Pós-Graduação em Ciência da Computação

"METODOLOGIA E USO DE TÉCNICAS DE EXPLORAÇÃO E ANÁLISE DE DADOS NA CONSTRUÇÃO DE DATA WAREHOUSE"

Por

ROBERTO ÂNGELO FERNANDES SANTOS

Dissertação de Mestrado

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE, SETEMBRO/2002

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ROBERTO ÂNGELO FERNANDES SANTOS

METODOLOGIA E USO DE TÉCNICAS DE EXPLORAÇÃO E ANÁLISE DE DADOS NA CONSTRUÇÃO DE DATA WAREHOUSE

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco.

Orientador: Prof. Décio Fonseca

Banca: Prof. Dr. Paulo Jorge Leitão Adeodato-UFPE
Prof. Dr. Marcus Costa Sampaio – UFPB
Prof. Dr. Décio Fonseca – UFPE

Apoio: CNPq.

RECIFE, SETEMBRO 2002

EPÍGRAFE

A mente, como o lar, é mobiliada pelo proprietário, portanto, se sua vida for fria e árida, a culpa será somente dele.

Louis L'Amor

Dedico esta dissertação a todos que de alguma forma participaram e contribuíram da elaboração.

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus, pela minha existência.
A toda minha família, em especial meu falecido pai Augusto Ângelo e a minha mãe Sônia Santos, que me ensinaram o valor dos estudos em minha vida.
A minha noiva por sua paciência e pelo seu apoio.
A minha tia que me acolheu em seu lar e me trata como se fosse seu filho.
Um muito obrigado a meu orientador Doutor Décio Fonseca, que me apoiou e acreditou em mim, mesmo antes do início da jornada. Sem ele tudo seria muito mais difícil ou talvez nem acontecesse.
Agradeço ao meu colega de trabalho, professor e amigo Paulo Adeodato pela orientação informal e pelas longas discussões sobre o assunto aqui abordado.
Meu agradecimento a todos os professores que sempre me deram incentivo. Em especial aos meus mestres Prof. João Gualberto e Profª.Simone Branco.
A empresa Neurotech que me deu oportunidade de ter acesso à tecnologias de manipulação e tratamento de dados, bem como acesso à aplicações práticas e para estudos de casos reais.
Ao Núcleo de Tecnologia da Informação da UFPE (NTI-UFPE) pela disponibilidade dos dados dos COVEST e pelo espaço cedido em seu laboratório.
Agradecimento a Vitor Guedes por me ajudar em algumas implementações da dissertação, Ana Soraya por me ajudar com a formatação e correção, a Hélio por discutir assuntos relacionados à dissertação, tornando meu trabalho menos solitário.
E por fim agradeço a todos, não citados aqui, mas que me apoiaram e me ajudaram de alguma forma.

SUMÁRIO

1 – INTRODUÇÃO 12

1.1 – CONTEXTO 12
1.2 – MOTIVAÇÃO 13
1.3 – OBJETIVOS 15
1.4 – ORGANIZAÇÃO RESUMIDA DA DISSERTAÇÃO 16

2 – O AMBIENTE REDIRIS 17

2.1 – INTRODUÇÃO 17
2.2 – CONCEITOS BÁSICOS 18
2.3 – A DINÂMICA REDIRIS 19
2.4 – O COMPONENTE PRÉ-PROCESSADOR NO REDIRIS 21

3 – REVISÃO BIBLIOGRÁFICA 23

3.1 – CONSTRUÇÃO DE DATA WAREHOUSE 23
3.1.1 – Metodologia Baseada em Dados 23
3.1.2 – Banco de Dados Amostra Viva de INMON 27
3.1.3 – Granularidade 29
3.1.4 – Integração de Dados EM DW 30
3.2 – MINERAÇÃO DE DADOS 31
3.3 – TÉCNICAS E ALGORITMOS PARA PRÉ-PROCESSAMENTO DE DADOS 34
3.4 – QUALIDADE DOS DADOS 36
3.4.1 – Qualidade dos Dados no DW 37
3.4.1 – Processo de Qualidade de Dados 38
3.5 – EXPLORAÇÃO E ANÁLISE DE DADOS NO PRÉ-PROCESSAMENTO 44
3.5.1 – Visualização dos Dados no Pré-Processamento 45
3.5.2 – Distribuição de Frequência 47
3.5.3 – Dist. de Frequência na Exploração e Análise de Dados 50

4 – METODOLOGIA FASTCUBE 53

4.1 – INTRODUÇÃO 53
4.2 – VISÃO GERAL DA METODOLOGIA 54
4.3 – MAPEAMENTO E INTEGRAÇÃO DE DADOS 55
4.4 – FRAGMENTAÇÃO DOS DADOS 57
4.5 – ANÁLISE DOS DADOS 58
4.6 – TRATAMENTO DOS DADOS 61
4.7 – PROTOTIPAÇÃO RÁPIDA 63

5 – UMA IMPLEMENTAÇÃO DO FASTCUBE 71

5.1 – UM CICLO DO FASTCUBE NA ARQUITETURA DO REDIRIS 71
5.3 – ARQUITETURA 72
5.4 – O MODELO DE DADOS E METADADOS 75
5.3 – QUALIDADE DOS DADOS 79
5.3 – TÉCNICAS DO PRÉ-PROCESSAMENTO - UM MODELO EXTENSÍVEL. 81
5.3 – TÉCNICAS IMPLEMENTADAS NO REPOSITÓRIO DE TÉCNICAS. 84
5.4 – A INTERFACE E NAVEGAÇÃO DO PROTÓTIPO 87
5.5 – POSSIBILIDADES COMPLEMENTARES DO MODELO 91

6 – ESTUDO DE CASO – COVEST 93

6.1 – OBJETIVOS E REQUISITOS DO DATA MART - COVEST 93
6.2 – FERRAMENTAS UTILIZADAS 93
6.3 – DESCRIÇÃO DOS DADOS 94
6.4 – GRANULARIDADE 96
6.5 – SELEÇÃO E INTEGRAÇÃO DOS DADOS 97
6.6 – MANIPULAÇÃO E TRATAMENTO DOS DADOS 100
6.7 – GERAÇÃO DO MODELO E POVOAMENTO 101
6.8 – EXTRAÇÃO E ANÁLISE DOS RESULTADOS 102

7 – CONSIDERAÇÕES FINAIS 109

7.1 – RESULTADOS E CONCLUSÕES 109
7.2 – TRABALHOS CORRELATOS 111
7.3 – TRABALHOS FUTUROS 112

ANEXO I - ALGORITIMO DE SUGESTÃO DE DIMENSÕES 114

ANEXO II - ALGORITMO DE MATCHING MELHORADO 117

REFERÊNCIAS BIBLIOGRÁFICAS 120

LISTA DE ILUSTRAÇÕES

Figura 2.1 – Arquitetura do REDIRIS..... 20

Figura 3.1 – Etapas da fase METODO2 24

Figura 3.2 – Banco de Dados Amostra Viva..... 28

Figura 3.3 – Problemas Típicos de Integração de Dados..... 31

Figura 3.4 – Etapas do Processo KDD [FAY96]..... 32

Figura 3.5 – Taxonomia das Técnicas de Mineração de dados [AUR97]..... 33

Figura 3.6 – Fórmula para Normalização segundo a Distribuição..... 35

Figura 3.7 – Fórmula para Normalização segundo a Amplitude..... 35

Figura 3.8 – Medição da Qualidade na Construção de um DW [CAM01]. 39

Figura 3.9 – Critérios de Qualidade..... 40

Figura 3.10 – Critérios de qualidades e as tarefas realizadas em um DW [JAR00]..... 41

Figura 3.11 – Dimensões de qualidade propostas por [JAR97a] 42

Figura 3.12 – Dados representados na Forma Gráfica [ALM01]..... 47

Figura 3.13 – Distribuição de Frequência do campo INSCRICAO (chave primária) 51

Figura 4.1 –Um ciclo na metodologia FastCube..... 54

Figura 4.2 – Fragmentação aplicada a uma tabela de várias colunas. 58

Figura 4.3 – Tela de análise e tratamento de dados do NeuralScorer. 61

Figura 4.4 – Um exemplo de um modelo de classes de metadados para tratamento de dados. 63

Figura 4.5 – Seleção dos fragmentos (atributos) relevantes para o Data Mart. 65

Figura 4.6 – Uma tabela de fato e dimensões com apenas um atributo 66

Figura 4.8 – Modelo com *surrogate key* e dimensões montadas. 69

Figura 4.9 – SQL Simplificado de carga de dimensões e fatos..... 70

Figura 5.1 – Um ciclo no REDIRIS. 71

Figura5.2 – Arquitetura básica da implementação do FastCube 72

Figura 5.3 – Principais pacotes do FastCube..... 73

Figura 5.4 – Diagrama de classes do pacote dataManipulation..... 74

Figura 5.5 – Diagrama de classes do pacote Storage. 75

Figura 5.6 – Principais pacotes de dados e metadados 76

Figura 5.8 –Principais cenários para o processo de qualidade dos dados. 81

Figura 5.9 – Fluxo da aplicação de uma técnica no módulo Pré-Processador do REDIRIS. 82

Figura 5.10 – Aplicação de várias técnicas em cascata. 84

Figura 5.11 – Diagrama de classes de implementação das técnicas. 86

Figura 5.12 – Tela inicial do protótipo FastCube. 87

Figura 5.13 – Tela de criação aplicação. 88

Figura 5.14 – Tela de abertura de aplicação 88

Figura 5.15 – Tela de fragmentação da *TabelaAmostra.* 89

Figura 5.16 – Tela de pré-processamento de dados. 89

Figura 5.17 – Tela de montagem de modelo dimensional. 90

Figura 5.18 – Tela de sugestão de dimensão. 91

Figura 6.1 – Modelo Lógico Relacional dos dados de entrada. 96

Figura 6.2 – Modelo estrela do protótipo final do COVEST. 101

Figura6.3– Gráficos considerando a variável MOTIVO_CURSO 102

Figura 6.4 – Gráficos considerando a variável Renda Familiar 102

Figura 6.5 – Gráficos considerando a variável Renda Familiar 103

Figura 6.6 – Gráficos considerando a variável Renda Familiar 103

Figura 6.7 – Tela do SAGENT com gráfico de Aprovação X Cursos X Rede. 104

LISTA DE TABELAS

Tabela 3.1: Algumas tarefas de KDD e suas técnicas de mineração de dados [AUR97] 32

Tabela 3.2: Características e métricas geralmente usado na qualidade dos dados [HUF96]. 43

Tabela 3.3 – Dados apresentados na forma tabular [ALM01] 46

Tabela 3.4 – Distribuição de frequência 1. 48

Tabela 3.5 – Distribuição de frequência 2. 48

Tabela 3.6 – Distribuição de frequência 3. 48

Tabela 3.7 – Distribuição de frequência de estados. 49

Tabela 3.8 – Sugestão de tratamento 1 49

Tabela 3.9 – Sugestão de tratamento 2. 49

Tabela 4.1 – Quadro comparativo de aspectos de construção de DW dos principais autores. 70

Tabela 6.1 – Atributos da tabela desnormalizada 100

Tabela 6.2 - Estatística geral de aprovação 105

Tabela 6.3 - Estatística de aprovação segundo sexo e opção 105

Tabela 6.4 - Estatística de aprovação segundo Ar-Condicionado, Computador e Internet 105

Tabela 6.5 - Estatística de aprovação segundo posse de computador 105

Tabela 6.6 - Estatística de aprovação segundo sexo 106

Tabela 6.7 - Estatística de aprovação segundo domínio de língua estrangeira 106

Tabela 6.8 - Estatística de aprovação segundo quantidade de vestibulares prestados 106

Tabela 6.9 - Conhecimento descoberto pela equipe diretamente no Data Mart. 106

Tabela 6.10 - Estatística de aprovação segundo sexo e participação da renda familiar 107

Tabela 6.11 – Relação entre número de familiares e participação em ensino público e privado 108

LISTA DE ABREVIATURAS E SIGLAS

DCBD	- Descoberta de Conhecimento em Base de Dados
DW	- Data Warehouse
DWQ	- Data Warehouse Quality
ETL	- Extraction, Transformation and Loading
KDD	- Knowledge Discovery in Databases
KNN	- K Nearest Neighbours
OLAP	- On-Line Analitical Process
REDIRIS	- Reuse Environment on Data Integration, Reuse and Quality in Information Systems
SAD	- Sistemas de Apoio à Decisão
SGBD	- Sistema Gerenciador de Banco de Dados
SQL	- Structured Query Language
UML	- Unified Modeling Language
XML	- eXtensible Markup Language

RESUMO

O volume de informações a ser trabalhado na tomada das decisões gerenciais supera largamente a capacidade do processamento humano, mecânico e dos sistemas transacionais atuais, exigindo ferramentas de apoio à decisão mais adequadas aos novos desafios gerenciais. Mesmo aplicando-se modelos de decisão tidos como adequados, uma grande parte das implementações de Sistemas de Informação não atingem os resultados esperados, o que levam muitos deles ao fracasso total ou parcial. Acredita-se que com obtenção de resultados rápidos se possa conseguir um maior envolvimento do usuário final, o que segundo os especialistas diminui bastante a possibilidade de fracasso.

Esse trabalho visa a utilizar técnicas de análise e exploração de dados na construção de soluções de Sistemas de Apoio à Decisão, em especial na construção de Data Warehouse(DW). Aproveita-se o conhecimento adquirido com a aplicação dessas técnicas, mostrando a sua importância nas diversas fases de sua construção de um DW. Propõe-se e implementa-se uma metodologia chamada FASTCUBE, que é baseada em um modelo de pré-processamento de dados. Ela incorpora de maneira rápida os metadados extraídos diretamente da massa de dados. Acelerar e sedimentar a compreensão do problema, sempre levando-se em consideração a qualidade dos dados, durante todas as suas fases é um dos pontos forte dessa metodologia. O seu objetivo final é acelerar o processo de visualização do modelo de decisão, através de um protótipo de modelo dimensional, com dados operacionais amostrados no início do processo e tratados durante o mesmo.

ABSTRACT

This effort sights to use analysis techniques and data exploration in the construction of solutions of Support Decision Systems, especially in the construction of Data Warehouse (DW). It is Utilized the knowledge acquired with these techniques application, showing its importance in the several stages of its construction of a DW. It is suggested and implemented a methodology called FASTCUBE, which is based on a pre-processing data model. It incorporates in fast way metadata extracted directly of the data mass. Accelerating and sediment the comprehension of the problem, always, considering the data quality, this is one of the strong points of this methodology during all the phases. Its final objective is accelerate the process of visualization of the decision model, through a dimensional model prototype, with operational data showed in the beginning of the process and treated during itself.

1 – INTRODUÇÃO

1.1 – CONTEXTO

A multiplicação de situações complexas e de baixa previsibilidade, as reações administrativas habituais – estabelecer a ordem, racionalizar, sistematizar e controlar – se revelam cada vez mais ineficazes, impondo a renúncia de antigos paradigmas. A velocidade das comunicações e o volume das informações a serem trabalhadas nas tomadas das decisões gerenciais superam largamente a capacidade do processamento humano e mecânico, exigindo ferramentas de apoio à decisão mais adequadas aos novos desafios gerenciais.

É indiscutível a importância e o destaque que os pesquisadores ligados à Administração e a Tecnologia da Informação [DRU99] [MAR99] dão à manipulação dos dados na moderna administração. O sucesso de qualquer organização nos dias de hoje está condicionado à capacidade de analisar, planejar e reagir rapidamente às mudanças no ambiente ao qual ela está inserida [DRU99], tornando-se imprescindível à captação e ao processamento de dados internos e externos das empresas em um menor tempo possível.

O crescimento, nos últimos anos, do nível de informatização das organizações tem como resultado o crescimento substancial no volume de dados armazenados, disponibilizando os elementos necessários para a obtenção de informações necessárias para reações diante das "constantes mudanças". É preciso que as organizações saibam tirar proveito de toda massa de dados disponível. Essa não é uma tarefa fácil, pois faz-se necessário todo um tratamento diferenciado de dados para que esses possam ser usados em nível de apoio à decisão.

Apesar da melhora crescente do nível de informatização das empresas, os grandes esforços ainda estão concentrados no seu nível operacional. Grande parte dos dados das empresas foram projetados para atender a esse nível organizacional. O lado positivo desse processo é o grande volume de dados que foram e estão sendo acumulados pelas organizações. Essa enorme quantidade de dados operacionais não pode ser usada diretamente pelo nível decisório da empresa, porque fica difícil para o especialista manipular e compreender esses dados na tarefa de formular testes de hipóteses. Porém, se bem utilizado, grandes volumes de dados com qualidade vão corresponder a um maior potencial de informação. Nesse cenário é que introduzimos os Sistemas de Apoio à Decisão (SAD), especialmente construídos para disponibilizar informações e dar subsídios à decisão. Eles

surgem a partir dos sistemas transacionais da empresa, mas podem (e devem) absorver informações de outras fontes internas ou externas à organização.

"O ambiente de dados para suporte aos processos de gerência e tomada de decisão é fundamentalmente diferente do ambiente convencional de processamento de transações. No coração deste ambiente está a idéia do Data Warehouse, integrando e consolidando dados disponíveis em diferentes acervos para fins de exploração e análise, ampliando o conteúdo informacional destes acervos para atender às expectativas e necessidades de nível estratégico na empresa." [CAM00]

1.2 – MOTIVAÇÃO

Uma boa parte das tentativas de implementações de Sistemas de Apoio à Decisão (SAD) resulta em fracasso [BRI02] [KEL97], O Data Warehouse é uma técnica de apoio à decisão, que também tem tido dificuldade de gerar bons SADs[SIN01]. Para se garantir o sucesso de um Data Warehouse (DW), fundamentalmente, tem-se que escolher corretamente a estratégia a ser adotada [INM97]. Essa deve ser adequada as necessidades específicas do ambiente onde o DW será implementado. Segundo [INM97], é melhor começar com pequenos Data Warehouses, sem muitos dados e depois incrementá-los. [CAM00] reforça esta idéia afirmando que a estratégia mais adequada é uma abordagem evolucionária, apresentado-a como sendo: "uma maneira prudente, segura e eficiente de se desenvolver o DW".

Uma grande parte das implementações de Sistemas de Informação de Apoio à Decisão não atingem os resultados esperados, acarretando atraso de prazos, aumento de custos ou simplesmente não atende aos requisitos do usuário. Um grande número desses nem chegam a ser implementados ou são apenas implementados parcialmente. O que se observa são tentativas errôneas de projetos de Data Warehouse, dentre as quais tem-se como principal erro a tentativa de implementações de Data Warehouses globais. Pode-se destacar ainda alguns fatores de fracassos que podem ser conseqüência ou não da tentativa de construção de Data Warehouses globais, são eles: a falta de conhecimento do negócio, tentativa de uso de tecnologias do modelo transacional, baixa qualidade dos dados e falta de conhecimento, dificuldades na construção de modelos que suportem decisão e pensar em apoio à decisão somente no final do processo de construção do DW.

Uma maneira de pensar em apoio à decisão é a adoção de uma metodologia baseada em dados, desde o início da modelagem. Normalmente, as metodologias de construção de SAD não possuem uma abordagem de utilização direta dos dados, resumindo-se ao entendimento de metadados (Diagrama de Entidade Relacionamento, por exemplo) sem se preocupar com as informações que podem ser extraídas diretamente dos dados, nem com o uso e real identificação dos dados.

Partindo do princípio que os sistemas de apoio à decisão são construídos a partir dos dados operacionais, porque não usá-los diretamente durante o processo de análise e projeto? Os dados podem ser fontes de informações valiosas no entendimento do problema, servindo como uma rica e real fonte de informações, auxiliando na construção do modelo, verificando a qualidade e até mesmo servindo de suporte para a decisão da viabilidade do sistema. O fato é que com auxilio dos dados podemos entender melhor o negócio e muitos problemas que são encontrados no final do processo implantação podem ser identificados no início do processo.

Em geral, os pesquisadores e o mercado estão de acordo que as vantagens obtidas com a adoção de particionamento do Data Warehouse [INM97][KIM98] [MAR99][SIN01], sendo esse um dos principais motivos de sucesso de um DW. O particionamento de um DW em Data Marts diminui sua complexidade de construção. Dentro desse raciocínio, é possível diminuir ainda mais essa complexidade, se utilizarmos para análise, projeto e prototipação um sub-conjunto (amostra) dos dados do Data Mart. A utilização de amostras para construção de Data Warehouses é conhecida como Banco de Dados de Amostra Viva [INM97]. Mas essa técnica não é muito explorada pelos autores, sendo citada apenas como um DW de consultas estatísticas elaborado a partir de uma amostra de um DW completo.

Investigar os dados induz ao gestor a pensar mais sobre os dados que são importantes em todos os níveis (decisório, operacional, estratégico), contribuindo para o maior entendimento da própria organização, porque pensar sobre os dados é pensar sobre o negócio da empresa.

1.3 – OBJETIVOS

Esse trabalho visa utilizar técnicas de análise e exploração de dados na construção de soluções de Sistemas de Apoio à Decisão, mostrando como elas podem ser importantes nas diversas fases de construção de um Sistema de Apoio à Decisão. Propõe-se uma metodologia chamada FASTCUBE, baseada em pré-processamento de dados, que incorpora rapidamente metadados extraídos diretamente da massa de dados, mostra a sua real situação, ajuda no seu entendimento, descobre e soluciona problemas no início do processo de análise e modelagem.

Através da incorporação de metadados que possibilitem a medição da qualidade dos dados, pode-se verificar a real situação dos dados, possibilitando a aplicação de métricas para a avaliação da qualidade dos dados. Essas métricas direcionarão alguns dos próximos passos a serem tomados, podendo inclusive indicar o abandono do processo em vista da qualidade dos dados. Essa verificação será não só importante para a construção de um novo Sistema de Apoio à Decisão, mas também servirá para validar e melhorar todo o processo de coleta de dados da organização.

As possibilidades de uso dessa metodologia são bem amplas, podendo ajudar em todas as fases de construção, apoiando o especialista na identificação as regras do negócio, transformando os dados em informações estratégicas. Outro aspecto relevante desse trabalho é a prototipação rápida de um Data Mart que é um dos resultados obtidos pela aplicação da metodologia. Propõem-se que a construção da solução seja feita de forma iterativa por garantir um processo evolutivo e mais seguro. O modelo deve ser enriquecido a cada iteração, não sendo necessário que o especialista termine completamente uma fase para começar a outra. É importante chegar ao final da primeira iteração da metodologia, porque o usuário só terá condições de expressar as suas necessidades com clareza após o primeiro ciclo.

Esse trabalho não propõe uma solução “end-to-end”, e sim um conjunto de técnicas e de uma metodologia que serão dispostas no ambiente integrado de construção de Sistemas de Informação REDIRIS (Research Environment on Data Integration, Reuse and Quality in Information Systems), que deverão ser usadas juntamente com todas as outras possibilidades encontradas no ambiente.

Os resultados desse trabalho são validados através da construção de protótipos, a partir de um laboratório de CASES, com o uso de implementações próprias e diversas

ferramentas para materializar o ambiente com as suas diversas características, demonstrando assim a utilidade e amplitude de uma metodologia de construção baseada em dados e as técnicas por ela empregada.

1.4 – ORGANIZAÇÃO RESUMIDA DA DISSERTAÇÃO

Esse documento está organizado da seguinte forma:

- No Capítulo1 é a introdução do trabalho, contextualizando-o, descrevendo seus objetivos e motivações.
- No Capítulo2 é apresentado o ambiente REDIRIS e suas principais características, destacando a importância desse trabalho para o ambiente.
- No Capítulo3 são abordados aspectos metodológicos mais relevantes para o entendimento desse trabalho.
- No Capítulo4 é apresentada uma metodologia de construção de Data Mart, discutindo os passos necessários para seu uso.
- No Capítulo5 é apresentado um ambiente onde foi possível simular e validar a metodologia proposta no Capítulo4.
- No Capítulo6 é feito um estudo de caso com os dados do COVEST como uma forma de validar a metodologia.
- No Capítulo7 são feitas algumas considerações finais apresentando os resultados obtidos. Nesse capítulo também são mostrados alguns trabalhos correlatos e os possíveis trabalhos futuros.

2 – O AMBIENTE REDIRIS

Os projetistas de Data Warehouse precisam utilizar-se de técnicas apropriadas para desenvolvimento e implantação de Sistemas de Informação, porque as organizações demandam cada vez mais informações de qualidade e cada vez mais rápido. Essa demanda só pode ser suprida por um ambiente voltado para esse fim, construído e/ou adaptado em tempo hábil utilizando sempre o conhecimento adquirido em implementações anteriores. É nesse contexto que surge a idéia de um ambiente integrado chamado de REDIRIS (*Reuse Environment on Data Integration, Reuse and Quality in Information Systems*). Neste capítulo será apresentado o ambiente REDIRIS de forma simplificada, para contextualizar e situar os mecanismos desenvolvidos neste trabalho.

2.1 – INTRODUÇÃO

O desenvolvimento de SAD, encorajado por todo o avanço no campo de Data Warehouse, vem tornando-se uma motivação comum de pesquisadores na área [VAS00]. Contudo, a notória complexidade relacionada à construção de um esquema dimensional eficiente, adicionou uma considerável quantidade de projetos mal sucedidos [BRI02] [KEL97][SIG97] reivindicando por uma reflexão mais profunda nas estratégias adotadas no desenvolvimento. Alguns fatores relevantes como o pouco conhecimento do negócio, necessidade do limite de gerência, abordagens de projeto inadequadas, fraca qualidade dos dados e falta do conhecimento dos dados, aparecem sendo uma causa comum das possíveis falhas. Assim, o reuso de modelos e soluções, cuja eficiência já tem sido reconhecida, leva a uma perspectiva promissora para sistemas de apoio à decisão, baseados na premissa que estes sistemas dividem um conjunto de conceitos em comum [KIM96]. Tal abordagem envolve a identificação dos objetos, operadores e relações que compõem um domínio de aplicação [NEI80] (ou de uma família de aplicações) como meios de gerar novos exemplares de soluções pré-existentes. Na Engenharia do Domínio [ARA91] técnicas e métodos suportam inteiramente o processo de capturar, organizar e melhorar a informação com respeito a um domínio SAD particular, enquanto encapsula a dado coletado em componentes reusáveis e modelos. Mais recentemente, outras tecnologias floresceram no suporte de reuso do domínio-específico tal como o metadado [SVV99], reuso de bibliotecas [MCD89], mega programação [BOE92], modelagem orientada a aspectos [AOP98], testes padrões [BUS96], e estruturas de frameworks [FAY99]. Apesar da atrativa perspectiva, o desenvolvimento de apoio à decisão

baseado no reuso não tem feito parte das mais recentes pesquisas nesta área, de acordo com o exame descrito em [VAS00].

Por outro lado, muitos problemas normalmente encarados no último estágio de um projeto de SAD poderiam ter sido tratados em etapas adiantadas da construção se a investigação apropriada dos dados fosse conduzida. De fato, as investigações de dados induzem projetistas a considerarem meticulosamente a qualidade de dados em todos os níveis de decisão e contribuem para uma compreensão melhor da organização e do potencial do projeto. Técnicas de inteligência artificial, tais como a mineração de dados [FAY96], ajudam na descoberta da informação latente que eventualmente reside nos dados operacionais, além de auxiliar nas etapas de projeto e implementação de um SAD. Em um SAD, a complexidade de cada análise pode ser consideravelmente reduzida se nós dividirmos os dados em unidades menores (de acordo com similaridades de negócio) onde cada amostra de dados mais restrita pode ser extraída e analisada. Esta abordagem também está de acordo com princípio do *Data Mart,* defendido por muitos autores [KIM96] [INM97] [MAR99].

REDIRIS guarda uma grande similaridade com Ambientes de Projeto Orientados a Domínio - Domain-Oriented Design Environments (DODE) - [FIS94] como uma Odisséia – Odyssey- [BWM99] estendendo à automatização do reuso das soluções sob a arquitetura baseada em componente. REDIRIS considera o processo de desenvolvimento reusável, o qual aponta para a redução da complexidade inerente à definição de soluções e assim construir em um tempo menor e com mais qualidade aplicações de suporte a decisão.

2.2 – CONCEITOS BÁSICOS

Alguns princípios básicos regem a concepção e a construção do ambiente REDIRIS. A seguir são apresentados esses princípios.

Primeiro Princípio. Qualidade como o maior fator de sucesso, desde os processos de captura e análise dos dados, até a entrega de uma nova solução de apoio à decisão.

Segundo Princípio. O processo de desenvolvimento SAD clássico é inadequado. Conseqüentemente, incorporação do conhecimento de negócio e exigências de gerência em estágios avançados deste processo, aliados ao uso de métodos de desenvolvimento avançado contribuem para impedir que o processo total falhe.

Terceiro Princípio. A dinâmica de um SAD requer velocidade, respostas confiáveis, que pedem a implementação de um esquema de reuso eficiente, assim como ela encaixa processos de decisão, acoplando modelos (semi) automáticos de métodos de geração e ferramentas inteligentes.

Quarto Princípio. O ciclo de desenvolvimento de um SAD deve perseguir a interatividade, características dinâmicas e incrementais, e requer avaliação durante todos os estágios do processo, com uma atenção especial para as mudanças estratégicas que podem ocorrer dentro da organização.

Quinto Princípio. A especificação de metadados e domínios de aplicação, aliados com a sua gerencia eficiente remanesce os grandes desafios de um ambiente de construção de um SAD.

Acoplado com mecanismos de já consagrados de engenharia de software e construção de sistemas de apoio à decisão, o REDIRIS introduz técnicas de inteligência artificial [CHE01], [PYL99], [HAN01] ao ciclo de desenvolvimento de um SAD como um caminho para conseguir objetivos na qualidade de dados. Entre outros aspectos, essas técnicas relacionadas a aplicações de indução de regras para descobrimento e avaliação do conhecimento de negócio, casamento e correlação de algoritmos (matching) para tratar anomalias de dados, algoritmos de associação para identificação de hierarquias dimensionais, e métodos estáticos para desenvolvimento de soluções de apoio à decisão de alta qualidade.

2.3 – A DINÂMICA REDIRIS

A arquitetura REDIRIS é descrita na Figura 2.1. O *Módulo de Captura e Análise* responde para os dois processos sobre a captura de aspectos herdados de uma aplicação de domínio, e a análise de sistemas legados para avaliar seus potenciais para reuso e qualidade de dados operacionais.

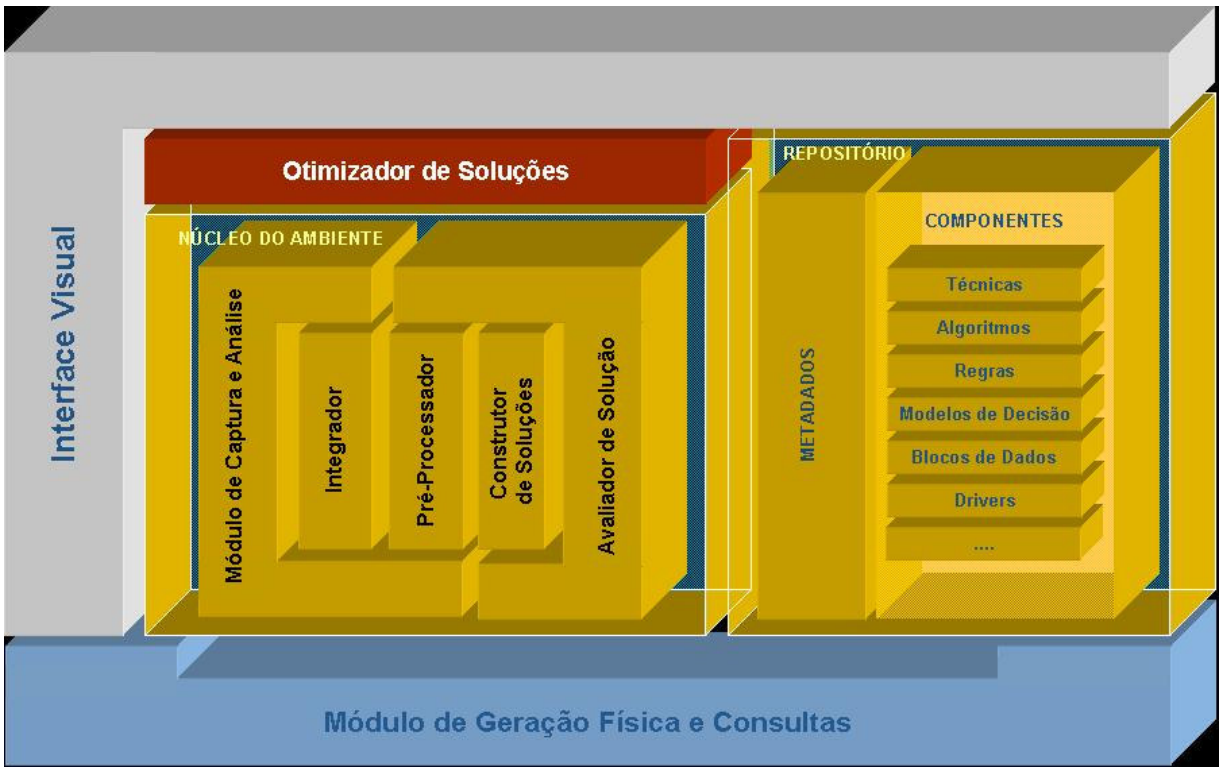

Figura 2.1 – Arquitetura do REDIRIS

O *Módulo Integrador* tem como principal funcionalidade a integração de dados, principalmente quando provêm de bases distribuídas e heterogêneas. Esse módulo Integrador fornece ao REDIRIS os dados em um formato padrão pré-estabelecido tornando diluindo e tornando transparente toda a complexidade envolvida na integração dos dados de entrada de uma solução. O *Módulo Pré-Processador* executa a tarefa importante de promover técnicas de análises de dados para ajudar na avaliação do potencial que os dados poderão fornecer para o apoio à decisão. O *Módulo Construtor de Soluções* é responsável pela elaboração conceitual das soluções de apoio à decisão, ele pode elaborar uma solução a partir do zero ou se basear em soluções pré-existentes do repositório REDIRIS utilizando reuso de componentes de domínio semelhantes para essa nova solução. Para que o *Módulo Construtor* seja capaz de fazer reuso de aplicações antigas o REDIRIS propõe que todo o conhecimento gerado durante o processo e os resultados obtidos a cada construção de solução seja carregado no repositório. O *Módulo Otimizador* provê ajuda aos especialistas na busca pelo melhor conjunto de configuração para um determinado modelo de decisão. Para essa configuração esse módulo se baseia em requisitos de funcionamento que devem estar em conformidade com as características especificadas da solução. O *Módulo de Geração Física e Consultas* é responsável pela implementação física dos modelos estabelecidos pelo *Módulo Construtor,* por exemplo, se esse último determinar a construção de um modelo dimensional, o *Módulo de*

Geração Física é se responsabilizará pela execução dos scripts de criação e carga no banco de dados. Finalmente, todos os componentes REDIRIS acima mencionados são controlados por uma iterativa *Interface Multifuncional* que tem o desafio de personalizar todo o processo interativo para diferentes perfis de usuários, acesso inteligente aos componentes de domínio, integração de unidades de ambiente, e recuperação eficiente de dados em diferentes níveis abstratos.

2.4 – O COMPONENTE PRÉ-PROCESSADOR NO REDIRIS

Para essa dissertação o módulo de pré-processamento é o mais importante, porque esse trabalho tem no seu centro uma metodologia focada na manipulação de dados para construção de Data Warehouses. Apesar da característica marcante de processamento de dados desse trabalho e consequentemente da metodologia, todos os módulos desse ambiente são abrangidos para implementação dessa metodologia. No Capítulo 5 é mostrado um ciclo por todos os componentes do REDIRIS envolvidos, focando principalmente a importância nesse trabalho.

O módulo PRÉ-PROCESSADOR do ambiente REDIRIS é o responsável por toda a captura de metadados a partir dos dados de entrada. Ele é quem faz toda e qualquer manipulação de dados através de análise, limpeza e transformação dos dados. Ele executa operações relacionadas com os dados de entrada; fornece metadados para a modelagem dimensional, verifica a qualidade e enriquece os dados de entrada.

A análise dos dados é feita durante todas as etapas que envolvem dados no REDIRIS, porém é o módulo pré-processador que gera a maior parte dos metadados de suporte a essa operação. Ele também provê acesso às técnicas e algoritmos de limpeza e transformação de dados. A limpeza e transformação dos dados envolvem a verificação da consistência das informações, a correção de possíveis erros, o preenchimento ou a eliminação de valores nulos e redundantes e a geração de novos atributos derivados.

O pré-processador é composto basicamente por um conjunto de processos, que manipulam e geram metadados a partir de uma amostra de dados. Esses processos utilizam uma extensa biblioteca de técnicas e algoritmos de tratamento de dados, localizados no Repositório do REDIRIS. Todas essas técnicas e algoritmos são executados a partir dos

metadados que as representam, o que possibilita a inserção de novas técnicas, tornando o pré-processador um módulo extensível a qualquer nova implementação requerida.

No capítulo seguinte será apresentada uma revisão bibliográfica.

3 – REVISÃO BIBLIOGRÁFICA

Neste capítulo serão apresentados os aspectos conceituais e metodológicos, mostrando os principais assuntos relacionados ao tema desta dissertação.

3.1 – CONSTRUÇÃO DE DATA WAREHOUSE

Nessa sessão, serão discutidos tópicos relevantes de trabalhos relacionados à construção de um Data Warehouse, como: uma breve revisão sobre granularidade e integração de dados, uma metodologia baseada em dados [INM97] e Banco de Dados Amostra Viva [INM97]. Esse capítulo apresentará ainda uma pequena revisão de sobre mineração, pré-processamento, qualidade e exploração de dados. Todos esses tópicos foram abordados por se tratar de assuntos importantes na construção de um DW e principalmente para elaboração desta dissertação.

3.1.1 – METODOLOGIA BASEADA EM DADOS

Diversos autores propõem metodologias para desenvolvimento e construção de Data Warehouse. [KIM98] criou a metodologia *Bussiness Dimensional Lifecycle.* [GOL98] defende o uso de uma metodologia com suas etapas de projeto tendo como base um modelo conceitual gráfico, chamada de *Dimensional Fact Model(DFM).* [GIL00] propõe uma metodologia baseada nos princípios da Orientação a Objetos. Todas essas metodologias são importantes e possuem seus pontos fortes e fracos, entretanto a metodologia de maior relevância para esse trabalho é a metodologia baseada em Dados, descrita em [INM97], que mostra o fluxo de construção de um DW partindo dos processos e dados existentes, com um forte apelo ao reaproveitamento de dados e ao reconhecimento da redundância.

Uma metodologia baseada em dados constrói novas implementações sobre o código e os dados já existentes, aproveitando o que já existe e criando o que não existe, considerando que essas novas implementações podem e deverão ser aproveitadas no futuro. Essa metodologia é de uso geral e possui 3 (três) fases, a primeira trata os sistemas e o processamento operacional, a segunda é usada para construção de Data Warehouse e a terceira parte descreve a utilização iterativa do Data Warehouse. Dentro do contexto desse trabalho a fase mais relevante é a que descreve a construção do Data Warehouse(METOD2).

