

Universidade Federal de Santa Catarina
Programa de Pós-Graduação em
Engenharia de Produção

**UMA ARQUITETURA PARA DISTRIBUIÇÃO
DE COMPONENTES TECNOLÓGICOS
DE SISTEMAS DE INFORMAÇÕES
BASEADOS EM DATA WAREHOUSE**

Denilson Sell

Dissertação apresentada ao
Programa de Pós-Graduação em
Engenharia de Produção da
Universidade Federal de Santa Catarina
como requisito parcial para obtenção
do título de Mestre em
Engenharia de Produção

Florianópolis

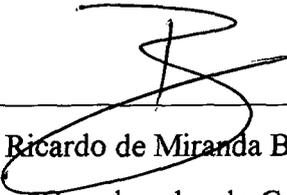
2001

Denilson Sell

**UMA ARQUITETURA PARA DISTRIBUIÇÃO
DE COMPONENTES TECNOLÓGICOS
DE SISTEMAS DE INFORMAÇÕES
BASEADOS EM DATA WAREHOUSE**

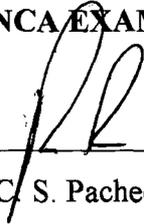
Esta dissertação foi julgada e aprovada para a
obtenção do título de **Mestre em Engenharia de
Produção no Programa de Pós-Graduação em
Engenharia de Produção** da
Universidade Federal de Santa Catarina.

Florianópolis, 21 de março de 2001.



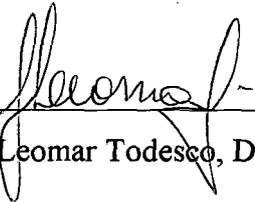
Prof. Ricardo de Miranda Barcia, Ph. D
Coordenador do Curso

BANCA EXAMINADORA

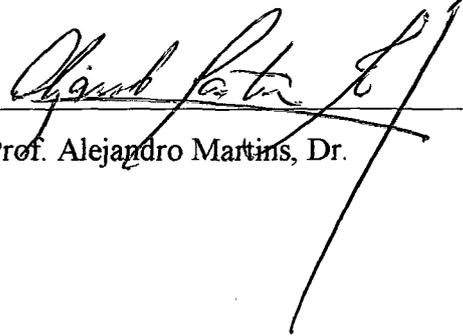


Prof. Roberto C. S. Pacheco, Dr.

Orientador



Prof. José Leomar Todesco, Dr.



Prof. Alejandro Martins, Dr.

AGRADECIMENTOS

Desejo manifestar minha enorme gratidão às pessoas que contribuíram direta e indiretamente para a realização deste trabalho.

Primeiramente a Deus, que sempre me deu forças e iluminou o meu caminho. Por ter me acompanhado, por ter me dado a vida, saúde, bons amigos, uma ótima família e uma esposa maravilhosa.

Ao meu orientador, professor Roberto C. S. Pacheco e ao professor Alejandro Martins, pessoas formidáveis que contribuíram muito para o desenvolvimento do trabalho e para a minha formação como pessoa e pesquisador.

Aos grandes amigos, dos grupos Consulte e Stela, e das empresas CELESC e Construtel, que acompanharam e contribuíram para o meu trabalho, em especial ao Carlos C. X. Santos.

À minha família, que mesmo acompanhando a minha luta de longe, foi fundamental para o meu triunfo.

E, em especial, a quem dedico este trabalho, minha esposa Graciele K. Sell, pela compreensão, carinho, apoio e incentivo.

SUMÁRIO

LISTA DE FIGURAS	vii
LISTA DE QUADROS	viii
RESUMO	ix
ABSTRACT	x
1 INTRODUÇÃO	1
1.1 APRESENTAÇÃO	1
1.2 JUSTIFICATIVA	2
1.3 OBJETIVO GERAL	3
1.4 OBJETIVOS ESPECÍFICOS	3
1.5 METODOLOGIA	3
1.6 ESTRUTURA DO TRABALHO	5
2 SISTEMAS DE INFORMAÇÕES	6
2.1 INTRODUÇÃO	6
2.2 DEFINIÇÃO	6
2.3 IMPORTÂNCIA	9
2.4 CLASSIFICAÇÃO	10
2.5 TECNOLOGIA DE INFORMAÇÃO	12
2.6 CONCLUSÃO	14
3 DATA WAREHOUSING	16
3.1 INTRODUÇÃO	16
3.2 DEFINIÇÃO	16
3.3 FASES TÍPICAS NO DATA WAREHOUSING	19
3.3.1 Planejamento	19
a Definição do Projeto	20
b Justificativa	21
c Planejamento da Equipe e Atividades	23
d Identificação Detalhada dos Requisitos	23
3.3.2 Modelagem Lógica e Física	24
a O Modelo Dimensional	26
b Granularidade e Particionamento	28
c Agregados	30
3.3.3 Preparação e Desenvolvimento de Ferramentas de Extração	31
a Coleta	32
b Transformação	33
c Incorporação	34
3.3.4 Desenvolvimento de Ferramentas de Aplicações Cliente	35
3.3.5 Metadados	36
3.4 METODOLOGIAS DE DESENVOLVIMENTO	37
3.5 CONCLUSÃO	42
4 A ARQUITETURA PROPOSTA	44

4.1	INTRODUÇÃO	44
4.2	ARQUITETURAS	44
4.2.1	Modelos de Arquiteturas	47
4.3	A ARQUITETURA PROPOSTA	50
4.3.1	Extração	52
4.3.2	Área de Armazenamento	54
4.3.3	Metadados	58
4.3.4	Área de Apresentação	60
4.4	CONCLUSÃO	62
5	APLICAÇÃO: SISTEMA DE APOIO À DECISÃO PARA ONCOLOGIA	65
5.1	INTRODUÇÃO	65
5.2	ABORDAGEM DO PROBLEMA	65
5.3	A ARQUITETURA DO SISTEMA	67
5.4	A EXTRAÇÃO DOS DADOS PARA O AMBIENTE	68
5.5	O ARMAZENAMENTO	71
5.6	A ÁREA DE APRESENTAÇÃO	73
5.7	CONCLUSÃO	76
6	CONCLUSÃO E TRABALHOS FUTUROS	79
7	REFERÊNCIAS BIBLIOGRÁFICAS	82
ANEXO A		86
ANEXO B		87
ANEXO C		90

LISTA DE FIGURAS

<i>Figura 1 - Processamento em um Sistema de Informação</i>	8
<i>Figura 2 - Níveis do Conhecimento</i>	10
<i>Figura 3 - Processamento Básico em um Data Warehouse</i>	18
<i>Figura 4 - Fases Típicas no Desenvolvimento de um Data Warehouse</i>	19
<i>Figura 5 - Etapas do Planejamento</i>	20
<i>Figura 6 - Análise de Quadrante para Definição de Prioridades</i>	21
<i>Figura 7 - Semelhança do Modelo Dimensional a um Cubo</i>	27
<i>Figura 8 - O Modelo Dimensional</i>	28
<i>Figura 9 - Processo de Extração</i>	31
<i>Figura 10 - Fases do Desenvolvimento Tradicional</i>	37
<i>Figura 11 - Fases do Desenvolvimento do Data Warehouse Global</i>	38
<i>Figura 12 - Fases do Desenvolvimento dos Data Marts Legados</i>	40
<i>Figura 13 - Fases do Desenvolvimento dos Data Marts Incrementais</i>	41
<i>Figura 14 - Componentes da Arquitetura</i>	45
<i>Figura 15 - A Arquitetura Top-Down</i>	48
<i>Figura 16 - A Arquitetura Bottom-Up</i>	48
<i>Figura 17 - A Arquitetura BUS</i>	49
<i>Figura 18 - Componentes da Arquitetura Proposta</i>	51
<i>Figura 19 - Os Módulos de Coleta</i>	53
<i>Figura 20 - Os Módulos de Incorporação</i>	54
<i>Figura 21 - Matriz de Data Marts e Dimensões</i>	55
<i>Figura 22 - Área de Armazenamento</i>	57
<i>Figura 23 - Área de Apresentação</i>	60
<i>Figura 24 - A Arquitetura Geral do Sistema</i>	67
<i>Figura 25 - Iteração dos Componentes de Extração do Sistema</i>	69
<i>Figura 26 - Área de Apresentação do Sistema</i>	74
<i>Figura 27 - Tela Principal do Sistema</i>	87
<i>Figura 28 - Seleção de Filtros</i>	88
<i>Figura 29 - Visualização Através de Tabelas</i>	88
<i>Figura 30 - Visualização Através de Gráficos</i>	89
<i>Figura 31 - Visualização Através de Mapas</i>	89
<i>Figura 32 - Modelo de Dados do Sistema Câncer On-Line</i>	90

LISTA DE QUADROS

<i>Quadro 1 – Características dos Sistemas de Informações</i>	12
<i>Quadro 2 - Histórico da Utilização da Tecnologia de Informação</i>	14
<i>Quadro 3 - Diferenças entre Dados de Sistemas Operacionais e Data Warehouse</i>	26
<i>Quadro 4 - Vantagens e Desvantagens da Metodologia Tradicional</i>	38
<i>Quadro 5 – Vantagens e Desvantagens do Data Warehouse Global</i>	39
<i>Quadro 6 – Vantagens e Desvantagens dos Data Marts Legados</i>	40
<i>Quadro 7 – Vantagens e Desvantagens dos Data Marts Incrementais</i>	41
<i>Quadro 8 – Regras para Extração Contidas no Metadados</i>	58
<i>Quadro 9 – Metadados da Área de Armazenamento</i>	59
<i>Quadro 10 – Metadados da Área de Apresentação</i>	59
<i>Quadro 11 – Escopo das Informações Contidas no Data Warehouse</i>	72
<i>Quadro 12 – Papéis em um Projeto de Data Warehouse</i>	86

RESUMO

A demanda por informações estratégicas para apoio à decisão tem sido um dos grandes motivos para investimentos em soluções informatizadas. Apesar dos intensos investimentos, ao final da maioria dos projetos as necessidades do tomador de decisão não são satisfeitas, principalmente devido às constantes mudanças dos cenários no processo decisório.

Um dos principais fatores para o fracasso de muitos sistemas é o enfoque orientado a “projeto”. Em um ciclo de vida de *software* orientado a “projeto”, encara-se o desenvolvimento como uma seqüência rígida de tarefas, com pouco enfoque na criação de uma arquitetura flexível a mudanças.

Devido à flexibilidade e ao foco num processo evolutivo contínuo, a utilização de uma metodologia de desenvolvimento baseada em *data warehousing* orientada a “processo”, viabiliza o desenvolvimento de sistemas de informações permitindo o acompanhamento das mudanças de necessidades dos tomadores de decisões.

Neste trabalho, propõe-se uma arquitetura que torna possível o uso de uma metodologia de *data warehouse* baseado em um “processo” contínuo de desenvolvimento e adequação às necessidades do usuário. A arquitetura distribui os componentes físicos do *data warehouse* em diversos módulos, o que permite grande flexibilidade e rapidez no desenvolvimento, possibilitando um fácil acompanhamento das necessidades dos usuários, além de outros benefícios.

Baseado na arquitetura proposta, o trabalho resgata conceitos afetos a sistemas de informações e a *data warehousing*, além da apresentação dos resultados obtidos no desenvolvimento de um sistema de apoio à decisão para oncologistas brasileiros.

ABSTRACT

The demand for strategical information supporting decision has been one of the main reasons to invest in computer-based solutions. In spite of heavy investments, the decision-maker needs are not fulfilled at the end of most of the projects, especially due to constant changes of the decision-making process scenarios.

One of the main factors that contribute to the failure of many systems is the oriented focus on a 'project'. In the lifespan of software oriented towards a 'project', development is seen as an unbroken sequencing of tasks with little focus on the creation of an architecture flexible about changes.

