensões
Descrição	Usuário acessa opção Visualizar para consultar as dimensões através do cubo de decisão.
Ator	Administrador/Aluno
Pré-condição	Usuário deve fazer <i>login</i> na ferramenta. Ferramenta deve ter pelo menos um <i>Data Mart</i> cadastrado. Ferramenta deve ter pelo menos uma tabela de fato cadastrada. Ferramenta deve ter pelo menos uma tabela de dimensão cadastrada.
Fluxo principal	a) usuário opta por visualizar os dados; b) ferramenta apresenta tela com as tabelas de dimensão e seus atributos; c) usuário clica na dimensão ou no atributo que deseja visualizar e arrasta para a tabela de visualização de dados; d) ferramenta apresenta dados cadastrados na tabela de dimensão selecionada.
Cenário – Visualização	A ferramenta mostra os registros cadastrados nas dimensões selecionadas.
Pós-condição	Usuário visualizou os dados da dimensão selecionada ou atributo.

Quadro 19 – Descrição do caso de uso Cadastrar dimensões

No Quadro 20 apresenta-se o caso de uso "Unificar arquivos txt ou XML".

Nome do Caso de Uso	UC10 – Unificar arquivos txt ou XML
Descrição	Usuário acessa a opção Unificar, para efetuar leitura de arquivos txt ou XML.
Ator	Administrador/Aluno
Pré-condição	Usuário deve fazer <i>login</i> na ferramenta. Ferramenta deve ter pelo menos um <i>Data Mart</i> cadastrado. Ferramenta deve ter pelo menos uma tabela de fato cadastrada. Ferramenta deve ter pelo menos uma dimensão cadastrada. Dados devem ter sido exportados de um sistema legado e estarem disponíveis em formato txt ou XML.
Fluxo principal	a) usuário seleciona opção Unificar; b) ferramenta solicita arquivo; c) usuário informa arquivo que deve ser unificado; d) ferramenta apresenta tela solicitando valores que deverão ser unificados; e) usuário informa valores que deverão ser unificados; f) ferramenta gera arquivo XML com os registros unificados.
Cenário – Visualização	A ferramenta apresenta o log da unificação de forma resumida.
Pós-condição	Usuário unificou dados fornecidos em arquivo txt ou XML.

Quadro 20 – Descrição do caso de uso Unificar arquivos txt ou XML

No Quadro 21 apresenta-se o caso de uso "Efetuar carga de dados".

Nome do Caso de Uso	UC09 – Efetuar carga de dados
Descrição	Usuário acessa a opção Povoar, para efetuar a carga dos dados já unificados.
Ator	Administrador/Aluno
Pré-condição	<p>Usuário deve fazer <i>login</i> na ferramenta.</p> <p>Ferramenta deve ter pelo menos um <i>Data Mart</i> cadastrado.</p> <p>Ferramenta deve ter pelo menos uma tabela de fato cadastrada.</p> <p>Ferramenta deve ter pelo menos uma dimensão cadastrada.</p> <p>Dados devem estar unificados.</p>
Fluxo principal	<p>a) usuário utiliza opção Povoar;</p> <p>b) ferramenta apresenta tela com os arquivos já unificados disponíveis no banco;</p> <p>c) usuário informa o(s) arquivo(s) que serão importados;</p> <p>d) ferramenta efetua leitura dos arquivos unificados;</p> <p>e) ferramenta insere dados na base de dados.</p>
Cenário – Visualização	A ferramenta apresenta log da carga dos dados de forma resumida.
Pós-condição	Usuário efetuou a carga de dados.

Quadro 21 – Descrição do caso de uso Efetuar carga de Dados

APÊNDICE B – Dicionário de dados do Modelo de Entidade e Relacionamento

O quadro 22 apresenta a tabela e os atributos da tabela ArquivoUnificacao, utilizada para armazenar os arquivos unificados.

Tabela	Descrição			
ArquivoUnificacao	É utilizada para armazenar os arquivos unificados, é desta tabela que os dados são buscados para realizar a importação.			
Atributos da tabela	Descrição	Tipo	Tamanho	Chave primária
Id	Número seqüencial gerado automaticamente para cada registro da tabela.	int		Sim
Arquivo	Arquivo unificado.	xml		Não
Nome	Nome do arquivo para auxiliar na sua localização para que possa ser importado.	varchar	50	Não
Data	Data em que o arquivo foi unificado.	datetime		Não

Quadro 22 – Tabela de arquivos unificados

O quadro 23 apresenta a tabela Sala e os seus atributos.