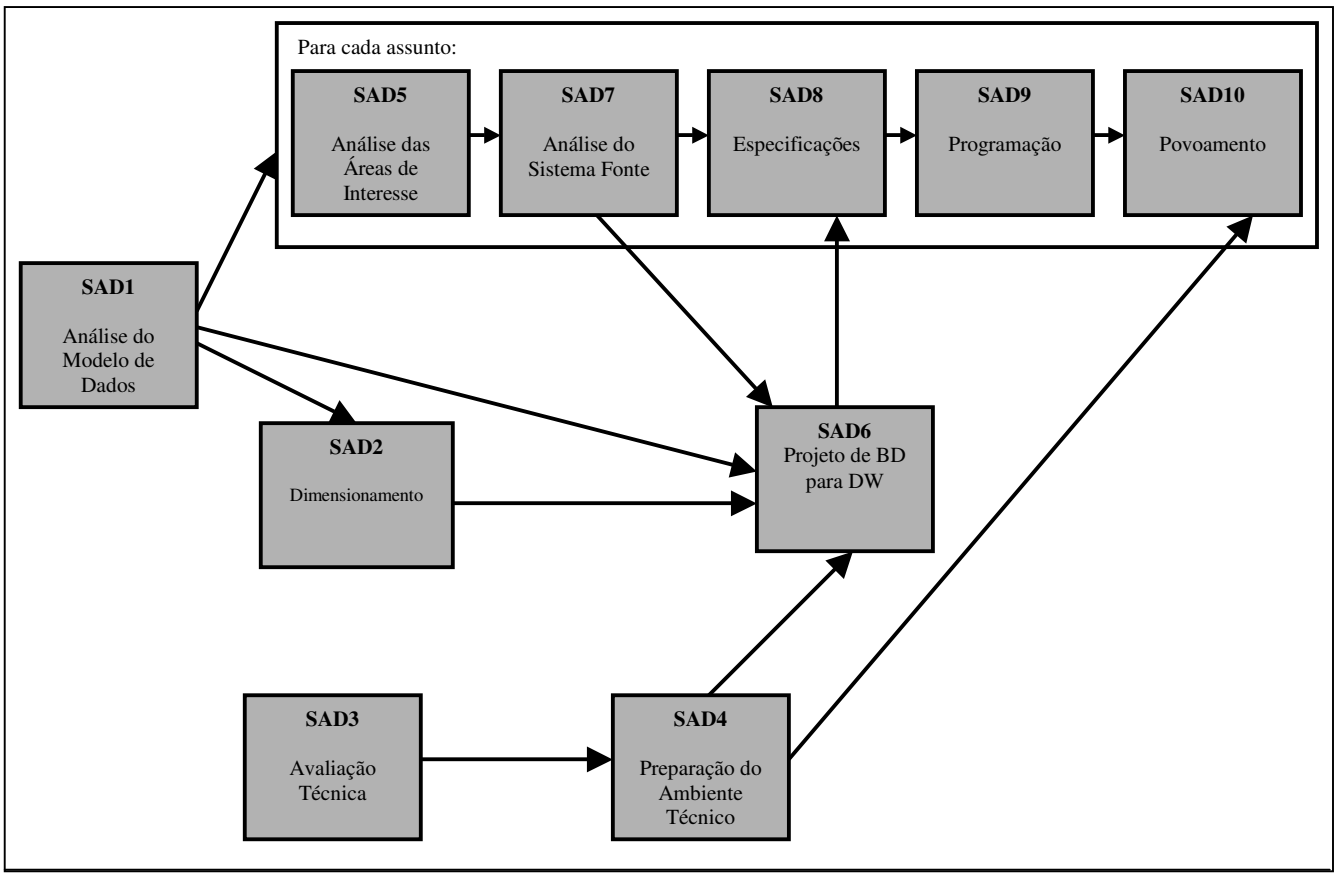

Figura 3.1 – Etapas da fase METODO2

A fase de construção de um Sistema de Apoio à Decisão da metodologia baseada em dados, que tem como ênfase o armazenamento central de dados em um Data Warehouse, é chamada de METOD2 e possui 10(dez) atividades, aqui descritas como SAD1 a SAD10 (Figura 3.1) [INM97]. Essas atividades não possuem um fluxo plano e linear, sendo executadas de maneira iterativa. Elas podem ser repetidas em situações diversas, diferentemente do processo cíclico de repetição para refinamento.

As atividades são as seguintes:

- **SAD1** – Análise do Modelo de Dados

Essa etapa, que é normalmente executada apenas uma vez, identifica as principais áreas de interesse do SAD e delimita as fronteiras do modelo de dados . É feita uma análise nos dados operacionais existentes, os dados primitivos e os derivados são separados. Para cada área de interesse são identificados os atributos, as chaves, os relacionamentos entre agrupamentos de atributos, os dados repetidos e

os tipos de dados. É gerado um documento descrevendo os dados que farão parte do SAD e os dados que permanecerão apenas operacionais.

Como resultado dessa atividade deve ser definido um modelo de dados, que atenda a todos os critérios especificados dentro de um padrão de qualidade definido.

- **SAD2** – Dimensionamento

Uma vez que o modelo de dados (SAD1) tenha sido elaborado e que possua a qualidade, o próximo passo é a análise de granularidade (dimensionamento). O dimensionamento visa estimar logo no início do processo se o volume de dados será um problema relevante a ser analisado. Nessa etapa, é projetada em termos brutos, a quantidade de dados que o DW vai armazenar. Como resultado dessa etapa tem-se a análise da necessidade da existência de vários níveis de granularidade. Se o DW não for destinado a conter uma grande quantidade de dados, não se faz necessário planejar vários níveis de granularidade. Se for detectado que o DW vai possuir vários níveis de granularidade, a definição de quais serão esses níveis, faz parte do resultado dessa etapa.

- **SAD3** – Avaliação Técnica

Essa atividade está diretamente ligada às definições técnicas do Data Warehouse, normalmente é executada apenas uma vez e não existe uma atividade como pré-requisito que é fundamental para execução. Será avaliada a capacidade de: gerenciar grandes quantidades de dados, organizar e permitir que dados sejam manipulados de maneira flexível, verificar a conformidade com um modelo de dados e a sua capacidade de carga de grandes volumes de dados periodicamente.

- **SAD4** – Preparação do Ambiente Técnico

Essa atividade consiste em identificar tecnicamente como acomodar os requisitos gerados pela Avaliação Técnica (SAD3). Essa atividade trata diretamente os assuntos relacionados com as restrições técnicas que tem que ser eliminadas. São questões técnicas como: a rede que vai ser usada, a quantidade de DASD (Direct Access Storage Device) necessária, o sistema operacional que vai gerenciar o

DASD, o software usado para gerenciar o Data Warehouse, o próprio Data Warehouse.

- **SAD5** – Análise das Áreas de Interesse

Essa atividade é normalmente executada apenas uma vez por cada projeto de povoamento que exista e após a atividade de Análise do Modelo de Dados. Agora é selecionada a área de interesse a ser povoada. A primeira área de interesse a ser selecionada deve ser suficientemente grande para ter sentido e suficientemente pequena para ser implementada. Uma área pode também ser selecionada parcialmente. O resultado desse passo é um escopo de empreendimento em relação a uma área de interesse (ou assunto).

- **SAD6** – Projeto de Data Warehouse

Depois de feita a Análise do Modelo de Dados, a Análise do Sistema Fonte e o Dimensionamento deve ser feito o projeto físico de banco de dados para o warehouse. O DW é projetado com base em um modelo de dados já definido. Em seguida é necessário pensar em: como acomodar as diferentes granularidades (caso existam), orientar os dados para os principais assuntos da empresa, projetar a variação em relação ao tempo que cada registro de dados apresenta, desnormalizar dados caso seja necessário (por questões de desempenho) e cria os meios de carga dos dados do sistema operacional para o Data Warehouse. Quando esse passo é apropriadamente executado, o resultado é um DW que contém uma quantidade de dados gerenciável, que podem ser carregados, acessados, indexados e pesquisados de modo eficiente.

- **SAD7** – Análise do Sistema Fonte

Após ser feita a Análise das Áreas de Interesse, ou seja, que tenha sido definido o assunto a ser povoado, essa atividade consiste em identificar nos ambientes de sistemas existentes a origem dos dados para um assunto específico. É nesse ponto que as questões de integração começam a ser tratadas, pois as fontes podem ser originadas de diversos lugares. O resultado dessa atividade é o mapeamento dos dados operacionais para o ambiente SAD.

- **SAD8** – Especificações

Após a interface entre o ambiente operacional e o ambiente SAD ter sido delineada, agora é necessário especificar os programas que efetivamente irão realizar esse trabalho. Isso só é possível porque na atividade de Análise do Sistema-Fonte todos os dados operacionais já foram mapeados e durante a atividade de Projeto de Data Warehouse já foi definido. Esse passo permite que a extração e a integração sejam feitas de forma mais eficiente. O resultado dessa atividade é a efetiva descrição dos programas e tarefas de aplicativos que serão usados para a carga do SAD.

- **SAD9** – Programação

Essa atividade só pode ser executada após ser feita a especificação e ela inclui todas as atividades padrão de programação em linguagem de programas ou em linguagem de aplicativos de *Extraction, Transformation and Loading* (ETL) como desenvolvimento de pseudocódigo, codificação, compilação, agendamento e bateria de testes. O resultado é a criação de todos os programas para extração e integração, incluindo a perspectiva de tempo. Após essa atividade o próximo passo natural é o Povoamento

- **SAD10** – Povoamento

Esse passo consiste em executar o que foi gerado com a Programação, tratando questões como: frequência de povoamento, eliminação de dados povoados, execução de programas de verificação de obsolescência de dados povoados, gerenciamento de vários níveis de granularidade e renovação de dados de amostra viva, esse banco de dados será abordado na próxima sessão. O resultado deve ser um DW povoado e funcional.

3.1.2 – BANCO DE DADOS AMOSTRA VIVA DE INMON

Outro conceito importante e que também será abordado nesta sessão é o Banco de Dados “Amostra Viva” descrito em [INM97]. Ele consiste em um Data Warehouse montado a

partir de um subconjunto de dados (amostra) ou dados levemente resumidos. O termo "Amostra Viva" é derivado do fato de ser montado a partir de uma amostra de dados de um banco de dados maior e da necessidade periódica de ser renovado (Figura 3.2).

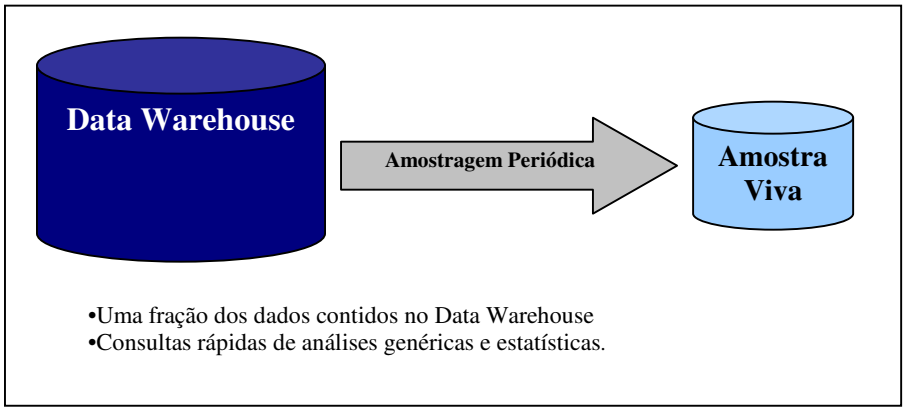

Figura 3.2 – Banco de Dados Amostra Viva

Um exemplo onde uma aplicação Banco de Dados Amostra Viva pode ser muito útil é numa análise estatística de população. Porém, existem limitações e problemas específicos dessa configuração. Caso deseje-se buscar por registros específicos e individuais de dados, corre-se o risco deste registro pertencer a massa do DW inteiro, mas não está presente na amostra. Por isso essa configuração não é indicada para consultas onde essa situação pode existir. Um outro problema é que por se tratar de uma amostra aleatória, existe a possibilidade da amostra não representar a massa inteira. Nesse caso, deve-se prover mecanismos para se verificar e validar a representatividade da amostra. A amostragem deve ser feita sempre seguindo métodos estatísticos que verificam a representatividade da amostra após a coleta.

A produtividade do analista de SAD depende da velocidade de resposta da análise que está sendo feita. Consultas sobre um banco de dados baseado em amostra é muito mais eficiente em termos de performance, pois a base contém apenas uma fração da massa total do DW (exemplo: 1/1.000 ou 1/10.000). Com isso, um analista que leva horas para fazer uma determinada análise envolvendo a varredura do banco de dados, pode fazê-la em minutos ou até mesmo em segundos numa amostra. Essa situação piora quando consideramos a atividade de exploração de informações em sistemas de apoio à decisão é baseada em consultas heurísticas, onde o analista estuda um resultado, para fazer novas consultas.

Um aspecto muito criticado com relação a consultas em cima de amostras estatísticas é a produção de respostas relativamente "incorretas". Por exemplo, considere uma pesquisa

feita em 100 milhões de registros e chega-se a conclusão que 53,7% são homens; essa mesma pesquisa é feita em 25 mil registros e o resultado é que 54,1% são homens. Apesar da imprecisão das respostas, o tempo de processamento das consultas na amostra poderá compensar a pequena margem de erro. Essa compensação se torna mais evidente, caso as análises sejam heurísticas, ou seja consultas gerando mais consulta, o que torna a imprecisão irrelevante diante do esforço e tempo economizado. Mesmo que se deseje números exatos, é possível utilizar um Banco de Dados Amostra Viva para as investigações heurísticas, determinando quais as consultas são relevantes e depois submetê-las ao DW completo para os resultados serem exatos.

No Banco de Dados Amostra Viva [INM97] a base de dados resultante é gerada a partir de uma amostra de um Data Warehouse. O seu uso é restrito e indicado apenas para agilizar as consultas ou para usuários que demandem por análises genéricas e estatísticas. Mais adiante será visto que a idéia de amostragem para geração de um Data Mart com proporções menores é defendida, porém essa amostragem deverá ser feita a partir dos dados operacionais, antes mesmo que o DW esteja pronto.

3.1.3 – GRANULARIDADE

A definição do nível de granularidade se constitui um dos passos mais importantes da construção de um DW [INM97] [KIM97]. A granularidade representa o nível de detalhamento dos dados contidos no DW. Quanto menor o nível de detalhes mais alto é o nível de granularidade e vice-versa. A importância do nível de granularidade em um DW determina o volume de dados armazenados e o tipo de consulta que pode ser atendido e deve ser balanceado de acordo com o detalhamento requerido nas consultas [INM97]. [KIM97] defende que em quase todas as situações um menor nível de granularidade para suas dimensões deve ser usado, para atender ao maior número de consultas possíveis, inclusive as não esperadas. Ele também argumenta que isso torna o DW muito mais “resistente” a novos elementos de dados. No entanto, essa adoção de uma granularidade muito baixa além de aumentar o volume de dados armazenados no DW pode também elevar a complexidade das consultas finais, se uma parte das consultas pretendidas envolvem operações mais complexas do que simples agregações. A escolha inadequada pode comprometer todo o resto do projeto. No capítulo 6 será abordado um exemplo prático onde uma granularidade muito baixa pode não ser aplicável.

Se tomarmos como base um modelo estrela, a definição da granularidade corresponde ao menor grão que será armazenado na tabela de fatos e a granularidade da tabela de dimensão não poderá ser menor do que a da tabela de fatos. N o entanto se essa for maior não irá acarretar qualquer contradição lógica [KIM98], mas poderá causar a perda de informações úteis para o usuário.

A metodologia FastCube que será apresentada no Capítulo 4, está em conformidade com Kimball [KIM97], pois ele afirma que a forma mais rápida de identificar as dimensões é na definição da granularidade da tabela de fatos.

> "Uma definição cuidadosa do grão determina as dimensões primárias da tabela de fatos. Geralmente, será possível em seguida adicionar outras dimensões ao grão básico da tabela de fatos, sendo que essas dimensões adicionais devem produzir um único valor para cada combinação das dimensões primárias. Se for constatado que uma dimensão acrescentada viola o grão gerando registros adicionais, então a definição do grão deve ser revisada para acomodar a dimensão adicional" (KIMBALL, 1997:27).

Essa citação sugere que granularidade deve ser definida e em seguida incrementar o modelo dimensional pouco a pouco. De forma análoga a metodologia FastCube propõe uma técnica que direciona a construção do modelo, após a definição do nível de uma granularidade, de forma iterativa e incremental.

3.1.4 – INTEGRAÇÃO DE DADOS EM DW

Uma das principais características de um Data Warehouse é o seu aspecto integrador de dados [INM97]. Os dados que estão em um ambiente de warehouse são em principio integrados. Quando se imagina que o Data warehouse é composto por um gama de fonte de dados diferentes e com características diferentes, conclui-se que deve haver uma grande necessidade de adequação dos dados a esse novo ambiente integrado. A necessidade da integração está presente em muitas formas: na padronização dos nomes, na forma de mensurar o atributo e na consistência dos diversos códigos.

Os contrastes e diferenças encontradas nas diversas fontes de dados revelam também a falta de padronização no modelo de construção e implementação dos modelos de dados. Essa falta de padrão é verificada não só entre fontes de diferentes organizações, como também de uma mesma empresa. A Figura 3.3 caracteriza alguns desses problemas encontrados:

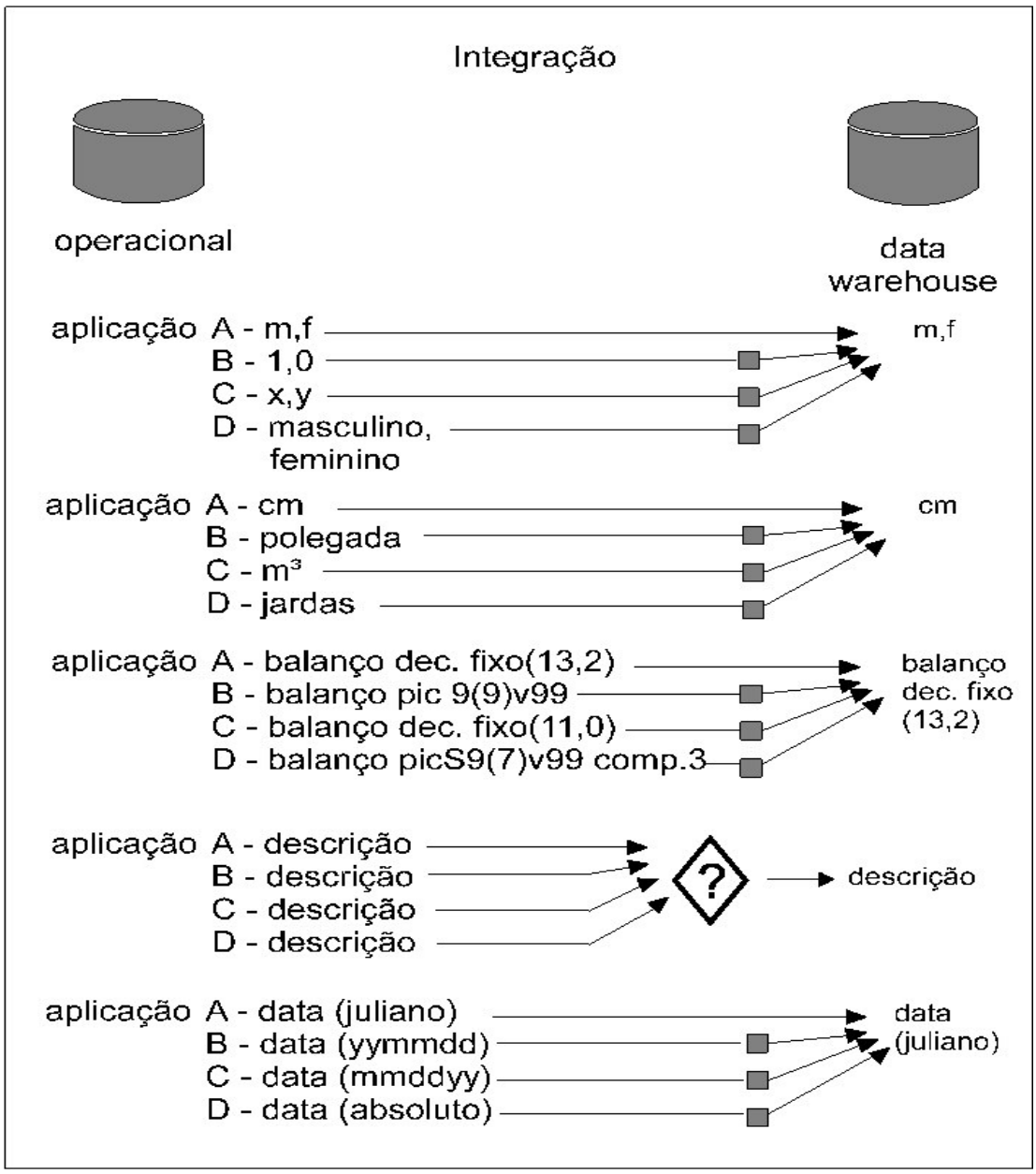

Figura 3.3 – Problemas Típicos de Integração de Dados

3.2 – MINERAÇÃO DE DADOS

A proposta de extrair conhecimento de banco de dados surgiu devido à explosão do crescimento da quantidade de dados armazenados em meios magnéticos e da necessidade de aproveitá-los, motivada pela "necessidade de conhecimento".

Mineração de dados é um dos passos de um processo maior denominado de Descoberta de Conhecimento em Base de Dados (DCBD) – Knowledge Discovery in

Databases (KDD) –, que é realizado por ferramentas computacionais em desenvolvimento para crescentes volumes de dados [KUR98]. As etapas do processo de KDD são mostradas pela Figura 3.4. Mineração de dados é um termo genérico utilizado para métodos e técnicas computacionais visando a extração de informações úteis de um grande volume de dados.

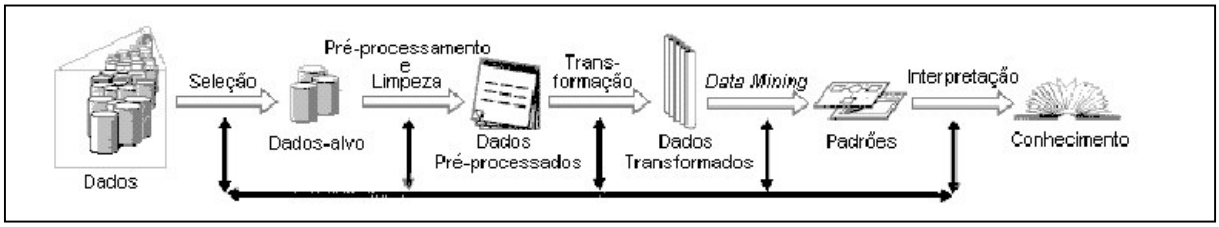

Figura 3.4 – Etapas do Processo KDD [FAY96]

A mineração de dados é considerada a principal fase do processo de KDD. Essa fase é exclusivamente responsável pelo algoritmo minerador, ou seja, o algoritmo que diante da tarefa especificada, busca extrair o conhecimento implícito e potencialmente útil dos dados. A mineração de dados é, na verdade, uma descoberta eficiente de informações válidas e não óbvias de uma grande coleção de dados [BIG96].

Durante a fase de Mineração de Dados necessita-se definir a técnica e o algoritmo a ser utilizado em função da tarefa proposta. A Tabela 3.1 mostra as principais tarefas de KDD e as técnicas mais utilizadas para mineração de dados.

Tarefas de KDD	Técnicas
Associação	Estatística e Teoria dos Conjuntos
Classificação	Algoritmos Genéricos, Redes Neurais e Árvores de Decisão
Clustering	Redes Neurais e Estatística
Precisão de Séries Temporais	Redes Neurais, Lógica Nebulosa e Estatística

Tabela 3.1: Algumas tarefas de KDD e suas técnicas de mineração de dados [AUR97]

Uma vez escolhido o algoritmo a ser utilizado, deve-se adaptá-lo ao problema proposto e implementá-lo através de uma ferramenta existente ou pelo desenvolvimento específico de uma nova ferramenta. A partir daí inicia-se o processo de mineração de dados, onde serão apresentados diversos padrões, que serão interpretados para geração do conhecimento.

Quando se fala em mineração de dados, não está se considerando apenas consultas complexas e elaboradas que visam ratificar uma hipótese gerada por um usuário em função dos relacionamentos existentes entre os dados, mas principalmente a descoberta de novos fatos, regularidades, restrições, padrões e relacionamentos pouco visíveis ao especialista.

Na fase de mineração dos dados, o executor da tarefa pode utilizar várias ferramentas e técnicas para que o seu objetivo seja bem sucedido, esta envolve diversas áreas e técnicas, além dos principais algoritmos. A Figura 3.5 mostra uma taxonomia da fase de mineração de dados. Os algoritmos estão representados pelo símbolo (·), enquanto que as caixas representam áreas e técnicas.

Técnicas de mineração de dados são muito relevantes no processo de construção de DW, tanto na busca de informações não triviais, como no auxílio à análise e correção dos dados.

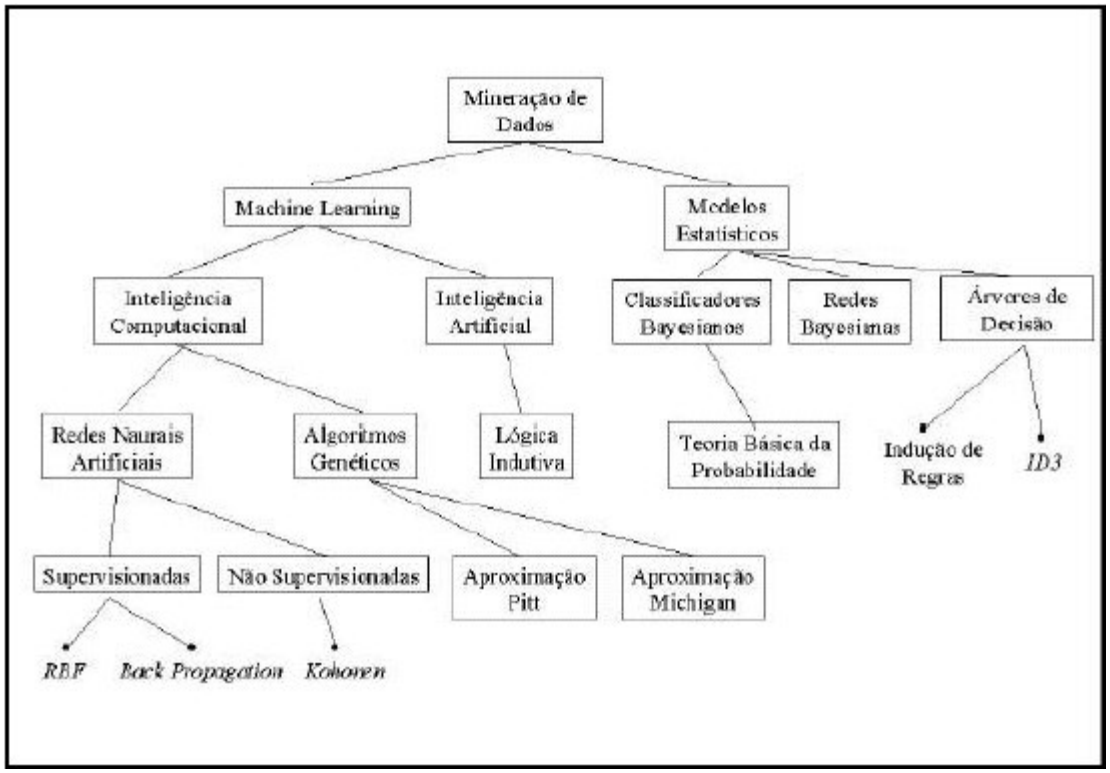

Figura 3.5 – Taxonomia das Técnicas de Mineração de dados [AUR97]

3.3 – TÉCNICAS E ALGORITMOS PARA PRÉ-PROCESSAMENTO DE DADOS

Muitos dos problemas encontrados na qualidade dos dados em uma base ocorrem pela fragilidade dos critérios da verificação na entrada da informação. Em muitos casos, as aplicações não prevêem os possíveis erros na digitação ou carga de dados nos SGBD`s (como datas inválidas ou referência a dados não cadastrados na tabela de origem) e permitem o armazenamento de informações sem qualidade. Após a identificação da origem do erro da informação, as regras envolvidas no processo de armazenamento do dado devem ser revistas e ajustadas, ou até mesmo reescritas, para solucionar o problema.

As rotinas de pré-processamento envolvem todo processo de transformação dos dados visando torná-los compatíveis, legíveis para análise (exemplo normalizar um campo) ou para melhorar a sua qualidade. Essa última opção é muito utilizada quando não são possíveis ajustes na fonte original (operacional) de dados, necessitando assim um pré-processamento mais rigoroso.

De maneira geral, pode-se dizer que os motivos que levam os dados a serem transformados nas rotinas de pré-processamento são:

- Melhoria na apresentação dos dados.
- Baixo grau de correção – dados incorretos.
- Abrangência inadequada – não atende às demandas do usuário, precisa ser completado.
- Falta de consistência – dados conflitantes.
- Baixo grau de completeza – dados incompletos (nulos).
- Incoerência com regra de negócio ou fora do domínio
- Precisão inadequada.

A seguir serão apresentadas algumas técnicas mais comuns para tratamento e transformação de dados. Para entendimento das técnicas a seguir considere que o tratamento de dados é por coluna individualmente ou por coluna

A *Normalização* tem o propósito de minimizar os problemas oriundos do uso de unidades e dispersões distintas entre variáveis. As variáveis podem ser normalizadas segundo a amplitude ou segundo a distribuição dos seus valores (ver Figuras 3.6 e 3.7). A

normalização segundo a amplitude deve ser usada quando existem unidades diferentes ou dispersões muito heterogêneas. A normalização distribucional é interessante em situações como: remoção de distorções de valores aberrantes e obtenção de simetria. Essa técnica é muito usada para beneficiar a mineração de dados como redes neurais, algoritmos genéticos, KNN, clustering...[CAR01] [PYL99] [WIT99] [HAN01] [FAY96].

A técnica de *Agrupamento* possibilita a junção de um conjunto de ocorrências seguindo um determinado critério. Esse critério pode até ser um algoritmo inteligente de Clustering ou um peso de uma rede neural [PYL99].

$$\sqrt{x} \qquad \log(x) \qquad -\frac{1}{x}$$

Figura 3.6 – Fórmula para Normalização segundo a Distribuição

$$a)\ y = \frac{x - m}{s} \qquad b)\ y = \frac{x - \min}{\max - \min}$$

$$c)\ y = \frac{x}{10^k}, \text{ para o menor k tal que } \max\left(\left|\frac{x}{10^k}\right|\right) < 1$$

Figura 3.7 – Fórmula para Normalização segundo a Amplitude

A *Discretização* dos dados faz com que o domínio dos atributos seja reduzido possibilitando a melhor compreensão e/ou visualização dos dados [AUR97] [PYL99] [WIT99]. Ela também é muito importante em pré-processamento para Data Mining. Essa técnica é considerada uma espécie de agrupamento. Exemplo: transformar o campo idade em maior 18 anos e menor 18 anos.

A técnica de *Agrupamento Outros* consiste em agrupar em apenas uma classe as ocorrências de menor frequência. É muito utilizada para melhorar tanto a visualização das classes mais relevantes quanto o aprendizado em mineração de dados, valorizando as classes de maior ocorrência [WIT99] [PYL99].

O algoritmo de *Similaridade ou Matching* visa procurar ocorrências que deveriam ser iguais, mais por algum problema na entrada de dados (erro de digitação, por exemplo) foram cadastrados de maneira diferente. Através de uma regulagem de precisão pode-se dar pesos a comparação entre dois strings e determinar se são a mesma ocorrência ou não.

Na aplicação de *Banco de Conhecimento* existem algumas formas de uso do conhecimento adquirido em tratamentos de colunas similares anteriores. Uma dessas aplicações muito usada, é a comparação de 2 variáveis de alta correlação onde uma determina a outra, exemplo estado e cidade. Tendo a variável cidade preenchida corretamente e um

banco com todas as possíveis cidades e seu respectivo estado, é possível com um certo grau de acerto determinar o estado correspondente. Uma outra maneira de utilizar essa técnica é guardar todas as ocorrências certas ou as anomalias de um atributo e o seu valor correto, assim as ocorrências podem ser comparadas a uma base de dados e o valor correto a ser substituído.

A *Codificação de dados* consiste em pegar uma ocorrência discreta e transformar em numérica. Essa codificação pode ser de valores discretos para valores binários, para o código termômetro ou simplesmente normalizados [AUR97] [PYL99]. Essa técnica é muito usada para gerar atributos da tabela FATO. Porém, seu maior uso é para pré-processar dados que servirão de entrada para técnicas de Data Mining que trabalham apenas com dados numéricos.

Existem outras possibilidades que não enquadram bem nas técnicas acima, mas que são muito usadas como a divisão de informação, por exemplo, separar a sigla de estado da cidade, ou mesmo separar os dígitos do CEP para melhorar o entendimento de localização. O inverso também é verdadeiro, pois muitas vezes deseja-se a junção de dois campos de maneira matemática (renda declarada mais renda não declarada = renda total) ou de uma outra forma.

As possibilidades são ilimitadas porque o tratamento de dados pode ser obtido através da aplicação de diversas técnicas em seqüência. Torna até difícil fazer uma classificação, visto que algumas técnicas são na verdade a especialização de outra mais genérica ou correspondem à aplicação de duas ou mais técnicas em conjunto.

O conhecimento da potencialidade destas técnicas poderá ajudar o projetista em todas as fases de um projeto, que se precisa manipular dados. Elas podem ajudar na interpretação dos dados, no tratamento de anomalias dos dados, na validação de regras de negócio, descoberta de novas características dos dados, e até mesmo na elaboração de formulário mais eficientes para entrada de dados operacionais. Nessa dissertação serão implementadas algumas técnicas de pré-processamento no análise, manipulação e tratamento de dados.

3.4 – QUALIDADE DOS DADOS

O alto custo da baixa qualidade de dados é imensurável e impacta em todas as áreas das instituições e do cotidiano das pessoas. Os custos envolvem: tempo, dinheiro e

prejuízos físicos. A qualidade dos dados deve ser requerida em todos os níveis das organizações, seja ele operacional, tático ou estratégico. Com o advento dos sistemas de apoio à decisão, a qualidade dos dados tem sido um assunto que preocupa os especialistas, porque os SAD são muito dependentes dos dados operacionais de entrada (que normalmente não possuem uma boa qualidade) e dos modelos e técnicas de transformações desses dados em modelos para apoio à decisão.

3.4.1 – QUALIDADE DOS DADOS NO DW

Ainda hoje a fase que mais demanda tempo em um projeto de construção de Sistema de Apoio à Decisão [CHE01] é a fase de preparação e pré-processamento de dados. No entanto, a grande preocupação das equipes de desenvolvimento de DW está no projeto do modelo e da aplicação do usuário final (ferramenta OLAP) [CAM01]. Como conseqüência se tem um número muito grande de erros no processo de tratamento dos dados, refletindo diretamente na qualidade final dos dados, levando ao usuário final (analistas), quase sempre, a perda de confiança no DW.

A preocupação com a qualidade dos dados das fontes de informação preocupa os especialista, mas na prática pouco se tem realizado.

> "Outra função que deve ser executada na implementação do Data Warehouse é assegurar que a qualidade dos dados adquiridos das origens de produção seja alta o suficiente para satisfazer as necessidades de informação da organização" (INMON, 1998:267).

Apesar dessa preocupação, [INM98] sugere que os dados sejam corrigidos em suas fontes, cabendo especificamente ao Analista de Qualidade de Dados (DQA - Data Quality Analyst) a identificação e a comunicação aos responsáveis pelos dados de suas fontes. Sabe-se que essa postura algumas vezes não pode ser aplicada, porque nem sempre os dados podem ser corrigidos nas suas origens. Um exemplo disso é quando os dados são cedidos de uma organização para outra, onde a que recebe não tem nenhum vínculo direto com a organização que cedeu os dados. Os dados existem, logo uma alternativa é fazer o tratamento dos dados antes da carga do DW, no processo chamado de pré-processamento de dados. A política de consertar os dados sem recorrer as suas fontes, através de uma rotina de pré-processamento no Data Warehouse [CHE01] é geralmente defendida por pesquisadores que usam o Data Warehouse, como fonte de dados para fazer Mineração de Dados. Uma política correta de construção de DW é fazer sempre a medição da qualidade dos dados que estão

entrando no processo de construção do DW e medir a qualidade dos dados ao final do processo de ETL.

O sentido da qualidade é bastante abrangente. Em apoio à decisão a qualidade está associada: a consistência e a coerência dos dados e a facilidade de entendimento e de tomada de decisão. Os projetos de Data Warehouse falham por diversas razões, mas todas elas podem ser resumidas em uma única falha, a falta de qualidade [ENG99]. São componentes de falta de qualidade: definição de uma arquitetura pobre de dados, dificuldade para integrar dados de diferentes fontes, dados departamentais definidos sem consistência, a ausência de dados, a imprecisão de dados, as respostas lentas de solicitações, a ausência do usuário no processo de construção do data warehouse. Sendo a informação o produto do Data Warehouse, a presença dessas falhas implicará diretamente na qualidade dos resultados obtidos através do mesmo.

3.4.1 – PROCESSO DE QUALIDADE DE DADOS

A qualidade é uma saída diretamente dependente do objeto (físico, conceitual ou lógico) de análise que se deseja mensurar. Ela depende de todos os níveis e pespectivas do Data Warehouse [QUI98]. E ainda deve ser analisada na perspectiva de metas de qualidade de diferentes etapas da implementação [QUI98] [CAM01]. A Figura 3.6 ilustra a medição de qualidade nas diversas etapas de construção do DW, na perspectiva do fluxo de dados.