Due to flexibility and focus on a continuous evolutionary process, the use of a development methodology based on data warehousing oriented towards 'project' seems to make the development of information systems, allowing the accompanying of the decision-makers' changes of needs.

This work proposes an architecture that makes it possible to use a data warehousing methodology based on a continuous process of development and suitability for the user's needs. The architecture distributes the components of the data warehouse in different modules which in turn allows a great deal of flexibility and quickness in the development, making it easy to wait on the users' needs, besides other benefits.

Based on the proposed architecture, this work recovers concepts entrusted with information systems and data warehousing, besides it presents the results obtained from the development of a decision-making system for the Brazilian Association of Oncology.

1

INTRODUÇÃO

1.1 APRESENTAÇÃO

Em tempos de globalização e grande concorrência, os sistemas de informações apresentam-se como um dos fatores essenciais para a sobrevivência do negócio. Segundo Mcgee, Prusak (apud Mecheln, 1997), o mundo industrializado vem enfrentando a transição de uma economia industrial para uma economia de informação. Neste tipo de economia, o sucesso é determinado pelo que se sabe e não pelo que se possui. Quaisquer vantagens competitivas que pudessem ter sido adquiridas pelas economias de escala poderiam ser contrabalançadas através do desenvolvimento e uso efetivo da informação.

Atualmente, a tecnologia da maioria dos novos projetos de sistemas de informações foi ou está sendo desenvolvida através de uma técnica denominada *data warehousing*, que tem como propósito a concepção de sistemas pela estruturação de um *data warehouse*¹. Segundo o instituto IDC, em 2000 foram gastos mais de 5 bilhões de dólares apenas em soluções de *data warehousing* baseadas em pacote (Bort, 2001).

Mesmo utilizando-se da técnica, muitos dos sistemas fracassam. Segundo o instituto Standish, o percentual de projetos cancelados chega a 31%, enquanto que o percentual de projetos entregues com atraso e orçamento estourado chega a 53% (Data Knowledge, 2001). Já segundo Zimmer (2001), o percentual de fracasso total ou de projetos entregues com grande atraso chega a 50%. Grande parte destes percentuais se deve ao enfoque orientado a “projeto” e não a “processo” dado a implementação. Em uma metodologia baseada em um ciclo de vida orientado a “projeto”, encara-se o

¹ Inmon (1997) define *data warehouse* como “um conjunto de dados baseado em assuntos, integrado, não volátil e variável em relação ao tempo, de apoio às decisões gerenciais”.

desenvolvimento como uma tarefa orientada a um ciclo de interações rígidas, onde pouca ênfase sobre flexibilidade é dada na concepção da arquitetura do sistema.

O grande problema é que mudanças nas necessidades dos usuários são constantes o que incorre em abandono por parte do usuário ou inviabilização devido à falta de perspectiva de acompanhamento das mudanças de requisitos nas metodologias adotadas e dificuldade em realizar mudanças nos componentes do *data warehouse*, dada a estruturação rígida imposta pela arquitetura adotada.

O objetivo deste trabalho é discutir as principais metodologias utilizadas e apresentar uma arquitetura a ser empregada em conjunto com uma metodologia baseada em *data warehouse* orientado a “processo”, onde o projeto é construído de forma incremental e modular através da estruturação de *data marts*² integrados. Deste modo, os primeiros resultados são apresentados rapidamente ao usuário, sem perder a visão integrada do negócio, além de permitir uma rápida absorção de novos requisitos, dada a flexibilidade proporcionada pela arquitetura.

1.2 JUSTIFICATIVA

A utilização das metodologias e arquiteturas usuais mostrou através de inúmeros projetos fracassados, ser inadequada para a implementação de grandes sistemas de informações.

Através da aplicação da arquitetura proposta associada a uma metodologia de desenvolvimento incremental nos projetos tecnológicos de sistemas de informações, pode-se apresentar resultados de forma mais rápida e acompanhar as mudanças de requisitos dos usuários finais. Isto é possível devido à independência dos módulos componentes da arquitetura, justificando assim, os investimentos em projetos de tecnologia da informação.

² Data marts são pequenos conjuntos do data warehouse baseados em assuntos específicos (Inmon, 1997) e (Kimball, 1998b).

1.3 OBJETIVO GERAL

Desenvolver uma arquitetura modular para organização do *data warehouse*, de forma a torná-lo flexível a mudanças nos requisitos de projeto ou no ambiente operacional.

1.4 OBJETIVOS ESPECÍFICOS

- apresentar os elementos teóricos relacionados ao projeto de sistemas de informações e à técnica *data warehouse*;
- apresentar e comparar as metodologias utilizadas para desenvolvimento de sistemas de informações;
- discutir o modelo de arquitetura proposto, apresentando de forma detalhada os módulos e sua adequação à metodologia baseada em *data marts* incrementais proposta por Kimball (1998b); e
- descrever a aplicação da arquitetura em um sistema de informações para a oncologia.

1.5 METODOLOGIA

A metodologia adotada no trabalho fundamenta-se em cinco etapas, sendo:

1. estudo sobre a teoria de sistemas de informações;
2. estudo sobre a teoria acerca de *data warehousing* e das principais metodologias empregadas nesta técnica;

3. definição da arquitetura para distribuição dos componentes do *data warehouse*;
4. desenvolvimento de um sistema de informações utilizando a arquitetura proposta; e
5. refinamento da arquitetura com base nos resultados obtidos na implementação do sistema de informações.

Na primeira etapa são identificados os principais conceitos afetos a sistemas de informações e à tecnologia de informação, enfocando o processo decisório e o papel da tecnologia sobre estes sistemas.

Na segunda etapa são levantados os aspectos teóricos acerca de *data warehousing*, estudo dos componentes de um sistema de informações com tecnologia baseada em *data warehouse*, fases do processo de extração, organização da base de dados e publicação da informação. Nesta etapa ainda são levantadas as principais metodologias para *data warehousing*, enfocando fases, vantagens e desvantagens de cada uma.

Na terceira etapa é discutida a arquitetura proposta e são apontadas as melhorias que podem ser obtidas com o seu emprego sobre a distribuição dos componentes tecnológicos de um sistema de informação desenvolvido através de *data warehousing*. A arquitetura é definida baseando-se sobre os preceitos da metodologia de *data warehousing* baseada em *data marts* incrementais, por se tratar de uma das metodologias mais completas de acordo com o estudo das metodologias.

A quarta etapa procura validar o emprego da arquitetura no desenvolvimento de um sistema de informações baseado em *data warehousing* para a Sociedade Brasileira de Oncologia Clínica (SBOC). Nesta etapa é discutida a distribuição dos componentes do sistema seguindo a arquitetura e são apresentados os resultados alcançados.

Na última etapa o modelo da arquitetura é revisado tendo em vista os resultados obtidos na implementação do sistema de informações. Nesta etapa, a arquitetura recebe alguns ajustes e incorporações, de modo a se tornar mais flexível.

1.6 ESTRUTURA DO TRABALHO

O trabalho trata de um estudo dirigido à aplicação de uma arquitetura diferenciada para o desenvolvimento tecnológico de sistemas de informações. Desta forma, os capítulos foram divididos de acordo com os temas pertinentes à realização deste trabalho.

No segundo capítulo é apresentada uma introdução sobre sistemas de informações, modalidades de sistemas de informações e conceitos básicos necessários para o entendimento do processo decisório e dos elementos envolvidos nestes sistemas, com um enfoque maior sobre a tecnologia.

No terceiro capítulo é apresentada a fundamentação teórica de projetos de sistemas de informações baseados em *data warehousing*. O objetivo é descrever sucintamente os principais conceitos necessários para a implementação da técnica.

No quarto capítulo é detalhada a arquitetura proposta, seus módulos e relação com os conceitos abordados no segundo e terceiro capítulos. Serão apresentadas considerações em relação ao seu emprego, tendo em vista as metodologias de desenvolvimento utilizadas.

No quinto capítulo, é apresentado o estudo de caso realizado a partir da arquitetura proposta em um sistema de apoio à decisão para os oncologistas brasileiros.

No sexto capítulo serão apresentados a conclusão e trabalhos futuros.

2 SISTEMAS DE INFORMAÇÕES

2.1 INTRODUÇÃO

A arquitetura proposta neste trabalho é uma alternativa para o desenvolvimento dos componentes tecnológicos de sistemas de informações. Assim sendo, neste capítulo será apresentada uma introdução sobre sistemas de informações, modalidades de sistemas de informações e conceitos básicos necessários para o entendimento do processo decisório. Em seguida serão abordados os conceitos acerca de tecnologia da informação, sua evolução e importância para as organizações.

O objetivo é introduzir os principais elementos envolvidos nestes sistemas, com destaque à tecnologia, dada a relação do tema deste trabalho com a organização de parte dos elementos tecnológicos envolvidos em um sistema de informações. A discussão sobre a implementação de sistemas de informações segue no Capítulo 3, onde será discutida a aplicação da técnica *data warehousing*.

2.2 DEFINIÇÃO

Segundo Laudon & Laudon (1998, p.21), “um sistema de informações pode ser tecnicamente definido como um conjunto de componentes inter-relacionados que coleta (ou recupera), processa, armazena, e distribui informação para a tomada de decisão em uma organização”. Gil (apud Mecheln, 1997, p.14), sob um enfoque mais humanístico, afirma que:

"um sistema de informações consiste em pelo menos uma pessoa, com certas características psicológicas, que enfrenta um problema dentro de um contexto organizacional para o qual necessita de dados; com a finalidade de obter uma solução, esses dados são tratados criando-se informações geradas, distribuídas e entregues segundo um modo de apresentação".

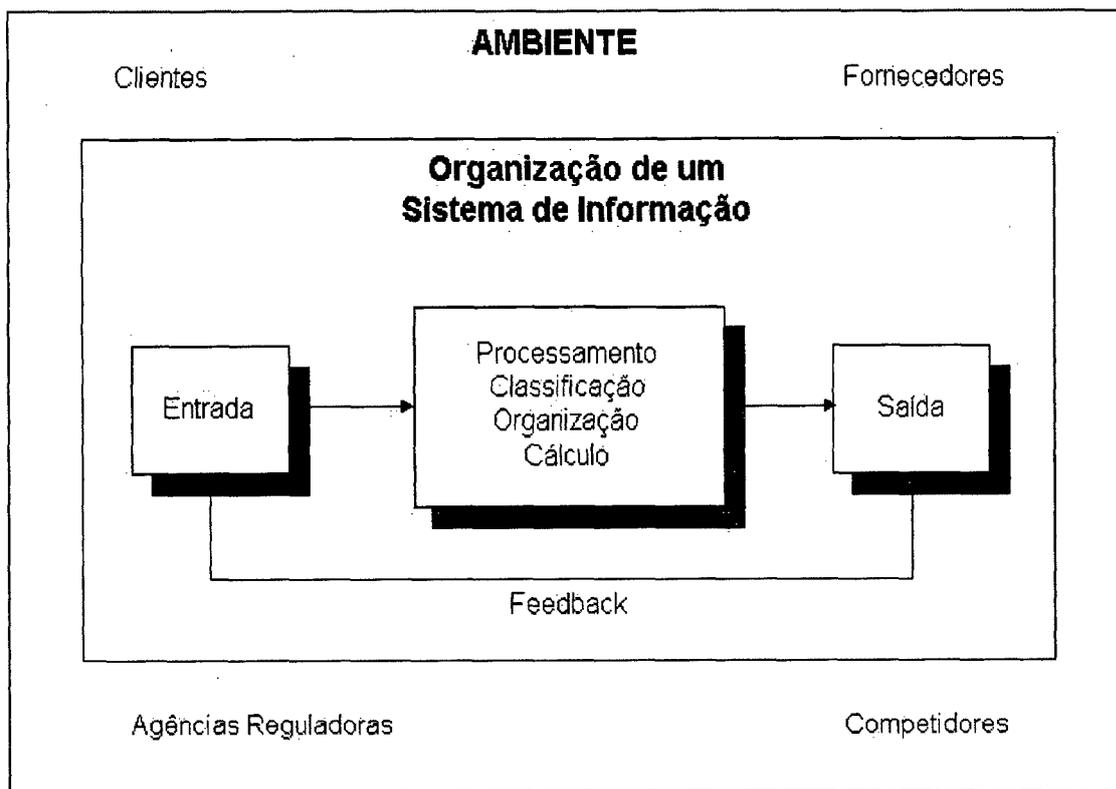
Segundo Verstraet (2000, p.1), "os componentes que constituem um sistema de informações são: pessoas, procedimentos e dados". Pessoas seguem procedimentos para manipular dados e criar informações. Dados são um conjunto de observações. Eles são uma coleção de medidas sobre alguns aspectos dos negócios. Os dados são processados para produzirem informações.