Tabela	Descrição			
Sala	É utilizada para que ao cadastrar-se, o usuário selecione a sua sala, assim é feito controle, apenas usuários com sala ativa podem acessar a ferramenta.			
Atributos da tabela	Descrição	Tipo	Tamanho	Chave primária
Id	Número seqüencial gerado automaticamente para cada registro da tabela.	int		Sim
Nome	Nome da sala, esta informação é apresentada para que o aluno selecione a sua sala.	varchar	50	Não
Descricao	Descrição da sala.	varchar	100	Não
Situacao	Situação da sala para que possa ser feito o controle, evitando acesso de alunos de outros semestres.	bit		

Quadro 23 – Tabela de Sala

O quadro 24 apresenta a tabela Usuario e os seus atributos.

Tabela	Descrição			
Usuario	É utilizada para realizar o cadastro de usuários, permitindo a realização do controle de acesso à ferramenta.			
Atributos da tabela	Descrição	Tipo	Tamanho	Chave primária
Id	Número seqüencial gerado automaticamente para cada registro da tabela.	int		Sim
Nome	Nome do usuário.	varchar	50	Não
Sobrenome	Sobrenome do usuário.	varchar	50	Não
Login	<i>Login</i> utilizado para acessar a ferramenta.	varchar	50	Não
Senha	Senha que é utilizada no acesso a ferramenta.	varchar	50	Não
Administrador	Define se o usuário é administrador da ferramenta, permitindo controlar as funcionalidades a que o usuário tem acesso.	bit		
Sala	Sala do usuário	int		Não
Empresa	Empresa do usuário.	int		Não

Quadro 24 – Tabela de Usuario

O quadro 25 apresenta a tabela Empresa e os seus atributos.

Tabela	Descrição			
Empresa	É utilizada para realizar o cadastro de empresas, permitindo assim visualizar os dados carregados por cada empresa, o que facilita a avaliação nas atividades propostas pelo docente.			
Atributos da tabela	Descrição	Tipo	Tamanho	Chave primária
Id	Número seqüencial gerado automaticamente para cada registro da tabela.	int		Sim
Sala	Sala do usuário logado ao realizar o cadastro da empresa.	int		Não
Nome	Nome da empresa.	varchar	50	Não

Quadro 25 – Tabela de Empresa

O quadro 26 apresenta a tabela Regiao e os seus atributos.

Tabela	Descrição			
Regiao	Armazena as regiões em que as vendas podem ser realizadas.			
Atributos da tabela	Descrição	Tipo	Tamanho	Chave primária
Id	Número seqüencial gerado automaticamente para cada registro da tabela.	int		Sim
Nome	Nome da região.	varchar	50	Não

Quadro 26 – Tabela de Regiao

O quadro 27 apresenta a tabela Cliente e os seus atributos.

Tabela	Descrição			
Cliente	Armazena os clientes para os quais as vendas podem ser realizadas.			
Atributos da tabela	Descrição	Tipo	Tamanho	Chave primária
Id	Número seqüencial gerado automaticamente para cada registro da tabela.	int		Sim
Nome	Nome do cliente.	varchar	50	Não
CPF	CPF para pessoa física.	varchar	11	Não
CNPJ	CNPJ para pessoa jurídica.	varchar	14	Não
DataNascimento	Data de nascimento para pessoa física.	datetime		Não
Sexo	Sexo para pessoa física.	varchar	1	Não

Quadro 27 – Tabela de Cliente

O quadro 28 apresenta a tabela Fornecedor e os seus atributos.

Tabela	Descrição			
Fornecedor	Armazena os fornecedores das vendas.			
Atributos da tabela	Descrição	Tipo	Tamanho	Chave primária
Id	Número seqüencial gerado automaticamente para cada registro da tabela.	int		Sim
Nome	Nome do fornecedor.	varchar	50	
CPF	CPF para pessoa física.	varchar	11	Não
CNPJ	CNPJ para pessoa jurídica.	varchar	14	Não

Quadro 28 – Tabela de Fornecedor

O quadro 29 apresenta a tabela Tempo e os seus atributos.