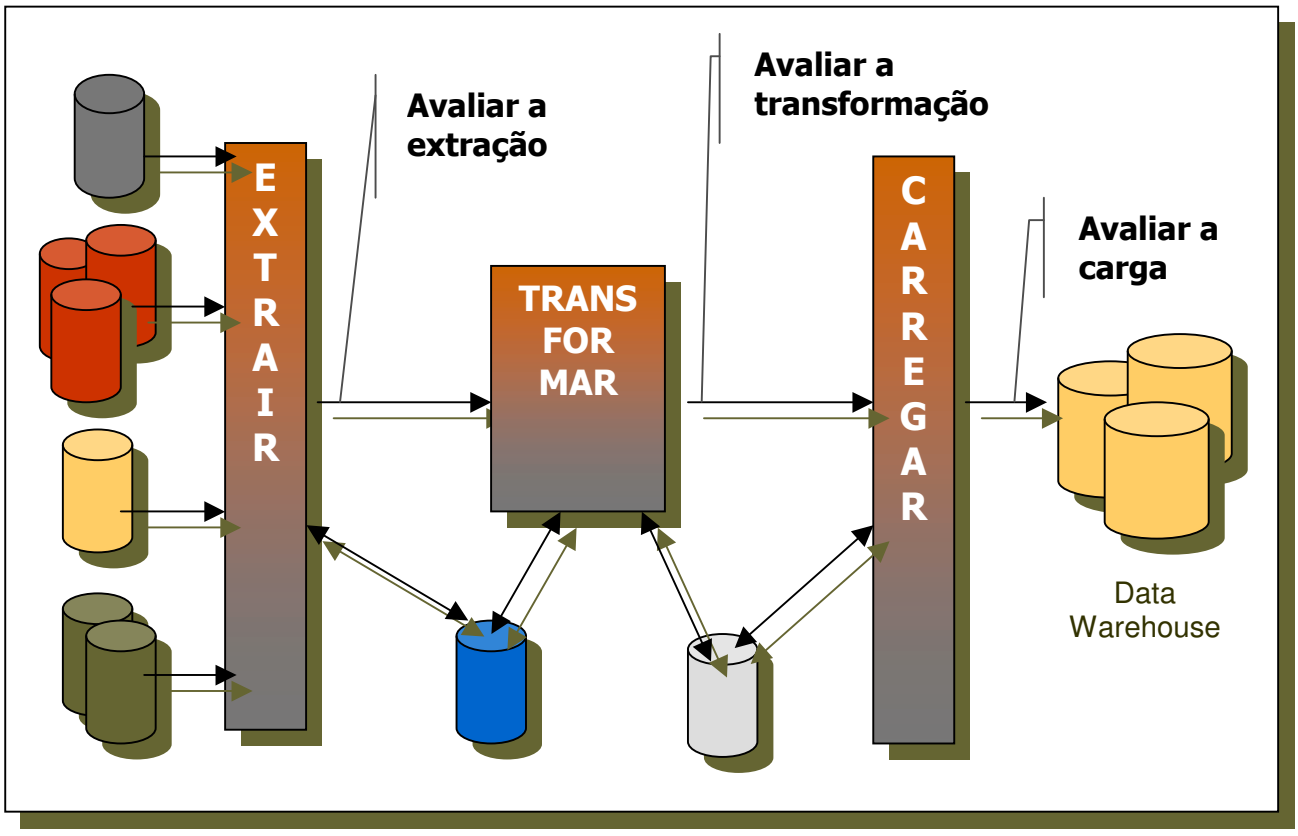

Figura 3.8 – Medição da Qualidade na Construção de um DW [CAM01].

Existem diferentes papéis de usuários em um ambiente de Data Warehouse e cada um desses usuários possui visões diferentes do que é a qualidade no seu escopo de trabalho. Para o Analista Decisor que utiliza uma ferramenta de busca OLAP, ele precisa de respostas relevantes para suas buscas, ele está interessado com a qualidade dos dados armazenados, com seu tempo útil e com a facilidade de buscá-los através das ferramentas OLAP. O Administrador de um Data Warehouse precisa de facilidades como: relatórios de erro, acessibilidade de metadados e conhecimento do tempo útil dos dados, a fim detectar mudanças e razões para elas, ou problemas com a informação armazenada. O Designer do Data Warehouse precisa medir a qualidade do esquema da estrutura do Data Warehouse (existente ou a ser produzida) e, também, a qualidade dos metadados. Além disso, ele ainda precisa definir padrões de avaliação do software para testar os pacotes de software que ele deve considerar para comprar.

A qualidade de dados deve também ser vista sob a ótica dos consumidores de dados. Em Data Warehouse o nível de exigência é maior pelo risco de se tomar decisão com dados inapropriados. O projeto TQDM (Total Data Quality Management) [WAN96] desenvolvido pelo MIT (Massachust Institute Technology) tem como principal objetivo o de prover uma base formal para a qualidade de dados. Foram definidos diversos critérios de qualidade, o grau de importância de cada critério e uma estrutura hierárquica dos critérios. Foram definidos 118 (cento dezoito) critérios, que foram consolidados em 15(quinze) dimensões agrupadas em quatro categorias: intrínseca, contextual, representacional e de acessibilidade, conforme descritas abaixo:

- Intrínseca: corresponde à qualidade inerente aos dados.
    - Precisão (valores corretos das fontes/número total de valores).
    - Objetividade
    - Taxa de erro (grau de correção)
    - Reputação dos dados e fonte de dados (confiabilidade)
- Contextual: critérios associados ao contexto das tarefas que estão sendo realizadas.
    - Valor agregado (benefícios associados ao uso)
    - Relevância dos dados
    - Tempo de atualização

- Completude

- Representacional: associada à maneira de se representar os dados, seus formatos e significados.
  - Facilidade de interpretação
  - Facilidade de entendimento
  - Representação consistente
  - Representação concisa

- Acessibilidade: corresponde a disponilibidade e forma de acesso aos dados.
  - Grau de acessibilidade (percentual de tempo de disponibilidade de dados)
  - Segurança de acesso (nível de privacidade entre fontes e consumidores)

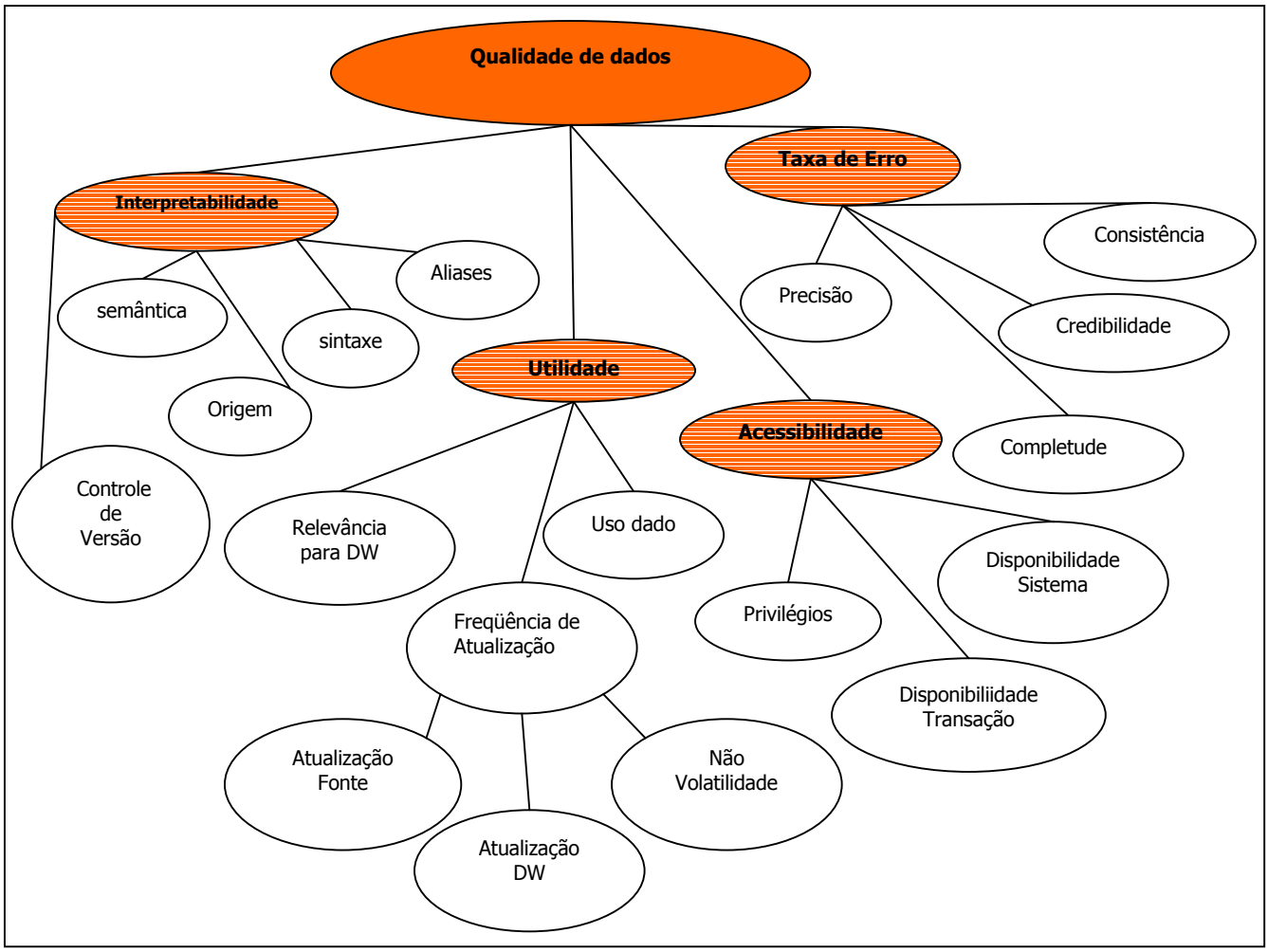

Figura 3.9 – Critérios de Qualidade

A qualidade dos dados que são armazenados no "warehouse", não é, obviamente, um processo por se só; contudo, é influenciado por todos os processos que ocorrem no ambiente do "warehouse". O projeto DWQ (Data Warehouse Quality) [JAR97][JAR00] propõe um modelo formal de qualidade de dados com critérios para data warehouse e o relacionamento entre estes e as tarefas realizadas em um DW, inspirados nos critérios definidos em [WAN96], conforme Figuras 3.9 e 3.10. Os principais objetivos DWQ: Enriquecer as semânticas de meta banco de dados com modelos formais de qualidade de informação DW, enriquecer as semânticas dos modelos de recursos de informação para habilitar propagação de mudança incremental e resolução de conflito e enriquecer as semânticas dos modelos de esquema de DW para capacitar os projetistas e os otimizadores de consulta e tirar proveito da natureza temporal, espacial e agregada de dados de DW.

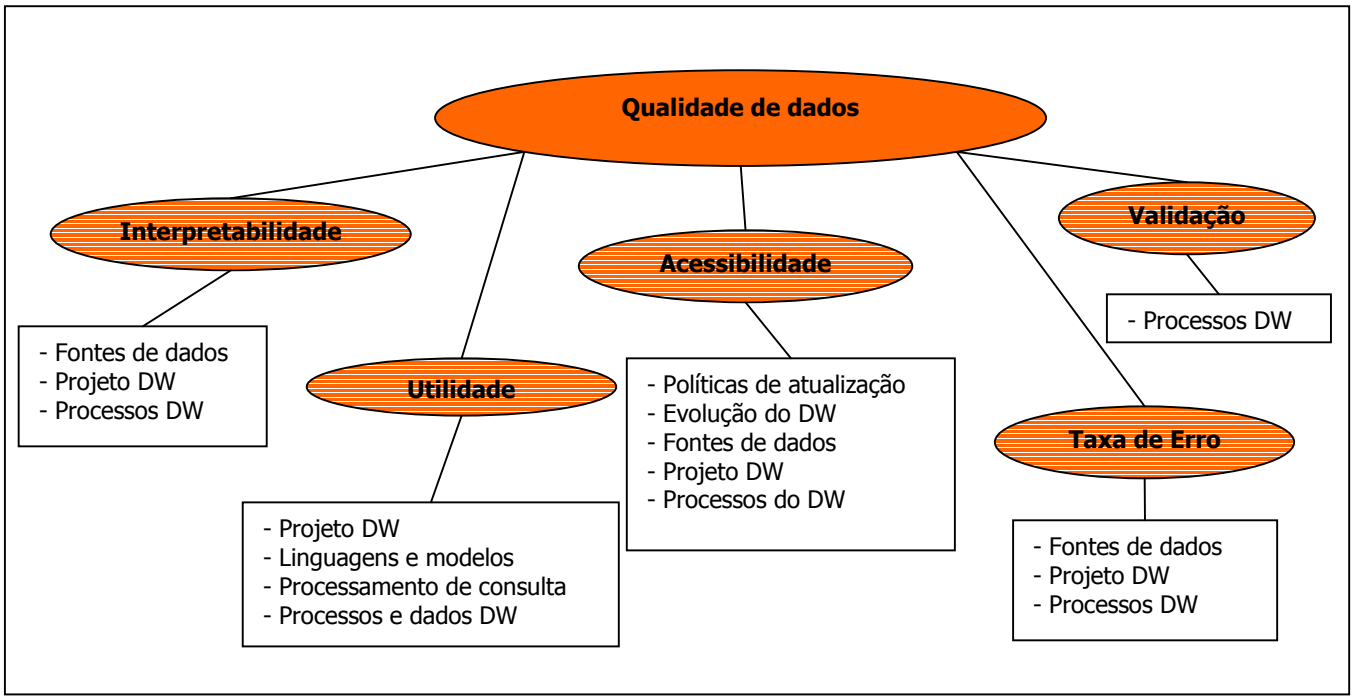

Figura 3.10 – Critérios de qualidades e as tarefas realizadas em um DW [JAR00]

Especificamente para qualidade de dados em [JAR97a] propõe o uso de apenas um subconjunto de dimensões propostas em outros modelos. Ainda segundo [JAR97a] a noção de qualidade dos dados, em sua maior parte, deve ser tratada como um segundo nível de dimensão, chamada credibilidade (believeability). As dimensões básicas de qualidade de dados são mostradas na Figura 3.11e explicadas a seguir.

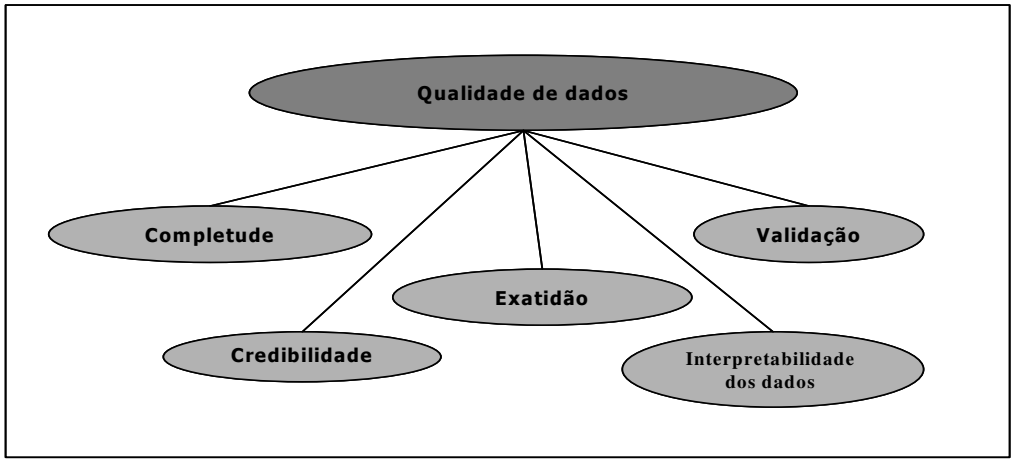

Figura 3.11 – Dimensões de qualidade propostas por [JAR97a]

A dimensão da qualidade dos dados não está relacionada diretamente com o processo do Data Warehouse, ela é produto de uma consulta diretamente às propriedades dos dados armazenados sem avaliar diretamente esquemas ou modelos. Conseqüentemente, está relacionada a perspectiva física da arquitetura representando os dados armazenados e de seus elementos em todos os níveis. A dimensão da completude descreve a porcentagem da informação do mundo real incorporada às fontes e/ou ao warehouse. Por exemplo, a completude poderia avaliar se o tamanho de uma cadeia de caracteres (string) que descreve um endereço corresponde ao tamanho do atributo necessário para representar o endereço. A dimensão de credibilidade descreve a confiabilidade da fonte fornecedora da informação. A dimensão de exatidão descreve a precisão do processo de introdução de dados que aconteceu nas fontes (cadastros, cargas etc). A dimensão de consistência descreve a coerência lógica da informação. A dimensão de interpretabilidade dos dados é concernida com a descrição dos dados, como: a disposição de dados dos sistemas legados e dos dados externos, a descrição da tabela para bases de dados relacionais, chaves primárias e estrangeiras, os pseudônimos, os domínios e a explanação de valores codificados.

O primeiro passo para assegurar a qualidade é definir a expectativas de qualidade dos dados necessária para dar suporte as metas do negócio do SAD [HUF96]. Para saber se essas expectativas foram cumpridas é necessária a definição de métricas capazes de avaliar a qualidade dos dados. Nesse contexto que entra a definição de metadados para dar suporte ao processo de qualidade de dados. Em [JEU98] é apresentado um modelo de metadados para gerencia de qualidade em Data Warehouses em especial seguindo os requisitos do DWQ [JAR98] [JAR00].

Características de qualidade de dados	Descrições	Exemplo de Métrica
Exatidão	Grau de acerto entre um conjunto de valores dos dados e um conjunto correspondente de valores corretos.	Percentagem dos valores que estão corretos quando comparados à característica do objeto descrito pelos dados.
Completude	Grau de preenchimento dos valores que estão presentes nos atributos que os requerem.	Percentagem dos dados que têm os valores incorporados neles.
Consistência	Concordância ou coerência lógica entre os dados levando em conta variações ou contradições de ocorrências.	Percentagem das condições combinadas do valor ou das condições satisfeitas derivadas do valor.
Relatividade	Concordância ou coerência lógica que permite a correlação razoável em comparação com o outros dados similares.	Percentual de condições de integridade referencial satisfeitas.
Tempo de Atualização (timeless)	O dados que são fornecidos em tempo requerido ou em tempo especificado.	Percentagem dos dados disponíveis dentro de uma janela de tempo especificada inicialmente (dias ou horas).
Unicidade	Valores dos dados únicos, que devem comportar-se como chaves candidatas. Cada valor é único de seu tipo.	Percentagem dos registros com violações da unicidade (valor duplicado da chave candidata).
Validade	Conformidade dos valores dos dados que são editados para a acessibilidade, reduzindo a probabilidade do erro.	Percentagem dos dados que têm os valores que ficam dentro de seus domínios respectivos de valores permissíveis.

Tabela 3.2: Características e métricas geralmente usado na qualidade dos dados [HUF96].

A Tabela 3.2 descreve um conjunto de características geralmente referenciado sobre qualidade dos dados e identifica uma métrica útil para especificar as exigências de qualidade nas medições nos dados reais [HUF96]. Em vez de definir a medida para todos os dados requeridos, é mais importante identificar as características que causam a situação de risco. A etapa seguinte é definir a medida para cada um destes elementos. É útil neste momento alinhar a análise do risco com o objetivo da aplicação. Por exemplo, se o objetivo da aplicação de um SAD for otimizar operações do transporte e manipulação para a entrega do produto, a completude dos atributos que representam o peso e local de entrega do produto, tornam-se bastante importantes.

Existem muitas propostas para a garantia de qualidade em um DW, com soluções das mais simples as mais complexas [HUF96] [JAR98] [JAR98a] [FIO00] [MAC00a] [MAC00b], cabe ao projetista de DW fazer uma avaliação da que melhor se enquadre em seu projeto, mas nunca esquecer que a qualidade tem que ser um dos pontos chaves do projeto de DW. No capítulo 5 abordaremos como é utilizada a qualidades de dados em nosso trabalho.

3.5 – EXPLORAÇÃO E ANÁLISE DE DADOS NO PRÉ-PROCESSAMENTO

A Exploração e Análise de Dados são elementos importantes que facilitam a preparação dos dados. Um dos objetivos de seus pré-requisitos é conseguir uma boa visualização dos mesmos, e quase sempre é necessário fazer algum uso de operações sobre os dados, preparando-os de tal forma, que possam ser mais bem visualizados e principalmente entendidos. Tais operações podem ser realizadas utilizando técnicas estatísticas, Data Mining ou KDD (Knowledge Discovery in Data Base) [CHE01] [HAN01]. Só com a visualização dos dados é possível fazer uma boa avaliação, possibilitando o emprego das técnicas adequadas de tratamento dos dados. Gerar novas colunas que possibilitem uma melhor visualização ou mesmo aproveitamento da informação também podem ser feitas através do tratamento de dados. Portanto o processo de análise e exploração em geral é iterativo e se mistura com processo de tratamento dos dados.

O processo de consultas OLAP e de mineração de dados, corresponde na verdade a aplicação de técnicas de exploração de dados em praticamente todas as etapas desses processos, para que a todo o momento e com objetivos diferentes, possamos visualizar os dados de uma forma entendível ou direcionada para o objetivo da fase correspondente. O processo de análise e exploração dos dados começa do tratamento dos dados. Por exemplo, quando começamos a etapa de pré-processamento de dados, seja para construção de um DW ou para Mineração de Dados, já está sendo realizada a análise e exploração de dados, porque essa etapa consiste em saber: que dados usar, como integrar as bases de dados, qual o nível de qualidade e preenchimento dos dados, de que maneiras esses dados deverão ser formatados e aproveitados... Quando termina a formatação dos dados para um DW ou para Mineração de Dados, começa-se outra fase que também consiste em análise e exploração dos dados, porém essa agora através de consultas OLAP na base do DW ou analisando os resultados após a aplicação de algoritmos ou técnicas de Mineração nos dados tratados.

Em [CAM01] são mostrados alguns métodos de Análise e Exploração de Dados para preparação e avaliação da qualidade dos dados na construção de Data Warehouse.

- *Análise de Domíni*o, identifica e organiza o conhecimento sobre o domínio dos dados, visando entendê-lo e representá-lo através de modelos chamados de modelos de domínio. No processo de análise de dados essa fase é importante porque permite identificar: os tipos de dados, intervalos de valores, uso de valores "default", valores não legítimos, dados de valores diferentes e de mesmo significado, revelar a presença de valores suspeitos, valores tecnicamente válidos, porém fora das regras de negócio.

- *Análise de Plenitude dos Dados*, esse método é comparável a um cabeçalho dos dados, onde são armazenadas metadados relacionados com as quantidades ou percentagem de dados nulos, inválidos, válidos, espaços, colunas incompletas. Esses metadados podem servir como entrada para análises de qualidade de dados.

- *Análise de Integridade Estrutural*, esse método visa verificar a integridade da chave primária, situação que pode acontecer se a base de dados não tiver esse recurso ou não foi definido, ou então em possíveis junções de dados. A integridade referencial também é analisada nessa fase, principalmente se a integridade referencial não é obrigatória nas bases operacionais. A cardinalidade também é verificada nessa fase.

- *Análise de Aderência às Regras do Negócio*, nesse método é altamente recomendável a aplicação de algoritmos inteligentes e/ou de mineração de dados para descobrir as regras implícitas nos dados, identificar combinações inválidas em campos correlatos para tratamento dessas anomalias (regras de associação), validar regras conhecidas, analisar a aplicação de operações aritméticas em dados e avaliar regras de seqüência lógica ou cronológica dos dados.

- *Aderência às Regras de Transformação*, visa validar as regras de construção das estruturas de dados destino e o conteúdo dessas estruturas (conteúdo dos dados) após a fase de ETL do Data Warehouse.

3.5.1 – VISUALIZAÇÃO DOS DADOS NO PRÉ-PROCESSAMENTO

Quando se trata de grandes volumes de dados, uma tarefa a qual devemos dar muita importância é a visualização dos dados, isso porque o ser humano necessita de

informações condensadas e/ou particionadas. Isto decorre do fato de que o ser humano tem capacidade restrita de memorização. Ele tem dificuldade de memorizar os resultados parciais até o momento de serem utilizados [MEN99]. Por esse motivo busca-se formas mais condensadas de apresentar dados. A sumarização das colunas de uma tabela em outras menores com o intuito de visualizar através da distribuição de frequência é uma das técnicas possíveis. Uma outra técnica que pode ser usada em conjunto ou separado é a transformação dos dados tabulares em informações gráficas já que, um gráfico pode condensar milhares de dados.

Os seres humanos em geral têm uma maior facilidade de processar dados visuais. Por isso, o processo de visualização de dados torna-se muito importante no pré-processamento de dados e na apresentação dos resultados. Usando métodos adequados de visualização, os seres humanos podem extrair resultados importantes a partir de dados complexos, em poucos milisegundos. Boas técnicas de visualização podem auxiliar o cérebro humano a processar informações complexas de forma muito rápida [ALM01]. Para melhor explicar esse conceito, considere-se a Tabela 3.3, onde mostra dados na forma tabular. Nela não se consegue visualizar de forma clara o que ela tem a informar. O padrão nela apresentado é de difícil memorização e interpretação por seres humanos. Esses mesmos dados mostrados de forma tabular podem ser mais bem interpretados, como mostra a Figura 3.12 [ALM01].

X	**0**	**0,259**	**0,5**	**0,707**	**0,866**	**0,966**	**1**	**0,966**	**0,866**	**0,707**
Y	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**
X	**0,5**	**0,259**	**0**	**-0,259**	**-0,5**	**-0,707**	**-0,866**	**-0,966**	**-1**	**-0,966**
Y	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
X	**-0,866**	**0,707**	**-0,5**	**-0,259**	**0**	**0,259**	**0,5**	**0,707**	**0,866**	**0,966**
Y	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**

Tabela 3.3 – Dados apresentados na forma tabular [ALM01]

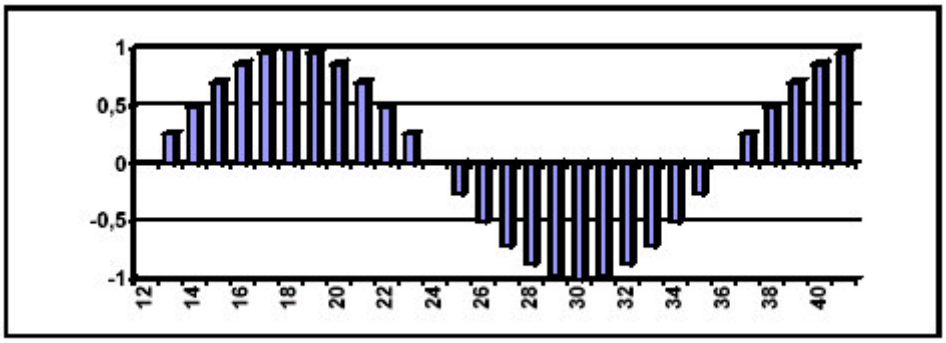

Figura 3.12 – Dados representados na Forma Gráfica [ALM01]

O que foi dito até agora é que recursos usados para a melhoria na visualização dos dados, são muito importantes para o entendimento humano, portanto quando se está manipulando dados é importante procurar as melhores formas para entendê-los e conseqüentemente tratá-los da melhor maneira possível, reduzindo a complexidade do processo.

3.5.2 – DISTRIBUIÇÃO DE FREQUÊNCIA

A distribuição de frequência é um método estatístico na qual em uma tabela condensa uma coleção de dados conforme as frequências de ocorrência dos dados (repetições de seus valores) [SIL02].

Tomemos como entrada uma tabela ***Ex1*** (Tabela primitiva) com elementos que não foram numericamente organizados, observa-se que sem uma ordenação é difícil formar uma idéia exata do comportamento do grupo como um todo. A maneira mais simples de organizar os dados para uma melhor visualização é através de uma ordenação crescente ou decrescente de seus elementos. Como resultado dessa ordenação temos a tabela ***Ex2***(Rol).

- Ex1: 45, 41, 42, 41, 42 43, 44, 41, 50, 46, 50, 46, 60, 54, 52, 58, 57, 58, 60, 51.
- Ex2: 41, 41, 41, 42, 42 43, 44, 45, 46, 46, 50, 50, 51, 52, 54, 57, 58, 58, 60, 60.

Após a ordenação já é possível entender melhor como os valores estão distribuídos. Por exemplo, já se pode saber os valores máximo e mínimo mais facilmente.

Apesar da ordenação melhorar a visualização, esta pode ser ainda melhorada, se for feita uma distribuição de frequência. No exemplo da tabela abaixo observa-se com maior facilidade a determinação dos valores mínimo e máximo dos dados. Se for feita uma

ordenação por frequência é possível também determinar mais facilmente os valores mais ou menos freqüentes.

Dados	Frequência
41	3
42	2
43	1
44	1
45	1
46	2
50	2
51	1
52	1
54	1
57	1
58	2
60	2
Total	**20**

Tabela 3.4 – Distribuição de frequência 1.

Dados	Frequência
41	3
42	2
46	2
50	2
58	2
60	2
43	1
44	1
45	1
51	1
52	1
54	1
57	1
Total	**20**

Tabela 3.5 – Distribuição de frequência 2.

Quando o tamanho da amostra é elevado e a quantidade de elementos distintos é grande ou quando se deseja frequências mais relevantes, o mais apropriado é efetuar o agrupamento dos valores em vários intervalos de classe, cuja distribuição é denominada de *Distribuição de frequência com intervalos de classe.*

Classes	Frequências
41 |------- 45	7
45 |------- 49	3
49 |------- 53	4
53 |------- 57	1
57 |------- 61	5

Tabela 3.6 – Distribuição de frequência 3.

Ao fazer esse agrupamento ganha-se em simplicidade, porém se perde em detalhes. O que se pretende com esse aumento das frequências relativas é realçar o que existe de essencial nos dados, e tornar possível ou melhorar o uso de técnicas analíticas para sua

descrição, até porque a estatística e outros métodos de análise de dados tem por finalidade analisar um conjunto de dados, desinteressando-se por casos isolados.

A distribuição de frequência com intervalos de classes é conhecida também como discretização de dados numéricos sendo muito abordada em referências de Data Warehouse e Data Mining, principalmente na fase de pré-processamento de dados [HAN01] [PYL98].

Dados	Frequências
Bahia	1001
Pernambuco	903
São Paulo	4
Sergipe	3
Mato Grosso	2
Francesco	1
XXXX	1
1234	1

Tabela 3.7 – Distribuição de frequência de estados.

Apesar da técnica de distribuição de frequência ser normalmente citada para discretização e tratamento de dados numéricos, essa técnica também é aplicável em dados do tipo categórico, ou seja, o seu uso possibilita uma boa visualização da distribuição dos dados de uma variável, servindo de suporte para identificação de erros e a definição de novos tipos de dados também para dados categóricos. Veja o exemplo na Tabela 3.7.

Na Tabela 3.7 observa-se que existem 3(três) estados com baixa frequência e 3 entrada de dados que devem ser caracterizados como sendo uma entrada errada. Um possível tratamento para esse caso seria a junção dos estados menos freqüentes e erros em 1(um) ou 2(duas) novas categorias para torna-los mais representativos. Além de tornar os dados mais representativos o agrupamento para diminuição das ocorrências distintas melhora a visualização. Nas Tabelas 3.8 e 3.9 são representados dois possíveis tratamentos para esses dados.

Dados	Frequências
Bahia	1001
Pernambuco	903
Outros	9
Erros	3

Tabela 3.8 – Sugestão de tratamento 1

Dados	Frequências
Bahia	1001
Pernambuco	903
Outros	12

Tabela 3.9 – Sugestão de tratamento 2.

3.5.3 – DIST. DE FREQUÊNCIA NA EXPLORAÇÃO E ANÁLISE DE DADOS

A distribuição de frequência é uma das principais técnicas de análise e exploração de dados. O uso dessa técnica em pré-processamento de dados é amplo e pode dar alguns dos principais subsídios para análise e exploração de dados. Ao longo desse trabalho, o uso da distribuição de frequência será constantemente citada de maneira direta ou indireta, porque os produtos derivados dessa técnica entram em todas as etapas da metodologia FastCube, que será apresentada no capítulo 4 dessa dissertação. Eles são usados como: instrumento para visualização dos dados, fonte geradora de metadados, entrada de técnicas de tratamento e modelagem de dados, medição de qualidade dos dados.

A seguir, estão ilustradas algumas das possíveis aplicações do método de Distribuição de Frequência no pré-processamento de dados:

- *Verificação de violação de chave única*. Isso se deve ao fato das fontes do DW poderem derivar de sistemas não relacionais, onde é possível a violação se houve algum dado onde a frequência é maior do que 1 no campo que deveria ser chave única. Para facilitar essa análise, recomenda-se que os dados venham ordenados de forma decrescente pela frequência. (Figura 3.13)

- *Análise de verificação de nível de preenchimento (valores nulos).* Pode avaliar o nível de preenchimento de determinadas variáveis, comparando a quantidade de ocorrências nulas no campo, em relação ao total de registros. Um determinado campo numérico com muitos valores zerados pode ser um indicativo de problemas com os dados.(Figura 3.13)

- *Valores fora do domínio*. Se um campo aceita apenas uma faixa de valores e apresenta valores fora desse domínio, possivelmente existem erros no sistema de coleta de dados. Normalmente em domínios pequenos após a distribuição de frequência os valores que não fora da faixa de domínio são facilmente identificados com a distribuição de frequência. Com essa operação determina-se o quanto e qual são os valores que estão fora da

faixa de uma determinada coluna. Ex: Idade = 1000 e idade = 2, para uma faixa de “18 a 99” ou Sexo = 1 e 0, para valores válidos “F” e “M”. Esse caso do campo Sexo pode indicar que houve problemas na integração dos dados, que pode ter sido oriundo de mais de uma fonte, ou pode indicar a necessidade rever a forma como os dados são coletados nos formulários dos sistemas operacionais.

A Figura 3.13 representa uma interface de um software que usa a distribuição de frequência automática nas colunas de uma tabela, sendo essa distribuição a base para tratamento dos dados.

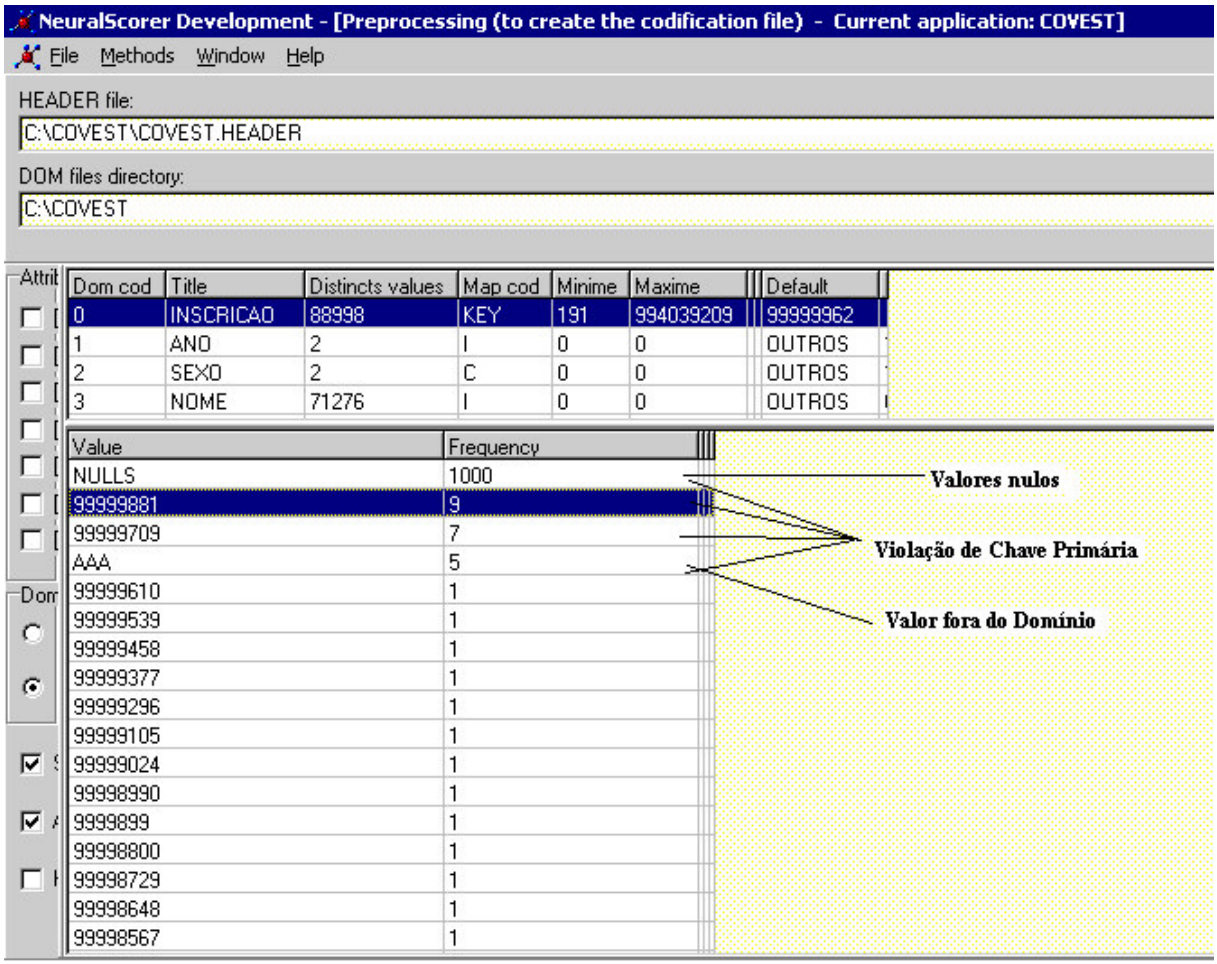

Figura 3.13 – Distribuição de Frequência do campo INSCRICAO (chave primária)

Nesse capítulo foram apresentados alguns dos principais conceitos necessários ao entendimento e elaboração dessa dissertação. No próximo capítulo será apresentada uma metodologia de construção de DW proposta nessa dissertação, denominada de FASTCUBE.

4 – METODOLOGIA FASTCUBE

Surgida inicialmente para tratamento de dados para Banco de Dados de Apoio à Decisão a metodologia FASTCUBE, a ser descrita nesse capítulo, evoluiu e tornou-se uma alternativa simples à construção e/ou prototipação de DW. Ela passa por todas as etapas de construção de um modelo dimensional.

4.1 – INTRODUÇÃO

A metodologia FASTCUBE não foi feita para tentar englobar todos os aspectos da construção de um DW e sim fornecer subsídios para que se implemente em um curto espaço de tempo um DW Estatístico, um Data Mart ou um protótipo de DW. Sendo baseada em dados e o seu principal artefato que é a própria massa de dados, pode-se chegar ao fim do processo de modelagem com um esquema dimensional com dados, pronto para consultas OLAP.

Feita inicialmente como uma metodologia para tratamento de dados, é natural que ela aborde alguns dos principais problemas relacionados com o tratamento dos dados. Durante todo o processo de modelagem (análise e projeto) ela produz uma representação mais realista sobre a situação dos dados. Percebeu-se que durante o tratamento de dados alguns dos elementos deste tratamento de dados poderiam ser usados para se ir além de uma simples avaliação e tratamento de dados. Os dados e metadados envolvidos no processo de tratamento de dados, aliados aos conhecimentos adquiridos pelo especialista durante esse processo, poderiam ser usados para geração de um modelo de dados populado.

Traçar um panorama sobre os dados, identificar os possíveis tratamentos a serem executados e gerar um protótipo multidimensional é dos principais focos dessa metodologia. A metodologia também já prevê mapeamento de todo o processo de tratamento de dados (ETL), podendo esse ser usado para auxiliar consertos nas bases de dados operacionais ou simplesmente para efetuar futuras cargas no DW.

4.2 – VISÃO GERAL DA METODOLOGIA

A metodologia FastCube possui características iterativas como a maioria das metodologias atuais de desenvolvimento, devendo ser executada várias vezes para chegar ao resultado desejado. Quanto maior o número de iterações maior será o refinamento do produto final. O projetista deverá parar quando atingir o nível desejado (ou máximo) de qualidade a partir da interação com os usuários. Uma fase não precisa estar totalmente terminada para avançar à próxima, mesmo porque, as fases seguintes podem ajudar na melhor definição das fases anteriores. A Figura 4.1 mostra como seria uma iteração nessa metodologia. Observa-se que as fases Fragmentação, Análise dos Dados e Tratamento de Dados possuem uma natureza cíclica entre si, isso porque todos os tratamentos dos dados possíveis nessa metodologia são feitos a partir da construção de novos fragmentos.