A partir das definições apresentadas, pode-se abstrair que o processamento básico realizado por um sistema de informação é baseado em três etapas: a coleta de dados, o processamento e a disponibilização da informação, em um dado ambiente (Feliciano Neto, Shimizu, 1996). A informação obtida no processamento será utilizada então para a geração de conhecimento. Este processo é ilustrado na Figura 1.

French (apud Mecheln, 1997) apresenta os estágios pelos quais as informações são processadas e classificadas. São eles:

- *dados*: fatos relacionados e seus valores;
- *informação*: relacionamento de entidades para formular um contexto;
- *conhecimento*: relacionamento de informação ligada às regras de negócio para formular a perícia;
- *julgamento*: sobre decisões ligadas ao contexto global para evidenciar regras proveitosas.

Figura 1 - Processamento em um Sistema de Informação



Fonte: Laudon & Laudon (1998)

Os dados são transformados em informações através de um processo de *análise*, tais informações transformam-se em *conhecimento* e esse *conhecimento*, através de um processo de *síntese*, leva ao *julgamento* (Feliciano Neto, Shimizu, 1996).

A tecnologia da informação auxilia nos três primeiros estágios, ou seja, dados, informação e conhecimento, sendo que o julgamento acerca das decisões cabe principalmente ao analista das informações. Através da tecnologia, essas decisões passarão a fazer parte da base de casos ou conhecimento da empresa para futuras decisões (Cruz, 1996) e (Mecheln, 1997).

2.3 IMPORTÂNCIA

Com a evolução da automação dos processos, com as constantes mudanças culturais e outros fatores internos e externos à empresa, o termo “administração eficaz” nunca foi tão almejado.

Druker (apud Mecheln, 1997) cita que a fonte da riqueza é algo especificamente humano: conhecimento. Se a pessoa aplicar o conhecimento nas atividades que já sabe executar, chama-se de produtividade. Se for aplicado a atividades novas e diferentes, chama-se de inovação. Somente o conhecimento permite atingir essas duas metas.

O objetivo da informação é melhorar a qualidade das decisões e seus resultados na solução de problemas. Sistemas de informação utilizados desta maneira, mudam a natureza da organização e da indústria onde operam.

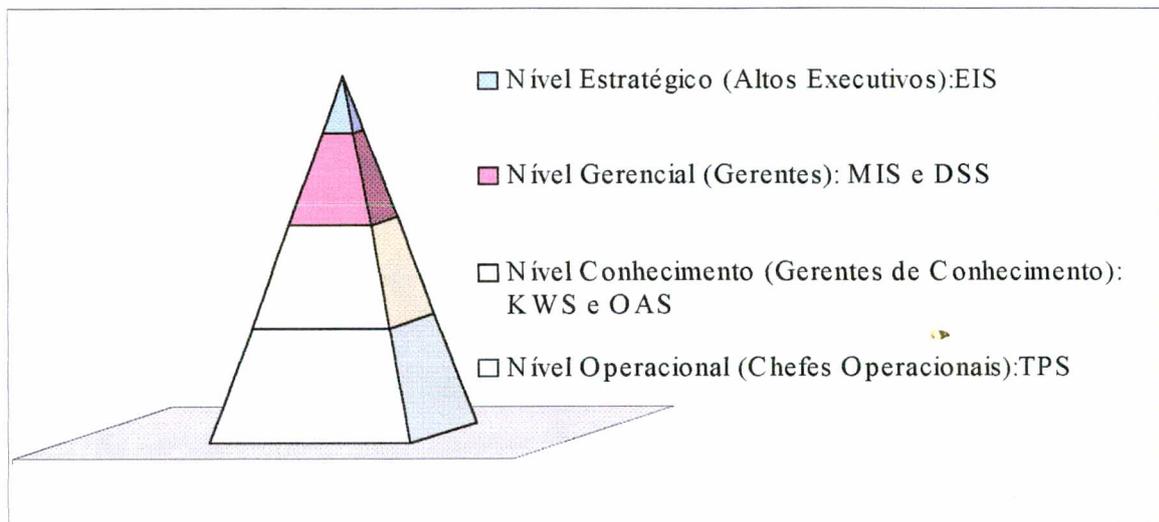
Nada é estático no ambiente de negócios. Como uma organização melhora ou inova com sistemas de informação, outras organizações são forçadas a responder. Se elas não respondem, o mercado força a correção para esta inação, e elas perderão parte do mercado e sofrerão outras adversidades. Então, os competidores devem responder às mudanças causadas pelos sistemas de informação.

Assim, é um engano considerar um sistema de informação particular como uma solução permanente. Em vez disso, os bem sucedidos profissionais de negócios verão a introdução de novos sistemas de informação como um processo repetitivo. A introdução de novos sistemas que adicionam valor de novos modos, tornar-se-á um aspecto regular da operação empresarial.

2.4 CLASSIFICAÇÃO

Os principais tipos de sistemas de informações são apresentados na Figura 2. Cada nível contempla as necessidades de conhecimento de um determinado público alvo (Laudon & Laudon, 1998).

Figura 2 - Níveis do Conhecimento



Os sistemas de informações do nível operacional são utilizados para o controle do fluxo das atividades básicas da organização, tal como vendas, fluxo de caixa, controle de materiais e outros.

Os sistemas de informações do nível de conhecimento apóiam o processo de coleta e armazenamento de novo conhecimento associado ao negócio, para a administração da continuidade das tarefas cotidianas.

Os sistemas de informações do nível gerencial são utilizados periodicamente no monitoramento, controle e tomada de decisões, pelos gerentes de nível médio da organização. Neste nível figuram ainda sistemas de informações baseados em simulações e perguntas, como sistemas *what-if*.

Os sistemas do nível estratégico são utilizados pelos executivos para a realização do planejamento estratégico. Proporcionam a visão necessária da empresa para o planejamento das próximas ações em face do ambiente externo.

De acordo com os níveis organizacionais, pode-se classificar os seis principais tipos de sistemas de informações como segue, e no Quadro 1, são apresentadas as principais características dos tipos de sistemas de informações (Laudon & Laudon, 1998).

No nível operacional figura o *TPS, Transaction Processing Systems* ou sistemas de processamentos transacionais.

No nível de gerência de conhecimento figuram o *KWS, Knowledge Work Systems* ou sistema de controle de conhecimento e o *OAS, Office Automation Systems* ou sistemas de automação de escritório.

No nível gerencial, figuram o *DSS, Decision Support Systems* ou sistemas de suporte à decisão e o *MIS, Management Information Systems* ou sistemas de administração das informações.

No nível executivo figura o *ESS, Executive Support Systems* ou sistema de suporte aos executivos.

Independentemente do tipo, o sistema de informação tem uma grande relação com o aspecto tecnológico. A tecnologia auxilia na *busca* dos dados necessários, na *geração* de informação e no *armazenamento* do conhecimento.

Quadro 1 – Características dos Sistemas de Informações

Sistema	Entradas	Processamento	Saída	Usuários
<i>ESS</i>	Dados Agregados, externos e internos.	Gráficos, simulações, interações.	Projeções, tabelas.	Executivos
<i>DSS</i>	Baixo volume de dados ou grande volume otimizado para análises	Interações, simulações, análises	Relatórios especiais, análises de decisão, tabelas	Gerentes, equipe da gerência
<i>MIS</i>	Sumários de dados, Grande volume de dados	Relatórios rotineiros, análises de baixo-nível	Sumários e relatórios de exceções	Gerentes médios
<i>KWS</i>	Base de conhecimento	Modelagens e simulações	Modelos, gráficos	Equipe de administração de conhecimento
<i>OAS</i>	Documentos, cronogramas	Gerência de documentos e cronogramas	Documentos, cronogramas e cartas	Equipe de administração de conhecimento
<i>TPS</i>	Transações, eventos	Sorteio, listagem, agregação e atualização	Relatórios detalhados, listagens e sumários	Equipe operacional e supervisores

Fonte: Adaptado de Laudon & Laudon (1998)

2.5 TECNOLOGIA DE INFORMAÇÃO

Tecnologia de informação é uma das bases que sustentam os sistemas de informações. Pode-se entender como sendo: computadores, *software*, redes de comunicação eletrônica pública e privadas, rede digital de serviços, tecnologia de telecomunicações, protocolos de transmissão de dados e outros serviços (Tait, 2000).

O objetivo de um sistema de informação deve ser a integração entre negócios, sistemas e tecnologia da informação. Esta integração deve ser feita de maneira com que os negócios sejam suportados por sistemas de informações baseados em tecnologia de informação ou através do uso do sistema de informação ligando clientes e fornecedores ou ainda pelo planejamento estratégico da empresa (Tait, 2000).

Segundo Verstraet (2000, p.2), "historicamente, os sistemas de informações estão mais ligados à tecnologia do que nas pessoas ou procedimentos. Contudo, a tendência é a maior ênfase nas pessoas e procedimentos". Ao invés de ver como uma tecnologia se adapta aos negócios, o enfoque passa a ser na análise dos problemas da organização (aprendizagem organizacional), que considera as maneiras efetivas de melhorar as organizações em função de seus objetivos, usando ou não uma tecnologia de informática (Laudon & Laudon, 1998).

Talvez devido à curta história do processamento de dados comercial, o uso de sistemas de informações ainda é acompanhado por uma expectativa por vezes irreal e idealista. A simples adoção do uso da tecnologia da informação é tomada como garantia de que esteja sendo feito adequado controle das atividades da empresa (Feliciano Neto, Shimizu, 1996). A evolução da tecnologia de informação aliada ao suporte das atividades das organizações é retratada por Reinhard (apud Tait, 2000), conforme Quadro 2.

Seguindo a evolução da utilização da tecnologia de informação retratada no Quadro 2, nota-se que com o passar dos anos, a tecnologia de informação está cada vez mais ligada aos conceitos gerenciais propostos pelo sistema de informação. Isto se deve ao fato de que a qualidade das informações estratégicas tem sido um dos grandes diferenciais para as empresas modernas.

Analisando a evolução da tecnologia utilizada pelos sistemas de informações, pode-se afirmar que, com a evolução do *hardware*, bancos de dados e linguagens de programação, tornou-se possível atender a demanda por informações gerenciais de maneira mais rápida (Inmon, 1997). Mais recentemente, o surgimento de técnicas como *data warehousing*, contribuiu para a construção de *softwares* orientados para sistema de informação. Esta técnica será detalhada no capítulo seguinte.