Tabela	Descrição			
Tempo	Armazena quando a venda ocorreu.			
Atributos da tabela	Descrição	Tipo	Tamanho	Chave primária
Id	Número seqüencial gerado automaticamente para cada registro da tabela.	int		Sim
Dia	Dia em que a venda ocorreu.	int		Não
Mes	Mês em que a venda ocorreu.	int		Não
MesExtenso	Mês por extenso em que a venda ocorreu.	varchar	20	Não
Trimestre	Trimestre em que a venda ocorreu.	int		Não
Ano	Ano em que a venda ocorreu.	int		Não

Quadro 29 – Tabela de Tempo

O quadro 30 apresenta a tabela Produto e os seus atributos.

Tabela	Descrição			
Produto	Armazena o que foi vendido.			
Atributos da tabela	Descrição	Tipo	Tamanho	Chave primária
Id	Número seqüencial gerado automaticamente para cada registro da tabela.	int		Sim
Nome	Nome do produto.	vachar	50	Não
Descricao	Descrição do produto.	vachar	100	Não
Categoria	Categoria do produto.	vachar	50	Não

Quadro 30 – Tabela de Produto

O quadro 31 apresenta a tabela Conceitos e os seus atributos.

Tabela	Descrição			
Conceitos	Armazena o embasamento teórico para permitir a consulta através da árvore.			
Atributos da tabela	Descrição	Tipo	Tamanho	Chave primária
Id	Número seqüencial gerado automaticamente para cada registro da tabela.	int		Sim
Assunto	Título para localizar o conteúdo na árvore de conceitos.	vachar	150	Não
Conteudo	Embasamento teórico.	vachar	max	Não
IdPai	Id do nível superior, permite montar a árvore com diferentes níveis hierárquicos.	int		Não

Quadro 31 – Tabela de Conceitos

O quadro 32 apresenta a tabela ExplicacaoFigurasDW e os seus atributos.

Tabela	Descrição			
ExplicacaoFigurasDW	Armazena a explicação que é exibida na figura, presente na ferramenta, explicando o funcionamento de um DW.			
Atributos da tabela	Descrição	Tipo	Tamanho	Chave primária
Id	Número seqüencial gerado automaticamente para cada registro da tabela.	int		Sim
Codigo	Código de parte da figura.	vachar	50	Não
Explicacao	Explicação de parte da figura.	vachar	100	Não

Quadro 32 – Tabela de ExplicacaoFigurasDW

O quadro 33 apresenta a tabela Unificacao e os seus atributos.

Tabela	Descrição			
Unificacao	Armazena os valores que devem ser unificados.			
Atributos da tabela	Descrição	Tipo	Tamanho	Chave primária
Id	Número sequencial gerado automaticamente para cada registro da tabela.	int		Sim
Tabela	Tabela da coluna que deve ser alterada.	varchar	50	Não
Coluna	Coluna que deve ser alterada.	varchar	50	Não
ValorAntigo	Valor que deve ser alterado	varchar	50	Não
ValorNovo	Valor para o qual a coluna deve ser alterda.	varchar	50	Não

Quadro 33 – Tabela de Unificacao

APÊNDICE C – Questionário aplicado em sala

A figura 53 apresenta o questionário de avaliação da ferramenta aplicado em sala na aula de TIGN.

Avaliação do ambiente de ensino em Data Warehouse utilizado na disciplina de TIGN

* Required

1) Você está em qual semestre? *

Primeiro

2) A hierarquia das informações está adequada? Segue uma sequência lógica? *

Não Atende

3) No caso de procurar uma funcionalidade, encontrou a mesma com facilidade? *

Não Atende

4) No preenchimento de um formulário caso algum campo obrigatório não esteja preenchido, ao tentar enviar o sistema apresenta mensagem alertando o usuário? *

Não Atende

5) Em caso afirmativo na resposta anterior, ao retornar ao formulário para correção, os campos preenchidos corretamente são mantidos? *

Não Atende

6) É fácil consultar os dados? *

Não Atende

7) Ao ocorrer algum evento em que o usuário necessite ser informado a mensagem apresentada é coerente? *

Não Atende

8) O software cumpre de forma objetiva a tarefa de complementar a explicação teórica sobre Data warehouse apresentada em sala? *

Não Atende

9) Qual nível de poluição Visual do ambiente? *

1 2 3 4 5

Nenhum Altamente poluído

Figura 53 – Questionário de avaliação da ferramenta