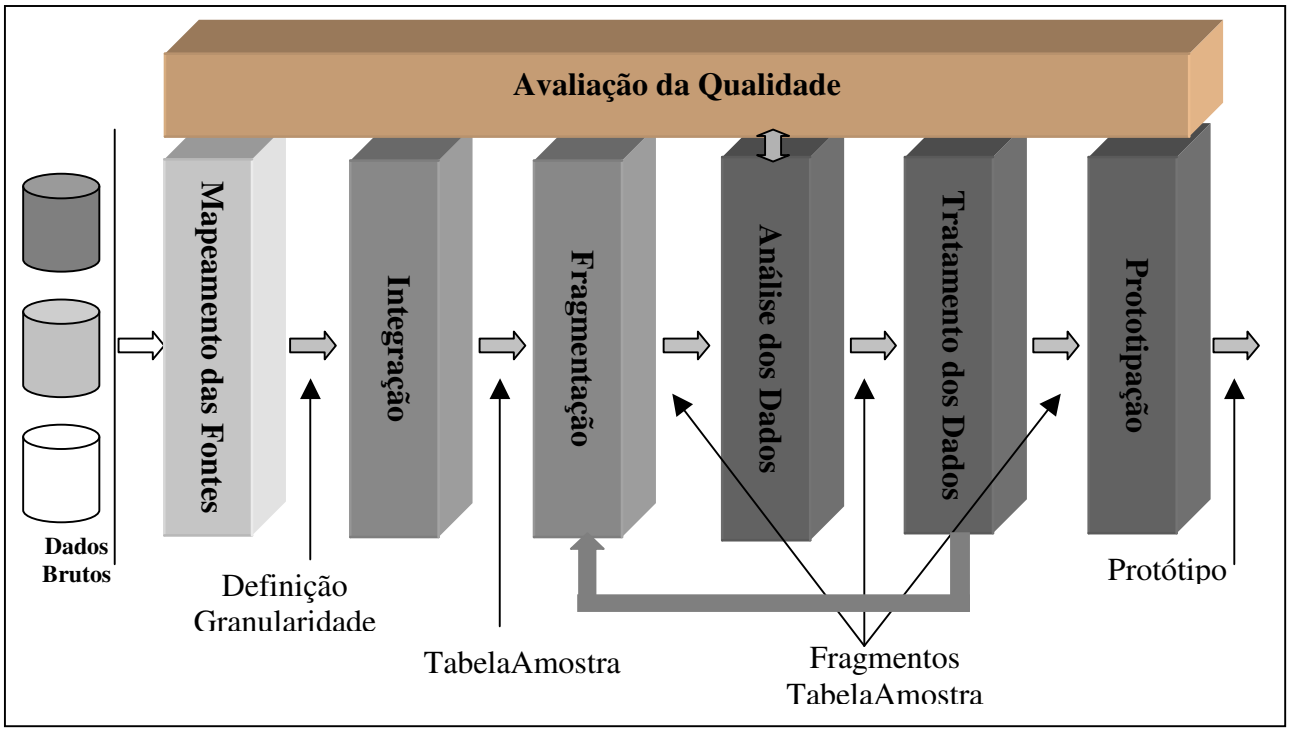

Figura 4.1 –Um ciclo na metodologia FastCube.

Pode-se resumir uma iteração da metodologia da seguinte forma:

- Após o mapeamento das fontes e a seleção dos dados que serão utilizados, tem-se a definição da granularidade dos dados. Com esse mapeamento e a definição da granularidade é feita a integração dos dados em uma tabela desnormalizada. Essa tabela deve ter apenas uma amostra representativa dos dados. [PYL99] descreve

algumas técnicas que podem ser empregadas para fazer uma amostragem. Essa tabela é aqui denominada de *TabelaAmostra*.

- Aplicando-se o método de distribuição de frequência [SIL02], descrito no Capítulo 3, em cada uma das colunas da *TabelaAmostra* são construídos os *Fragmentos* de colunas.

- O especialista analisa os dados e metadados das colunas. A maioria dos metadados foi gerada a partir do processo de fragmentação e o próprio *Fragmento* pode ser considerado um metadado.

- Através da aplicação de algumas técnicas de pré-processamento [PYL98] [CHE01] [HAN01] [INM97] sobre os *Fragmentos e as colunas da TabelaAmostra,* é feito o tratamento e enriquecimento dos dados, tendo como resultado a geração de novas colunas tratadas na *TabelaAmostra.*

- Novamente o especialista analisa os dados, para avaliar os resultados obtidos e o potencial de incremento de qualidade que deverá ser obtido, com a correção dos problemas nas fontes originais ou na etapa de ETL, caso seja impossível tratar as fontes originais.

- Além de ajudar no tratamento e na avaliação dos dados, os *fragmentos* e seus metadados poderão ser usados para o auxílio na montagem de um protótipo de um modelo dimensional.

- A qualidade dos dados é medida em todas as etapas do processo, porém a fase de Análise dos Dados tem uma maior interação com a Análise da Qualidade, porque todo o tratamento dos dados é baseado no que se deseja obter com os dados e na qualidade dos mesmos. O foco da qualidade nessa dissertação é nos dados.

4.3 – MAPEAMENTO E INTEGRAÇÃO DE DADOS

As etapas de mapeamento e integração de dados fazem parte do processo de preparação dos dados da metodologia. Uma tabela desnormalizada é o artefato principal de saída dessas etapas e o artefato principal de entrada no processo de pré-processamento de dados. Com o uso de uma tabela desnormalizada a granularidade desejada já está embutida,

tornando assim possível a implementação de um modelo de tratamento de dados baseado no acréscimo de novas colunas nessa mesma granularidade. A construção dessa tabela envolve aspectos abordados no capítulo 3, como a integração dos dados e definição da granularidade. Definir o nível de granularidade é um dos primeiros passos e um dos mais relevantes da metodologia, pois a sua definição está diretamente ligada ao objetivo final do Sistema de Apoio à Decisão. Os impactos e aspectos relacionados com a adoção de um nível de granularidade também abordados no Capítulo 3.

Após a definição da granularidade, é importante fazer uma análise das fontes de dados disponíveis, identificando as tabelas e os atributos relevantes que deverão fazer parte do modelo inicial. Ao se adotar um nível de granularidade é natural que atributos derivados sejam criados para contemplar as transformações dos atributos que estão em granularidades diferentes nas bases transacionais ou que precisem de algum tipo de pré-tratamento para ser útil. A criação de atributos derivados deve ser feita na integração dos dados. Mas a idéia é que não se tente fazer essa tarefa por inteiro no primeiro ciclo da metodologia, pois ao fim do primeiro ciclo muitos atributos derivados que não foram definidos inicialmente certamente aparecerão. O esforço para que eles apareçam certamente será bem menor que tentar defini-los sem a primeira iteração. É desejável que a montagem dessa tabela desnormalizada leve em conta todos os aspectos da integração de dados, mas que seja feita de maneira iterativa, isso porque a idéia da metodologia é que a etapa de integração seja auxiliada em uma estratégia evolucionária. Um caso típico onde não se deve perder muito tempo com a integração, é quando se tem pouca informação e documentação sobre os dados, e se deseja obtê-la através de uma primeira análise exploratória dos dados. A definição final dessa tabela desnormalizada pode acontecer de uma só vez, mas o que normalmente se observa é o projetista tendo uma definição final somente depois de algumas iterações. A própria metodologia pode facilitar o processo final de integração dos dados, direcionando e informando sobre diversos problemas ocorridos e como estes devem ser resolvidos.

A integração de dados é um dos requisitos de entrada do FastCube. O nível de integração a ser aplicado nos dados deve ser ajustado para a necessidade imediata, o que quer dizer que não devem ser aplicadas todas as técnicas possíveis na integração no início do processo. Caso contrário, fugiria do objetivo central da metodologia, que é voltado para a obtenção de resultados rápidos. No entanto, vale ressaltar que o processo de preparação dos dados irá ganhar produtividade e qualidade se os dados estiverem devidamente integrados.

O mínimo requerido por parte da integração é que essa etapa gere uma tabela desnormalizada de acordo com a granularidade pretendida, englobando a amostragem de uma ou mais fontes de dados. Várias técnicas de amostragem poderão ser utilizadas durante o processo de integração. Para garantir que a amostragem é representativa do universo de dados, alguns algoritmos podem ser usados tais como: “Stratified Sampling”, “Every nth Observation”, “First n Observations” e “Cluster” [SAS99]. A próxima etapa da metodologia é a fragmentação dos dados.

4.4 – FRAGMENTAÇÃO DOS DADOS

Utilizando o método de distribuição de frequência, abordado no Capítulo 3, a partir de cada coluna da *TabelaAmostra* é gerado um *fragmento* com as ocorrências distintas e a frequência de cada coluna da *TabelaAmostra*. Para melhor entendimento do processo podemos compará-la ao comando SQL a seguir, sendo aplicado para cada coluna da *TabelaAmostra*:

```
CREATE TABLE FRAGMENT_X AS
SELECT COLUNA1, COUNT(*)
FROM SAMPLETABLE_X
GROUP BY COLUNA1
```

Essa é apenas a idéia básica da distribuição de frequência, um processo muito usado por estatísticos e pela maioria dos métodos e softwares de tratamento e limpeza de dados [GFSS00] [GFSS99] [VDO01] [CAM01]. Esse método também possibilita a obtenção de outros metadados, e serve como entrada de alguns algoritmos e técnicas de tratamento de dados. A visualização da frequência de ocorrências de uma coluna permite ao usuário uma avaliação do preenchimento daquela coluna (Figura 4.2). A análise visual é complementada com as informações (metadados), que podem ser obtidas com base na distribuição de frequência. Esses metadados contemplam informações de preenchimento como: quantidade de ocorrências distintas, nível de preenchimento, valor máximo e mínimo da variável (se numérico), quantidade de zeros...

A fragmentação dos dados além de auxiliar na visualização dos dados pelo projetista (Figura 4.2), indicando alguns dos procedimentos que deverão ser tomados para melhoramento dos dados, pode servir também como entrada para muitas técnicas de tratamento de dados, pois uma boa parte dessas técnicas utilizam os dados e metadados que

são gerados no processo de fragmentação. Essas técnicas de pré-processamento foram abordadas no Capítulo 3 e algumas implementações foram feitas no capítulo 5.

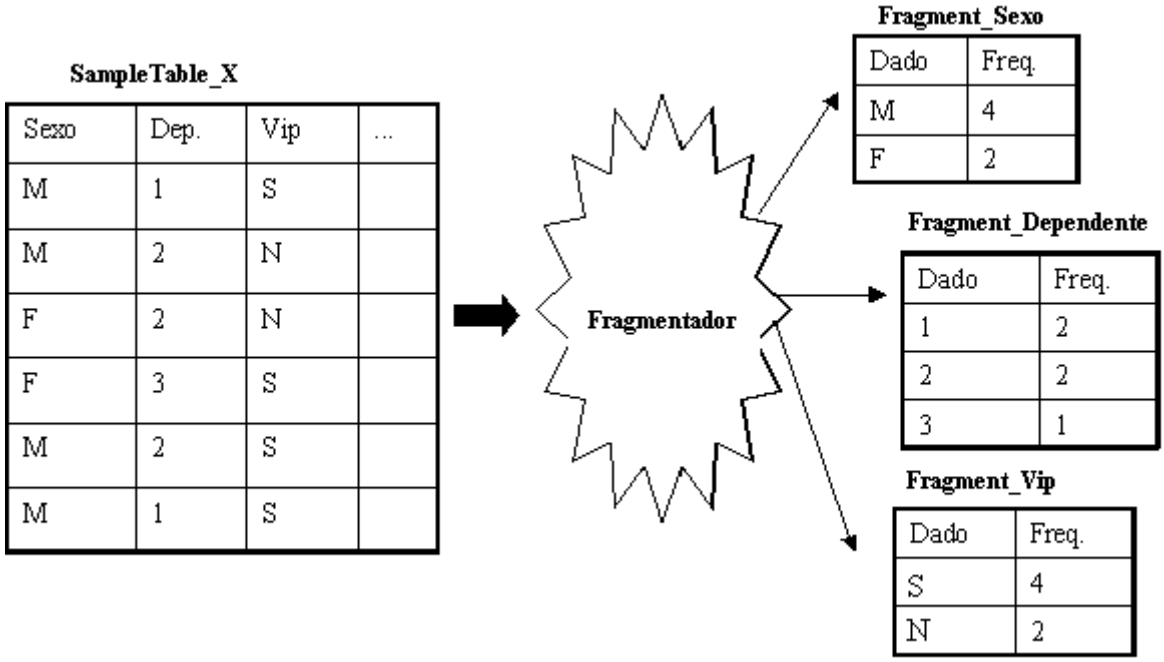

Figura 4.2 – Fragmentação aplicada a uma tabela de várias colunas.

A próxima etapa a ser vista é a Análise dos Dados, ela se baseia nos dados e metadados gerados pela fase de fragmentação.

4.5 – ANÁLISE DOS DADOS

Como já foi dito, a amostra é fragmentada e os dados e metadados gerados por essa fragmentação, devem ser visualizados em uma interface bem projetada, que possibilita a usuário não somente entender, inferir e manipular os dados, como também identificar as causas dos problemas encontrados. A análise dos dados é realizada coluna a coluna e o especialista deve julgar o nível de qualidade de cada coluna e decidir o que fazer com ela. Alguns aspectos são levados em consideração nessa análise como:

- Se a coluna não possui um bom preenchimento e se deve ser aproveitada como está.
- Se a coluna possui problemas de preenchimento, mas pode ser recuperada a partir das fontes originais ou por alguma técnica de limpeza.
- Se a coluna é realmente inútil e não pode ser recuperada

- Se a coluna tem má qualidade devido ao processo de integração que foi inadequado.
- Se a análise indicar problemas na captura dos dados através das fontes operacionais, o especialista poderá interferir no processo, sugerindo soluções, visando corrigir o processo de captura de dados.
- Se o Data Warehouse é voltado (principalmente) para mineração de dados ou para análises estatísticas, com dados que não podem ser recuperados nas suas fontes, normalmente o tratamento dos dados é feito via técnicas de limpeza de dados. Algumas dessas técnicas já foram abordadas anteriormente.
- Se for sugerida a criação de novas colunas derivadas.
- Se for sugerida a eliminação de informação redundante ou irrelevante.

Para todas essas análises é importante que métricas de qualidade sejam definidas previamente para que o especialista tenha um apoio efetivo para ajudar a decidir. Métricas tais como: percentual mínimo de valores nulos em colunas, percentual aceitável de preenchimento errados em colunas. Essas métricas levam em conta um limite por variável (coluna), com base na influência da amostra de dados. Alguns desses metadados de qualidade podem e devem ser preenchidos a partir dos metadados dos fragmentos gerados após a etapa de fragmentação. Esses metadados devem conter informações sobre a coluna, a *TabelaAmostra* de entrada e também sobre a solução completa. Isso possibilita fazer um diagnóstico mais amplo da entrada. Alguns desses metadados não estão diretamente ligados ao processamento de dados ou não podem ser capturados sem a ajuda de um especialista. A idéia é que se possa automatizar ao máximo o processo, flexibilizando ajustes nas medidas capturadas automaticamente e deixando aberta as que devem ser especificadas pelo especialista. Alguns metadados de qualidade são sugeridos no próximo capítulo. As distribuições de frequência e os metadados deverão ajudar na difícil tarefa de análise dos dados pelo especialista. Um gráfico de histograma de coluna poderá ser mais uma importante ferramenta de auxílio nessa análise dos dados, principalmente na visualização das faixas numéricas. Através de um histograma também é possível identificar valores numéricos e categóricos fora de um domínio.

Durante todo esse processo, o especialista estará pensando sobre os dados da empresa. Pensar sobre os dados é pensar sobre o negócio da empresa, é entender como ela funciona, ou como deveria funcionar, é descobrir as falhas, é entender o processo, é buscar formas de melhorá-los. Ainda nessa etapa também é possível e aconselhável o uso de técnicas de Mineração de Dados para um melhor entendimento dos dados. O uso de técnicas como

Indução de Regras, que é um mecanismo automático de descoberta de padrões nos dados [HAN01] [WIT99] [MAR99] [AUR97], é altamente aconselhável ainda na fase de análise. A indução pode ser feita diretamente na Tabela Amostra em cima das colunas selecionadas (inclusive as já tratadas). Esses padrões podem: validar algumas regras já levantadas do negócio, descobrir novas regras, apontar problemas com os dados, descobrir procedimento incorretos. A origem de problemas com os dados poderá fazer uma espécie de pré-mineração dos dados antes mesmo do final do processo de construção do Data Warehouse.

O processo de verificação da qualidade dos dados deve ser feito principalmente em duas etapas nesse processo. Uma logo após a fragmentação dos dados, para medir a qualidade dos dados de entrada no processo; a outra após o tratamento e limpeza dos dados, pois durante o processo de pré-processamento os dados podem ser enriquecidos, o que pode possibilitar um aumento na qualidade. Essa medição no final do processo mostra a qualidade final dos dados que serão usados nas consultas do DW.

A Figura 4.3 mostra a interface do NeuralScorer®, um software de mineração de dados que possui algumas funcionalidades de visualização, análise e tratamento de dados [NEU02]. Esse software foi baseado em uma tese de dissertação e procura cobrir todas as etapas do de desenvolvimento de uma solução de mineração de dados[MON99]. Ele foi usado para algumas de nossas análises com permissão para realização de algumas customizações na interface do software em conjunto com a equipe da Neurotech com o objetivo de aumentar a produtividade na análise, na compreensão e no tratamento de dados. Após essa customização a interface ficou bastante funcional e conseguiu reunir em uma só tela todas colunas da tabela a ser tratada, a distribuição de frequência de cada coluna e os possíveis tratamentos e transformações para cada coluna. Essa tela é um bom exemplo de apresentação de dados, sendo ela um dos resultados práticos de trabalhos realizados durante a elaboração dessa dissertação, pois a Neurotech adotou algumas das sugestões. Anteriormente para se atingir todas essas funcionalidades atuais seria necessária a navegação em diversas telas do NeuralScorer®.

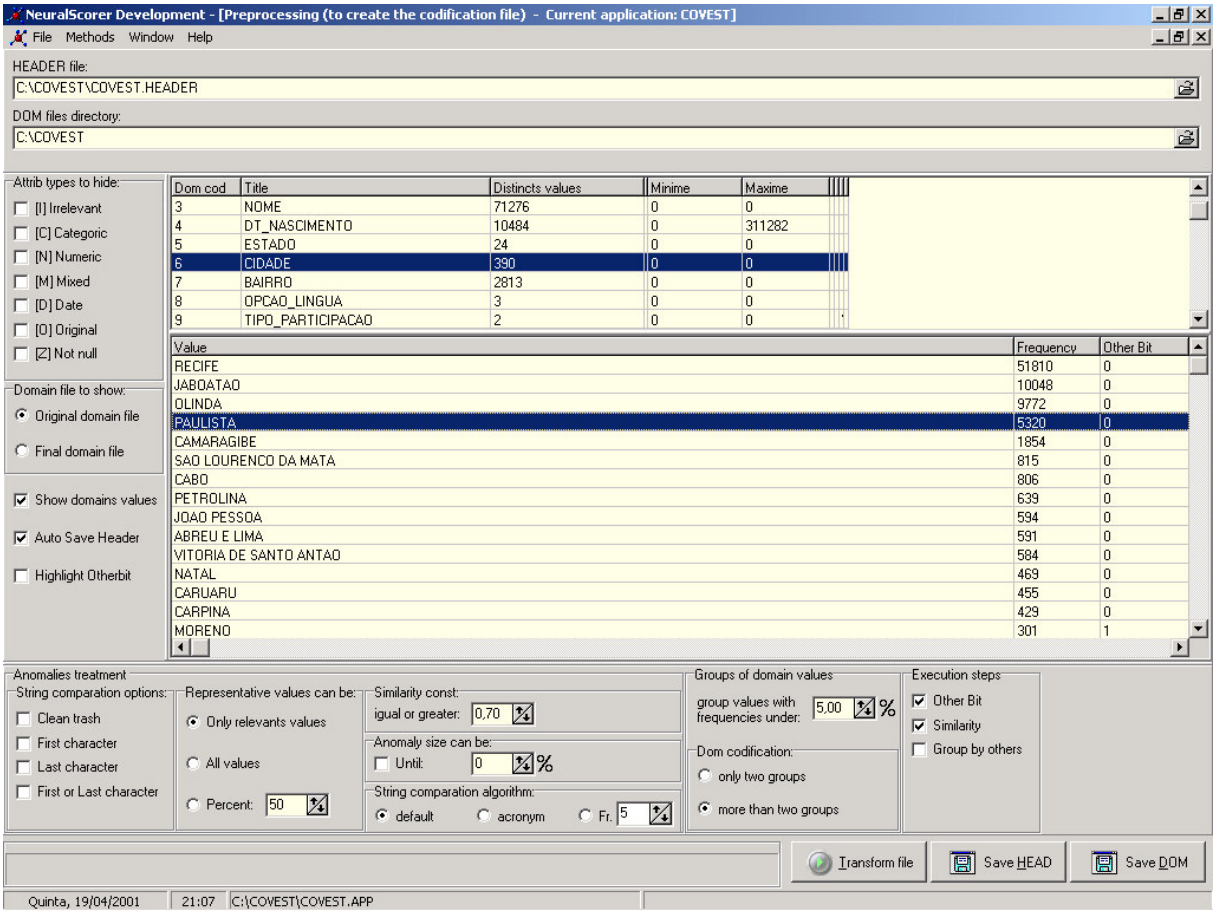

Figura 4.3 – Tela de análise e tratamento de dados do NeuralScorer.

Durante a análise dos dados deve-se ter sempre em mente que as colunas podem ser tratadas a medida em que é analisada; necessariamente não se termina a análise de todas as colunas para se começar o tratamento de dados. Normalmente a análise e tratamento são feitos em paralelo coluna a coluna um depois do outro.

4.6 – TRATAMENTO DOS DADOS

Esse processo é executado quase que em paralelo com a análise dos dados e deve ser feito coluna a coluna. O tratamento e a análise são etapas que às vezes se misturam, pois para que se faça o tratamento é sempre necessária a análise dos dados. É trabalho do especialista verificar a possibilidade de aplicação de algumas técnicas para melhorar ou adaptar uma coluna. O método de tratamento de dados da metodologia é simples, aplica-se uma técnica de pré-processamento em uma determinada coluna e tem-se uma ou mais colunas

tratadas e/ou modificadas como resultado. As colunas geradas são acrescentadas na *TabelaAmostra*, juntamente com as demais. Novas colunas sempre serão geradas a partir do uso das técnicas de pré-processamento, mas as originais nunca serão apagadas. As colunas derivadas (geradas) podem existir (materializadas) ou não (virtuais), isso porque algumas colunas precisam de várias derivações para serem geradas e seria muito dispendioso em volume de dados manter todas as colunas intermediárias. Algumas técnicas sugeridas para tratamento de dados já foram discutidas no capítulo 3. Literaturas específicas de pré-processamento para Mineração de Dados [HAN01] [CHE01] [PYL99] e os manuais e artigos de softwares de limpeza e tratamento de dados [NEU01] [SAS99] [GFSS99] [GFSS00] [VAS00] [VAS01] são fontes de informação usadas para ilustrar o tratamento de dados usando técnicas de pré-processamento.

Na metodologia, a coluna que se deseja tratar é submetida individualmente ou em conjunto com outras colunas às técnicas de tratamento de dados. Como resultado é gerada uma nova coluna na *TabelaAmostra* e conseqüentemente um novo fragmento. As colunas e os seus fragmentos originais da *TabelaAmostra* nunca serão apagados. Essas colunas podem no máximo ser marcadas como não ativas para visualização. Essa característica é importante, pois se deseja que todo o processo de tratamento de dados seja registrado. Uma coluna derivada de diversos tratamentos deve ser resultado de diversas transformações, logo, todo o processo de tratamento, incluindo parâmetros e técnicas utilizadas, deve ser guardado, para que em qualquer momento este processo possa ser refeito com outros dados. A Figura 4.4 sugere uma estrutura simples de metadados em UML capaz de armazenar as transformações de cada fragmento. O armazenamento desses metadados de tratamento possibilitará que o Analista de Qualidade de Dados do DW oriente os responsáveis pelas bases operacionais. Guardar a história durante o processo de transformação servirá para orientar as alterações que foram feitas nos dados para melhoria da qualidade das fontes operacionais (limpeza de dados nas fontes operacionais) e indicará muitas transformações que deverão ser feitas no processo de ETL definitivo. Uma estrutura completa dos metadados para suporte à metodologia será mais detalhada no Capítulo 5.

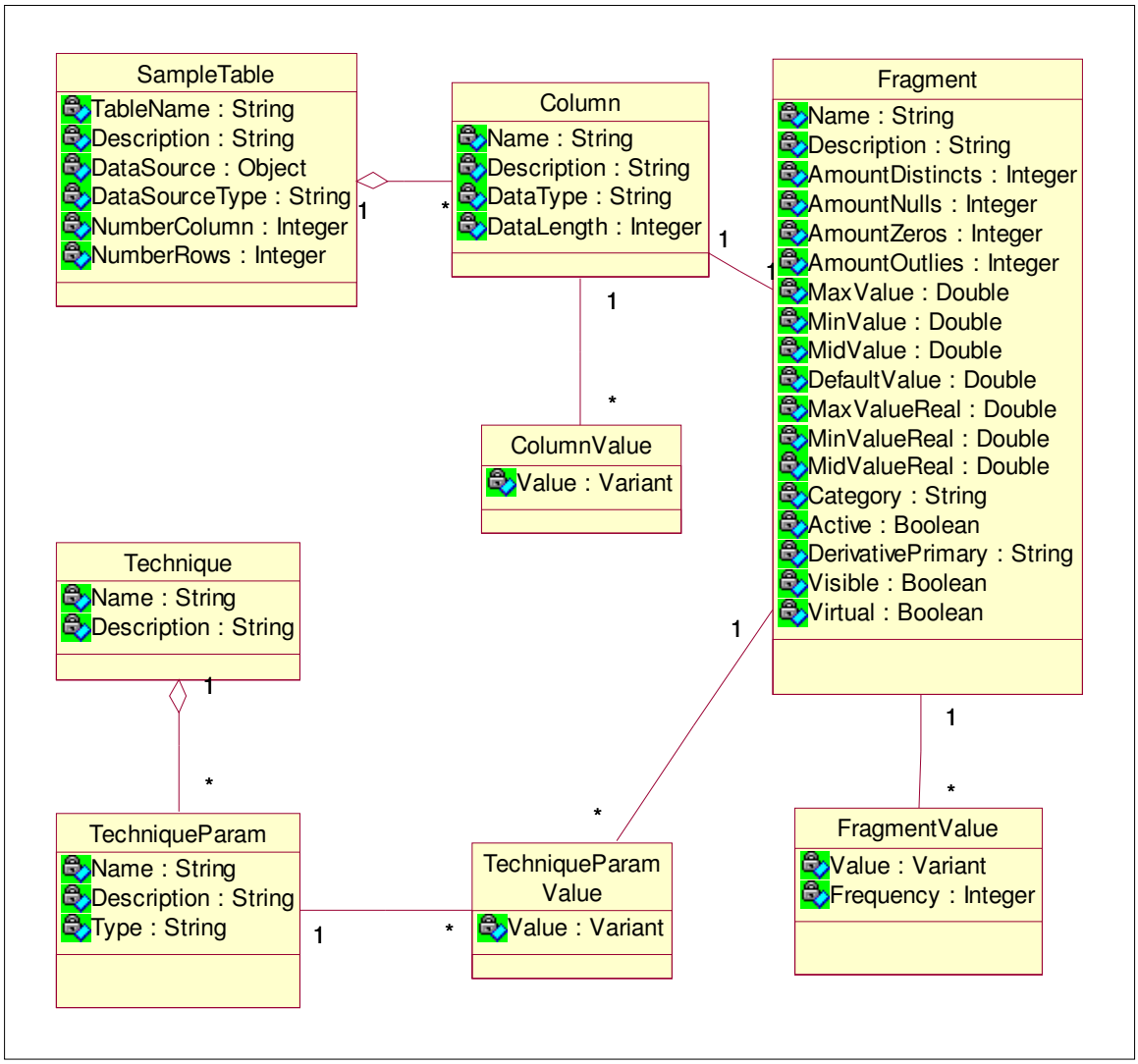

Figura 4.4 – Um exemplo de um modelo de classes de metadados para tratamento de dados.

Após o tratamento de dados, o próximo passo será a prototipação de um modelo dimensional para suportar os dados tratados.

4.7 – PROTOTIPAÇÃO RÁPIDA

Essa metodologia foi inicialmente desenvolvida para formatação de um modelo de manipulação de dados, onde seria possível fazer a verificação da qualidade e um tratamento efetivo dos dados, com o registro de todos os processos de transformações sofridos pelos mesmos. Porém descobriu-se que com o conhecimento adquirido durante esse processo pelo especialista, aliado aos metadados gerados a partir da fragmentação podia-se facilmente

chegar a um modelo multidimensional. A seguir será descrito como ocorre o processo de modelagem baseado no FastCube.

Para melhor ilustrar a fase de prototipação, será usado um mini-exemplo com os dados de um Data Mart referente a informações de candidatos que prestaram vestibular. No capítulo 6, o modelo dimensional completo será apresentado, bem como os resultados de consultas realizadas diretamente na base de dados final do Data Mart.

A geração do modelo dimensional é baseada no princípio de que cada coluna (fragmento) é um atributo do modelo em potencial. Inicialmente o especialista escolhe apenas as colunas que farão parte do seu Data Mart. Na Figura 4.5 algumas colunas da *TabelaAmostra* são separadas para a primeira montagem do modelo dimensional. O raciocínio utilizado para seleção dos fragmentos que se tornarão atributos do modelo dimensional deve ser intuitivo, pois a essa altura já se está bem familiarizado os dados do negócio. Depois da seleção das colunas relevantes é necessária a classificação dessas colunas em fato ou dimensão. Essa classificação assemelha-se a montagem de um pequeno quebra-cabeças, onde será usado todo conhecimento que o especialista acumulou durante as etapas passadas ou mesmo durantes iterações anteriores. O que se apresenta aqui é a identificação dos atributos, que farão parte das dimensões e fatos, pois na metodologia FastCube a definição de fato e dimensões é feita após o tratamento de dados.

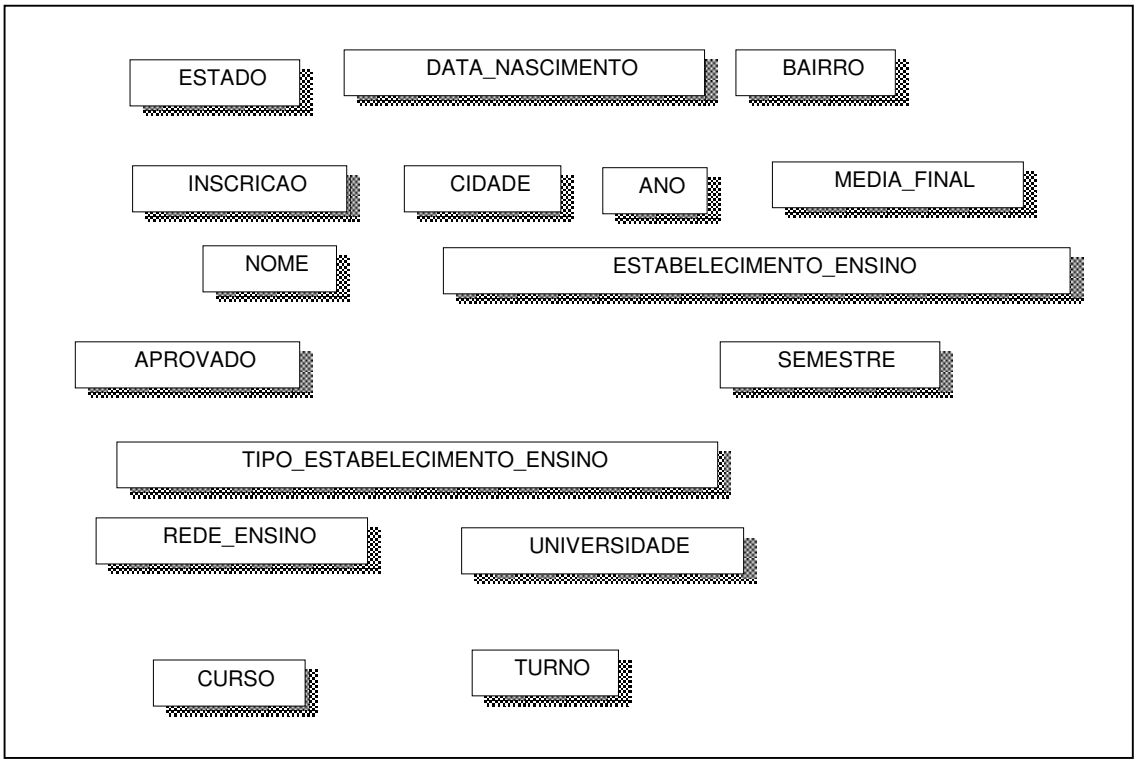

Figura 4.5 – Seleção dos fragmentos (atributos) relevantes para o Data Mart.

A metodologia FastCube sugere que a análise para descobertas do fato e dimensões seja feita através das características dos atributos no contexto de modelagem dimensional. A essa altura, com o conhecimento do especialista, aliado as características inerentes aos atributos que compõem os fatos, é possível identificar e mapear a tabela de fato do Data Mart. De maneira genérica os atributos candidatos a tabela fato são normalmente numéricos e os atributos candidatos à dimensão são geralmente do tipo textual e/ou discretos [KIM97] [KIM98]. Então, baseado nas características genéricas que definem o que atributos farão parte da tabela de fato, monta-se essa tabela. Para efeito de metodológico os demais atributos serão considerados dimensões e cada tabela de dimensão seria formada por apenas um atributo. A Figura 4.6 ilustra uma tabela de fato com os atributos de aprovação e de média final do candidato e as tabelas de dimensão com apenas um atributo.

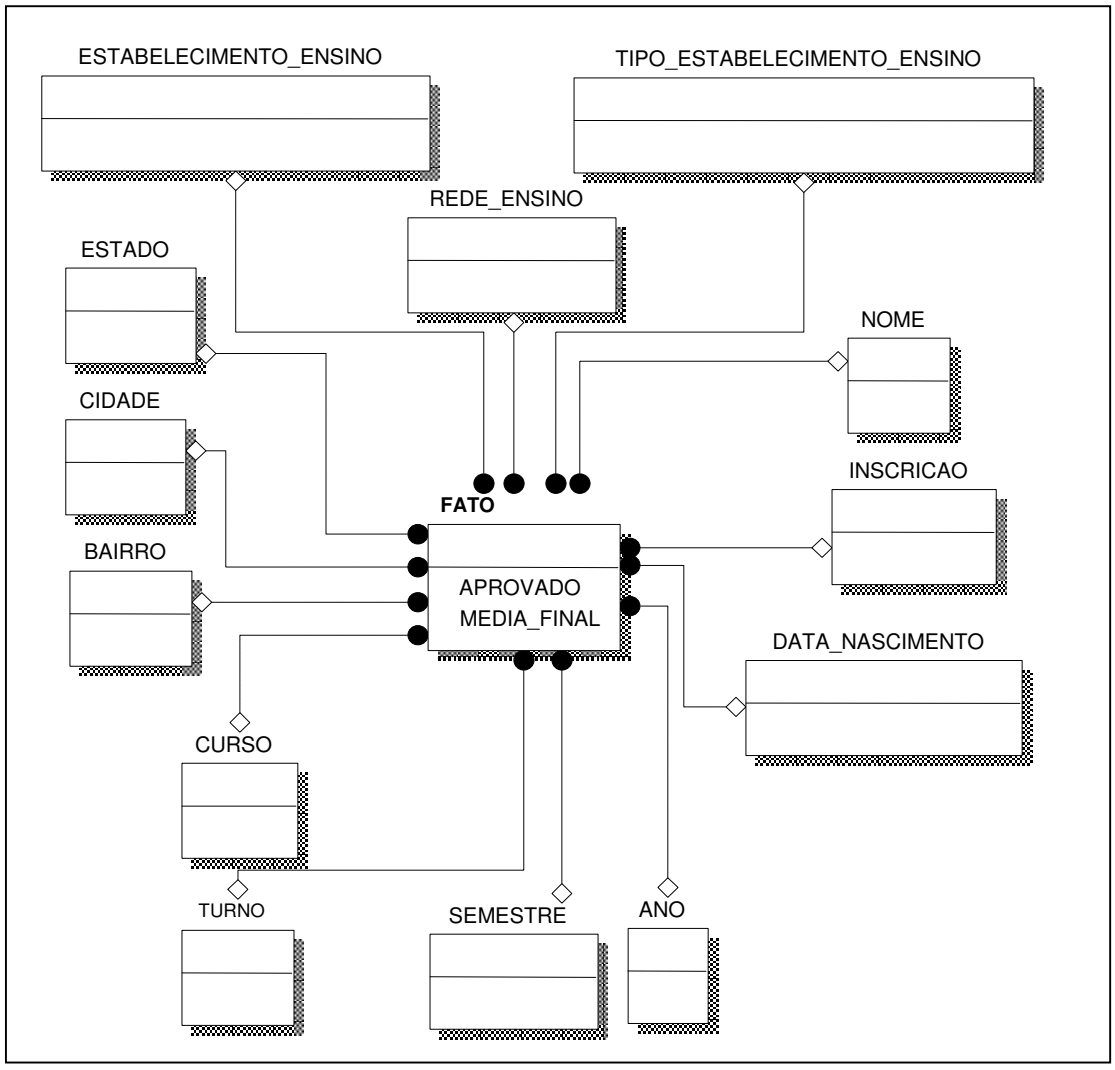

Figura 4.6 – Uma tabela de fato e dimensões com apenas um atributo

Agora se faz necessário o agrupamento das dimensões de único atributo, em dimensões com hierarquia e/ou um agrupamento semântico de vários atributos. Para que seja feito o agrupamento de atributos com o intuito de gerar as dimensões, é necessário um profundo conhecimento semântico das colunas que participarão do modelo. A maioria desse conhecimento já foi adquirida durante o processo de tratamento dos dados e deve ser complementada com ajuda de metadados e da documentação dos sistemas fonte. Vale ressaltar que a interatividade com os usuários de negócio deverá ocorrer durante todo o processo. Existem técnicas que podem auxiliar essa modelagem, sugerindo a construção de elementos do modelo(fato e dimensões), mas essas técnicas não descartam a interação com o especialista. A Figura 4.7 mostra um possível agrupamento para as dimensões da Figura 4.6,

como resultado tem-se o surgimento das dimensões: LOCAL, PREFERENCIA, TEMPO, CANDIDATO E ESTABELECIMENTO DE ENSINO.