Quadro 2 - Histórico da Utilização da Tecnologia de Informação

Década	Características
60	<ul style="list-style-type: none"> • Empresas iniciam o uso de tecnologia de informação; • Poucas opções tecnológicas (<i>software</i> e equipamento); • Processos de construção de aplicativos trabalhosos, com pouco suporte de ferramentas; • Necessidade de metodologias para atender demanda de forma rápida; • Automação de rotinas manuais; • Escassez de mão-de-obra técnica; • Desenvolvimento com caráter artesanal.
70	<ul style="list-style-type: none"> • Aumento do impacto dos sistemas nas empresas; • Analistas passam a considerar: conceitos de desenvolvimento organizacional; processo decisório; adoção de inovações; aprendizagem; interface humano-computador; relacionamento entre profissionais de tecnologia de informação e usuário; • Estímulo à construção de sistemas de apoio à decisão.
80	<ul style="list-style-type: none"> • Mudanças no ambiente externo das empresas; • Terceirização; • Sistemas interorganizacionais; • Arquitetura de sistemas; • Desenvolvimento de sistemas considerando-se aspectos: econômicos, legais, políticos, culturais.
90	<ul style="list-style-type: none"> • Tecnologia de informação como centro da estratégia empresarial; • Conhecimento como fonte de geração de valor.

Fonte: Tait (2000)

O uso da tecnologia de informação orientado por um sistema de informação estruturado e planejado possibilita à organização um maior controle dos processos internos e orienta a administração estratégica. Em face do mercado globalizado e de uma concorrência cada vez mais acirrada, torna-se cada vez mais importante a adoção de soluções baseadas neste preceito.

2.6 CONCLUSÃO

Os sistemas de informações há muito são empregados nas empresas por exercerem um papel fundamental na realização das atividades. A empresa precisa ficar atenta às mudanças externas e ao seu funcionamento. Com a crescente competitividade,

sobreviverão as empresas que estiverem melhor preparadas, onde os sistemas de informações constituem um dos principais recursos.

Neste capítulo foram abordados os conceitos básicos acerca de sistemas de informações, sua classificação e adequação aos níveis organizacionais. Sobre cada nível organizacional poderá ser aplicado pelo menos um tipo de sistema de informações, preparado para auxiliar na realização das tarefas pertinentes ao nível.

Um dos principais componentes de um sistema de informações é a tecnologia de informação. A evolução dos sistemas de informações está diretamente ligada à evolução do *hardware*, bancos de dados, linguagens de programação, e de técnicas para desenvolvimento de *softwares* como *data warehousing*, a qual é abordada no capítulo seguinte.

3 DATA WAREHOUSING

3.1 INTRODUÇÃO

O conceito de *data warehousing* (desenvolvimento de sistemas de informações baseados em *data warehouse*), embora surgido recentemente, baseia-se em idéias que vinham sendo aplicadas em vários sistemas de informações há muitos anos (Inmon, 1997). A técnica ganhou mais força com o surgimento e evolução de várias tecnologias e metodologias, que facilitam a implementação de sistemas de informações.

A arquitetura proposta foi elaborada para a distribuição dos componentes de um *data warehouse*, e pode ser aplicada em conjunto com a metodologia de desenvolvimento baseada em *data marts* incrementais. Tem por objetivo orientar a implementação dos vários componentes tecnológicos existentes em um sistema de informações.

Neste capítulo aborda-se a técnica *data warehousing*, seus principais elementos, fases da construção e aspectos implementacionais. O objetivo é fundamentar a apresentação da arquitetura no capítulo 4 e a discussão da aplicação no capítulo 5.

3.2 DEFINIÇÃO

Data warehousing é uma técnica de desenvolvimento do componente tecnológico de sistemas de informações onde a preparação dos dados e do ambiente é baseada em um

data warehouse. Segundo Inmon (1997, p.33), *data warehouse* é “um conjunto de dados baseado em assuntos, integrado, não volátil e variável em relação ao tempo, de apoio às decisões gerenciais”. Analisando esta definição podem-se abstrair as seguintes características:

- *data warehouse é baseado em assuntos*: o *data warehouse* é projetado com o intuito de fornecer informações estratégicas sobre o negócio, e não descrever os processos do negócio, como os sistemas operacionais³ da organização;
- *data warehouse é integrado*: ao projetar-se o modelo de dados do *data warehouse*, tem-se o cuidado de eliminar as redundâncias e as possibilidades de respostas ambíguas. O modelo é construído e organizado de forma a obter-se respostas únicas e certas, normalmente de forma independente da forma como os dados estão organizados nos sistemas operacionais;
- *data warehouse não é volátil*: a principal idéia na alimentação de dados no *data warehouse* é a de que em determinados períodos, sejam extraídos dados dos sistemas operacionais e armazenados no *data warehouse*. Uma vez armazenado, o dado não sofrerá alterações;
- *data warehouse é variável em relação ao tempo*: com o acúmulo de dados sobre diversos períodos, o *data warehouse* fornecerá subsídios para análises do negócio em tempos diferentes, possibilitando análises de regressões, tendências, etc.

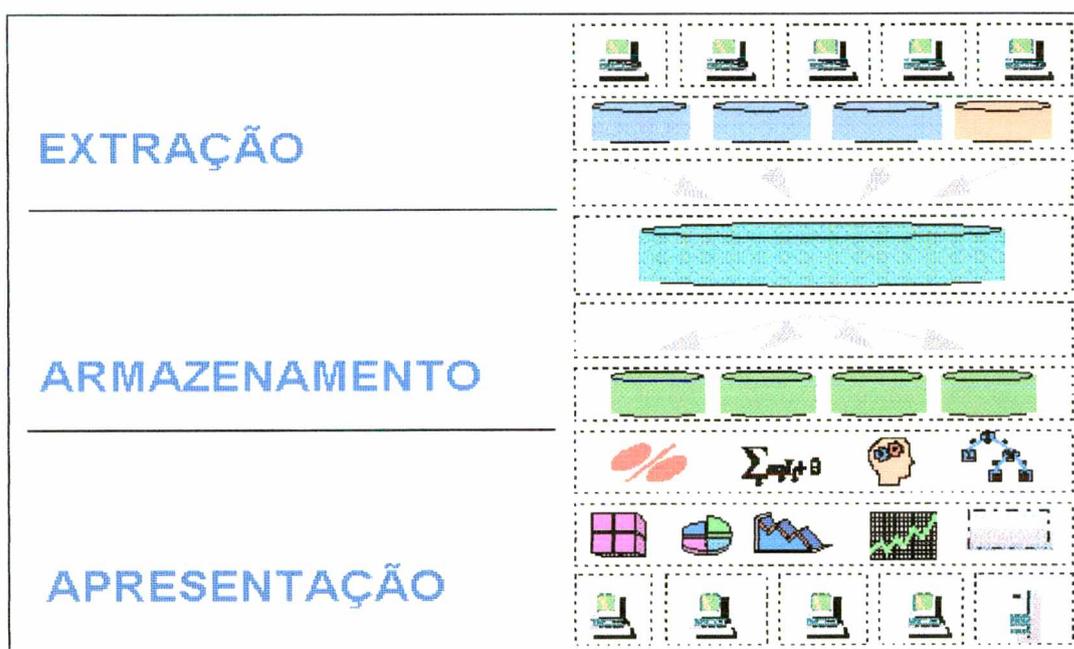
Um *data warehouse* bem projetado conterà todos os dados para responder as perguntas cruciais formuladas pela gerência (O que?, Quando?, Por quê?, E se?, etc ...). Ao contrário dos sistemas operacionais, o *data warehouse* é projetado para que estas respostas sejam fornecidas em tempo hábil para a tomada de decisões.

³ Sistemas operacionais são aqui considerados os programas voltados aos processos de operação da organização.

Para responder os questionamentos dos usuários em um *data warehouse*, parte-se do pressuposto de que o acesso aos dados seja efetuado sobre uma (ou várias) bases de dados consolidadas.

Para tanto, o esquema de funcionamento de um de *data warehouse* pode ser dividido em processos básicos, sendo estes a *extração de dados dos sistemas operacionais*, o *armazenamento dos dados* e a *apresentação de informações*, conforme ilustrado na Figura 3 (Palma, 1998).

Figura 3 - Processamento Básico em um Data Warehouse



Fonte: Adaptado de Armstrong (2000)

Por *extração de dados*, entende-se a concepção ou aquisição e parametrização das ferramentas que realizarão as tarefas de coleta, limpeza, transformação e migração dos dados operacionais ao *data warehouse*. A realização de todas as tarefas desta fase constitui um dos processos mais morosos e delicados no *data warehousing* (Kimball, 1998b) e (Inmon, 1997).

Por *armazenamento dos dados*, entende-se a concepção do repositório das informações, que é o núcleo do ambiente do *data warehouse*. Neste estarão

representados todos os dados extraídos dos sistemas operacionais, necessários para o processo de tomada de decisão (Inmon, 1997) e (Kimball, 1998b).

Por *apresentação de informações* entende-se a concepção ou aquisição e parametrização das aplicações clientes e do servidor que atenderá às requisições de dados junto ao *data warehouse* e disponibilizará as informações resultantes (Tanler, 1998) e (Corey, Abbey et al, 1998).

3.3 FASES TÍPICAS NO DATA WAREHOUSING

O desenvolvimento do *data warehouse* pode ser decomposto em processos distintos conforme Figura 4 (Giovinazzo, 2000) e (Debevoise, 1999).

Figura 4 – Fases Típicas no Desenvolvimento de um Data Warehouse



A título ilustrativo, a Figura 4 apresenta um *data warehouse* concebido através de um enfoque utilizado na concepção de sistemas operacionais, apenas para retratar os processos envolvidos nesta e em outras metodologias.

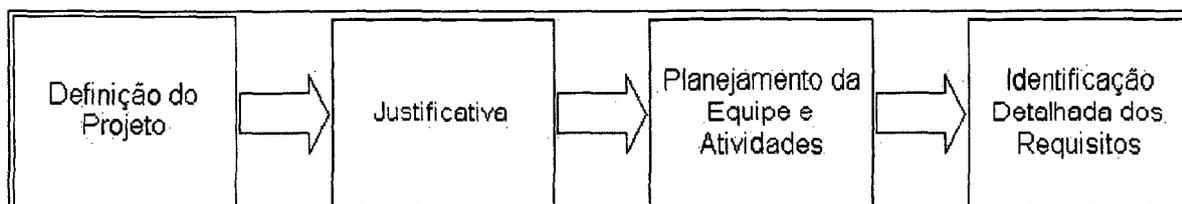
No capítulo seguinte serão abordadas as metodologias para desenvolvimento de um *data warehouse*. A seguir, serão abordados os processos ilustrados na Figura 4.

3.3.1 Planejamento

Como o próprio nome propõe, nesta fase é planejada a construção do sistema. Constitui uma das fases mais importantes, pois qualquer falha na delimitação de escopo, identificação de necessidades ou erro na especificação dos recursos pode resultar na

inviabilização total do projeto. As etapas constituintes desta fase são ilustradas na Figura 5 e em seguida cada qual será abordada (Kimball, 1998b).

Figura 5 – Etapas do Planejamento



a **Definição do Projeto**

Nesta importante fase do planejamento, é identificada a existência e a origem de demanda. Logo, será realizado um levantamento da empresa, onde serão identificados os potenciais usuários e suas necessidades.

Segundo Kimball (1998b), para o sucesso de um projeto de *data warehouse* é necessário que a empresa esteja realmente engajada no projeto, devido à necessidade de acompanhar os concorrentes ou estratégia de crescimento. Outro fator importante é a parceria entre comunidade de usuários e profissionais de informática e a observação da cultura de utilização de informação na empresa (Chuck, Dirk et al., 1998).