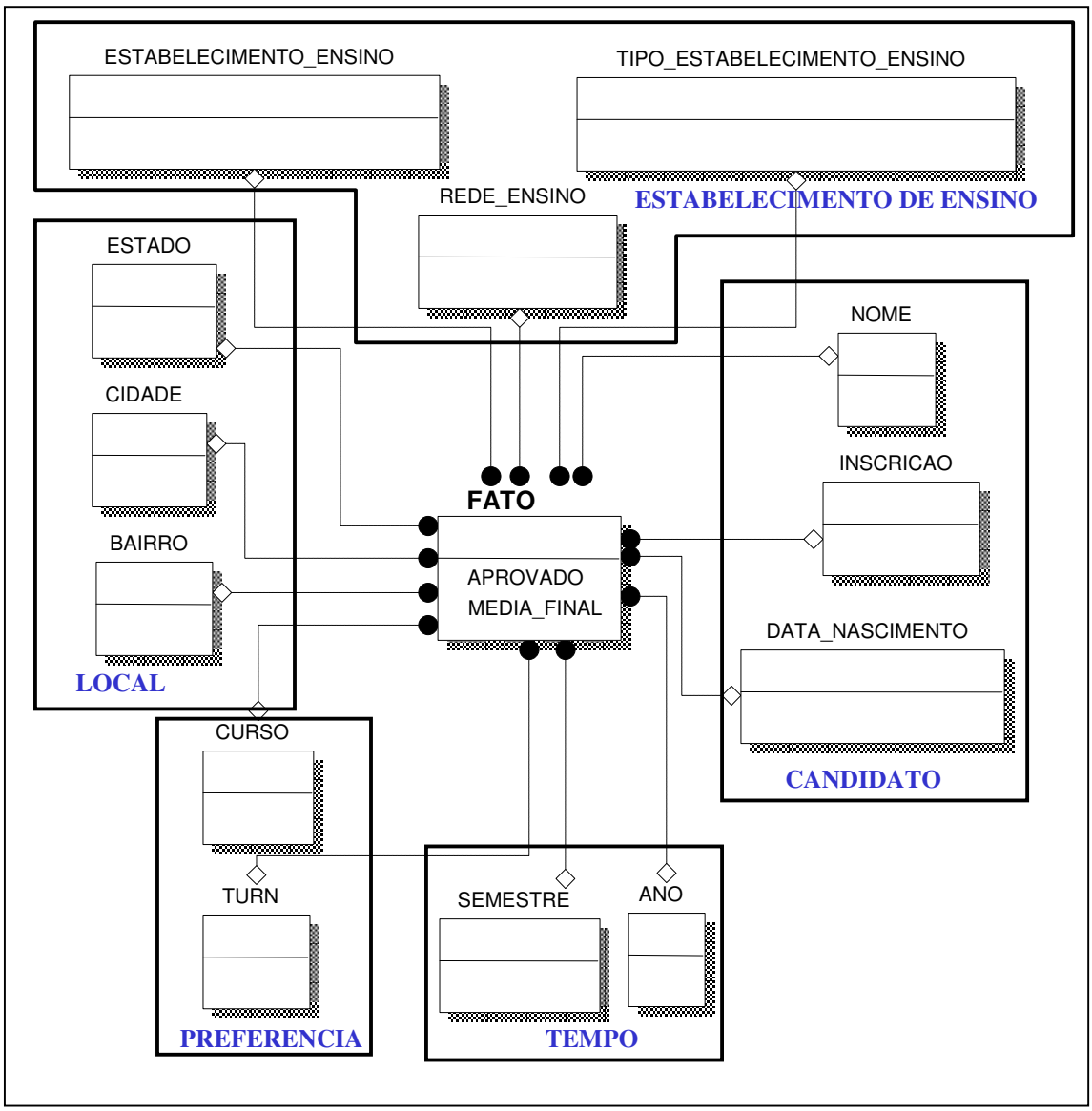

Figura 4.7 – Agrupamento para formação das dimensões.

Baseadas nas características (metadados) dos elementos multidimensionais (fatos e dimensões) e no fato de que os próprios dados estão presentes no ambiente de modelagem, a montagem do Data Mart pode ser auxiliada por técnicas e algoritmos para sugerir modelos a partir dos dados. Essa ajuda fornecida pelos dados deve ser complementada com ajustes feitos diretamente pelo analista com o auxilio dos metadados dos dados operacionais. Mas existem muitas limitações ao uso de tais técnicas de sugestão de modelo baseada em dados, porque não se consegue extrair todas as informações necessárias diretamente dos dados, pois podem existir muitas lacunas e armadilhas na semântica dos dados. A aplicação de uma técnica que

descubra através de correlação de variáveis, possíveis dimensões que possuem hierarquias (LOCAL, por exemplo) pode confundir-se em casos em que a correlação existe, mas semanticamente pertenceriam a dimensões diferentes. Por exemplo, em um sistema de vendas a correlação entre os atributos que indicam o local de residência na dimensão do CLIENTE e os atributos da dimensão LOCAL da loja, poderia sugerir pertencerem a uma mesma dimensão.

Qualquer algoritmo que seja apenas baseado em dados não consegue identificar atributos de dimensões que não possuam hierarquias bem definidas como, por exemplo: atributos que representem dados pessoais de um cliente não hierarquizáveis. Um algoritmo de sugestão de dimensões e descoberta de hierarquia, a partir dos dados foi desenvolvido para testes nessa dissertação e sua implementação encontra-se no ANEXO I. Ele é muito útil em situações em que se tem dúvidas sobre a hierarquização de atributos, então usa-se esse algoritmo para se determinar se atributos possuem uma hierarquia entre si, o que pode sugerir que pertençam a uma mesma dimensão.

Durante o processo de montagem de modelo dimensional, se o número de colunas for muito grande, é aconselhável para a diminuição da complexidade que sejam selecionadas um número menor de colunas para ser dado início ao processamento e posteriormente as demais colunas sejam acrescentadas ao modelo. A Figura 4.8 mostra o resultado final da simulação do processo de montagem do modelo dimensional descrito.

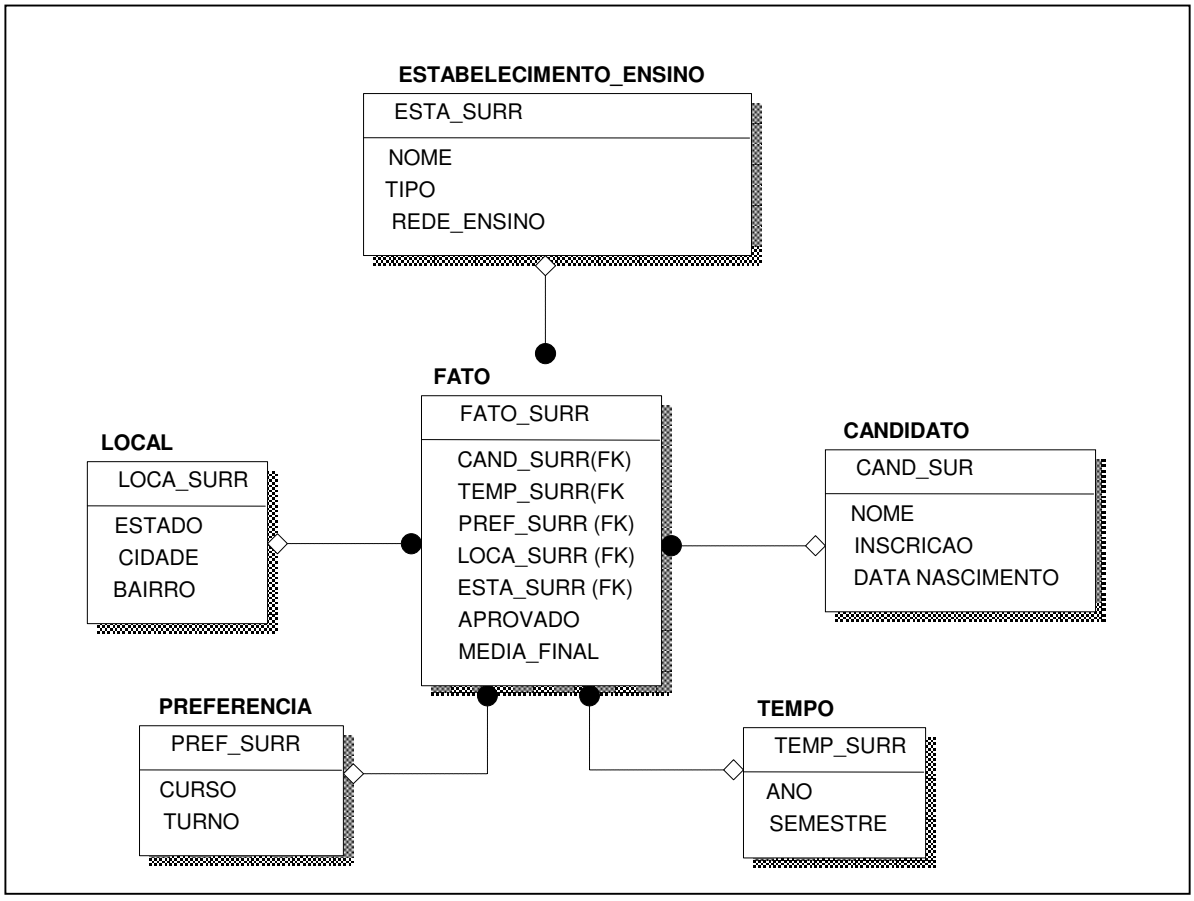

Figura 4.8 – Modelo com *surrogate key* e dimensões montadas.

Após a definição do modelo dimensional o próximo passo natural é a geração física desse modelo em um banco de dados. Tendo todos os metadados associados ao modelo gerado, a criação do script ou mesmo a criação direta das tabelas em um banco de dados se torna uma tarefa extremamente simples e que deverá ser totalmente automatizada.

Os dados que farão parte da tabela de fato e das tabelas de dimensões já estão carregados, então eles podem ser facilmente carregados no novo modelo criado. Todas as dimensões podem ser geradas a partir da *TabelaAmostra*, bastando fazer uma seleção distinta (SELECT DISTINCT) nas colunas que farão parte da tabela de dimensão, acrescentando uma chave substituta (*surrogate key*) [KIM98]. E como essa a *TabelaAmostra* já está na granularidade a ser implementada no Data Mart, todos os atributos selecionados como fato podem ser carregadas diretamente para a tabela de fatos do modelo, mapeando e acrescentando os relacionamentos com as chaves primárias das dimensões (*surrogate keys*). Devido a essa simplicidade no modelo de carga, esse pode ser totalmente automatizado, já que em um mesmo ambiente dispõem-se dos dados de entrada e dos metadados do Data Mart.

Um modelo de dados e metadados para suportar a metodologia é mostrado no capítulo 5. Na Figura 4.9 é apresentado um exemplo simplificado, de como seria essa carga de dimensões e fatos a partir da *TabelaAmostra*, sem levar em consideração a geração das chaves primárias e das chaves estrangeiras de um modelo estrela.

```
INSERT INTO DIMENSAO_LOCAL
SELECT DISTINCT ESTADO, CIDADE, BAIRRO
FROM TABELAAMOSTRA_COVEST

INSERT INTO FATO
SELECT APROVADO, MEDIA_FINAL
FROM TABELAAMOSTRA_COVEST
```

Figura 4.9 – SQL Simplificado de carga de dimensões e fatos.

Após a carga do Data Mart, já pode-se utilizar uma ferramenta OLAP para fazer as consultas analíticas em cima dos dados. No capítulo 6 algumas consultas OLAP são realizadas em cima do protótipo de Data Mart do estudo de caso. No próximo capítulo será apresentado uma implementação do FastCube.

A metodologia FASTCUBE faz parte de um conjunto de tecnologias desenvolvidas para construção de Data Warehouse dentro do contexto do Ambiente REDIRIS, descrito no Capítulo 3. Na Tabela 4.1 é apresentado um quadro comparativo entre alguns aspectos de construção de DW dos principais autores com a metodologia FASTCUBE.

Comparativo de alguns aspectos de contrução de DW dos principais autores					
Metodologia	**Prototipação**	**Uso de Dados**	**Amostragem**	**Scopo**	**Modelagem**
FASTCUBE	Forte	Durante todo processo	Início processo	Pequeno	Física
INMON	Opcional	Na carga	Opcional - a partir do DW	Grande	Conceitual e Física
KIMBALL	Opcional p/ Produtos	Na carga	Não	Grande	Conceitual e Física

Tabela 4.1 – Quadro comparativo de aspectos de construção de DW dos principais autores.

5 – UMA IMPLEMENTAÇÃO DO FASTCUBE

Essa implementação contempla apenas um subconjuto do Ambiente REDIRIS e um conjunto de possibilidades das técnicas e da metodologia FastCube, capaz de atender todos os requisitos necessários para tratamento de dados e modelagem de um protótipo de Data Mart. Essa configuração foi validada com a aplicação de um estudo de caso que será detalhado no próximo capítulo.

5.1 – UM CICLO DO FASTCUBE NA ARQUITETURA DO REDIRIS

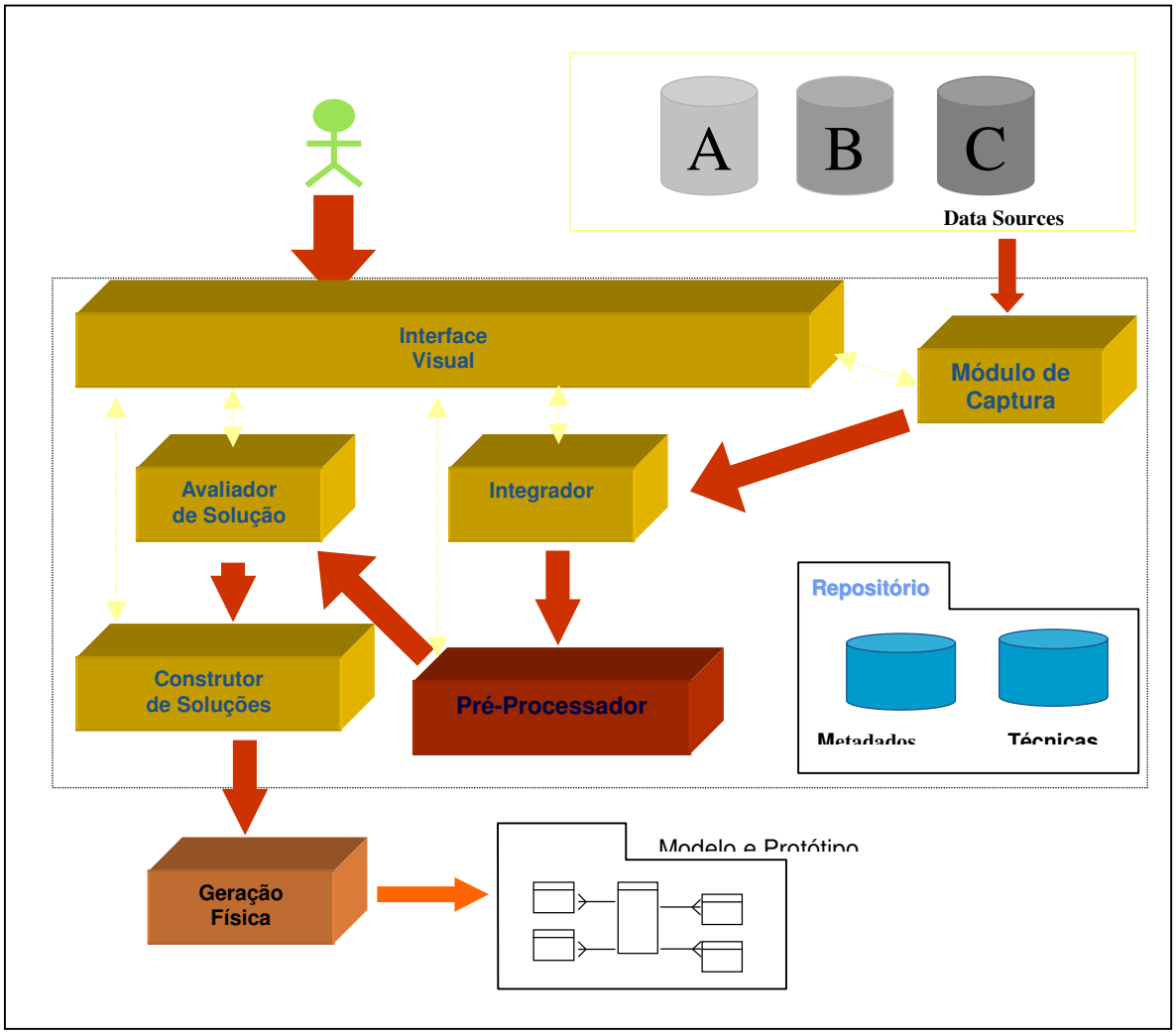

Figura 5.1 – Um ciclo no REDIRIS.

Dentro do Ambiente REDIRIS as técnicas e metodologias até aqui apresentadas participariam praticamente de todos os módulos da sua arquitetura. Na Figura 5.1 é mostrado um ciclo pelos módulos do REDIRIS.

O módulo de Captura é responsável pelo mapeamento das fontes de dados operacionais extraindo os metadados necessários para o entendimento inicial dos dados, proventos metadados para o módulo Integrador. Esse módulo tem o papel de realizar as operações necessárias para a montagem da *TabelaAmostra.* O módulo Pré-processador é o mais diretamente associado ao trabalho, é nele que a *TabelaAmostra* será fragmentada, analisada e será feito o tratamento e a limpeza dos dados. O módulo Pré-processador também proverá diversos metadados que serão utilizados pelo módulo Analisador de Soluções para medição e verificação da qualidade dos dados. O módulo construtor de Soluções também utilizará os metadados e dados gerados na fase de pré-processamento para elaboração de um modelo dimensional baseado nos *fragmentos*(colunas) como já foi visto anteriormente. O gerador de solução será o materializador da solução, ficando com a função de gerar fisicamente o modelo e realizar a carga no protótipo dimensional. O Módulo de Interface se relaciona com todos os outros módulos, pois a intervenção humana, às vezes muito reduzida, se faz necessária em todas as etapas do processo.

5.3 – ARQUITETURA

Para implementação do FastCube criamos uma arquitetura conceitual simples, constituída de apenas 4 camadas, mas capaz de dar um nível bastante elevado de abstração para o entendimento do modelo.

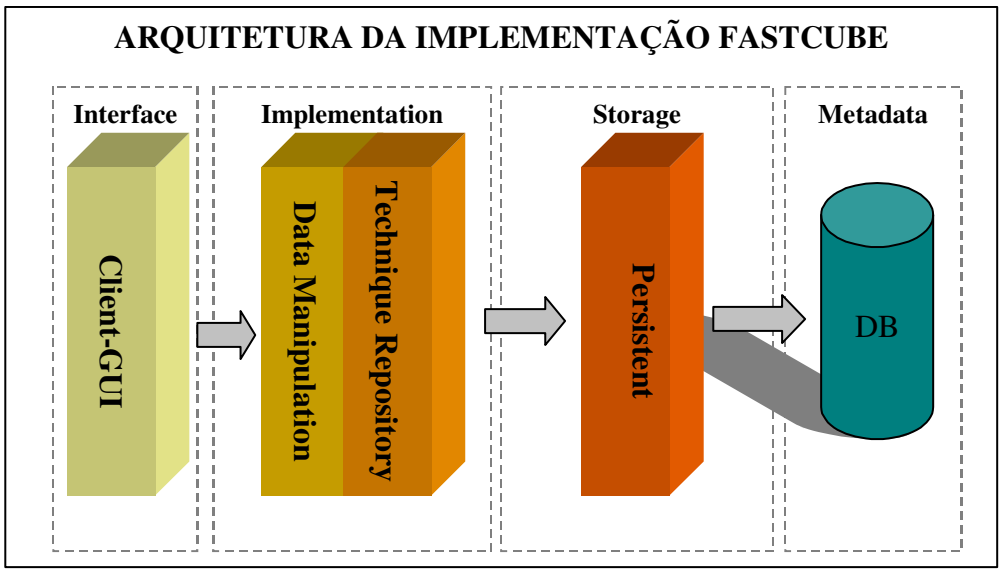

Figura5.2 – Arquitetura básica da implementação do FastCube

Cada uma dessas camadas foi implementada através de um conjunto pacotes e classes JAVA. A Figura 5.3 mostra a hierarquia dos principais pacotes.

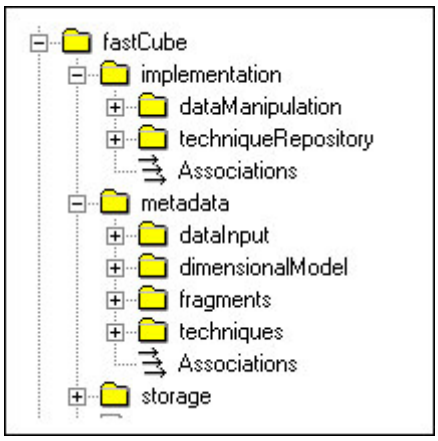

Figura 5.3 – Principais pacotes do FastCube.

A camada de *Implementation* que é representada pelo pacote de mesmo nome, possui 2 (dois) outros sub-pacotes. O pacote *dataManipulation* é responsável por todos os controles de operações de manipulação de dados, sempre manipulando os elementos da metodologia FastCube como *colunas*, *TabelaAmostra* e *fragmentos*. A Figura 5.4 mostra as classes e métodos do pacote *dataManipulation* e o seu relacionamento com uma interface e uma classe do pacote *Storage*. O pacote *techniqueRepository* é responsável pela implementação de todo o conjunto de técnicas do FastCube, representado assim um repositório de técnicas. Esse repositório de técnicas será melhor abordado mais adiante.

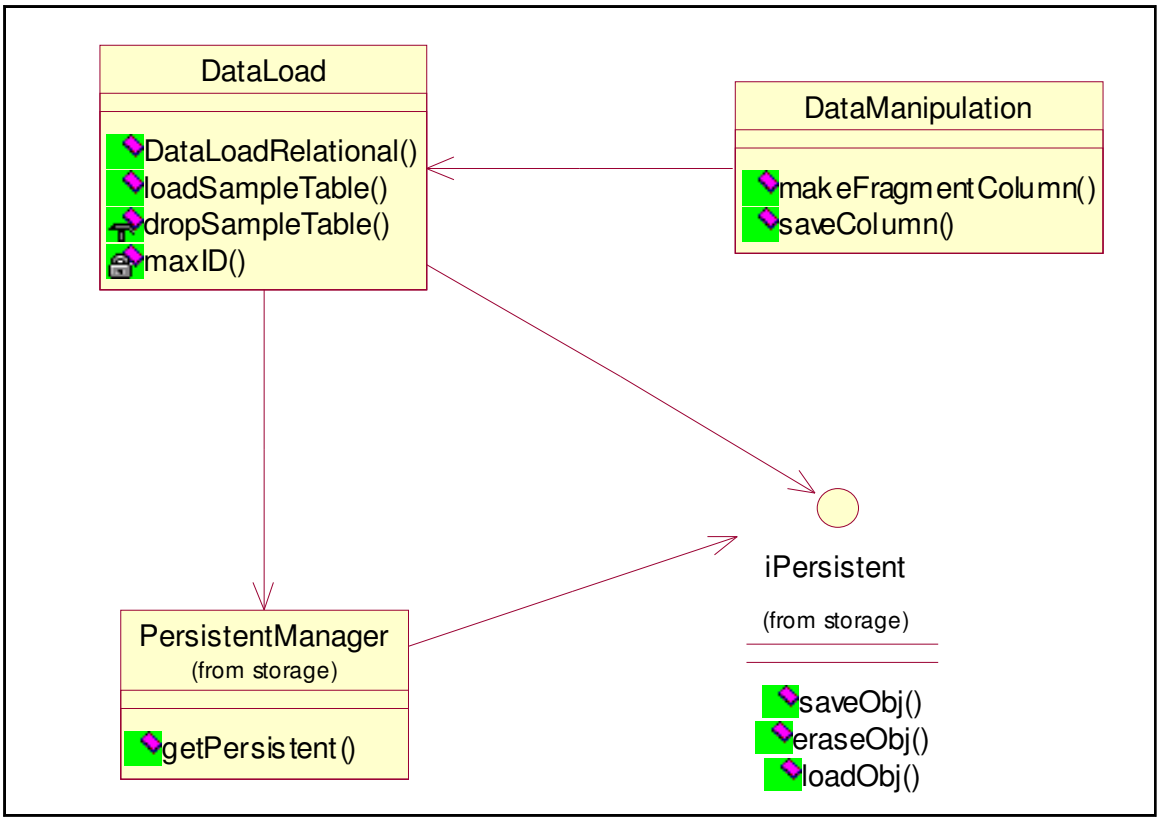

Figura 5.4 – Diagrama de classes do pacote dataManipulation.

A camada de *metadata* responsável por toda estrutura de dados e metadados da metodologia FastCube, seus sub-pacotes e suas classes serão abordadas mais detalhadamente na próxima sessão.

A camada *Persistent* é implementada pelo pacote *Storage*, esse provê a persistência dos dados e metadados do modelo. A estrutura de persistência do FastCube foi elaborada de maneira a ser extensível a qualquer tipo de armazenamento de Base de Dados. Para esse protótipo, todos os dados e metadados foram instanciados em um banco de dados relacional (ORACLE). A título de exemplificação foram indicadas na Figura 5.5, como seria a extensão do modelo para armazenamento de outros modelos de dados como XML e TXT, respectivamente nas classes *Persistent_XML* e *Persistent_TXT*. A Figura 5.5 mostra que as classes persistentes implementam uma interface em comum *iPersistent*. A classe *PersistentManager* é que gerencia a persistência e supre as demais classes do FastCube do serviço de persistência através de um objeto devolvido pelo método *getPersistent* que implementa a interface *iPersistente.*

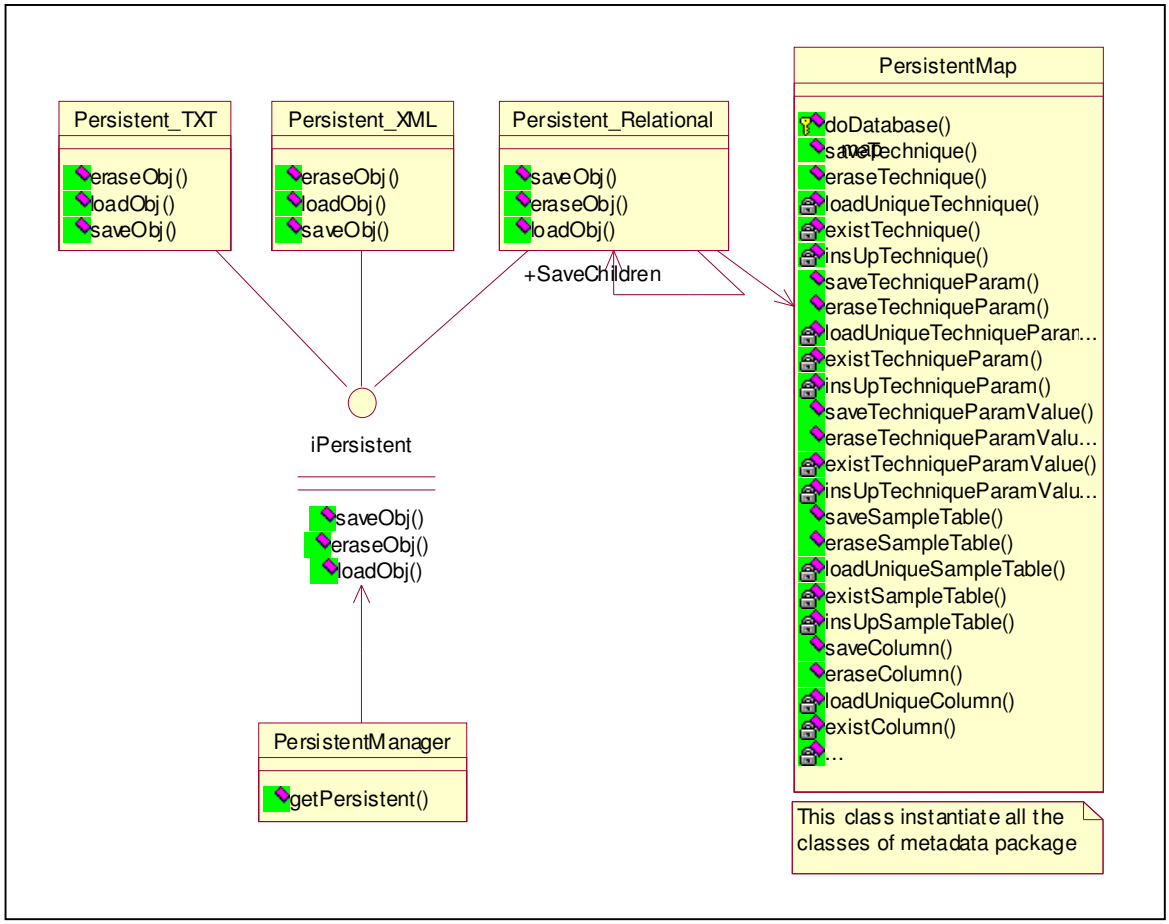

Figura 5.5 – Diagrama de classes do pacote Storage.

5.4 – O MODELO DE DADOS E METADADOS

Os principais requisitos para construção de um modelo que contemple a metodologia FastCube é a simplicidade e a completude. Esse modelo contempla as etapas do processos após o mapeamento e a integração dos dados. Foi projetado um modelo capaz de guardar todos os dados de entrada da metodologia (*TabelaAmostra*), guardar metadados de: características das colunas, mapeamento das transformações de dados e de modelo dimensional.

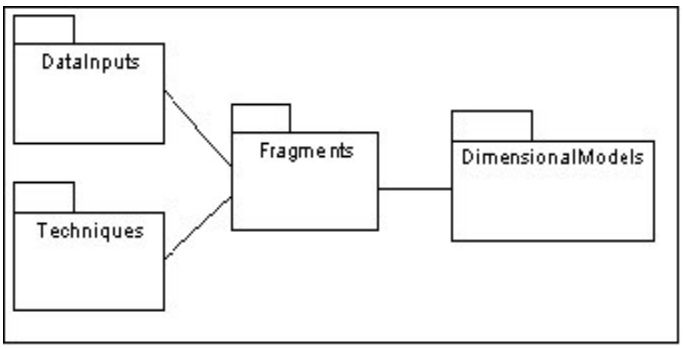

Figura 5.6 – Principais pacotes de dados e metadados

A Figura 5.6 mostra os pacotes das classes do modelo de dados e metadados. O pacote de dados de entrada ***DataInputs*** representa os dados e metadados da *TabelaAmostra,* ou seja ele armazena os dados de entrada da metodologia. Esse pacote possui 3 (três) classes:

- A ***SampleTable*** que guarda os metadados mais genéricos referentes a *TabelaAmostra* como o seu nome, descrição e a origem dos dados.

- A classe ***Column*** representa cada coluna da *TabelaAmostra* e possui atributos como o nome da coluna, descrição, tipo de dados e tamanho do dado.

- A ***ColumnValue*** pode ser considerada mais como uma coluna de dados do que metadados, pois essa classe diferentemente da ***SampleTable e da Column*** guarda os valores de dados de cada coluna da *TabelaAmostra*.

O pacote de Fragmentos ***Fragment*** representa os dados e metadados do resultado da fragmentação de cada coluna, através do método de distribuição de frequência somado a outros atributos de controle necessários a implementação. Essas classes podem ser consideradas também metadados das colunas da *TabelaAmostra*, pois nada mais são do que características extraídas das colunas e respectivos dados. Esse pacote é formado por duas classes:

- A classe ***Fragment*** representa algumas consolidações das colunas. Esse metadados são principalmente obtidos durante o processo de distribuição de frequência. São atributos como: nome e descrição, tipo, número de elementos distintos, total de valores nulos, total de valores fora da faixa (outlies), valores máximo, mínimo e médio (quando coluna numérica), indicador de visibilidade, indicador de uso, categoria (aditivo, semi-aditivo, não aditivo),

indicador de fragmento primário... Esses atributos são muito importantes quando está avaliando a qualidade dos dados.

- A classe ***FragmentValue*** guarda todos os valores distintos de cada coluna e suas respectivas ocorrências (frequência).

O pacote ***Techniques*** contempla o cadastro das técnicas e parâmetros que poderão ser usadas para as transformações dos dados no processo de pré-processamento, bem como armazena os valores dos parâmetros das técnicas que foram usadas para gerar uma coluna derivada, resultado de uma transformação. Esse pacote possui 3(três) classes:

- A classe ***Technique*** representa as técnicas que podem ser aplicadas para transformação dos dados (tratamento e limpeza). Nessa classe é que estão cadastradas as possíveis técnicas que poderão ser utilizadas. Os seus principais atributos são nome e descrição.

- A classe ***TechniqueParam*** os parâmetros de entrada das técnicas. Cada ocorrência ***TechniqueParam*** é um dos paramentros de uma técnica. Esses parâmetros podem ser números, caracteres ou qualquer elemento da metodologia FastCube como colunas e fragmentos. Seus principais atributos são nome, descrição e o tipo.

- A classe ***TechniqueParamValue*** é o valor de um parâmetro que foi aplicado para geração de uma nova coluna e conseqüentemente um novo fragmento. Cada fragmento que seja derivado deverá está associado a uma ou mais ocorrências de ***TechniqueParamValue.*** Seu principal atributo é o valor do objeto que foi passado como parâmetro para uma técnica.

O pacote ***DimensionalModels*** é a representação do modelo dimensional que será gerado a partir da modelagem FastCube. Esse pacote possui as classes básicas para geração de um Data Warehouse ou de um Data Mart. As suas classes são:

- A classe ***Attribute*** é gerada a partir de um fragmento que se deseja utilizar para fazer parte de uma dimensão ou de um fato. Segundo a metodologia, os atributos do modelo multidimensional devem derivar de um fragmento, mas

esse metamodelo pode contemplar definições de modelos multidimensionais sem fragmentos.

- A classe ***AttributeSet*** representa o uso de um atributo para uma determinada dimensão ou fato, isso porque um atributo derivado de um fragmento pode participar de vários Data Marts. Em um Data Mart um atributo pode participar da tabela de fato e em outro pertencer a uma dimensão. O atributo pode também diferentes nomes em diferentes situações. Essa classe possui os mesmos atributos de ***Attribute*** e mais a hierarquia se o atributo participar de uma dimensão que possua essa característica.

- As duas classes anteriormente apresentadas em conjunto com as classes ***Dimension, Fact, DataMart e DataWarehouse*** dão suporte a representação do modelo de dados DFM (Dimensional Fact Model) [GOL98] [GOL99].

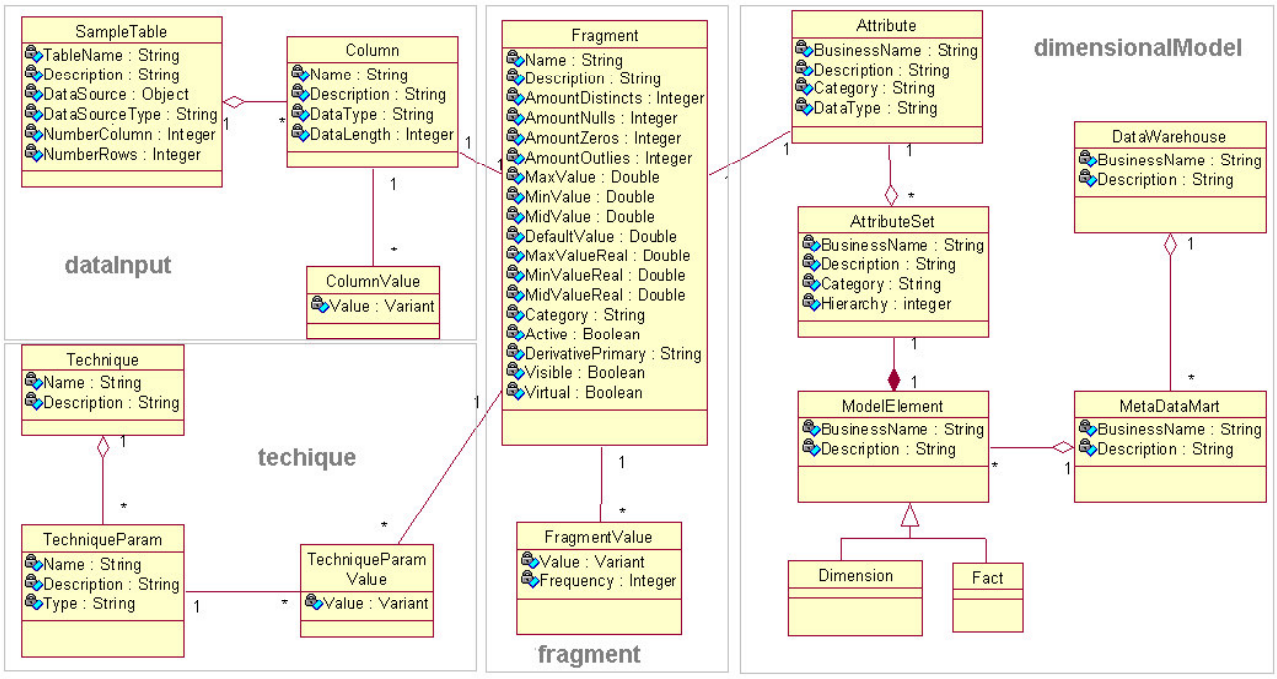

Figura 5.7 – Modelo de classes dos dados e metadados do FastCube.

A Figura 5.7 mostra as classes dos metadados e alguns dos seus principais atributos. Pode-se observar que a divisão em pacotes explicada anteriormente é bem natural e que mesmo nesse modelo mais completo essa divisão pode ser facilmente visualizada. Outra característica do modelo que pode ser facilmente notada é a classe ***Fragment*** servindo como

elemento de ligação entre os pacotes. Após a análise desse modelo de dados e de metadados, podemos perceber que ele contempla as etapas de fragmentação, análise/tratamento dos dados e a prototipação do modelo dimensional da Metodologia FastCube (Capítulo 4).

5.3 – QUALIDADE DOS DADOS

Conforme citado anteriormente, existe uma preocupação muito forte com a fidelidade dos dados do DW em relação as suas fontes originais em projetos de DW. Entretanto, a medição da qualidade dos dados adquiridos nas origens da produção tem que ser alta o suficiente para satisfazer as necessidades de informação da organização [INM98] [FIO00]. A idéia é que com a quantidade e principalmente com a qualidade dos dados de entrada, se consiga obter todos os requisitos necessários ao apoio à decisão. Na construção do DW. Normalmente os pesquisadores [MAR00] [ROT96] [KIM98] [INM97] dão maior destaque à medição de qualidade no final de todo o processo de ETL.

A verificação da qualidade dos dados tem que ser feita ao término da construção de cada nova carga do DW [INM97] [KIM98]. Uma dimensão de auditoria para registrar os passos da ETL se faz necessária [KIM98]. O acompanhamento da qualidade dos dados deve ser uma das tarefas ligadas ao Monitor de Dados [INM97]. Fica clara a preocupação com relação a compatibilidade entre os valores dos dados fontes e dos dados gerados para o DW, levando pouco em consideração se os dados de entrada possuem qualidade satisfatória.

Em nosso projeto foi adotado um modelo de medição de qualidade mais simplificado [JAR97a], por considerar que o mesmo atende o contexto do trabalho, com as 5 dimensões (completude, credibilidade, exatidão, interpretabilidade e validação) sendo usadas para medir a qualidade dos dados em cada elemento de nosso modelo. Para cada *TabelaAmostra* e para cada coluna da mesma são avaliadas as 5 dimensões de qualidade propostas. Nos elementos do modelo dimensional essas dimensões também são avaliadas.

Uma boa parte dos projetos de DW não fazem uma análise dos dados durante o estudo de viabilidade e na modelagem. Normalmente se parte do principio de que os dados operacionais estão adequados ou podem ser tratados e usados, sem um estudo inicial de viabilidade do modelo.