Em caso de identificação de pouca necessidade, falta de patrocinadores⁴ influentes e motivados ou falta de cultura do uso de informação, recomenda-se o adiamento do projeto, antes de obter-se um fracasso no meio do projeto.

Nesta etapa deve-se focar brevemente os pontos estratégicos do negócio e como são acompanhados, bem como identificar os processos centrais da organização e quais são mais desejáveis para serem abordados no projeto.

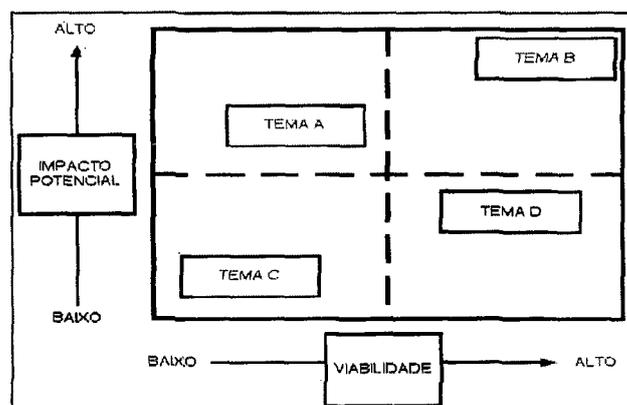
⁴ Usuários com respaldo político, responsáveis pela defesa do projeto e obtenção de recursos. Essenciais para um projeto de Data Warehouse (Kimball, 1998b).

Baseado neste levantamento, devem ser eleitas as prioridades a serem abordadas no projeto. Kimball (1998b) apresentou como técnica para eleição das prioridades uma análise em quadrantes, ilustrada na Figura 6.

Os principais temas são analisados e projetados no quadrante segundo sua viabilidade e impacto no negócio. Por viabilidade entende-se a disponibilidade de dados e recursos humanos, além da facilidade de desenvolvimento. No caso da Figura 6, o tema B é sem dúvida o aspecto mais indicado a ser abordado, devido ao alto impacto causado no negócio e à alta viabilidade.

Uma vez definido o escopo do projeto, deve-se documentá-lo para registrar os temas que serão abordados, bem como os que não irão, evitando-se assim futuras dúvidas.

Figura 6 – Análise de Quadrante para Definição de Prioridades



Fonte: Adaptado de Kimball (1998b)

b Justificativa

Definido o escopo, o passo seguinte é construir a justificativa do projeto. Nesta justificativa serão apresentados os custos e benefícios estimados.

Entre os custos, deverão ser considerados (Kimball, 1998b) e (Inmon, 1997):

- aquisição de *hardware* e licenças de *software*;
- despesas com manutenção do *hardware* e *software*;
- estimativa inicial de recursos humanos internos e externos;
- preparação do corpo técnico e treinamento de usuários;
- despesas com equipe de suporte;
- despesas com upgrades no *hardware*.

Para determinar os benefícios, é necessário investigar os ganhos no processo de tomada de decisão da corporação. Deve-se priorizar a receita e o lucro obtidos nas oportunidades geradas.

Uma forma proposta por Kimball (1998b) é a de quantificar em dinheiro os retornos esperados, realizando questionamentos com os futuros usuários em relação às informações que serão obtidas pelo sistema, tais como:

- o que aconteceria se você tivesse...?
- você é capaz de prever...?
- quanto custa para a empresa não saber...?

Com as respostas obtidas neste primeiro questionamento, deve-se continuar a investigação indagando até obter-se valores monetários de ganhos e cortes de despesas.

Após a estimativa dos custos e dos benefícios, deverá ser calculado o retorno do investimento (ROI). Deve-se salientar não só os retornos diretos, mas os retornos que

poderão ser obtidos futuramente nos processos decisórios. Kimball (1998b) sugere que se deva apresentar como um “custo de oportunidade”, ou seja, se o benefício estimado com a implantação do sistema é de R\$ 100.000,00 mensais, então a sua não disponibilização custará R\$ 100.000,00 mensais para a empresa.

c *Planejamento da Equipe e Atividades*

Tendo a aprovação para iniciar o projeto, deve-se definir a equipe e o plano de atividades do projeto.

A equipe será definida tendo em vista as atividades ou papéis constituintes de um projeto de *data warehouse*, levando-se em consideração que os membros da equipe assumirão mais de uma atividade/papel durante o projeto.

Cabe ao gerente do projeto definir os integrantes do projeto, observando os papéis existentes, recursos e tempo, além de esboçar o planejamento do projeto. Os principais papéis relacionados por Kimball (1998b) são apresentados no anexo A de acordo com a ordem em que entram em cena no decorrer do projeto.

No planejamento do projeto serão detalhadas as tarefas, prazos, status e responsáveis. Na especificação das tarefas deve-se atentar para o maior desmembramento possível em sub-tarefas, a fim de ter um maior controle e acurácia na estimativa de tempo (Kimball, 1998b) e (Inmon, 1997).

d *Identificação Detalhada dos Requisitos*

Esta consiste numa tarefa crucial para o sucesso do projeto. Na identificação dos requisitos são levantadas de forma detalhada as necessidades dos usuários.

A atividade é coordenada de preferência pelo analista da aplicação cliente. Frequentemente o gerente do projeto, modelador de dados, e o desenvolvedor da aplicação cliente auxiliam os procedimentos de identificação dos requisitos.

Existem muitas maneiras de se realizar a coleta de requisitos, mas de forma geral, estas seguem basicamente dois princípios: dirigida pelos dados ou dirigida pelos requisitos (Chuck, Dirk et al., 1998).

Quando dirigida pelos requisitos, serão levantadas as funções e as necessidades dos usuários. A coleta de requisitos se dá através de entrevistas com executivos, gerentes intermediários e funcionários. Além das entrevistas, investigações mais minuciosas são realizadas, através da análise dos sistemas existentes, questionários, sessões *JAD (Joint Application Design)* ou *brainstorming* e análise do plano estratégico de negócios (Inmon, 1997) e (Chuck, Dirk et al., 1998).

Quando dirigido pelos dados, inicialmente serão levantados os dados existentes nos sistemas da organização, para verificar-se o que é possível oferecer. Neste caso corre-se o risco de não atender as reais necessidades dos usuários, mas dificilmente haverá problemas quanto a especificações que não possam ser implementadas por falta de dados.

Os resultados da definição dos requisitos são documentados e ao final da fase da coleta descrevem-se os requisitos do sistema em um documento que será revisado pelos futuros usuários.

3.3.2 Modelagem Lógica e Física

O repositório das informações constitui o núcleo do ambiente do *data warehouse*. Neste estarão contidos todos os dados extraídos dos sistemas operacionais, necessários para o processo de tomada de decisão.

A primeira fase no projeto do repositório é a definição do modelo de dados. Nesta, serão analisados os dados necessários e que podem ser obtidos junto aos sistemas operacionais. É então criado um modelo de dados corporativo, que difere do modelo corporativo operacional na medida em que são retirados os dados que são utilizados apenas no ambiente operacional, e na medida em que são incorporados elementos de tempo nas chaves, além do agrupamento dos dados conforme sua utilização.

Segundo Kimball (1998a), a tradicional abordagem relacional altamente normalizada⁵, utilizada com sucesso no projeto de sistemas para o ambiente operacional, não é adequada para a construção de um *data warehouse*. No modelo relacional, a estrutura é otimizada para recuperar, criar e atualizar registros individuais em tempo real e preservar a integridade dos dados. No *data warehouse*, a utilização dos dados é muito diferente. As consultas realizadas recuperam um grande número de registros e os resume, durante o processamento. Neste caso, a estrutura normalmente não é normalizada para evitar junções de muitas tabelas, obtendo-se assim uma performance superior.

O Quadro 3 relaciona algumas diferenças apresentadas por Inmon (1997), de enfoque dos dados entre os sistemas operacionais e o *data warehouse*.

Além do imprescindível aspecto da performance, o período em que os dados ficam armazenados no banco de dados deve ser levado em conta pelo modelo de dados do *data warehouse*. Em um sistema operacional, o dado é atualizado e normalmente não mantém um registro histórico das mudanças. Já em um *data warehouse*, o dado normalmente não é atualizado. As modificações de atributos são armazenadas em registros separados.

Deve-se considerar também a legibilidade do modelo de dados. Não existem ferramentas gráficas que expõem um modelo entidade-relacionamento ao usuário para que este possa montar suas pesquisas customizadas, devido a enorme confusão que um

emaranhado de tabelas propicia. Por estes motivos, deve-se considerar a adoção de um modelo mais adequado à realidade de um *data warehouse*.

Quadro 3 - Diferenças entre Dados de Sistemas Operacionais e *Data Warehouse*

Dados Operacionais	Dados em um <i>Data Warehouse</i>
Baseados em aplicações	Baseados em assuntos ou negócios
Detalhados	Resumidos ou refinados
Exatos em relação ao momento do acesso	Representam valores de momentos já decorridos ou instantâneos
Acessados uma unidade por vez	Acessados um conjunto por vez
Voltados para transações	Voltados para análise
Alta disponibilidade	Disponibilidade atenuada
Não contemplam a redundância	A redundância não pode ser ignorada
Estrutura fixa; conteúdos variáveis	Estrutura flexível
Pequena quantidade de dados usada em um processo	Grande quantidade de dados usada em um processo
Atendem às necessidades cotidianas	Atendem às necessidades gerenciais
Alta probabilidade de acesso	Baixa, ou modesta probabilidade de acesso

Fonte: Adaptado de Inmon (1997)

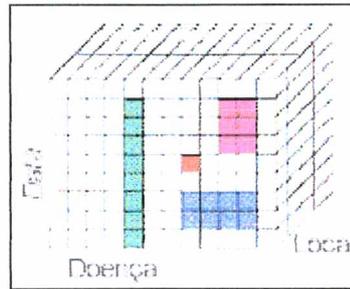
a **O Modelo Dimensional**

Muito se tem discutido sobre a perspectiva de utilização da abordagem dimensional para representação dos dados. O modelo dimensional lembra a idéia do cubo (Kimball, 1998a), contendo três ou mais dimensões cada uma representando um atributo diferente conforme apresenta a Figura 7.

Como exemplo, tomam-se as dimensões de análise de dados hospitalares *Local*, *Doença* e *Data*, e pode-se localizar um determinado fato, como o número de mortos ou atendimentos hospitalares. Desta forma o usuário pode “girar” o cubo para obter novos fatos a partir das dimensões.

⁵ Normalização se refere à premissa de que múltiplas tabelas são modeladas para representar relações um-para-muitos, muitos-para-muitos ou um-para-um. A normalização evita a duplicação de dados a fim de preservar a integridade e minimizar o volume de dados.

Figura 7 – Semelhança do Modelo Dimensional a um Cubo



Os projetistas têm implementado o modelo dimensional usando um banco de dados multidimensional, ou relacional através do esquema estrela (*star join schema*) (Kimball, 1999a). No centro figura uma tabela principal, a tabela de fatos, que contém os dados do negócio a serem analisados, e, em torno, várias tabelas descritivas, chamadas tabelas de dimensão. Cada tabela de dimensão possui uma única ligação com a tabela de fatos, a qual é feita através de chaves externas (Kelly, 2000).

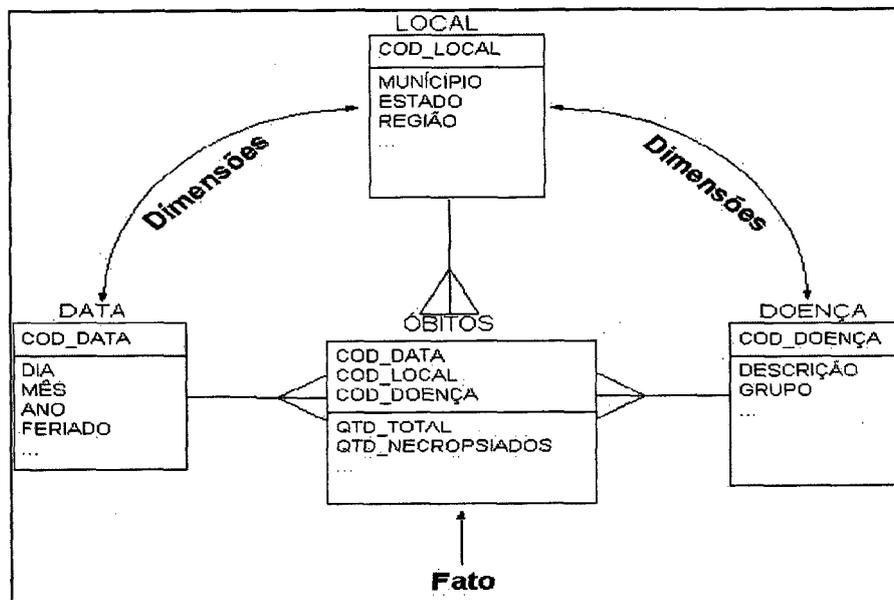
Uma dimensão é um atributo de um fato. Como exemplo de dimensões têm-se: período de tempo, regiões geográficas, produtos, promoções, etc. As tabelas de dimensão contêm dados textuais, ou numéricos descritivos, enquanto que as tabelas de fatos dados numéricos. A Figura 8 apresenta um modelo dimensional seguindo o exemplo anterior do cubo.

O modelo dimensional, além de agilizar o processamento das consultas, permite uma melhor visualização dos dados, devido à forma simples de organizá-los. Esta forma de organizar os dados ainda propicia a flexibilidade necessária para eventuais ajustes que se façam necessários no modelo (Kimball, 1998b), como:

- adicionar novos fatos à tabela de fatos, desde que correspondam ao mesmo nível de detalhe;
- adicionar novas dimensões;
- adicionar novos atributos às dimensões existentes;

- redefinir o nível de detalhe dos dados.

Figura 8 – O Modelo Dimensional



Na segunda fase da modelagem é construído o modelo físico, onde são incluídas características físicas e chaves. Tendo definido os atributos que figurarão nas tabelas de fatos e nas dimensões, resta ainda decidir sobre importantes aspectos de organização dos dados e performance, representados inicialmente pela definição da granularidade, do particionamento, e da definição de agregados.

b Granularidade e Particionamento

A granularidade refere-se ao nível de detalhe ou de resumo com o qual serão armazenados os dados no *data warehouse* (Inmon, 1997). A sua definição afeta diretamente o volume de dados do *data warehouse* e a qualidade das consultas que poderão ser feitas. Quanto maior o nível de granularidade menor o nível de detalhe e vice-versa.

Uma granularidade de alto nível garante maior rapidez nas consultas feitas pelos usuários, mas, em contrapartida, há uma diminuição da riqueza das informações. Por

outro lado, uma granularidade de baixo nível possibilita a obtenção de resposta a qualquer consulta, mas haverá um aumento do volume de dados, o que conseqüentemente, fará com que o tempo de resposta seja maior e que o investimento em *hardware* seja maior.

É necessário encontrar o ponto de equilíbrio. O nível adequado de granularidade deve ser definido de tal forma que atenda as necessidades do usuário, tendo como limitação os recursos disponíveis.

Para solucionar o “dilema da granularidade”, pode-se utilizar dois níveis de granularidade. Os dados mais recentes são armazenados em um nível de granularidade menor (maior detalhe) e, com o passar do tempo, estes são resumidos, criando um nível de granularidade maior. Por exemplo, pode-se definir que dados terão grão mensal para os anos mais recentes e, grão anual, para os anos menos recentes. Apenas quando o volume de dados for relativamente pequeno, não serão necessários níveis duais de granularidade (Inmon, 1997).

Já o particionamento refere-se à repartição dos dados em unidades físicas diferentes, de acordo com sua utilização, ou lógica do negócio.

Segundo Chuck, Dirk et al. (1998), o particionamento dos dados oferece:

- flexibilidade no acesso aos dados;
- facilidades no gerenciamento dos dados, tais como indexação, recuperação, pesquisa, entre outros;
- escalabilidade ao *data warehouse*;
- portabilidade, na medida em que aumentam as possibilidades de compartilhamento dos dados.

Alguns dos critérios mais utilizados no particionamento, entre outros são (Chuck, Dirk et al., 1998) e (Inmon, 1997):

- períodos;
- geografia (local);
- produto (ou linha de negócio);
- unidade de negócio.

c **Agregados**

Segundo Kimball (1998a, p.191), “Agregado é um registro de tabela de fatos que representa o resumo dos registros de nível básico da tabela de fatos”. Em outras palavras, sumários dos dados contidos no modelo original, organizados de forma a atender consultas rotineiras de forma mais ágil. Constitui uma das formas mais eficientes de melhorar a performance das consultas sobre o *data warehouse* (Neil, Schrader, et al, 1997).

Embora a definição de agregados ocorra mais freqüentemente durante a utilização por parte dos usuários, é válido projetá-los já na fase de definição do modelo de dados, para orientar dimensionamento de *hardware* e ajustes no próprio modelo, aproveitando também que os requisitos do usuário acabaram de ser definidos.

Deverão ser considerados candidatos os agregados que reduzirem consideravelmente o volume em relação ao modelo original, normalmente em torno da razão de 10 ou mais (Donald, 1997).

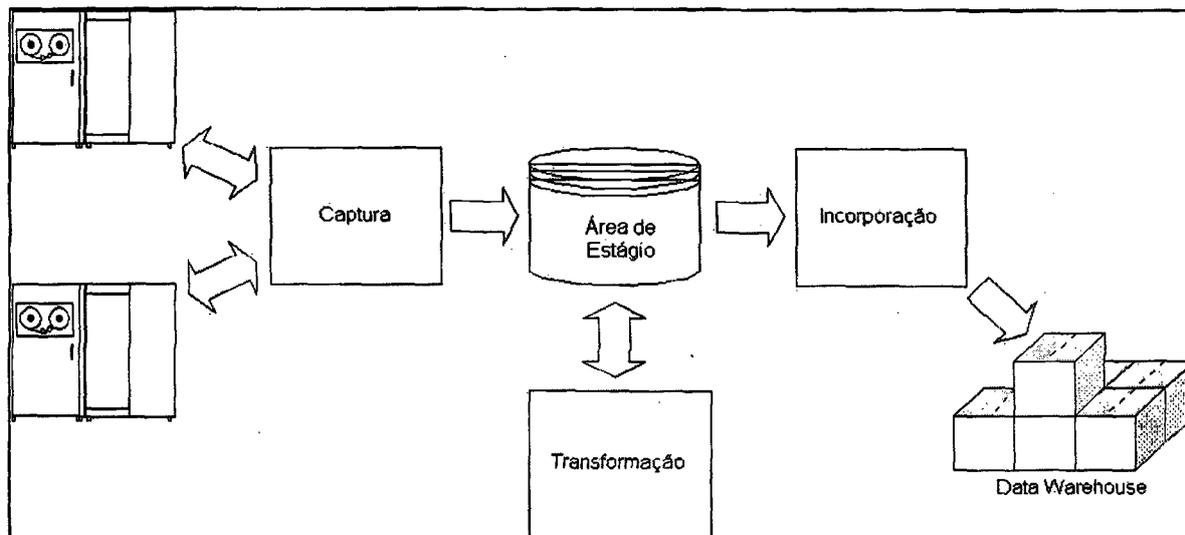
Existem duas abordagens para armazenar agregados: definindo novas tabelas de fatos ou definindo campos nível. A utilização da primeira abordagem torna mais simples a manutenção, a carga e a utilização dos dados (Kimball, 1998a).

3.3.3 Preparação e Desenvolvimento de Ferramentas de Extração

Extração é o processo de coleta dos dados existentes nos sistemas operacionais e sistemas externos para incorporação no *data warehouse*. Os dados são coletados, tratados e transformados no formato do modelo de dados do *data warehouse* e finalmente incorporados na base de dados, como representado na Figura 9 (Chuck, Dirk et al., 1998).

A definição e implementação da extração são de suma importância em um projeto de *data warehouse*, consumindo muito tempo para serem realizadas, tendo em vista a complexidade.

Figura 9 – Processo de Extração



a **Coleta**

É muito comum que os dados nos sistemas operacionais não sejam integrados. Também freqüente é o fato de que os mesmos dados sejam representados de maneira diferente entre os diversos sistemas operacionais. Segundo Inmon (1997), as principais diferenças encontradas na representação são:

- nomenclatura diferente para identificação do mesmo dado. Exemplo: filiação e nome do pai;
- mesma nomenclatura, mas definição diferente. Exemplo: no sistema A o sexo é definido como um caracter contendo “M” ou “F”. Já no sistema B, o sexo é definido pelo número “0” ou “1”;
- unidades de medida diferentes. Exemplo: no sistema A, a dimensão de um objeto é medida em centímetros, já no sistema B a dimensão é medida em polegadas.

Tendo os sistemas definições diferentes, é preciso que ocorra uma conversão dos dados de acordo com as definições do modelo de dados do *data warehouse*.

Mas além da representação dos dados de forma diferente, há ainda a preocupação com a plataforma em que os sistemas operacionais são mantidos. Serão necessários procedimentos especiais para conversão dos dados mantidos nestes para a plataforma em que o *data warehouse* será mantido.

O ciclo de dados num *data warehouse* define o período de atualização dos mesmos. O ciclo varia de acordo com a natureza dos dados, podendo ser diário, mensal, trimestral ou anual.

Quanto aos tipos de carga dos dados Inmon (1997) cita três tipos, sendo:

- carregamento de dados históricos;
- carregamento de dados de valor corrente no ambiente operacional;
- carregamento de alterações do *data warehouse*. a partir de atualizações que tenham ocorrido nos sistemas operacionais.

Quanto às técnicas Inmon (1997) cita cinco tipos sendo:

- pesquisa de dados por datas;
- pesquisa em um arquivo de informações sobre alterações desde a última carga;
- pesquisa nos arquivos de log;
- modificação no código dos sistemas operacionais de forma que estes, ao registrarem alterações, atualizem de forma automática o *data warehouse*;
- comparação entre arquivos de imagem, sendo comparado serialmente o arquivo de imagem retirado após a última extração com o arquivo de imagem tirado no momento da nova extração.

b **Transformação**

No processo de transformação, os dados coletados são tratados em uma área intermediária conhecida como *área de estágio*. Em geral, é necessário construir ou adquirir uma ferramenta personalizada para realizar a transformação (bem como o processo de coleta e incorporação) que, além de ser capaz de lidar com a

heterogeneidade dos bancos de dados e dos próprios dados, tenha as seguintes funcionalidades (Berson, 1997):

- remoção de dados indesejáveis;
- conversão para nomes e definições comuns dos dados;
- cálculo de dados personalizados e derivados;
- estabelecimentos de valores default para dados esquecidos;
- documentação da origem das mudanças de definições de dados no metadados;
- geração de chaves.

c **Incorporação**

O processo de incorporação consiste da carga no *data warehouse* dos dados mantidos na área de estágio, criada no processo de transformação. Existem três técnicas básicas de incorporação (Chuck, Dirk et al., 1998):

- alteração: nesta técnica os dados existentes nas tabelas do *data warehouse* são alterados pelos gerados pelo processo de transformação. Se as tabelas existentes não existem, serão criadas;
- inserção: nesta técnica os dados existentes na área de estágio serão inseridos às tabelas existentes no *data warehouse* sem sobrepor os existentes;
- inserção construtiva: nesta técnica os dados existentes na área de estágio serão inseridos nas tabelas do *data warehouse* e serão atualizados valores calculados que representam estados.