Definiu-se nesse trabalho que a medição de qualidade dos dados poderia ser feita fundamentalmente em 3 momentos do projeto (ver Figura 5.8):

- *Análise independente das fontes*: as bases operacionais que forneceram os dados para montagem do DW são avaliadas de forma separada, visando fazer um levantamento independente para cada fonte de dados. Esse processo pode ser executado no momento da decisão de se construir um DW e/ou a cada nova carga de dados.

- *Análise pós-integração:* as bases são avaliadas após a integração, possibilitando uma validação das possíveis técnicas empregadas que foram usadas no processo de integração ou determinando algumas necessidades que não forma contempladas no processo. Essa análise pode e deve feita a cada iteração. Essa avaliação é mais aplicável durante o processo de construção do DW, pois uma vez consolidado o processo de integração dificilmente a qualidade poderá ser alterada. Mesmo que ocorra algum problema quando o DW estiver em produção provavelmente essa falha aparecerá na análise ao término de cada do processo de ETL.

- *Análise ao término do processo de ETL:* encontrada em quase todas as literaturas de construção de Data Warehouse. Deve ser sempre executada ao termino de cada carga de dados. Essa carga pode ser para um protótipo com finalidades de avaliação de viabilidade ou a para uma carga de um DW que já esteja em produção.

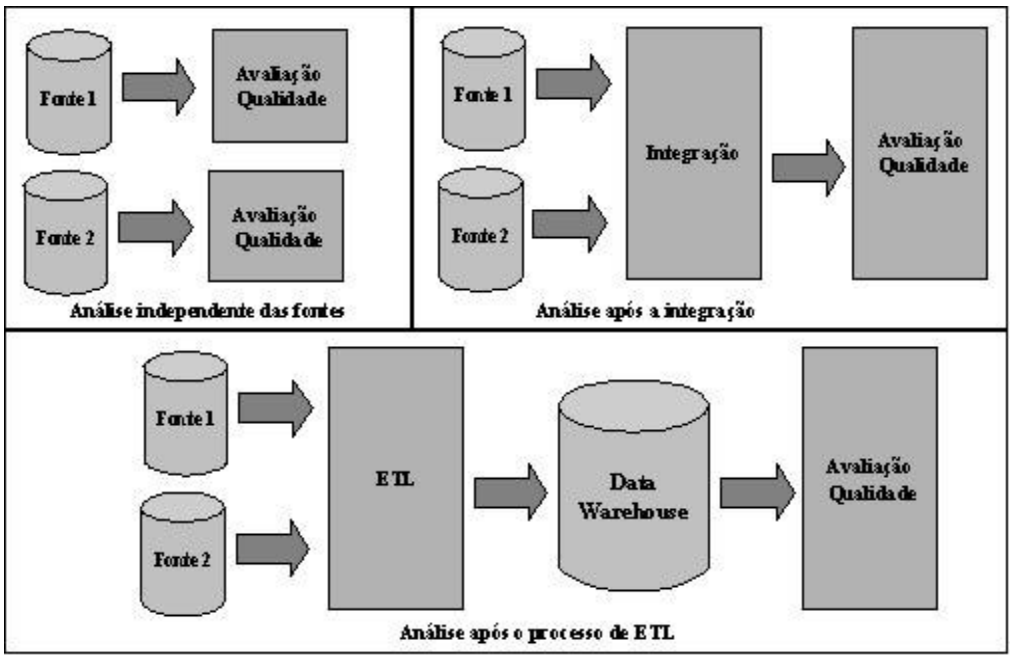

Figura 5.8 –Principais cenários para o processo de qualidade dos dados.

5.3 – TÉCNICAS DO PRÉ-PROCESSAMENTO - UM MODELO EXTENSÍVEL.

O modelo de técnicas elaborado para dar suporte ao pré-processamento do REDIRIS foi desenhado para ser simples, funcional e extensível. Todo o modelo é baseado no simples fato da geração de novas colunas da *TabelaAmostra* refletindo nessa nova coluna a aplicação de uma técnica de limpeza, tratamento ou transformação de dados. Ou seja, todas as técnicas têm como saída uma ou mais colunas, para serem adicionadas na *TabelaAmostra*. Com esse modelo, o repositório de técnicas se torna totalmente extensível, podendo a qualquer momento ser inseridas novas técnicas.

A partir da Interface visual, o usuário seleciona a(s) coluna(s) que deseja processar, seleciona a técnica especifica, que já deve ter sido cadastrada no modelo de metadados, e os seus parâmetros. Dentro do módulo de pré-processamento existem três funcionalidades básicas; a de interpretar a requisição de processamento da interface, a de processar a técnica especifica e a de persistir os dados e metadados. A Figura 5.9 ilustra esse processo.

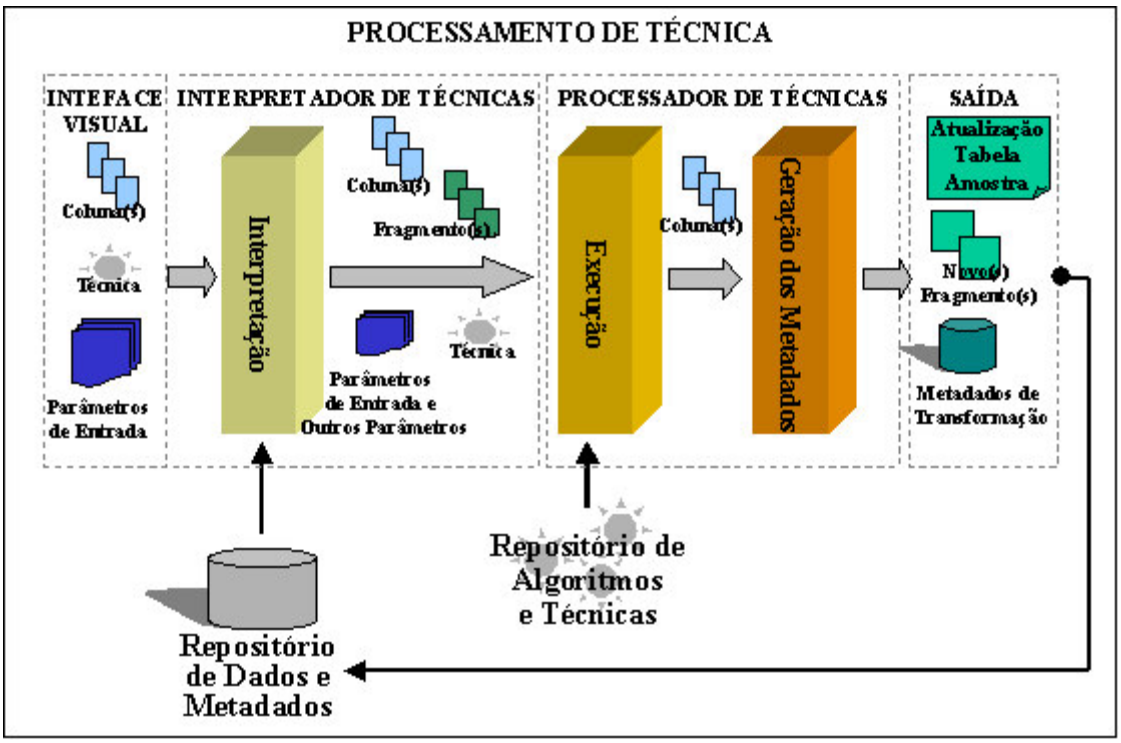

Figura 5.9 – Fluxo da aplicação de uma técnica no módulo Pré-Processador do REDIRIS.

O modelo prevê que as técnicas possam ter qualquer elemento do FastCube (colunas da *TabelaAmostra*, *fragmentos*) como parâmetro de entrada e mais outros parâmetros específicos de cada uma, porém a saída de cada técnica só pode ser uma lista de colunas a ser acrescentada diretamente na *TabelaAmostra* na mesma ordem. Com isso qualquer técnica pode ser adaptada, desenvolvida e acoplada diretamente no repositório de técnicas do REDIRIS, tornando-o totalmente extensível.

As técnicas poderiam trabalhar apenas com colunas de entrada e saída, isso tornaria a interface mais simples, mas foi observado que a maioria das técnicas de limpeza e tratamento de dados trabalha com elementos que já estão presentes no repositório de metadados em forma de *fragmentos*. Por esse motivo, as técnicas podem re-aproveitar os metadados já processados previamente, evitando códigos e processamento. E por esse motivo também é que o interpretador coloca a disposição das técnicas todos os elementos do FastCube.

Como a saída de cada técnica é apenas uma lista de colunas que deverá ser acrescentada na *TabelaAmostra*, então para cada nova coluna gerada temos também que gerar o fragmento específico. As informações que representam a geração dessa nova coluna

também têm que ser armazenadas através dos metadados de transformação. Essa tarefa serve para que se possa mapear posteriormente qualquer transformação de dados e também para dar suporte a colunas não materializadas.

A não materialização das colunas derivadas para armazenamento no Repositório é uma flexibilidade possível, ficando a critério do especialista quando essa deve ser ou não ser materializada. Essa funcionalidade é mais aplicada quando temos muitas colunas derivadas que não vão ser usadas na implementação final do Data Mart, porque são apenas elementos intermediários. Não materializar a coluna implica em armazenar apenas as informações necessárias para que se possa a qualquer instante chegar a essa mesma coluna. Esse recurso é comumente usado em técnicas de mineração de dados para se fazer à associação de dados categóricos a valores numéricos. Mas nesse caso para cada coluna é criada uma tabela que associe os valores categóricos distintos a valores numéricos [AUR99]. Isso reforça o argumento que a maioria das técnicas inteligentes de tratamento de dados fazem uso de alguma forma da distribuição de frequência ou de alguns dos seus elementos, como por exemplo, o uso dos valores distintos que é um elemento naturalmente retirado da distribuição de frequência já gerado e armazenado pela implementação.

Com esse modelo é possível a aplicação de várias técnicas em cascata para fazer um determinado tratamento de uma coluna, bastando apenas aplicar uma técnica por vez e o resultado de cada uma usar como entrada da seguinte. Considere o problema de tratamento de uma coluna que possui informações que naturalmente deveriam estar em duas colunas distintas; como exemplo o tratamento de um campo que possui dados de cidade e estado separados por um caracter qualquer. Deseja-se em primeiro lugar separar as duas colunas. Posteriormente aplica-se um algoritmo de similaridade nas duas colunas geradas para tratar alguns casos de erro de digitação. Após o tratamento de anomalias das duas colunas verifica-se que existem muitas cidades com baixa ocorrência (frequência) e deseja-se aplicar uma técnica que agrupe os elementos menos freqüentes da coluna cidade em um grupo chamado de OUTRAS. A Figura 5.10 ilustra como seria o fluxo necessário para essa operação com utilizando o mesmo exemplo descrito anteriormente.

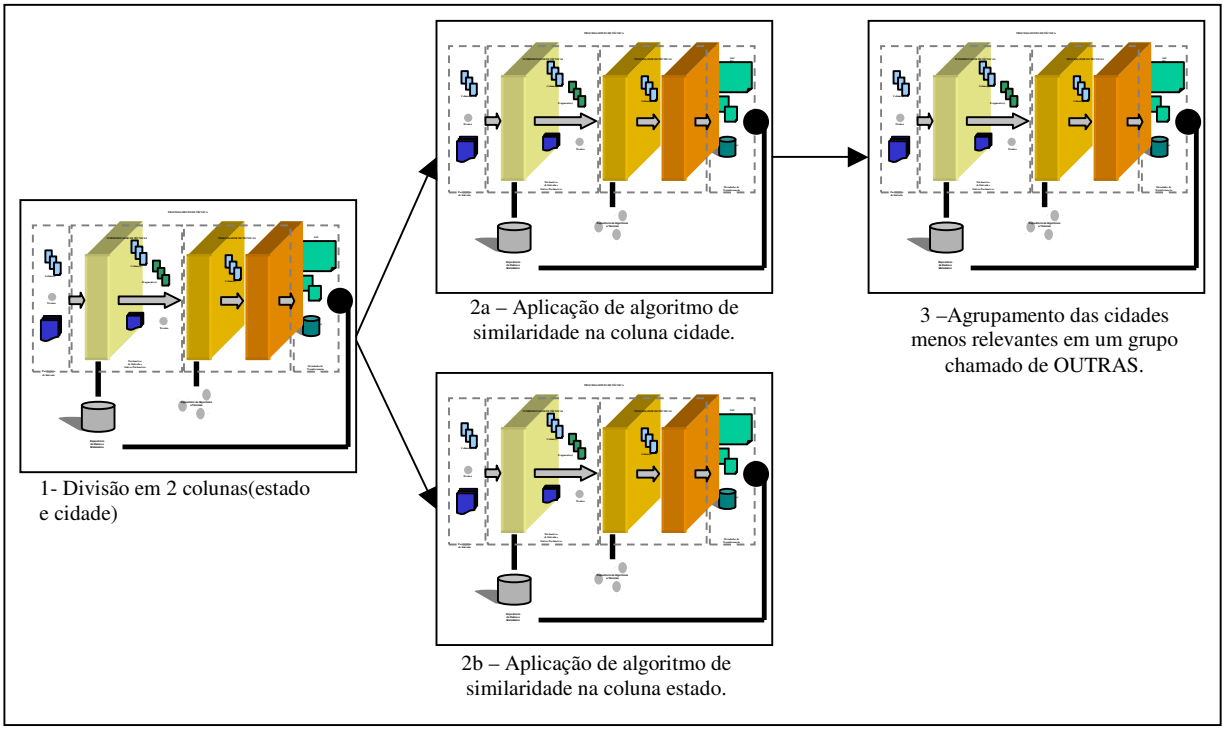

Figura 5.10 – Aplicação de várias técnicas em cascata.

5.3 – TÉCNICAS IMPLEMENTADAS NO REPOSITÓRIO DE TÉCNICAS.

Foi desenvolvido um modelo que não limita o uso de técnicas para pré-processamento de dados, mas para essa dissertação apenas um pequeno subconjunto de técnicas foi implementado. Esse subconjunto de técnicas e algumas ferramentas externas foram suficientes para se desenvolver um estudo de caso. Lembrando que essas técnicas são executadas após sua interpretação, pelo processador de técnicas (ver Figura 5.9), o que possibilita o uso de qualquer elemento do modelo de dados e metadados como parâmetro de entrada de uma técnica (fragmento, coluna).

Abaixo segue apenas uma descrição resumida de cada técnica implementada.

- *Desmembramento de Colunas* – essa técnica consiste em gerar colunas a partir de apenas uma coluna. Ela tem como entrada a coluna que se deseja desmembrar, o número de colunas de saída e um parâmetro para indicar se vai ser desmembrado por um separador específico, ou por largura fixa, o ultimo parâmetro e o separador, ou os tamanhos de cada coluna se for largura fixa.

- *Junção de Colunas* – essa técnica consiste em unir (concatenar) duas ou mais colunas. Ela tem como entrada as colunas que se deseja unir, e um separador caso deseje.

- *Agrupamento de Categoria não Relevante* – essa técnica consiste em juntar todos os elementos com frequência baixa em um grupo único. Normalmente essa técnica é usada para agrupar elementos pouco representativos, despoluindo a visualização de dados e descartando elementos pouco significativos de possíveis análises. Seus parâmetros de entrada são: a coluna, seu fragmento correspondente, o tipo do limite de agrupamento (por porcentagem do total de registros ou por uma frequência mínima) e o valor limite escolhido, podendo ser o valor da percentagem ou valor da frequência mínima.

- *Similaridade ou Matching* – esse algoritmo consiste em fazer um tratamento de anomalia com base na similaridade de seus valores, tentando identificar valores semelhantes que correspondam à mesma ocorrência. Foi projetado tendo como parâmetros de entrada: a coluna que se deseja tratar, o fragmento correspondente a essa coluna, o grau de precisão (número de 0 a 100), o flag de primeiro caráter igual, o flag de último caracter igual, flag de limpeza de lixo (retira todos os caracteres que não sejam letras, antes da comparação). Esse algoritmo em especial foi melhorado nesse trabalho, pois os seus similares normalmente possuem apenas o parâmetro de precisão. Com os novos parâmetros, notou-se que a taxa de acerto do algoritmo melhorou sensivelmente em relação as implementações originais. Esse algoritmo está descrito no anexo II dessa dissertação.

- *Discretização*: essa técnica visa transformar valores numéricos contínuos em intervalos de valores para melhor entendimento. Os paramentros de entrada são: a coluna numérica que se deseja discretizar e o método de discretização.

- *Divisão Intervalo*: essa técnica permite criar uma coluna com intervalos de valores de acordo com a distribuição de frequência da coluna, a coluna pode ser divida em n intervalos. Os parâmetros de entrada são: a coluna, o número de intervalos (se 4 quartil, se 10 decil, se 100 percentil).

- *Codificação Binária*: Transforma uma coluna em bits. A coluna a ser transformada pode ser numérica ou categórica. A entrada para essa técnica é uma coluna e seu

fragmento, método a ser aplicado, o método pode ser 1-de-N aplicado normalmente para valores categórico ou numérico discreto, Termômetro e Binário Padrão [AUR98] para colunas com tipo de dado numérico.

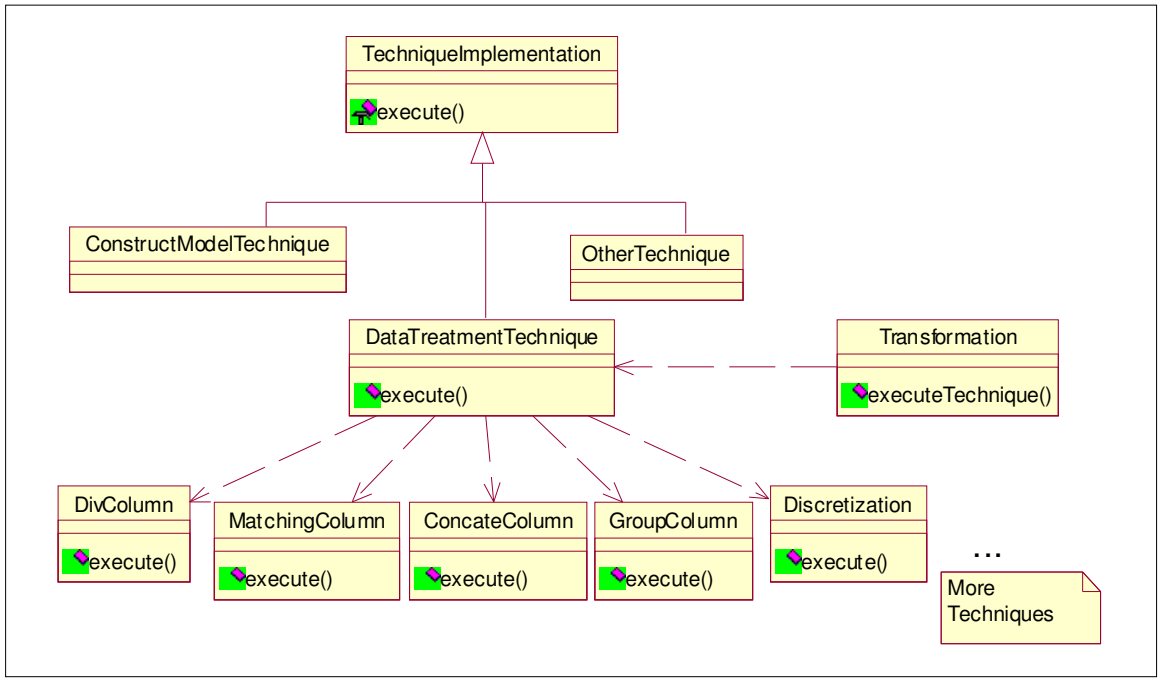

Figura 5.11 – Diagrama de classes de implementação das técnicas.

No diagrama de classes da Figura 5.11 está o modelo de classes adotado para implementação do repositório de técnicas do FastCube. A classe superior *TechniqueImplementation* possui as características de todos os tipos técnicas possíveis. Para nossa implementação foram codificadas as técnicas de tratamento de dados descritas anteriormente e uma técnica de construção de Modelo Dimensional descrita no anexo I.

Para cada técnica de tratamento de dados a ser implementada deve ser criada, uma classe que atenda essa nova funcionalidade. Essa classe deve ser carregada nos metadados do pacote *technique* (Figura 5.3) para que possa ser referenciada pela classe *DataTreatmentTechnique,* que é a responsável por gerenciar as técnicas disponíveis. Essa última classe trabalha recebendo solicitações execução de técnicas que são passadas como parâmetro. Então ela identifica a classe correspondente ao método solicitado como parâmetro e roda o método *execute()* correspondente a classe de técnica requerida. A classe T*ransformation* é apenas uma camada de abstração para isolar a aplicação das técnicas de tratamento de dados.

A implementação da técnica de auxilio a modelagem de dimensões, não foi registrada completamente no repositório do FastCube, pois existe a necessidade de um maior aprofundamento nas possíveis técnicas existentes e um modelo que possa contemplá-las. Como já foi dito anteriormente para essa implementação foi codificada apenas 1(uma) técnica que descobre possíveis hierarquias de atributos de dimensões, tendo como base a correlação de colunas, devolvendo como resposta uma matriz com os possíveis atributos que poderão formar algumas dimensões (anexo I).

5.4 – A INTERFACE E NAVEGAÇÃO DO PROTÓTIPO

A interface desse protótipo (figura 5.12) foi desenvolvida de maneira a contemplar a seqüência de passos da metodologia FastCube, tornando a navegação fácil e intuitiva. A interface foi desenvolvida em GUI com as classes Swing do JAVA.

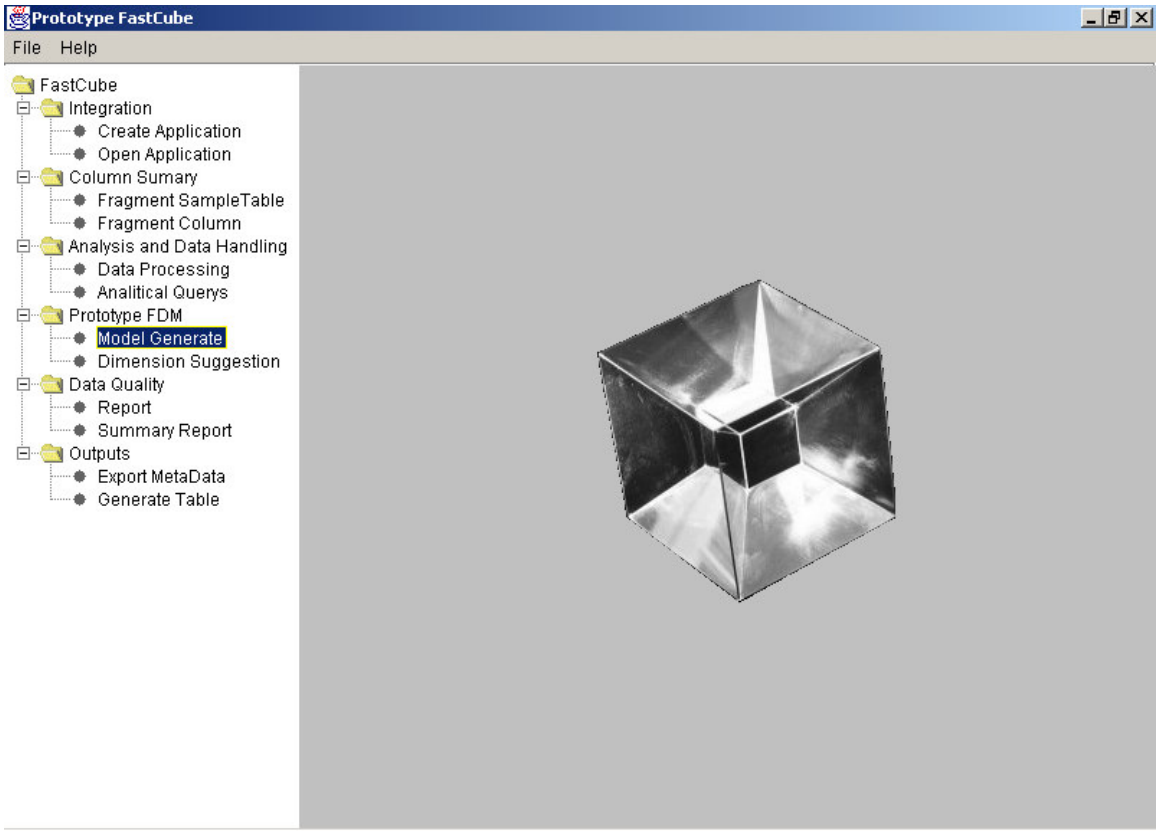

Figura 5.12 – Tela inicial do protótipo FastCube.

A tela inicial já vem com um menu a esquerda, onde pode ser verificada que as 4 (quatro) primeiras opções estão diretamente ligadas a seqüência do uso da metodologia. Em *integration* o usuário pode criar ou abrir uma nova aplicação. O ato de criar uma aplicação está diretamente ligada ao fato de fazer uma importação de uma tabela desnormalizada para o

nosso modelo de dados e metadados, que a partir desse momento a tabela se chamará *TabelaAmostra.* Para esse protótipo cada aplicação só pode está associada a uma *TabelaAmostra.* A tela de criação de uma aplicação com seus parâmetros é mostrada na Figura 5.13 e a tela de abertura de aplicação é mostrada na Figura 5.14.

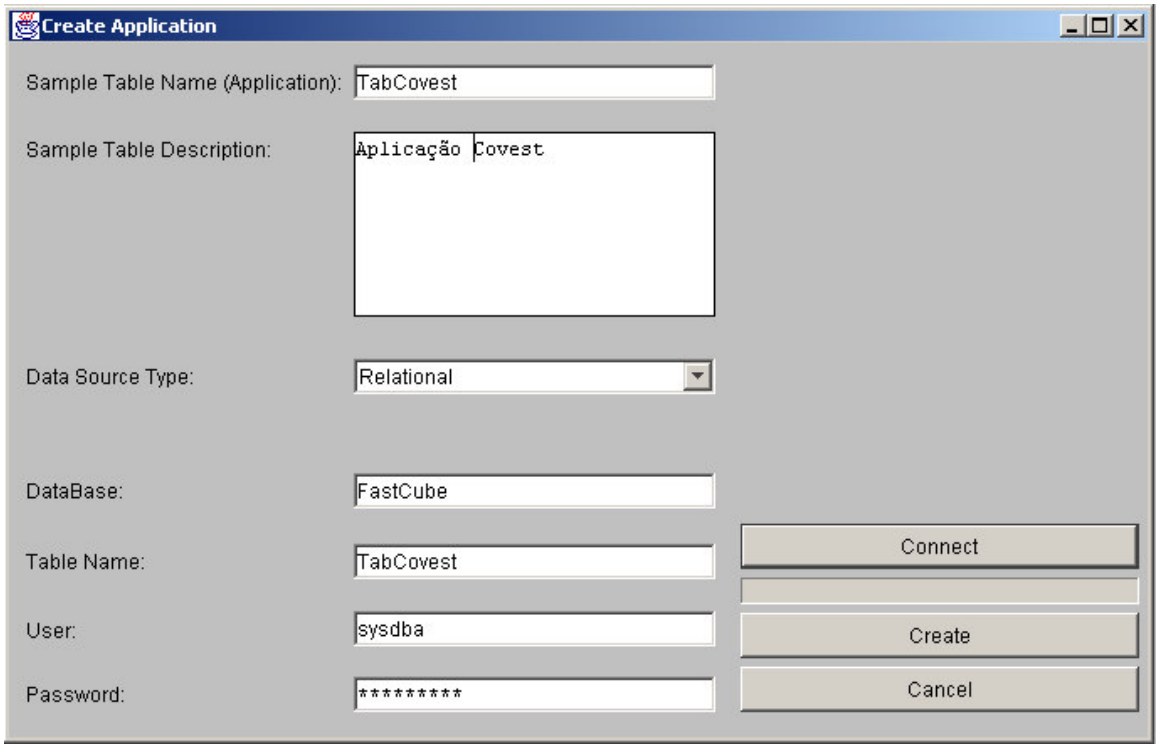

Figura 5.13 – Tela de criação aplicação.

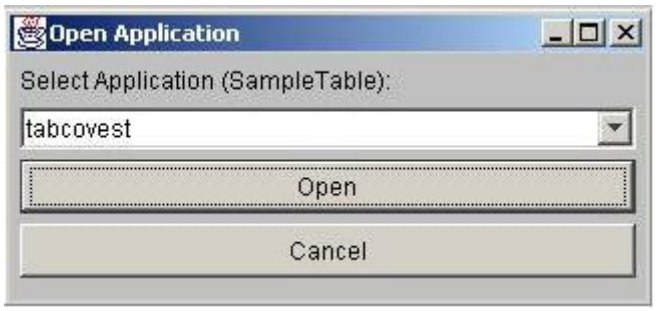

Figura 5.14 – Tela de abertura de aplicação.

Uma das principais funcionalidades do sistema é a fragmentação das colunas da *TabelaAmostra*, nesse protótipo a fragmentação é feita sempre para todas as colunas da *TabelaAmostra*. E após cada transformação que gere uma nova coluna também será gerado automaticamente o fragmento correspondente a coluna criada. A tela que faz a fragmentação inicial é mostrada na Figura 5.15.

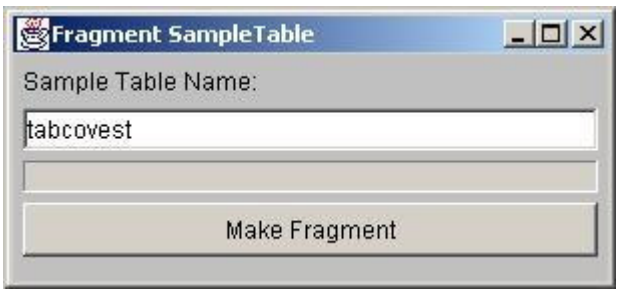

Figura 5.15 – Tela de fragmentação da *TabelaAmostra.*

A tela de pré-processamento de dados é a tela mais importante do protótipo, ela possui uma estrutura funcional que contempla todas as informações agregadas das colunas da *TabelaAmostra* na forma de fragmentos na tabela superior da tela (Figura 5.16). Na sua tabela imediatamente inferior a esquerda aparecem os valores possíveis (ocorrências) da coluna e sua frequência. Ao lado dessa tabela podemos selecionar a técnica que será aplicada em uma coluna e seus parâmetros.

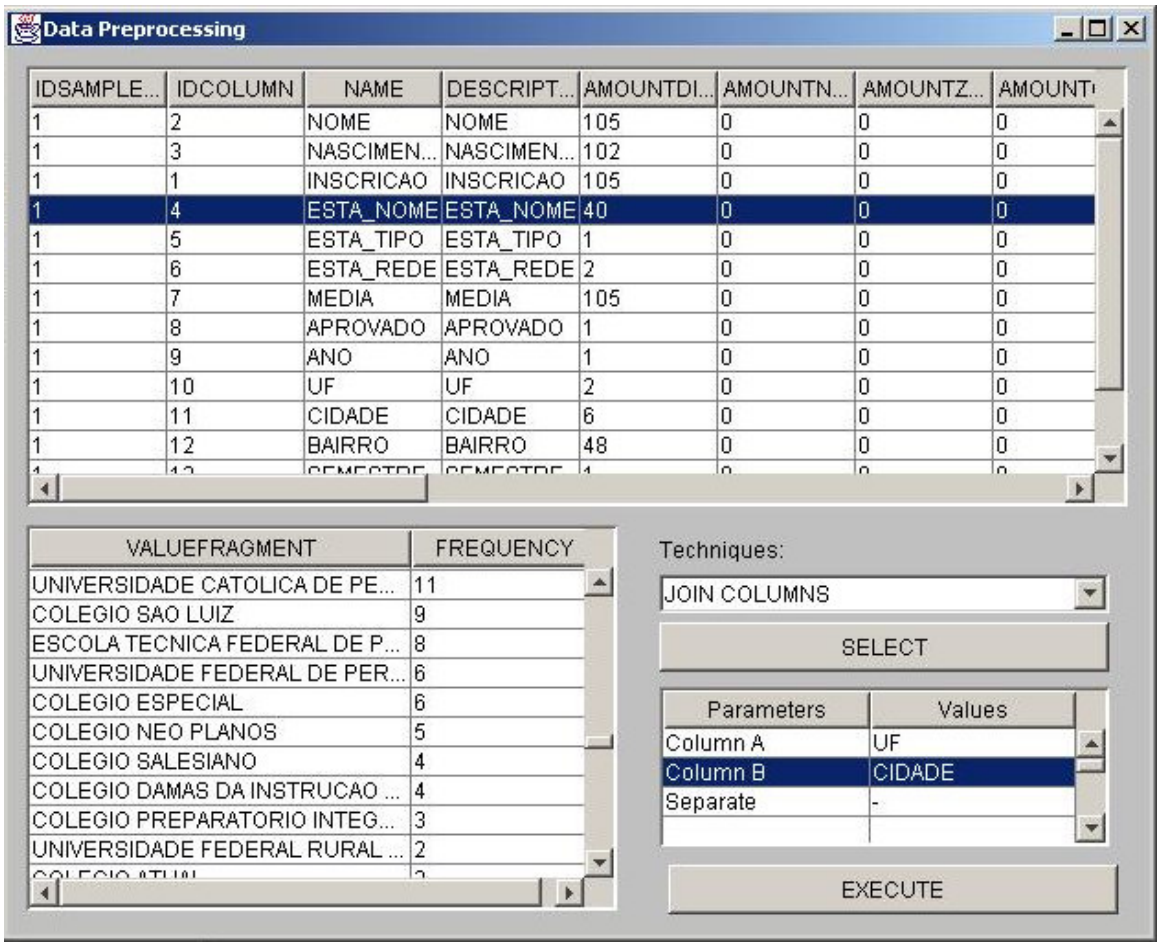

Figura 5.16 – Tela de pré-processamento de dados.

A tela de montagem de modelo dimensional (Figura 5.17) é feita para montagem de esquemas estrela em banco de dados relacional. A esquerda tem-se uma tabela com todos

os atributos disponíveis para participar do modelo, a esquerda temos as tabelas de dimensões e fato que serão preenchidas com os atributos selecionados na primeira tabela. Depois de selecionados os atributos que farão parte do modelo o sistema permite que esse modelo seja criado e fisicamente (botão *Generate*) e posteriormente carregado com dados (botão Data Load).

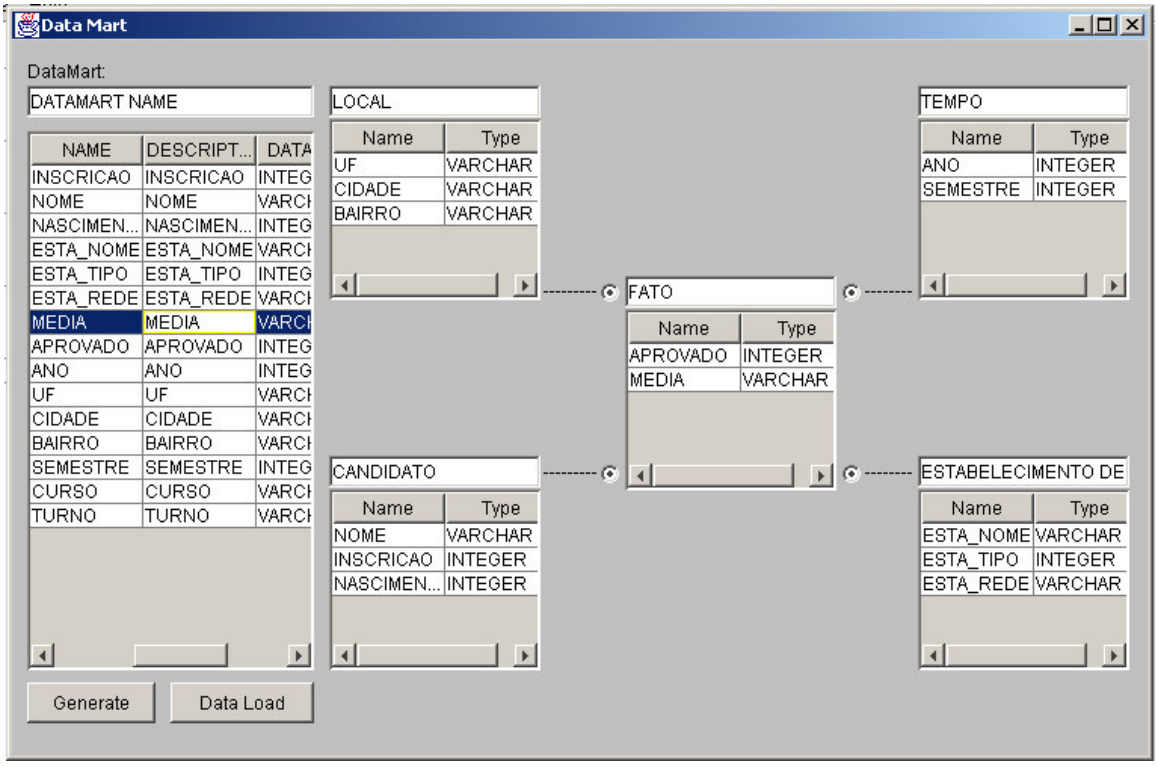

Figura 5.17 – Tela de montagem de modelo dimensional.

A tela de sugestão de dimensão (Figura 5.18) ainda pode ser bastante melhorada, mas já se pode ter idéia do potencial do uso de técnicas para essa finalidade. Ela é muito útil em casos que se deseja saber se existe hierarquia entre um conjunto atributos que sugerisse a criação de uma dimensão com todos ou com alguns deles. Essa tela dá suporte ao algoritmo do anexo I e funciona da seguinte maneira: na tabela mais a esquerda estão o conjunto de colunas disponíveis, na segunda tabela estão as colunas selecionadas para entrada na técnica e na terceira coluna o resultado da verificação, contendo apenas as colunas que deveriam pertencer a uma mesma dimensão e que possuem hierarquia entre si.

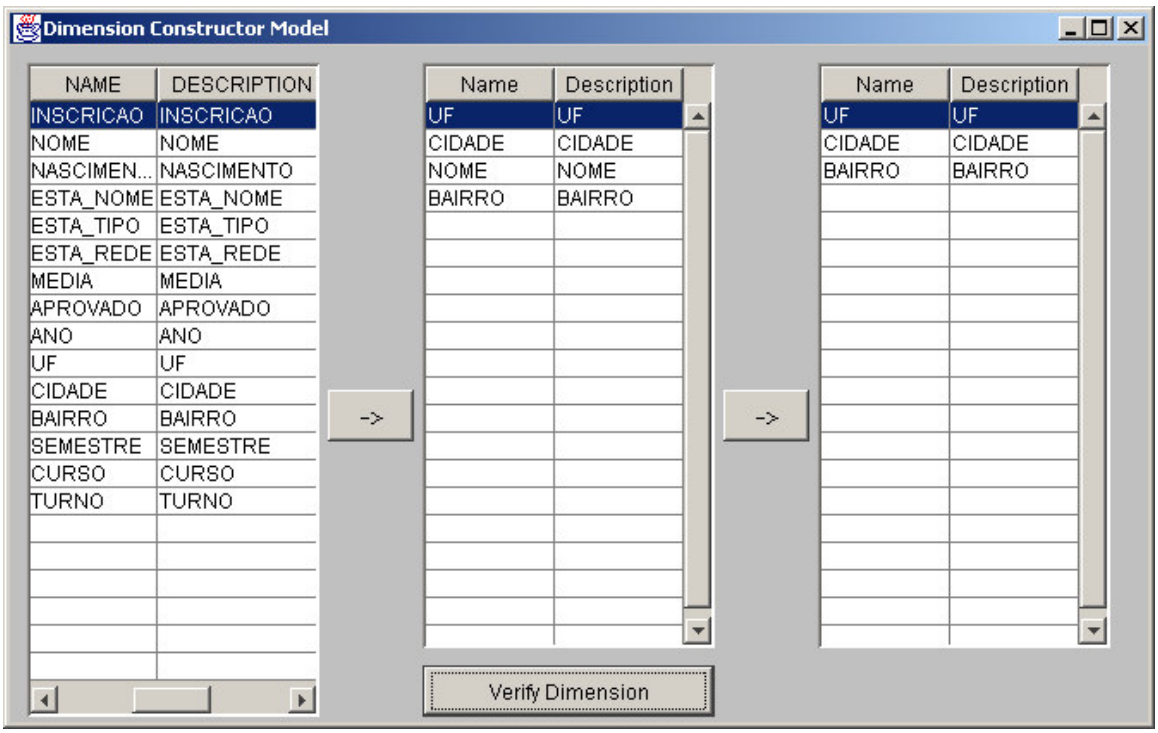

Figura 5.18 – Tela de sugestão de dimensão.