3.3.4 Desenvolvimento de Ferramentas de Aplicações Cliente

O valor do *data warehouse* pode ser medido sob o ponto de vista das informações que possam ser extraídas, visto que toda a arquitetura até aqui apresentada tem por fim a disponibilização de informações estratégicas no apoio a decisões. Dentre as formas que podem ser utilizadas para interagir com o *data warehouse* vale citar: OLAP, análises estatísticas e *data mining*.

Segundo Harrison (1997), o rótulo OLAP (*On-Line Analytical Processing* ou Processamento Analítico On-Line) foi reservado para análises multidimensionais. As consultas simples e relatórios representam as facilidades mais simples proporcionadas pelo OLAP. Tipos mais sofisticados de consultas permitem que o usuário “navegue” por níveis diferentes de resumo dos dados, até chegar ao nível de detalhe encontrado nos sistemas operacionais ou ainda combinar dados na mesma consulta para a detecção de relações entre dados, de forma multidimensional.

Uma operação muito útil realizada pelos OLAPs é o chamado *drill-down*. Ao realizarmos um *drill-down* estamos aumentando o nível de detalhe da informação ao longo de uma dimensão. Supondo uma consulta que resultou num relatório contendo o total de mortos por região, evidenciando um número elevado de mortes em uma região específica. Se efetuarmos um *drill-down* sobre esta região, visualizaremos o total de mortos para cada município que a compõe. A operação de *drill-up* percorre o caminho inverso, ou seja, do dado específico para o geral.

Quando o rótulo OLAP foi introduzido, os dados eram extraídos dos bancos de dados relacionais e carregados em bancos multidimensionais. O dado, então, era analisado através das funcionalidades citadas com extrema rapidez. Este método foi cunhado por MOLAP ou OLAP Multidimensional. Em seguida outro método denominado ROLAP ou OLAP Relacional foi introduzido com muito sucesso, devido à facilidade de mapeamento por utilizar o modelo relacional como base para as análises, e dispensar a utilização de um banco de dados multidimensional proprietário (Donald, 1997).

As análises estatísticas empregadas vão além da simples redução de dados a serem apresentados, como cálculos de média. Harrison (1997) citou que análises estatísticas mais sofisticadas incluem emprego de métodos de regressão para estabelecimento da relação entre variáveis dependentes (venda de produtos) com variáveis independentes (preço, clima, etc), e outros métodos.

Data mining ou mineração de dados pode ser conceituada como um processo de reorganização e análise dos dados para a descoberta de informações ocultas. O *data mining* utiliza técnicas de análises estatísticas bem como técnicas de inteligência artificial, como redes neurais, algoritmos genéticos, reconhecimento de padrões, conjuntos difusos, etc (Berson, 1997).

3.3.5 Metadados

O metadado constitui dados sobre os dados, ou um repositório de informações sobre as regras de formação dos dados, origem, modificações, etc. Ou seja, todo e qualquer dado necessário para atender às necessidades do projetista, do administrador do *data warehouse* e a utilização da informação por parte dos usuários finais.

Existem basicamente dois tipos de metadados: o metadados técnico, utilizado pelo corpo técnico (DBAs e desenvolvedores), e os metadados de negócio, utilizado pelos usuários finais (Brackett, 1996).

O metadados técnico agrupa definições do *data warehouse*, modelo de dados, extração de dados, programas e todas as descrições técnicas dos dados e suas operações que possibilitem a manutenção do *data warehouse*.

O metadados de negócios prevê uma descrição do negócio, regras de formação dos dados, cálculos e outros que auxiliem a interpretação da informação, além de frequência de atualização e outros.

3.4 METODOLOGIAS DE DESENVOLVIMENTO

Várias são as metodologias utilizadas para a implementação de sistemas de informações e variadas são as formas de como as mesmas são utilizadas. Apesar de muitos anos de desenvolvimento destes sistemas, não existe um consenso entre os desenvolvedores.

As variáveis afetadas pela escolha da metodologia são entre outras, o prazo, o retorno do investimento, satisfação do usuário, quantidade de implementação a ser refeita, requisitos de recursos e flexibilidade para adequações (Chuck, Dirk et al., 1998).

Existem correntes de desenvolvedores que defendem o desenvolvimento dos sistemas de informações utilizando a abordagem tradicional de desenvolvimento de sistemas operacionais. Nesta metodologia existem fases bem definidas e cada qual termina após um determinado número de iterações. A Figura 10 apresenta o ciclo de vida do desenvolvimento tradicional (Martin, 1991) e (Laudon & Laudon, 1998).

Figura 10 – Fases do Desenvolvimento Tradicional



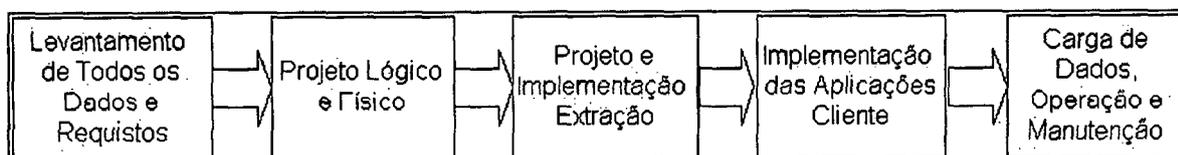
O ciclo de vida tradicional de concepção de sistemas é baseado na filosofia de “projeto”, onde existe um começo, meio e fim, com pouca ênfase sobre continuidade e refinamento. Nesta abordagem, normalmente no projeto lógico da organização de dados é adotado o modelo de Entidade-Relacionamento altamente normalizado.

Esta abordagem vem sendo utilizada há muitos anos e apresenta alguns problemas devido à sua estrutura rígida. O Quadro 4 apresenta a relação das principais vantagens e desvantagens da metodologia.

Quadro 4 - Vantagens e Desvantagens da Metodologia Tradicional

Vantagens	Desvantagens
Técnicas de modelagem dominadas pelo corpo técnico.	Dificuldade na realização de ajustes para adequação no modelo de dados, devido à forte dependência da implementação sobre o modelo.
	Fragilidade quanto a mudanças de cenários e necessidades dos usuários, devido à pouca previsão de novos ciclos de análise e implementação.
	A utilização do modelo entidade-relacionamento normalmente não é adequada a um sistema de informações devido à falta de clareza, performance inferior das consultas por conta da necessidade de efetuar-se muitas junções entre tabelas e ainda dificuldade em realizar-se mudanças na estrutura.

A metodologia baseada no *data warehouse* global, que foi utilizada no capítulo anterior para apresentar as fases existentes no *data warehousing*, é orientada especificamente para sistemas de informações, devido à utilização da técnica de *data warehouse*, bem como as metodologias que serão apresentadas em seguida. As fases constituintes de um *data warehouse* global são apresentadas na Figura 11 (Inmon, 1997).

Figura 11 – Fases do Desenvolvimento do Data Warehouse Global

Esta abordagem parte da idéia de desenvolvimento *Top-Down* (de cima para baixo). Inicialmente serão levantados todos os requisitos, fontes de dados, padrões, e outros, dos departamentos da empresa participantes da iniciativa, para então dar seqüência na implementação de toda a estrutura.

Apesar de ser orientado para sistemas de informações, esta abordagem apresenta algumas sérias desvantagens, mas também apresenta vantagens, como segue:

Quadro 5 – Vantagens e Desvantagens do *Data Warehouse* Global

Vantagens	Desvantagens
Como as fontes de dados são acessadas uma única vez e armazenadas em um único repositório, existe uma grande probabilidade de que os dados mantidos no <i>data warehouse</i> sejam dados confiáveis e coerentes, com representações unívocas.	O processo de análise e triagem de todos os dados existentes, análise de regras de negócio e todos os relacionamentos possíveis é demasiadamente custoso e demorado. Assim, o projeto só retornará algum resultado após muitos meses, tornando-se politicamente difícil de gerenciar.
Devido ao desenvolvimento da extração de dados ser feito uma única vez, evita-se replicação de esforços.	Os gastos iniciais com aquisição do <i>hardware</i> e infra-estrutura e formação da numerosa equipe técnica são demasiadamente elevados.
	O projeto e a implementação do processo de extração são complexos devido à enorme diversidade inicial de dados.

Devido ao fracasso de inúmeras implementações de *data warehouse* globais, empresas de consultoria apresentaram uma abordagem diferenciada para o desenvolvimento de *data warehouse*. Esta metodologia defende a construção de pequenos conjuntos do *data warehouse* baseados em assuntos específicos (*data marts*), para a apresentação de resultados em tempo bem mais reduzido (Vasconcelos, 1999).

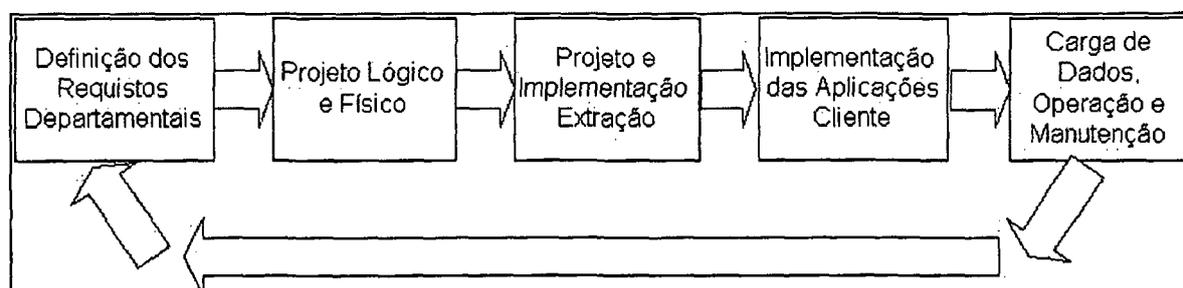
Ao contrário da abordagem anterior, esta parte do desenvolvimento *Bottom-Up* (de baixo para cima). Desta forma, serão implementados *data marts* individualmente, sem partir de uma análise de requisitos para toda a empresa, e com pouca perspectiva de integração entre eles.

Esta abordagem foi denominada *Legamart* ou *data mart* legado por Heckney (apud Vasconcelos, 1999), devido à falta de integração entre os *data marts*. As etapas de desenvolvimento da metodologia são retratadas na Figura 12 e as vantagens e desvantagens representadas no Quadro 6.

Uma nova abordagem para a implementação de *data marts* parte do princípio em que os requisitos devem ser bem definidos e que uma integração entre os *data marts* seja

planejada antes de começar a construção dos mesmos. Esta abordagem é conhecida como desenvolvimento baseado em *data marts* incrementais (Kimball, 1998b).