5.5 – POSSIBILIDADES COMPLEMENTARES DO MODELO

Durante a criação desse modelo de implementação da metododologia FastCube identificamos outras aplicações que não estão ligadas diretamente a construção de Data Warehouse.

O modelo pode ser usado para da verificação na qualidade de dados independentemente da origem de dados (operacional, DW, Data Mining...). Nos dados operacionais essa análise deve orientar possíveis melhorias no processo de captação de dados ou mesmo auxiliar o seu tratamento.

A saída padrão dessa implementação é um modelo dimensional, porém com um modelo de técnicas extensível nada impede que sejam desenvolvidas técnicas que gerem colunas que são dados transformados para se adequar a entrada de uma aplicação de Data Minig. Para esse caso, bastaria gerar uma *TabelaAmostra* com dados devidamente tratados e adequados a uma técnica especifica de mineração de dados. Isso é possível porque a maioria dos softwares de mineração de dados trabalha com uma tabela desnormalizada como entrada. Então, acrescentando novas técnicas para transformação dos dados no repositório do

REDIRIS, o modelo atual de técnicas contemplaria essa funcionalidade. Por exemplo: se desejássemos criar uma tabela para aplicar redes neurais com entrada entre 0 e 1, seria apenas necessário acrescentar técnicas que codificassem dados categóricos e que normalizassem dados numéricos. Essas técnicas criariam colunas na *TabelaAmostra* que seriam selecionadas e usadas para gerar uma nova tabela desnormalizada que serviria para treinamento de uma rede neural. Esse mesmo raciocínio pode ser aplicado para outras técnicas de mineração de dados como: algoritmos genéticos, algoritmos de indução de regras, métodos estatísticos e árvores de decisão.

6 – ESTUDO DE CASO – COVEST

Com o auxilio de algumas ferramentas de mercado e algumas implementações próprias, foi possível a aplicação e a validação da metodologia e das técnicas propostas nessa dissertação para obtenção de um Data Mart. Para esse estudo de caso, a implementação descrita no capítulo 5 foi utilizada juntamente com algumas ferramentas de mercado como o NeuralScorer e o Sagent[NEU02][SAG00]. Esse capítulo é a consolidação da metodologia apresentada no Capítulo 4 com a implementação do Capítulo 5, através de um estudo de caso real.

Para a validação dessa dissertação esse estudo de caso contempla a construção de um Data Mart baseado nos dados da Comissão de Vestibular (COVEST) realizado pela Universidade Federal de Pernambuco (UFPE) e Universidade Federal Rural de Pernambuco (UFRPE), tendo como foco a entrada dos novos alunos nos seus cursos de graduação.

6.1 – OBJETIVOS E REQUISITOS DO DATA MART - COVEST

O objetivo desse Data Mart é, baseado nos dados da COVEST, buscar de informações que caracterizem e tracem perfis dos candidatos; bem como medir o desempenho de grupos e identificar fatores relevantes que influenciam a aprovação dos candidatos aos cursos oferecidos pela UFPE e UFRPE.

Os perfis dos candidatos serão determinados levando-se em consideração os dados pessoais e as respostas ao questionário sócio-cultural, que é preenchido pelo candidato na ficha de inscrição para o concurso vestibular.

Espera-se com essa análise, identificar os principais fatores que determinam a situação de classificação do candidato ao concurso e a escolha profissional, através da primeira opção de curso do candidato.

6.2 – FERRAMENTAS UTILIZADAS

A principal ferramenta utilizada para geração desse estudo de caso foi a implementação do FastCube, descrita do Capítulo 5. Porém por entender que é interessante e produtivo a utilização dos bons recursos de ferramentas de mercado, fez-se uso de algumas para complementar a implementação. Entendemos que podemos tirar proveito do que existe

de melhor nas ferramentas, sem precisar re-inventar funcionalidades e que uma boa implementação deve ser aberta à integração com outras ferramentas.

Toda a simulação do modelo foi baseada em desenvolvimento próprio aliado às diversas ferramentas, integradas principalmente com rotinas feitas em linguagem JAVA, linguagem PL-SQL do ORACLE e algumas rotinas no software de construção de Data Warehouse SAGENT. O SAGENT é uma ferramenta de construção de Data Warehouse que faz toda parte de mapeamento e integração dos dados até as consultas OLAP. Do SAGENT também utilizamos as funcionalidades de mapeamento de fonte de dados e integração, para geração da *TabelaAmostra,* a partir dos diversos arquivos TXT, transformando-os em uma tabela única desnormalizada no ORACLE. Ainda do SAGENT foi usado o módulo de consultas OLAP para validação dos dados e modelo dimensional.

Algumas funcionalidades da ferramenta de Tratamento e Mineração de Dados NeuralScorer da Neurotech, como a visualização gráfica dos dados da *TabelaAmostra,* o indutor de regras e o algoritmo de árvore de decisão. A integração entre o protótipo e essa ferramenta foi facilitada, porque ela também trabalha com um conceitos muitos parecidos com os da metodologia FastCube, por exemplo o conceito de *Fragmento* é bem parecido com o de *Domain* dessa ferramenta [NEU02]. Outra semelhança é que todo o processo dessa ferramenta começa a partir de uma única tabela desnormalizada, alias, como em quase todos softwares de Data Mining. A grande desvantagem do uso do NeuralScorer em nosso estudo de caso é que seus dados e metadados estão em arquivos TXT, então tive-se que desenvolver algumas rotinas para carga e descarga dos dados no Banco de Dados ORACLE.

No ORACLE foram criadas todas as tabelas de dados e metadados utilizados nessa simulação e algumas partes da integração também foi feita em PL-SQL do ORACLE. O Erwin foi usado para visualizar os modelos diretamente do ORACLE. Fizemos uso de uma de suas características, que é a engenharia reversa de modelos. Então utilizamos principalmente a tabela de dados e metadados do ORACLE como interface para as diversas integrações.

6.3 – DESCRIÇÃO DOS DADOS

É comum nas universidades a aplicação de um formulário de inscrição e um questionário sócio cultural aos candidatos ao vestibular. Toda a informação dada pelo candidato é digitalizada e armazenada na base de dados da universidade. Essa base de dados

por si própria já fornece informações importantes para a universidade, pois esses dados já são coletados de maneira a direcionar investigações sobre o perfil dos candidatos. Entretanto a universidade estaria subtilizando esses dados se não fosse capaz de utilizá-los inteligentemente. Os dados foram cedidos para essa pesquisa e os resultados obtidos estão servindo como referencia para a própria administração da universidade, em especial ao Núcleo de Tecnologia da Informação-UFPE (NTI-UFPE).

Os dados brutos do COVEST 2000 ficam armazenados no Mainframe do CPD do NTI da UFPE em uma base de dados Hierárquica. Esses dados não estão acessíveis à pessoas não autorizadas pelo NTI, uma vez que algumas dessas informações sobre os candidatos são sigilosas. Os administradores de banco de dados do NTI disponibilizaram os dados em formato texto separados por virgulas.

Por se tratar de banco de dados hierárquico a integridade dos dados só pode ser garantida através dos programas que os geram e manipulam e como o sistema COVEST de controle do vestibular não tem como objetivo principal coletar de dados pessoais no formulário preenchido pelo candidato (incluindo o questionário sócio-cultural), a integridade desses dados não está sendo feita nos programas, o que compromete a qualidade das informações. A justificativa dada pelos técnicos é que se esse tratamento fosse feito o tempo de carga seria muito grande. Por isso eles não cuidam de aspectos relevantes para uma análise sócio-econômica, como padronizar o nome dos bairros, cidades ou mesmo validar as possíveis respostas do questionário. Isso reforça a necessidade de um tratamento mais elaborado para se extrair o máximo possível de informações desses dados.

A complementação dos dados enviados no formato TXT foi feita a partir do manual do candidato, porque nem todas as informações estavam em meio magnético. Em alguns casos também foi necessária a digitação de algumas tabelas a partir de informações contidas nesse manual para tornar o modelo mais completo.

Também foi elaborado um modelo lógico relacional, que não representava fielmente as tabelas fornecidas em TXT, mas fornecia uma abstração suficientemente boa para o entendimento dos dados. (ver Figura 6.1).

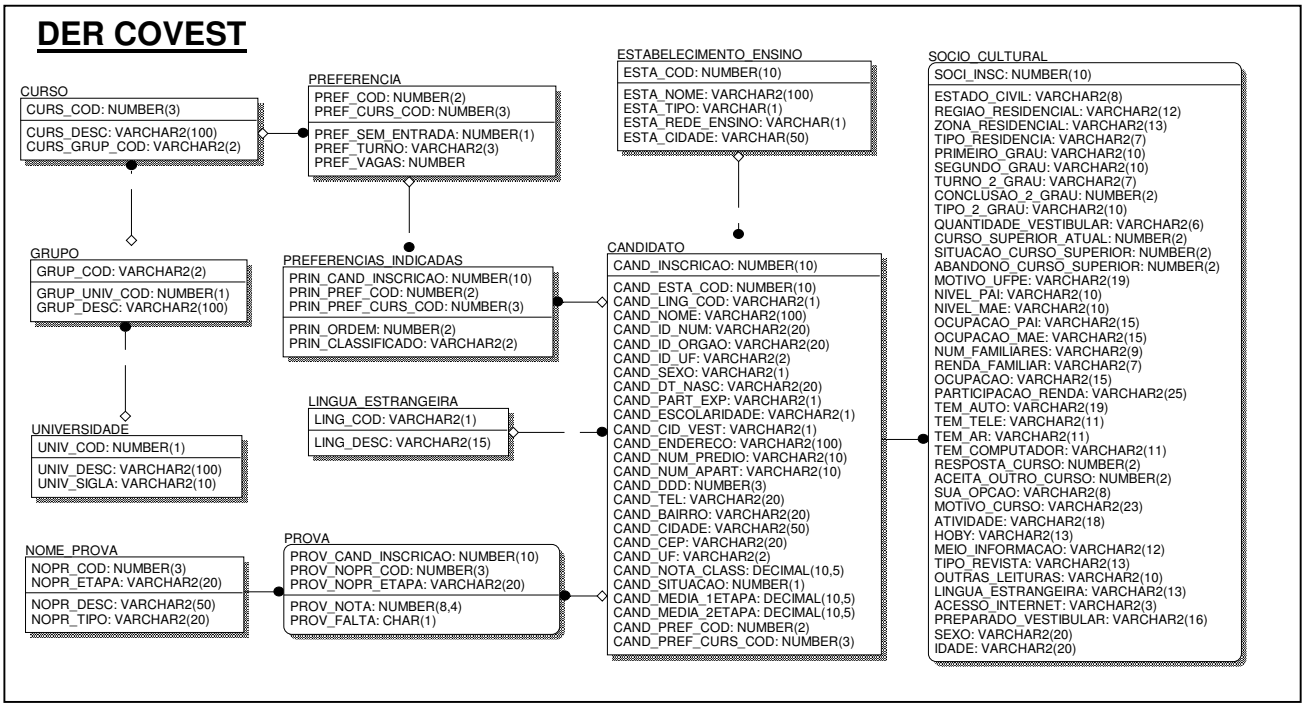

Figura 6.1 – Modelo Lógico Relacional dos dados de entrada.

6.4 – GRANULARIDADE

A determinação da granularidade se deu no nível de candidato, com agregações das notas das disciplinas em primeira e segunda etapa. Apesar da definição da granularidade nesse estudo de caso não ser muito complexa, foram avaliadas várias definições de granularidade.

A definição da granularidade influi diretamente no nível de performance das consultas, porque o volume de dados é diretamente ligado a essa definição. Porém a complexidade final das consultas também pode ser afetada se a definição da granularidade não for bem sucedida. Por exemplo, se as operações de consultas que evolvam consolidações não forem simples cálculos de agregações, as operações necessárias para realização dessas, podem comprometer todo o uso e a finalidade do Data Mart, pois aumentará consideravelmente a complexidade para formulação dessas consultas. No caso COVEST, se fosse determinada a granularidade no nível de nota de prova ou no nível de questão de prova, além de aumentar significativamente o volume de dados no DW, as consultas ficariam bem mais complexas, pois para se chegar a consolidações desejadas envolveria um conhecimento e

esforço muito maior do projetista de DW e do usuário final (OLAP), pois como se sabe o cálculo de score e a classificação de um candidato não é algo muito trivial.

Para o objetivo pretendido com os dados do COVEST, ficou definido que seria utilizada a granularidade no nível de candidato, pois essa atenderia a todos os objetivos definidos. Na próxima sessão serão descritas algumas consolidações e tratamentos necessários aos dados para adaptá-los a granularidade escolhida.

6.5 – SELEÇÃO E INTEGRAÇÃO DOS DADOS

A fase da seleção dos dados é iniciada quando a equipe de desenvolvimento e os usuários do DW selecionam dentre os dados o que será importante para o processo. Como indica a metodologia no Capítulo 4, a seleção e a integração não foi feita de forma definitiva, e sim de maneira incremental e iterativa. Foram selecionados os dados, juntamente com os especialistas do sistema COVEST. Inicialmente foi selecionado apenas um pequeno conjunto de atributos, que foi julgado relevantes para traçar o perfil dos candidatos.

Variáveis como número de CPF e RG dos candidatos aliadas ao ano do vestibular, foram usadas para determinar a qualidade de preenchimento de campos que deveriam ser chaves candidatas (esse tipo de análise já foi abordado em capítulos anteriores). Porém o preenchimento em si desses campos é irrelevante para o objetivo final desse projeto, pois não se quer avaliar dados individuais e sim tratar dos perfis dos candidatos. De qualquer forma essas variáveis podem fazer parte do Data Mart, caso se queira contemplar informações individuais de candidatos. Outras informações como notas de provas foram agregadas de maneira a simplificar o modelo.

A fase de seleção dos dados só pode avançar até antes de efetuada a primeira iteração. Somente a partir daí será possível verificar a relevância de alguns atributos, a necessidade de criação de novos atributos derivados e a captura de outros, antes considerados irrelevantes.

A partir dos atributos selecionados, foi gerada a tabela desnormalizada *TabelaAmostra*, contendo apenas um registro para cada candidato com todos os atributos considerados relevantes para essa primeira iteração.

Para a geração da *TabelaAmostra* foi preciso realizar transformações em algumas outras tabelas de relacionamento existentes na base. A tabela SOCIO_CULTURAL, por exemplo, possuía 40 registros para cada candidato, sendo que cada registro possuía três campos (inscrição, pergunta e resposta) para cada pergunta no questionário. A transformação foi feita de modo que a partir dessa tabela fossem retirados 40 campos de resposta do questionário. A tabela PREFERENCIAS_INDICADAS apresentava o mesmo problema. Havia mais de uma entrada para cada candidato, sendo indicada a ordem de preferência do turno escolhido. Nesta tabela, a solução foi bem parecida, porém mais simples, pois para efeito de construção do modelo foi considerada relevante apenas a primeira escolha de turno do candidato. Assim para a adequação da granularidade, diversas tabelas foram transformadas e tratadas nessa integração.

O resultado final dessa etapa é a montagem de uma tabela com os seguintes atributos descritos abaixo:

Atributo	Tipo	Descrição
Ano	Numérico	Ano do vestibular
Nome	Categórico	Nome do candidato
Sexo	Categórico	Sexo do candidato
Idade	Numérico	Idade do candidato
Estado	Categórico	Estado da residência
Cidade	Categórico	Cidade da residência
Bairro	Categórico	Bairro da residência
Opcao_Lingua	Categórico	Idioma estrangeiro escolhido na prova
Universidade	Categórico	UFPE ou UFRPE
Grupo	Categórico	Grupo que o curso pertence
Curso	Categórico	Curso escolhido como primeira opção
Turno	Categórico	Opção de turno do candidato
Estabelecimento_Ensino	Categórico	Nome do estabelecimento
Estabelecimento_Cod	Categórico	Código do estabelecimento
Estabelecimento_Tipo	Categórico	Tipo estabelecimento
Estabelecimento_Rede	Categórico	Se o estabelecimento de ensino é publico ou privado
Estabelecimento_Cidade	Categórico	Cidade do estabelecimento
Historia	Numérico	Nota na prova de história
Matemática	Numérico	Nota na prova de matemática
Português	Numérico	Nota na prova de português
Língua_Estrangeira	Numérico	Nota na prova de Ling. Estrangeira
Media_1Etapa	Numérico	Média 1ª Etapa do vestibular
Media_2Etapa	Numérico	Média 2ª Etapa do vestibular

Media_Final	Numérico	Média Final ou Score do vestibular
Situação	Categórico	Resultado final no concurso. Ex. Aprovado, eliminado, reprovado...
Aprovado	Numérico	Aprovado = 1 e Reprovado = 0
Q_Estado_Civil	Categórico	Estado Civil
Q_Sexo	Categórico	Sexo do candidato
Q_Idade	Categórico	Idade do candidato
Q_Regiao_Residencial	Categórico	Região da residência
Q_Zona_Residencial	Categórico	Zona da residência
Q_Tipo_Residencia	Categórico	Tipo da residência
Q_Primeiro_Grau	Categórico	Informações do 1grau
Q_Segundo_Grau	Categórico	Informações do 2grau
Q_Turno_Segundo_Grau	Categórico	Turno do 2grau
Q_Conclusao_2_Grau	Numérico	Ano de conclusão do 2grau
Q_Tipo_2_Grau	Categórico	Tipo de 2Grau
Q_Quantidade_Vestibular	Categórico	Quantos vestibulares prestou?
Q_Curso_Superior_Atual	Categórico	Fez ou vem fazendo curso superior
Q_Situação_Curso_Seperior	Categórico	Situação do curso superior
Q_Abandono_Curso_Superior	Categórico	Abandonou curso superior
Q_Motivo	Categórico	Motivo de escolha da UFPE/UFRPE
Q_Nivel_Pai	Categórico	Escolaridade do Pai
Q_Nivel_Mae	Categórico	Escolaridade do Mãe
Q_Ocupacao_Pai	Categórico	Ocupação Profissional Pai
Q_Ocupacao_Mae	Categórico	Ocupação Profissional Mãe
Q_Num _Familiares	Categórico	Número de membros da Família
Q_Renda_Familiar	Categórico	Renda Familiar em salários mínimos
Q_Ocupacao	Categórico	Ocupação profissional do candidato
Q_Participacao_Renda	Categórico	Participação na renda familiar
Q_Tem_Automovel	Categórico	Proprietário de automóvel
Q_Tem_Telefone	Categórico	Proprietário de telefone
Q_Tem_Ar	Categórico	Proprietário de ar condicionado
Q_Tem_Computador	Categórico	Proprietário de computador
Q_Aceita_Outro_Curso	Categórico	Aceita outra opção de curso que não a primeira
Q_Nivel_Decisao	Categórico	Certeza sobre o curso escolhido
Q_Motivo	Categórico	Motivo de escolha do curso
Q_Atividade	Categórico	Tipo de atividade que pratica
Q_Hobby	Categórico	Hobby
Q_Meio_Informacao	Categórico	Meio de informação que utiliza
Q_Tipo_Revista	Categórico	Tipo de revista que costuma ler
Q_Outras_Leituras	Categórico	Outros tipos de leitura
Q_Domínio_Lingua_Estrangeira	Categórico	Conhecimento em língua estrangeira

Q_Acesso_Internet	Categórico	Acessa a Internet
Q_Preparado_Vestibular	Categórico	Fez preparação especifica para o vestibular

Tabela 6.1 – Atributos da tabela desnormalizada

6.6 – MANIPULAÇÃO E TRATAMENTO DOS DADOS

De maneira geral os principais problemas encontrados com os dados foram a ausência de preenchimento, os valores fora do domínio e os erros de digitação. Mas esses problemas eram principalmente encontrados nas colunas referentes ao questionário sócio-cultural, pois os dados relevantes de avaliação e classificação do candidato estavam com um nível de qualidade excelente considerando sua completude, credibilidade e exatidão [JAR97a].

Ao ser montada a primeira *TabelaAmostra* já se sabia, por análises precedentes, que algumas colunas representavam a mesma informação, então com base na qualidade das colunas de mesmo conteúdo, foram escolhidas para eliminação as que tinham pior nível de preenchimento e/ou menor qualidade. Um exemplo foi a eliminação de **Q_Sexo, Q_Idade, Q_Estado_Civil** do questionário sócio-cultural. Elas foram escolhidas para serem excluídas, pois as informações colhidas em outras colunas de tabelas apresentaram melhor nível de preenchimento e melhor qualidade de dados.

No preenchimento de colunas como bairro, cidade e estado (UF) os erros de digitação foram muito comuns; erros que poderiam ser evitados se existisse um cadastro prévio onde os digitadores pudessem selecionar o local desejado em uma lista. O tratamento para esse tipo de campo foi a aplicação de um algoritmo de similaridade (matching), que de forma automática consegue recuperar muitos dados que poderiam ser considerados perdidos (exemplo: RE**FICE = RECIFE**).

Uma outra técnica muito comum aplicada em uma boa parte das colunas, inclusive após o uso da técnica de similaridade, foi o agrupamento de elementos de colunas com baixa frequência em um grupo chamado de OUTROS, isso foi realizado para redução da dimensionalidade da coluna e eliminação de ocorrências e baixa frequência que não possuem nenhum valor estatístico. Outro tipo técnica de agrupamento utilizado foi a criação de um grupo chamado Not Avaliabled (NA), onde os elementos que estavam fora do domínio da

coluna, eram substituídos por esse rótulo (NA). Essa codificação foi principalmente utilizada nas colunas do questionário sócio-cultural, porque todas as colunas do questionário possuíam uma boa parcela de valores de colunas fora do domínio.

6.7 – GERAÇÃO DO MODELO E POVOAMENTO

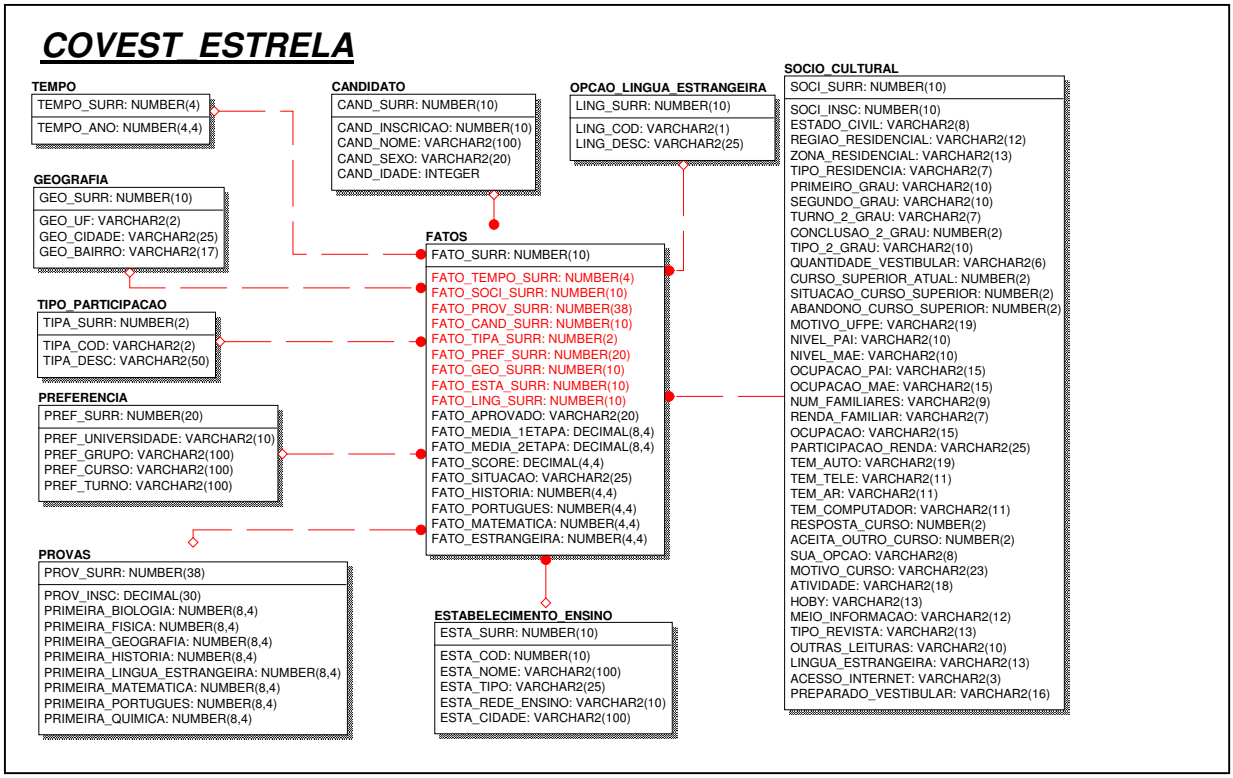

Figura 6.2 – Modelo estrela do protótipo final do COVEST.

Dando prosseguimento a metodologia FastCube, após o tratamento de dados o que se deseja é a utilização do mesmo para montagem de um modelo multidimensional que dê suporte aos requisitos do usuário. A essa altura o especialista já conhece muito bem a disponibilidade e a qualidade dos dados que possui e já tem, segundo a metodologia FastCube (sessão 4.6), a capacidade para escolher as colunas que farão parte do modelo dimensional, além classificá-las em fatos ou em dimensões de um Data Mart.

Para esse protótipo foi escolhido um subconjunto de colunas capaz de fornecer resposta a alguns questionamentos. Todos os metadados para geração desse modelo dimensional foram carregados no modelo de metadados, no pacote *DimensionalModels* (sessão 5.3), então a construção do modelo, sua geração no banco ORACLE já estava semi-automatizada pela ferramenta. O modelo dimensional final do protótipo COVEST é apresentado na Figura 6.2.

Além dos metadados do modelo dimensional também temos no nosso modelo de dados um pacote de classes que representam todos os dados da T*abelaAmostra*. A partir desse pacote (*DataInputs*) podemos carregar todos os dados das colunas selecionadas para fazer parte do modelo dimensional. Alguns resultados obtidos com o Data Mart COVEST são mostrados na sessão seguinte.

6.8 – EXTRAÇÃO E ANÁLISE DOS RESULTADOS

A partir do modelo de dados do Data Mart COVEST montado, foi possível a aplicação de uma ferramenta OLAP para a extração dos resultados, através de relatórios, tabelas e gráficos. A ferramenta escolhida para essa tarefa foi a SAGENT, pois essa ferramenta por um módulo de importação de dados que possibilita fazer qualquer consulta OLAP, inclusive com disponibilidade automática na WEB.

Após adaptação do SAGENT à base relacional (esquema estrela), gerada pelo protótipo do ambiente, os dados foram disponibilizados para qualquer tipo de consulta que se desejasse fazer.

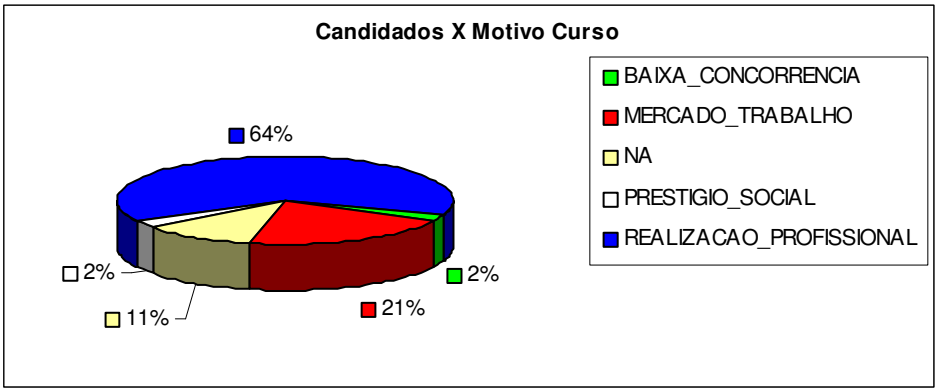

Figura6.3– Gráficos candidatos com a variável MOTIVO_CURSO

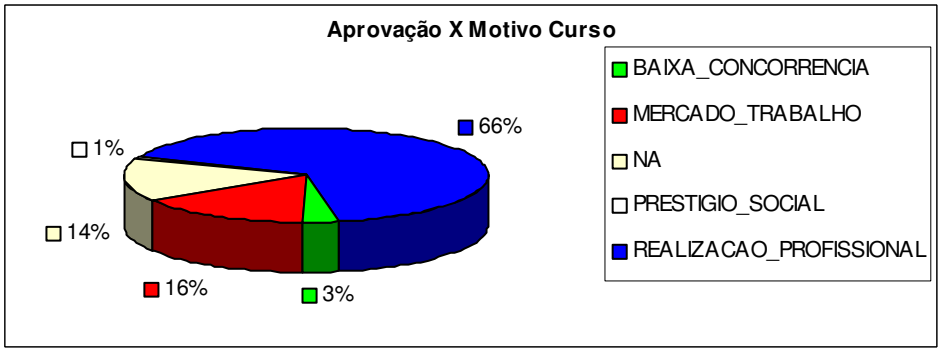

Figura 6.4 – Gráficos candidatos com a variável MOTIVO_CURSO

Os primeiros gráficos (Figuras 6.3 e 6.4) foram exportados do SAGENT para o Excel. Eles levaram em consideração a massa de candidatos e a massa de aprovados com a mesma variável, isso para que se tenham parâmetros de comparações.

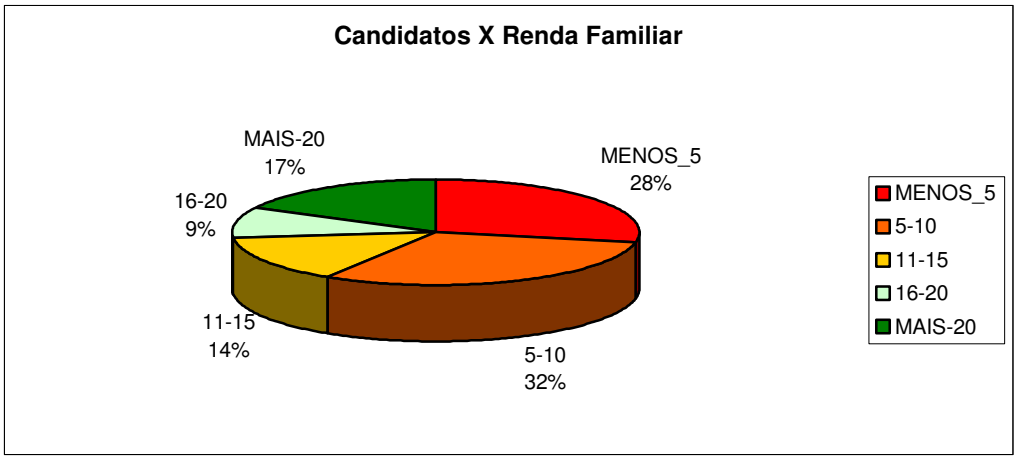

Figura 6.5 – Gráficos considerando a variável Renda Familiar

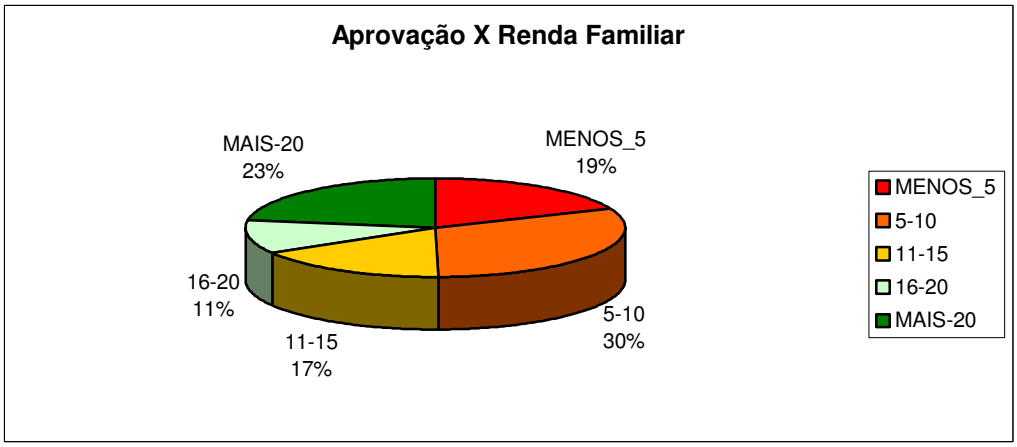

Figura 6.6 – Gráficos considerando a variável Renda Familiar

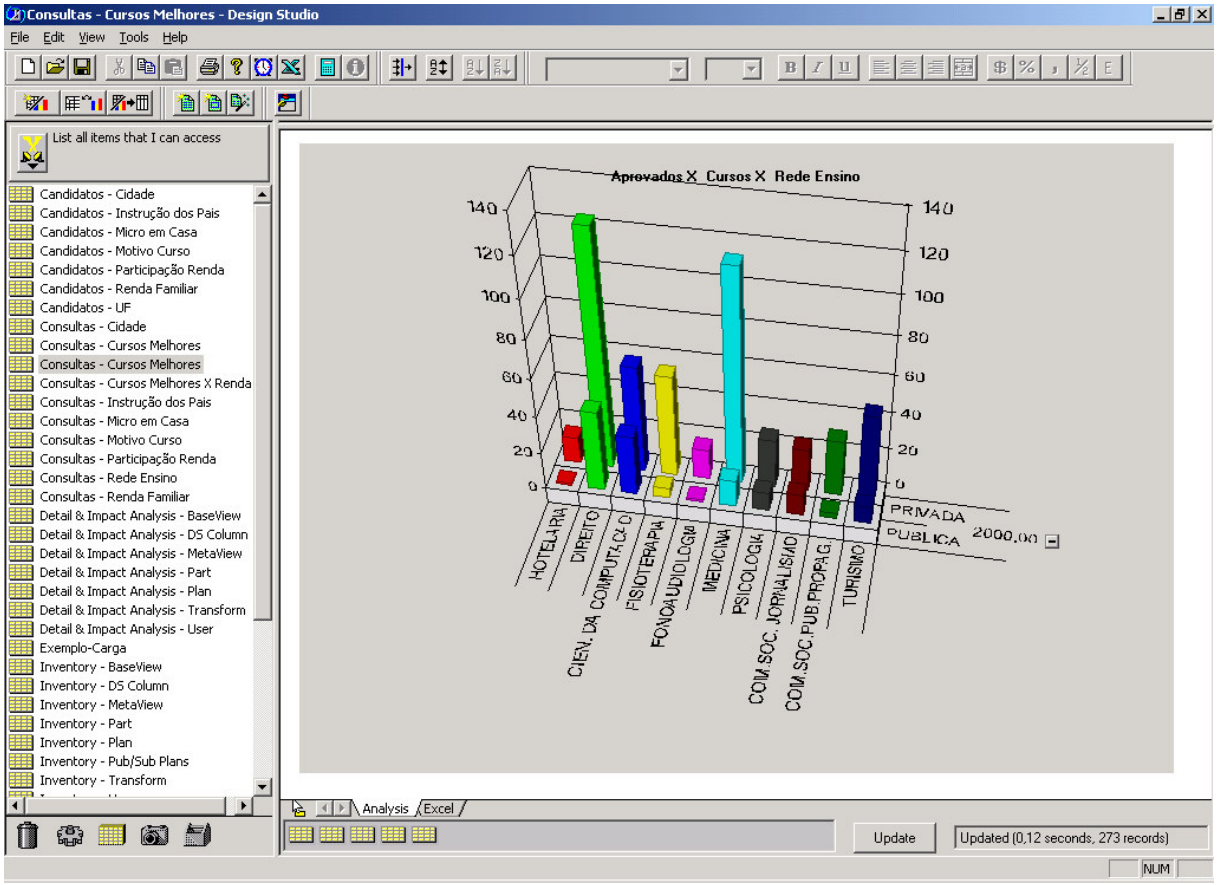

Figura 6.7 – Tela do SAGENT com gráfico de Aprovação X Cursos X Rede.

Após a disponibilização do ambiente de consulta, qualquer hipótese referente ao contexto do Data Mart poderia ser verificada. Apesar de não fazer parte do objetivo principal desse trabalho, foi gerada uma tabela de dados em formato flat (TXT) com os dados tratados, e submetida a um algoritmo de indução de regras. A partir daí foi possível verificar algumas das variáveis mais relevantes para a aprovação do candidato. Posteriormente algumas hipóteses puderam ser confirmadas diretamente nos dados do Data Mart. As tabelas abaixo mostram alguns dos principais resultados obtidos com esse protótipo.

Primeiramente, para analisar a estatística da influencia de um perfil na aprovação, é importante conhecer qual o resultado geral do nível de aprovação do vestibular. Esse resultado é apresentado na tabela 6.2.

Resultado do Vestibular	
Aprovação	13 %
Reprovação	87 %

Tabela 6.2 - Estatística geral de aprovação

A seguir serão apresentadas algumas regras sobre a aprovação dos candidatos e comentários sobre as mesmas.

Sexo	Sua Opção	Aprovação(%)	Reprovação(%)
F	Decidido	10,00	90,00
M	Decidido	14,59	85,41
F/M	Decidido	10,78	89,22

Tabela 6.3 - Estatística de aprovação segundo sexo e opção

O atributo "Sua Opção" é a resposta do questionário sócio cultural onde os candidatos revelavam como eles se sentiam em relação ao curso escolhido. Com base na regra gerada e nas duas outras regras retiradas do banco verificamos que as mulheres decididas em relação à sua escolha têm menor aprovação que os homens decididos(Tabela 6.3).

Ar-Condicionado	Computador	Acesso Internet	Aprovação(%)	Reprovação(%)
N	N	N	9,81	90,19
S	S	S	17,01	82,99

Tabela 6.4 - Estatística de aprovação segundo Ar-Condicionado, Computador e Internet

Com base em algumas regras geradas automaticamente pelo algoritmo indutor de regras, foram testadas as variáveis ar-condicionado, acesso a internet e computador juntas e constatou-se que o índice de aprovação aumenta consideravelmente em relação a estatística geral de alunos, quando eles possuem esses requisitos(Tabela6.4).

Computador	Aprovação(%)	Reprovação(%)
S	15,52	84,48
N	10,22	89,78

Tabela 6.5 - Estatística de aprovação segundo posse de computador

Na Tabela 6.5 a relevância do atributo computador isoladamente nos índices de aprovação. Fica claro que o atributo computador é de extrema relevância.

Sexo	Aprovação(%)	Reprovação(%)
M	14,93	85,07
F	10,10	89,90

Tabela 6.6 - Estatística de aprovação segundo sexo

Verificou-se na Tabela 6.6 que o índice de aprovação de homens neste concurso foi melhor do que o das mulheres. Essa análise só é relevante quando se sabe o percentual de participação geral dos candidatos quanto ao sexo.

Língua Estrangeira	Aprovação(%)	Reprovação(%)
Não	10,29	89,71
Domínio Médio	13,99	86,01
Domínio Bom	20,06	79,94

Tabela 6.7 - Estatística de aprovação segundo domínio de língua estrangeira

A análise da língua estrangeira isoladamente (Tabela 6.7) foi retirada diretamente do Data Mart, sem nenhuma indicação da ferramenta de indução de regras. A decisão de incluí-la foi acertada quando se supõe que a falta do domínio da língua estrangeira poderá ser um fator que agrava a aprovação. Constatamos que pessoas com um domínio bom da língua estrangeira possuem altos índices de aprovação.

Quantos Vestibulares	Aprovação(%)	Reprovação(%)
0	8,94	91,06
1	15,10	84,90
2	15,85	84,85
3	15,40	84,60
> 3	16,16	83,84

Tabela 6.8 - Estatística de aprovação segundo quantidade de vestibulares prestados

Na Tabela 6.8 fica evidente que a medida que cresce as tentativas de vestibular pelos candidatos, melhores são seus níveis de aprovação.

Tem Computador	Língua Estrangeira	Renda	Aprovação(%)	Reprovação(%)
S	Domínio Bom	Qualquer	22,10	77,90
S	Domínio Bom	Renda > 15 salários	23,29	76,71

Tabela 6.9 - Conhecimento descoberto pela equipe diretamente no Data Mart

Com base na observação de regras induzidas ou deduzidas e a análise de gráficos, resolveu-se analisar o Data Mart para identificar atributos que juntos fossem muito relevantes na aprovação de um candidato, o resultado esta na Tabela 6.9.

Além das regras relacionadas com a aprovação do candidato saiu-se um pouco do foco inicial da construção do Data Mart e foram identificadas outras regras que não estão diretamente ligadas com a aprovação no vestibular.

Sexo	Não Trabalham (%)	Aprovação(%)
F	76,00	10,10
M	64,89	14,93

Tabela 6.10 - Estatística de aprovação segundo sexo e participação da renda familiar

A partir da regra da Tabela 6.10 verificou-se que os homens que participaram do vestibular trabalhavam em maior proporção do que as mulheres.

TIPO_RESIDENCIA=”PRÓPRIA”,
TURNO_2_GRAU=”DIURNO”,
QUANTIDADE_VESTIBULAR=0
➔ PARTICIPACAO_RENDA=”NAO_TRABALHO”(91%)

Esta regra mostra que as pessoas com casa própria e que fizeram o segundo grau no período da manhã e que estão fazendo o seu primeiro vestibular não trabalham em 91% dos casos, enquanto que a média das pessoas que não trabalham no geral, fica em 72,3%.

RENDA_FAMILIAR=<5 ➔ TEM_AR =”NÃO”(90%)
RENDA_FAMILIAR=<5 ➔ TEM_AUTO=”NÃO”(77%)

Algumas suposições sobre a relação de renda com bens de consumo também podem ser testadas nesse modelo. Enquanto a porcentagem de alunos sem ar-condicionado em casa e sem automóvel no total é respectivamente 58% e 38,23%, os números sobem drasticamente entre os alunos com menos de 5 (cinco) salários na família chegando a 90% e 77% respectivamente.

Número de Familiares	Dos que estão no ensino Pública	Dos que estão no ensino Privado
2	4,45%	6,6%
3	13,87%	13,26%
4-6	74,25%	66,55%
>7	6,31%	12,20%

Tabela 6.11 – Relação entre número de familiares e participação em ensino público e privado

Na Tabela 6.11 é feita uma análise do tamanho da família dos alunos em relação ao tipo de rede de ensino ao qual eles freqüentaram (pública e privada).

7 – CONSIDERAÇÕES FINAIS

A qualidade dos dados das fontes de informação é em geral uma grande preocupação dos analistas e gestores já a algum tempo, mas pouca coisa é efetivamente realizada. Os especialistas começaram a levar a sério a qualidade e o tratamento dos dados, devido aos muitos fracassos e ao próprio amadurecimento dos Sistemas de Apoio à Decisão. Essa mudança está sendo refletida pela quantidade de produção científica gerada sobre esse assunto nos últimos 3 (três) anos. Esse trabalho é um reflexo dessa preocupação, aliada a uma materialização de resultados, que vão além da simples verificação de qualidade e de tratamento dos dados. Ele acrescenta novas fases após análise e tratamento de dados, expandindo o processo até a modelagem e implementação de um protótipo de Data Mart.

7.1 – RESULTADOS E CONCLUSÕES

Todo o estudo realizado visou o desenvolvimento e a implementação de metodologias e técnicas de construção de Sistemas de Apoio à Decisão, em especial no desenvolvimento de Data Marts baseadas em dados.

Esse trabalho tem uma abordagem única, uma vez que diferentemente das demais metodologias, ele tem como base a construção de modelos a partir do tratamento de dados. A metodologia FastCube produz uma grande contribuição principalmente para quem está começando na área de modelagem e construção de Sistemas de Apoio à Decisão, pela sua simplicidade e pela possibilidade de se gerar resultados mais rapidamente do que as metodologias tradicionais de construção de Data Warehouses, visto que essas normalmente são muito mais complexas e abrangentes e requerem maior tempo para absorção.

A simplicidade da metodologia FastCube se verifica no pequeno número de elementos básicos (*TabelaAmostra, Fragmento, Coluna*) necessários ao tratamento de dados, modelagem e construção de um Data Mart.

A implementação do ambiente que deu suporte a metodologia FastCube também foi desenvolvido utilizando conceitos de extensibilidade e simplicidade, não encontrada em ambientes de tratamento de dados conhecidos, seja ele acadêmico ou comercial. O repositório de técnicas para o tratamento de dados extensível foi desenvolvido de forma a se encaixar perfeitamente aos elementos básicos da metodologia FastCube.

Uma outra contribuição importante foi a construção do modelo simplificado de metadados, que contempla em um só modelo os dados de entrada, os metadados de transformações de dados e os metadados de modelagem dimensional (Figura5.7). Esse modelo conseguiu unir de forma harmônica elementos que normalmente são modelados separadamente.

Na implementação foi desenvolvido um modelo de classes extensível que pode ser estendida para carregar dados de entrada (carga DW) proveniente de qualquer tipo de base de dados (Relacional, OO, XML, TXT etc...). A persistência das classes dos metadados também contou com esse tipo de extensibilidade. Para implementação também foram desenvolvidas 2(duas) novas técnicas. Uma é uma melhoria de um algoritmo de similaridade, que se mostrou muito mais eficiente do que todos os algoritmos avaliados (anexo II). A outra técnica é um algoritmo criado para detectar correlação entre colunas e sugerir assim a junção de colunas em dimensões (anexoI).

A metodologia aqui aplicada está mais direcionada para a construção de Data Warehouse departamentais (Data Marts), Data Warehouse estatísticos [INM97] e principalmente para construção de protótipos em curtos intervalos de tempo (implementações rápidas). Os resultados obtidos dentro desse conceito foram mais do que satisfatórios, pois o intervalo de tempo para se modelar e construir um Data Mart utilizando a metodologia FastCube foi muito menor do que seguindo os modelos tradicionais de construção de DW. Isso foi verificado através do tempo gasto para treinamento de alunos de graduação em cursos de extensão, nas diversas metodologias (inclusive a FastCube) e o tempo gasto por esses alunos para fazer a implementação com elas.

Apesar de apresentado apenas um Estudo de Caso (COVEST), essa metodologia foi testada em diversos outras áreas, tento obtido bons resultados, segundo o relato de gestores habituados com métodos tradicionais de construção de DW. Até a data da submissão dessa dissertação, as técnicas aqui apresentadas já tinham sido aplicadas em: instituições financeiras para análise de concessão de cartão de crédito (com dados de 2 bancos), crédito para lojas de varejo (3 casos) e call center de uma empresa de telefonia (1 caso) obtendo em todos eles os resultados esperados.

Nesse trabalho, a qualidade, que é o primeiro princípio do REDIRIS, foi abordada de maneira objetiva, visando principalmente, a questão da qualidade dos dados em todas as

etapas do processo de construção do DW. Como já visto, a metodologia FastCube foge do processo clássico de construção de sistemas de Apoio à Decisão, pois ela é baseada em pré-processamento de dados. Com essa característica ela incorpora rapidamente em um modelo de metadados, todo o conhecimento contido nos dados, ao longo de toda a fase de tratamento e modelagem de dados. Ela faz uso da redução do volume de dados, com amostragem e uso Data Marts, em um processo iterativo e com alto grau de interatividade com usuário. Com isso, espera-se a obtenção de um modelo de dados para apoio à decisão em menores espaços de tempo e com maiores taxas acertos. Essas características citadas confirmam o compromisso desse trabalho com os princípios do REDIRIS do capítulo 2.

Para elaboração desse trabalho foi avaliado um grande número de trabalhos acadêmicos, ferramentas relacionadas à construção de DW e tratamento dos dados e mesmo entre esses (conforme referenciado ao longo dessa dissertação), a implementação do FastCube mostrou ser um software com alto grau de pragmatismo e aplicabilidade.

7.2 – TRABALHOS CORRELATOS

É abrangente o tema deste trabalho, pois trata de assuntos que por si só já geraram diversos trabalhos e publicações. Com isso, esse trabalho não pode ser comparado diretamente com nenhum outro trabalho. Como assuntos de destaque para a elaboração dessa dissertação tem-se: trabalhos na área de qualidade de dados, na construção de Data Warehouse e no pré-processamento de dados (limpeza e tratamento).

Um dos objetivos dessa dissertação é o tratamento e a limpeza dos dados, então apresentou-se um pouco mais sobre um software acadêmico que se propõe a realizar essas tarefas. O AJAX é um framework extensível que trabalha com apenas 5(cinco) operadores básicos para limpeza e transformação dos dados, onde o grupo que o desenvolveu estendeu o SQL para suportar essas operações [GFSS99] [GFSS00]. O *Maping Operator* é o operador responsável por gerar os subconjuntos de dados (projeção e seleção) criando chaves para fazer o link entre a fonte original e o subconjunto que será tratado. O *Mathing Operator* é um operador que aplica algoritmo de similaridade, com precisão ajustável tentando identificar ocorrências semelhantes que correspondem a uma só classe. O *Clustering Operator* é o operador capaz de aplicar algoritmos de agrupamento, que podem ser os mais variados possíveis. O *Merging Operator* é o operador que precisa ser usado para consolidar cada grupo (cluster) em um única tupla

após um agrupamento. O último operador é o *SQL View* responsável por permitir declarações de SQL unions e joins com algumas verificações de integridade.

Existem muitas diferenças entre o tratamento que o AJAX dá aos dados e esse trabalho de dissertação. O objetivo do AJAX é apenas tratar os dados, enquanto esse trabalho faz um tratamento voltado para a construção de um DW, não ficando apenas na limpeza de dados, chegando até a modelagem dimensional. O processo de verificação da qualidade no FastCube é feita de uma só vez através da importação dos dados para o modelo, então todos os metadados das colunas são gerados, independentemente se essa coluna será tratada ou não. Entretanto, o AJAX trabalha sobre demanda, processando e manipulando apenas os dados que forem sendo solicitados.

Existem outras ferramentas comerciais que englobam tarefas tratadas nesta dissertação, mas essas não estão organizadas da forma como foi abordada e neste trabalho. Normalmente essas ferramentas não possuem metodologias associadas a elas e tratam de um sub-conjunto de problemas como: qualidade, limpeza e tratamento de dados[DCI01], integração de dados [DCI01], construção de DW [CAM02] [SAG00], Mineração de Dados [WEK02] [NEU02]... Algumas ferramentas se propõem a tratar vários destes assuntos, mas para isso elas normalmente oferecem módulos que funcionam usualmente de forma independente ou complementar. O SAS [SAS99] [SAS02] é um bom exemplo para esse último tipo de ferramenta, pois ele surgiu inicialmente como uma ferramenta estatística e após melhorias constantes, conta hoje, com um pacote de software com diversos módulos como: qualidade e limpeza de dados, OLAP, Data Mining, Administração de DW e outros[SAS02] [INM01].

7.3 – TRABALHOS FUTUROS

Devido a grande abrangência do tema e do trabalho apresentado, não foi feito teste sobre a possibilidade de uso da metodologia para aplicação em grandes volumes de dados. Para ser usada nesse contexto, faz-se necessário um estudo mais aprofundado sobre a viabilidade e a possível adequação do modelo de dados e metadados. Também não foram realizados testes para implementações de vários Data Marts de um Data Warehouse em uma abordagem de construção incremental, mas apesar disso o modelo de metadados já foi construído para suportar tal funcionalidade.

Apesar da metodologia FastCube sugerir o uso de técnicas para auxilio de montagem semi-automática de modelos dimensionais, esse tema pode ser melhor explorado com pesquisas especialmente voltadas para identificar e criar técnicas especificas para esse propósito.

Mesmo tendo sido realizados alguns ensaios de uso de mineração de dados e de ser constatado que a metodologia FastCube e a sua implementação já atendem à requisitos de modelagem de dados para técnicas específicas de mineração de dados como redes neurais, algoritmos genéticos e outras, consideramos a formalização como sugestão relevante para continuidade desta dissertação. Mesmo com esses ensaios e a constatação que a modelagem e a implementação suportam facilmente a integração com técnicas de Mineração de Dados, é necessário um estudo maior para melhor adaptação e uso com esse fim. Na sessão 5.5 (Possibilidade Complementares do Modelo) já foi discutido o uso desse ambiente para Mineração de Dados, então a integração de ferramentas de mineração de dados ou mesmo implementações de algumas técnicas de mineração de dados seria algo muito enriquecedor para um ambiente de Apoio à Decisão como o REDIRIS. Existem outros pontos desse trabalho que podem ser ampliados como: uso de vários níveis de granularidade (usou-se apenas um nível de granularidade por Data Mart) e uso de um esquema de diretório para armazenamento dos dados da *TabelaAmostra.*

A implementação de um ambiente capaz de envolver todas as etapas do processo e a integração dos dados e metadados de diversas ferramentas para compor um ambiente único e integrado, afim de aproveitar o que há de melhor em diversas ferramentas e o desenvolvimento de outros módulos do processo de Apoio a Decisão seria uma continuação natural desta dissertação.

ANEXO I - ALGORITIMO DE SUGESTÃO DE DIMENSÕES

Esse anexo apresenta um algoritmo que pode ser usado na identificação de dimensões, baseado na hierarquia entre os atributos da dimensão. Ele se baseia na correlação entre as colunas, identifica se existe uma correlação e que coluna é hierarquicamente superior a outra. Esse algoritmo se baseia apenas nas relações de dependência entre as colunas não sendo para ele possível a identificação de atributos de dimensões que não sejam hierarquizadas.

Dado um conjunto de colunas como entrada do algoritmo, ele analisa todas as combinações de colunas 2 (duas) a 2 (duas) identificando suas correlações e hierarquias. A resposta do algoritmo é um vetor da estrutura abaixo.

```
Class Correlation
        Column: column;      // Coluna avaliada
        List: Bottom;        // Lista de colunas superiores hierarquicamente.
        List: Top;           // Lista de colunas inferiores hierarquicamente.
```

Com base no vetor de correlações superiores e inferiores das colunas, torna-se possível sugerir uma dimensão e a hierarquia de suas colunas, porque em Bottom tem as colunas que hierarquicamente superiores à column (coluna avaliada) e Top guarda todas as colunas hierarquicamente inferiores à column.

O núcleo do algoritmo é o cruzamento de todas as variáveis entre si, verificando para cada par se existe correlação e o sentido da hierarquia. A seguir é mostrada a sua estrutura simplificada em linguagem de programação.

```
function fDimensionTip(Columns: List): TCorrelationMatrix;
…
begin
  // Inicializa Matrix de correlação
    fInicializa(CorrelationMatrix);
...
  // Testa as correlações possíveis
 for i:=0 to Columns.Count-1 do
   for j:=i+1 to Columns.Count-1 do
     sReturn := fCorrelation(Columns[i],Columns[j], 80);
     if sReturn = 'Bottom' then
       CorrelationMatrix[i].Bottom.Add(Columns[j]);
       CorrelationMatrix[j].Top.Add(Columns[i]);
     else if sReturn = 'Top' then
       CorrelationMatrix[j].Bottom.Add(Columns[i]);
       CorrelationMatrix[i]. Top.Add(Columns[j]);
     else if sReturn = 'Nothing'
        Nothing
  result := CorrelationMatrix;
```

Percebe-se no algoritmo acima que todas as colunas de entrada são passadas através da lista Columns e submetidas duas a duas à função **fCorrelation()**. Essa função faz a correlação entre as duas colunas e retorna a string ‘Buttom’ ou ‘Top’ se existir hierarquia, ou retorna ‘Nothing’ se não existir nenhuma correlação. A função **fCorrelation()** gera uma terceira coluna que é a concatenação dessas duas colunas de entrada. Depois é feita a distribuição de frequência para todas as três colunas. Com a média da razão entre a frequência da coluna 3 e das duas demais frequências pode-se determinar se as colunas 1 e 2 são correlacionadas e qual é a direção dessa correlação. O algoritmo a seguir, mostra de forma simplificada uma possível implementação para essa função.

```
function fCorrelation(col1,col2: TstringList; nConst:Integer):string;
begin
…
// Gera col3
  for i:=0 to col1.Count-1 do
    col3.Add(col1[i] + '^' + col2[i]);

//  Faz a distribuição de frequencia das colunas
  fCol1 := fFrequency(col1);
  fCol2 := fFrequency(col2);
  fCol3 := fFrequency(col3);

  nFactorL := 0;
  nFactorR := 0;
// Verifica correlação col1 para col2
  for i:=0 to fCol3.Count do  // Loop para varrer fcol3
  …
    // Recupera substrings (fcol1 e fcol1) de fcol3
    value1 := subStr(fcol3[i].Value, 0, nLen1);
    value2 := subStr (fcol3[i].Value, nSeparator+1, nLen2);
    // Verifica a frequência em fcol1 e fcol2
    nFreq1 := fGetFrequency( fcol1, value1);
    nFreq2 := fGetFrequency( fcol2, value2);
    nFreq3 := fcol3[i].Frequency;
   // Faz o cálculo das razões
    nFactorL := nFactorL + ( nFreq3 / nFreq1 * 100);  // Fator na col1 esquerda
    nFactorR := nFactorR + ( nFreq3 / nFreq2 * 100);  // Fator na col2 direita
  end for;

    // Tira média
    nFactorL := nFactorL/i;
    nFactorR := nFactorR/i;

    // Compara com o parâmetro de entrada(nConst) estabelecido limite de correlação
    if ((nFactorL < nConst) and (nFactorR < nConst))     // Se não existe correlação
    or (col1.Count*nConst/100 < i  )                    // Tem muitos distintos
    then result := 'Nothing'
    else if (nFactorL >= nConst) and (nFactorL >= nFactorR) then result := 'Buttom'
    else result := 'Top';

end;
```

ANEXO II - ALGORITMO DE MATCHING MELHORADO

Existem vários algoritmos de matching, que normalmente comparam dois strings e retorna o nível de similaridade. Através de testes práticos foi selecionado o algoritmo similar100(abaixo) como sendo o mais eficaz dentro de um conjunto de testes realizados com diversos algoritmos. O similar100 apresentou uma taxa de acerto de 8% a 12% maior em média nas colunas que foram submetidas ao tratamento de similaridade, em relação aos demais algoritmos testados.

Para esse algoritmo tornar-se útil em nossa implementação e melhorar ainda mais a sua eficiência, foi necessária uma extensão de suas funcionalidades para adequá-lo às necessidades requeridas. A adequação e a flexibilidade desse algoritmo foi conseguida com a inserção de novos parâmetros de entrada. A inserção dessas novas funcionalidades no algoritmo fMatching tornou mais flexível o tratamento das colunas, com a possibilidade do analista de dados fazer tratamentos diferenciados dos dados, levando-se em conta as características de cada coluna para variar os parâmetros de entrada do algoritmo e obter melhores resultados. Em relação ao similar100 original, essa nova implementação mostrou um aumento de pelo menos 10% na taxa média de acerto, variando para mais ou para menos dependendo dos valores escolhidos pelo analista para os parâmetros de entrada.

Entrada:

- *String1* - Entrada 1
- *String2* - Entrada 2
- *VlLimite* - Valor da similaridade entre strings para consideradas iguais.
- *FlagFirst* –Define se o primeiro caracter deve igual nas duas strings.
- *FlagLast* - Define se o último caracter deve igual nas duas strings.
- *FlagClean* - Retira todos os caracteres que não sejam letras.

Saída:

- *Igual* - Retorna TRUE de é considerado igual e FALSE caso contrário.

Para melhor entendimento, a definição da função criada fMatching() de similaridade será aqui apresentada em linguagem estruturada e se utilizará da função similar100.

Na implementação em nosso ambiente esses algoritmos foram convertidos para JAVA e incorporado a classe *MatchingColumn*.

```
function fMatching(string1,string1:string; FlagFirst, FlagLast, FlagClean:boolean;
Vllimite:real):boolean;
begin

  // Testa se o primeiro caracter de cada string é igual
  if  ((FlagFirst) and ( CopyStr(string1,1,1) <> CopyStr(string2,1,1)) then
    Matching := FALSE;

  // Testa se o último caracter de cada string é igual
  else if ((FlagLast) and
          ( CopyStr(string1,length(string1),1) <> copyStr(string2,length(string2),1))) then
    Matching := FALSE;

  // Para deixar apenas letras em uma string criada a função fClean
  // Verifica se vai tirar o lixo antes de submeter a função Similar100
  else if FlagTrash then
    Matching := similar100(fClean(string1), fClean(string2)) >= vllimite

  // Aplica a função similar100 sem melhorias
  else if FlagTrash then
    Matching := similar100(string1, string2) >= vllimite;

end;
```

Aqui apresentamos a versão original de similar100 obtida na internet em linguagem pascal, pois ela se encontra mais didática.

```
function similar100(st1,st2:string):word;
{This one started the whole thing. Loosely based on an algorithm called
 SIMILAR.ASM  written by John W. Ratcliff and David E. Metzener
 only a lot more understandable. Returns percentage match. Pretty slow
 compared to the ASM versions. Case sensitive.
 Ron Nossaman Sept. 30 1994 }
var score:integer;

  procedure compare(s1,s2:string);
  var s1l,s1r,s2l,s2r,looker:integer;
  begin
    s1l:=1;s2l:=1;
    s1r:=length(s1);
    s2r:=length(s2);
    looker:=s2l;
  {increment s1, sweep s2}
    repeat
      if s1[s1l]=s2[looker] then
      begin          {got a match}
        inc(s1l);    {next position on everything}
        inc(looker);
        s2l:=looker;  {pull up starting position marker}
        inc(score);
      end else inc(looker); {no match, continue sweep}
      if looker&gt;s2r then    {looker swept past end of string}
      begin
        looker:=s2l;    {restore looker to last unmatched position}
        if s2l&gt;s2r then s1l:=s1r;
        inc(s1l);       {next char in first string for matching}
      end;
    until s1l&gt;s1r;
  end;
begin
  score:=0;
  compare(st1,st2);
  compare(st2,st1);
  score:=(score*100)div(length(st1)+length(st2));
  similar100:=score;
end;
```

REFERÊNCIAS BIBLIOGRÁFICAS

[ALM01] ALMEIDA, Márcio Oliveira – ***Uso de Interfaces Abundantes em Informação para Mineração Visual de Dados***, IV Workshop on Human Factors in Computer Systems (IHC´2001), Santa Catarina-BR, Outubro 2001.

[ARA91] ARANGO, G., Prieto, R. ***Domain Analysis Concepts and Research Directions. In "Domain Analysis and Software System Modeling",*** 1st ed., California, IEEE Computer Society Press Tutorial, pg. 09-25, 1991.

[AUR97] AURÉLIO, Marco et al - ***Descoberta de Conhecimento e Mineração de Dados***. Apostila. Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, 1999.

[BAT86] BATINI, C. and LENZERINI, M. - ***Comparative Analysis Of Methodologies For Database Schema Integration***, ACM Computing Surveys, New York, v.18, nº 4, pg.323-364, Dezembro 1986.

[BAU97] BAUER, A. e LEHNER, W. - ***The Cube-Query-Language (CQL) for Multidimensional Statistical and Scientific Database Systems***, International Conference On Database Systems For Advanced Applications, Melbourne, Australia, pg.263-272, Maio 1997.

[BCC98] BERGAMASCHI, S. Castano, DE CAPITINI di Vimercati, S., Montanari, S., Vincini, M. ***Intelligent Approach to Information Integration.*** In Formal Ontology in Information Systems, pgs253-267, IOS Press, 1998.

[BIGU96] BIGUS, J. P***., Data Mining with Neural Network – Solving Business Problems from Application Development to Decision Support***, McGraw-Hill, 1996.

[BOE92] BOEHM, B., SCHERLIS, W. ***Megaprogramming***. Proceedings of the DARPA Software Technology Conference, Arlington, VA, April, 1992.

[BRI02] BRIGGS, D., & ARNOTT, D**. *Decision support systems failure: An evolutionary perspective*** (Working Paper. No. 2002/01). Melbourne, Australia: Decision Support Systems Laboratory, Monash University, 2002.

[BUS96] BUSCHMANN, F. et al. ***Pattern-Oriented Software Architecture: A System of Patterns***. John Wiley & Sons, 1996.

[BWM99] BRAGA R., WERNER, C., MATTOSO, M. ***Odyssey: A Reuse Environment based on Domain Analysis.*** Proceedings of the IEEE Symposium on Application-Specific Systems and Software Engineering Technology (ASSET'99), IEEE CS Press, pg.50-57, Richardson, Texas, March, 1999.

[CAM97] CAMPOS, Maria Luiza e ROCHA, Arnaldo V. ***Data Warehouse***, XVII Congresso da Sociedade Brasileira de Computação, XVI Jornada de Atualização em Informática, Rio de Janeiro, 1997, pg 261.

[CAM00] CAMPOS, Maria Luiza. ***O Impacto da Qualidade de Dados no Sucesso do Projeto Data Warehouse***. Palestra – SUCESU-ES. Vitória-ES, 2001. Disponível na internet no endereço www.es.sucesu.org.br/eventos/agenda_passada.asp?cod_evento=30, último acesso 16/03/2002.

[CAM01] CAMPOS, Maria Luiza **–** ***Data Warehouse***. Mini-Curso – SBBD2000. João Pessoa, 2000.

[CAM02] CAMPOS, Maria Luiza, ARNALDO V. Rocha. ***Página do grupo DatAware. UFRJ***, 2002. Disponível na internet no endereço http://genesis.nce.ufrj.br/dataware/tutorial/tutorial.html, último acesso 22/07/2002.

http://genesis.nce.ufrj.br/dataware/

[CAR97] CARMO, M.B.; CUNHA, J.D.; ***Visualization of Large Volumes of Information Using Different Representations***. Proceedings IEEE Conference on Information Visualization -IV'97, 1997, 101-105.

[CAR98] CARMO, M.B.; CUNHA, J.D.; ***Filtragem e Escolha de Representação na Visualização de Informação.*** 8º Encontro Português de Computação Gráfica, Centro de Computação Gráfica – Coimbra, Fev 1998.

[CHA97] CHAUDHURI, S. e DAYAL, U. – ***An Overview of Data Warehousing and Olap Tecnology***, SIGMOD Record, New York, v.26, nº 1, pg.65-74, Março 1997.

[CHE99] CHEN, Zhengxin – ***Computional Intelligence for Decision Support***, CRC Press, USA, 1999.

[CHE01] CHEN, Zhengxin – ***Intellingent Data Warehousing: from Data Preparation to Data Mining***, CRC Press, USA, Dezembro 2001.

[CLE99] CLEMENTS, P., NORTHROP, L. M., et al. ***A Framework for Software Product Line Practice***. Version 2.0, Report from the Product Line System Program, Software Engineering Institute, Pittsburgh, PA, July, 1999.

[CZA00a] CZARNECKI, K., EISENECKER, U. ***Generative Programming: methods, tools and applications.*** Addison-Wesley, pg5-6, May, 2000.

[CZA00b] CZARNECKI, K., EISENECKER, U. ***Generative Programming: methods, tools and applications.*** Addison-Wesley, pg137-142, May, 2000.

[DCI01] GALHARDAS Helena, ***Data Cleaning an Integration***, site sobre ferramentas de limpeza e integração de dados – CEST 2001. Disponível na internet no endereço http://cosmos.inesc.pt/~hig/cleaning.html, último acesso 16/03/2002

[DOM99] MONTEIRO, Domingos Sávio M. P. – ***Discovery – Um Ambiente para Descoberta de Conhecimento e Mineração de Dados.*** Dissertação de Mestrado apresentada ao CIN-UFPE, Recife, dezembro 1999.

[DRU99] DRUCKER, Peter **–** ***Desafios Gerenciais para o Século XXI***, Pioneira, Rio de Janeiro, 1999.

[ENG99] ENGLISH, Larry P. – ***Inproving Data Warehouse and Business Information Quality: Methods for Reducing Costs and Increasing Profits***, John Wiley & Sons Inc., USA, 1999.

[FAY96] FAYYAD, Usama M. ***Advances in Knowledge Discovery and Data Mining***, AAAI Press – Los Angeles - USA, 1996.

[FAY99] FAYAD, M., JOHNSON, R. ***Domain-Specific Application Frameworks: frameworks experiences by industry***. John-Wiley & Sons, New York, NY, 1999.

[FIO00] RONALD Forino. ***Data Quality: The Meta Data Supply Chain.*** USA, DM Review**,** 2000.

[FIS94] FISCHER, G. ***Domain-Oriented Design Environments***. Automated Software Engineering – The International Journal of Automated Reasoning and Artificial

Intelligence in Software Engineering, Vol.1, No.2, pg177-203, June, 1994. [FUR86] FURNAS, G.; ***Generalized fisheye views***. Proceedings CHI'86, 1986, 16-23.

[GAL02] Site pessoal da pesquisadora Helena Galhardas PhD in Informatics from University of Versailles, disponível via internet no endereço http://caravel.inria.fr/~galharda/cleaning.html, último acesso 01/03/2002.

[GIO00] GIOVINAZZO, W. A. ***Object-oriented data warehouse design***. New Jersey: Prentice Hall, 2000.

[GOL99] GOLFARELLI, M.; MAIO, D.; RIZZI, S. ***Designing the Data Warehouse: Key Steps and Crucial Issues***. Journal of Computer Science and Information Management, v.2, n. 3, 1999.

[GOL98] GOLFARELLI, M.; MAIO, D.; RIZZI, S. ***The dimensional fact model: a conceptual model for data warehouses***. International Journal of Cooperative Information Systems, v.7, n. 2-3, p.215-247, 1998.

[GFSS99] H. Galhardas, D. Florescu, D. Shasha, and E. Simon. ***An extensible framework for data cleaning. Technical*** Report RR-3742, INRIA, 1999.

[GFSS00] H. GALHARDAS, D. Florescu, D. Shasha, and E. Simon. Ajax: ***An Extensible Data Cleaning Tool***. In Proceedings of ACM SIGMOD-2000, June 2000.

[HAN01] HAN, JIAWEI and KAMBER, MICHELINE. – ***Data Mining: Concepts and Techniques***, Morgan Kaufmann Publishers, USA, 2001.

[HJE01] HJELM, J. ***Creating the Semantic Web with RDF***. John Willey & Sons, New York, NY, 2001.

[INM93] INMON, W.H. ***Information System Arquiteture: Development in 90's***., John Wiley & Sons Inc., New York, 1993.

[INM97] INMON, W.H. ***Como Construir o Data Warehouse***, Campus, Rio de Janeiro, 1997.

[INM97a] INMON, W.H. & RICHARD D. HACKATHORN – ***Como Usar o Data Warehouse***, Infobook, Rio de Janeiro, 1997.

[INM98] INMON, W.H. ***Managing The Data Warehouse***, John Wiley & Sons Inc., New York, 1998.

[JAR97] JARKE M., Y. Vassiliou. ***Foundations of data warehouse quality: a review of the DWQ project.*** Proc. 2nd Intl. Conf. Information Quality (IQ-97),Cambridge, 1997.

[JAR00] JARKE M. et al. ***Foundations of data warehouse.*** Germany, Springer, 2000.

[JJQ98] M. JARKE, M.A. Jeusfeld, C. Quix, P. Vassiliadis: *Architecture and quality in data warehouses*. In Proc. 10th Intl. Conf. CAiSE*98, Pisa, Italy, June 8-12, 1998, pp. 93-113, Springer-Verlag, Berlin.

[JJQ99] M. Jarke, M.A. Jeusfeld, C. Quix, P. Vassiliadis. ***Architecture and quality in data warehouses: An extended repository approach.*** *Information Systems*, 24(3), pp. 229–253, 1999.

[KCH90] KANG, K., COHEN, S., HESS, J., NOWAK, W., Peterson, S. ***Feature-Oriented Domain Analysis (FODA) Feasibility Study***. Software Engineering Institute Technical Report, CMU/SEI-90-TR-21, Pittsburgh, PA, 1990.

[KEL97] KELLY, S. ***Data Warehousing in Action***. New York, John Wiley & Sons, 1997.

[KIM96] KIMBALL, Ralph – ***The Data Warehouse Toolkit***, John Wiley & Sons Inc., New York, 1996.

[KIM98] KIMBALL, Ralph – ***The Data Warehouse Lifecycle Toolkit***, John Wiley & Sons Inc., New York, 1998.

[KIM97] KIMBALL, Ralph – ***Dimensional Modeling Manifesto***, DBMS, Agosto 1997.

[KUR98] KURIKE, Raphael – Grupo de Sistemas Inteligentes – ***Mineração de Dado***, DIN - Departamento de Informática. UEM - Universidade Estadual de Maringá, 1998. Disponível na internet no endereço http://www.din.uem.br/ia/a_multid/mineracao/geral/

[MAC00a] DAVID Marco – ***Meta Data and Data Administration: Implementing Data Quality Through Meta Data Part 1,*** USA, DM Review, 2000

[MAC00b] DAVID Marco – ***Meta Data and Data Administration: Implementing Data Quality Through Meta Data Part 2,*** USA, DM Review, 2000

[MAR99] MARAKAS, George M. – ***Decision Support Systems in the 21st century***, Prentice-Hall Inc., USA, 1999.

[MCD89] MCDOWELL, R. C., CASSEL, K.A. ***The RLF Librarian: A Reusability Librarian based on Cooperating Knowledge-Based Systems***. Proceedings of the 4th Annual Rome Air Development Center Knowledge-Based Software Assistant Conference, Utica, NY, September, 1989.

[MCM91] MCMILLAN, J. ***Games, Strategies an Managers***. Oxford University Press, Oxford, England, 1991.

[MEN99] MENDONÇA, Manoel G.; SUNDERHAFT, N. L.. ***A State of the Art Report: MiningSoftware Engineering Data***. U. S. Department of Defense (DoD) Data & Analysis Center for Software, Rome, NY, USA, dec. 1999.

[MEY83] MEYER, Paul L. – ***Probabilidade Aplicações à Estatística***, LTC, Rio de Janeiro, 1983.

[NEI80] NEIGHBORS, J. M. ***Software Construction from Components***. PhD Thesis, TR-160, ICS Department, University of California at Irvine, USA, 1980.

[NEU02] NEUROTECH – ***Manual do NeuralScorer Deployment***, Neurotech/CESAR, Recife, Brazil, 2002.

[PYL99] PYLE, Dorian. – ***Data Preparation for Data Mining***, Morgan Kaufmann Publishers Inc., San Francisco, 1999.

[QUI98] M.A. Jeusfeld, C. Quix, M. Jarke: ***Design and Analysis of Quality Information for Data Warehouses***. In Proc. 17th International Conference on Conceptual Modeling (ER'98) , Singapore, Nov 16-19, 1998, Springer-Verlag, ISBN 3-540-65189-6, pp. 349-362 (pdf © Springer-Verlag).

[RTZ99] RAVAT, F., Teste, O., ZURFLUH, G. ***Towards Data Warehouse Design***. Proceedings of the International Conference on Information and Knowledge Management (CIKM), Missouri, USA, 1999.

[SAS99] SAS Institute Inc. ***Finding the solution to data mining: A map of the features and components of SAS Enterprise Miner software***, SAS Institute Inc., Cary, North Carolina.

[SAS02] ***SAS Institute Inc.*** - 2002, disponível na Internet via protocolo http, no endereço http://www.sas.com/, data do último acesso 21/05/2002.

[SAG00] SAGENT – ***Manuais do SAGENT***, SAGENT, New York, 2000.

[SIG97] SILVERSTON, L., INMON, W.H., GRAZIANO, K. ***The Data Model Resource Book***. John Wiley & Sons, New York, NY, 1997.

[SIL02] SILVA Paulo Cézar Ribeiro da - FACEV – Faculdade de Economia Vitória – AULA NET: "***Curso de Estatística on Line***", disponível na internet no endereço http://www.milenio.com.br/aeee/Disciplinas/Estatistica/estat%C3%ADstica.htm, último acesso 15/03/2002.

[SIN01] SINGH, Harry S. Sing. ***Data Warehouse – Conceitos, Tecnologias, implementação e Gerenciamento***, Makron Books, São Paulo, 2001.

[SVV99] STAUDT, M., VADUVA, A., VETTERLI, T. ***The Role of Metadata for Data Warehousing***. Institut für Informatik der Universität Zürich Technical Report, ifi-99-06, 1999.

[VAS00] VASSILIADIS, P. ***Gulliver in the Land of Data Warehouse: Practical Experiences and Observations of a Researcher***. Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW2000), Stockholm, Sweden, June, 2000.

[VDO01] VDOVJAK, R, HOUVEN, G-J. ***RDF Based Architecture for Semantic Integration of Heterogeneous Information Sources***. Proceedings of the International Workshop on Information Integration on the Web (WIIW'2001), pg51-57, Rio de Janeiro, Brazil, April, 2001.

[VJO01]Vijayshankar Raman and Joseph M. Hellerstein. ***Potters Wheel: An Interactive Framework for Data Cleaning and Transformation*.** VLDB, Roma, Italy, 2001.

[WAN96] WANG, Y.R., AND STRONG, D.M. ***Beyond accuracy: what data quality means to data consumers***, Journal on Management of Information Systems, vol. 12, no. 4, pp. 5-34, 1996.

[WIT99]WITTEN, Ivan H. and FRANK, Eibe. – ***Data Mining Practical Machine Learning and Techniques with JAVA Implementations***, Morgan Kaufmann Publishers Inc., San Francisco, 1999.

[WEK02] ***WEKA Project.*** 2002, disponível na Internet via protocolo http, no endereço http://www.cs.waikato.ac.nz/~ml/index.html, data do último acesso 30/06/2002.