Figura 12 – Fases do Desenvolvimento dos *Data Marts* Legados



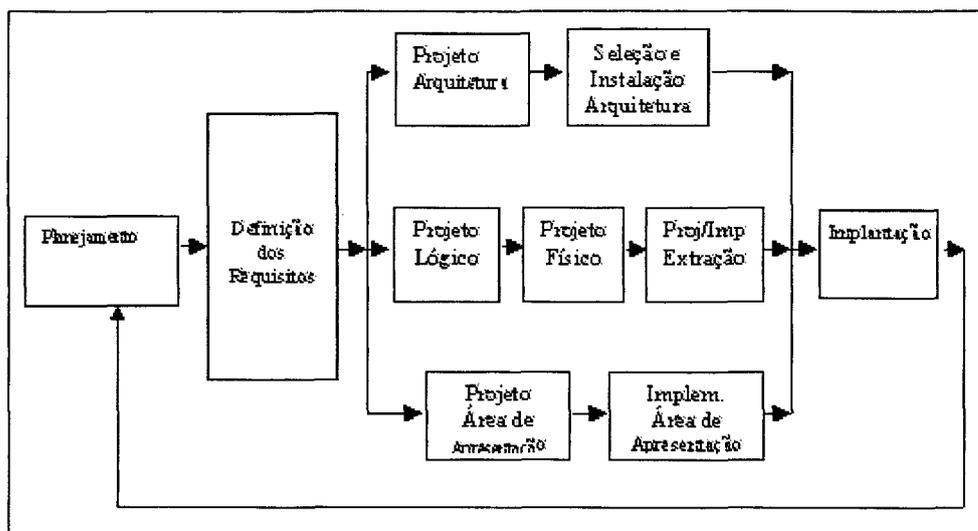
Quadro 6 – Vantagens e Desvantagens dos *Data Marts* Legados

Vantagens	Desvantagens
A apresentação dos primeiros resultados é feita de modo rápido e barato	Falta de planejamento de integração entre os <i>data marts</i>
Politicamente mais gerenciável devido ao orçamento reduzido	Grande probabilidade de incoerência nos dados, devido a diferentes representações e fontes de dados
	Esforços da concepção da extração replicados em cada <i>data mart</i>

Esta abordagem parte da combinação do desenvolvimento *Top-Down* e *Bottom-Up* e é baseada em um processo contínuo de desenvolvimento. Inicialmente, é realizada uma análise dos requisitos de forma global, de onde surge uma lista de *data marts* a serem implementados e como serão integrados. A partir daí, serão levantados os requisitos de um dos departamentos integrantes da iniciativa e implementado o *data mart* correspondente. O ciclo repete-se, incrementalmente, até que todos os *data marts* tenham sido implementados. O conjunto de todos os *data marts* desenvolvidos constitui o *data warehouse* da empresa.

A Figura 13 ilustra o processo de desenvolvimento desta abordagem e no Quadro 7 são retratadas suas vantagens e desvantagens.

Figura 12 – Fases do Desenvolvimento dos *Data Marts* Incrementais



Fonte: Adaptado de Kimball (1998b)

Quadro 6 – Vantagens e Desvantagens dos *Data Marts* Incrementais

Vantagens	Desvantagens
A apresentação dos primeiros resultados é feita de modo mais rápido e barato do que a abordagem global	Complicações políticas por conta da determinação da seqüência de implementação dos <i>data marts</i> e das prioridades de manutenção
A integração entre os <i>data marts</i> possibilita a unicidade de representação dos dados e informações mais confiáveis por não existirem redundâncias	Metadado mais complexo para gerenciar a distribuição e integração dos dados
Os mecanismos de extração são projetados uma única vez	Maior controle no nível de granularidade e nas manutenções das tabelas compartilhadas

Dentre as metodologias apresentadas, considera-se mais adequada a baseada em *data marts* incrementais, devido principalmente ao retorno rápido e à maior perspectiva de integração dos dados. Em seguida será apresentada a arquitetura proposta para a implantação em conjunto com a metodologia baseada em *data marts* incrementais.

3.5 CONCLUSÃO

Neste capítulo, apresentou-se uma revisão teórica sobre *data warehouse*. Foram mostrados os principais conceitos e componentes, com o propósito de fundamentar a elaboração e implementação do modelo proposto neste trabalho.

Sobre o processo de construção de um *data warehouse* pode-se concluir:

- planejamento do *data warehouse*: constitui uma das fases mais importantes do projeto. É nesta fase que são levantadas as necessidades dos usuários e determinados os limites do *data warehouse*. Um bom entendimento das necessidades iniciais apóia a construção de um *data warehouse* bem sucedido;
- modelagem: nesta fase é esboçado e construído o modelo de dados, de acordo com a definição dos requisitos. O modelo dimensional tem sido utilizado com muito sucesso em grande parte dos projetos devido à clareza, performance e flexibilidade, e é o modelo indicado. Importante observar a prática de definição de agregados, recurso imprescindível para a otimização da performance;
- projeto da extração de dados: fase normalmente responsável pelos maiores esforços e gastos no projeto. Nesta fase serão mapeados, extraídos, tratados e adicionados ao *data warehouse*, os dados contidos nos sistemas operacionais da empresa. A adoção de ferramentas especializadas e um dimensionamento de um tempo extra na concepção do cronograma pode ser decisivo para o sucesso desta fase;
- desenvolvimento das aplicações cliente : nesta fase serão concebidos os aplicativos que serão disponibilizados aos usuários finais para extração de informações do *data warehouse*. A construção de aplicativos de fácil utilização e de recursos gráficos para apresentação das informações é altamente indicada.

Os conceitos abordados neste capítulo serão referenciados nos próximos. No capítulo seguinte apresentam-se as metodologias para construção do *data warehouse* e a arquitetura proposta.

4 A ARQUITETURA PROPOSTA

4.1 INTRODUÇÃO

A escolha da metodologia e arquitetura de desenvolvimento deve ser a primeira decisão a ser tomada antes de se iniciar o desenvolvimento do sistema de informação. A escolha pode ser modificada após o início da implementação, mas normalmente esta decisão acarretará em grande volume de trabalho a ser refeito.

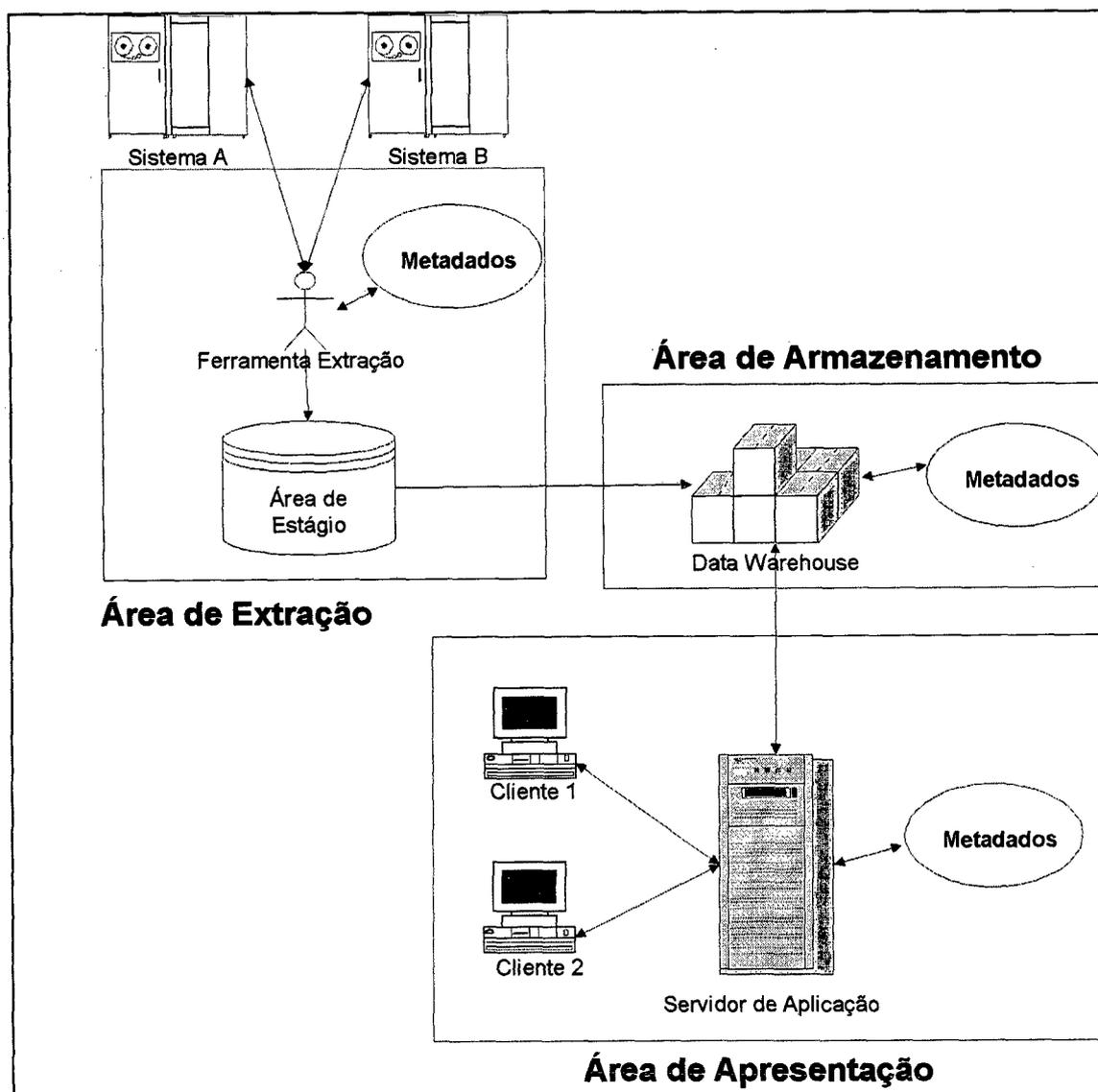
Neste capítulo serão abordadas as principais características das arquiteturas empregadas em sistemas de informações baseados em *data warehouse*. Em seguida será apresentada a arquitetura proposta, seus componentes, diferenças em relação às outras arquiteturas e benefícios da sua utilização, além da aplicação da metodologia de desenvolvimento baseada em *data marts* incremental.

4.2 ARQUITETURAS

A arquitetura determina como se dá a organização dos componentes do ambiente do *data warehouse*. A sua correta definição constitui uma tarefa crucial para o projeto, devido à grande dependência existente entre a implementação dos componentes e a sua organização.

Conforme apresentou-se no capítulo anterior, os componentes comuns existentes em um sistema de informação baseado em *data warehouse* são distribuídos em três áreas, sendo estas a área de extração, área de armazenamento e área de apresentação, conforme ilustra a Figura 14. As diferenças existentes entre as arquiteturas são resumidas à forma de implementação das áreas, principalmente da área de apresentação e de armazenamento.

Figura 14 - Componentes da Arquitetura



Na área de extração, a maioria das arquiteturas possui um único componente de extração que é responsável pela coleta dos dados operacionais para um única área de

estágio, além de efetuar a limpeza, transformação e inserção no *data warehouse* valendo-se do metadados (Kimball, 1998b), (Inmon, 1997), (Tanler, 1998) e (Debevoise, 1999).

Na área de armazenamento, algumas arquiteturas divergem quanto ao escopo e quanto à forma de organização dos dados. Dentre as formas mais comuns de organização de dados, podem-se destacar:

- *data warehouse* global: a área de armazenamento é composta por um único repositório (particionado ou não) que reúne os dados necessários a suportar toda ou grande parte da corporação. Deste repositório podem ser derivados *data marts* (Inmon, 1997) e (Chuck, Dirk et al.,1998);
- *data marts* isolados: a área de armazenamento é composta por vários *data marts* projetados de forma isolada, sem interligação (Vasconcelos, 1999) e (Chuck, Dirk et al.,1998);
- *data marts* integrados: a área de armazenamento é composta por vários *data marts* projetados de forma a manter a integração dos dados. A interligação dos dados faz-se através do metadados e pelo *software* gerenciador de banco de dados (Kimball, 1998b) e (Chuck, Dirk et al.,1998).

Na área de apresentação, a principal diferença encontrada na literatura se dá quanto à forma e ao local onde as requisições das aplicações clientes são processadas. As formas mais comuns são:

- conexão direta: as aplicações clientes conectam-se diretamente ao *data warehouse* para realizar as consultas e a lógica encontra-se nas próprias aplicações clientes apoiadas ou não por um metadados mantidos externamente (Tanler, 1998) e (Kimball, 1998b);
- conexão por servidor: as aplicações clientes submetem requisições de dados ao servidor (OLAP, WEB, servidor de aplicações, entre outros), no qual

reside a lógica do processamento. O servidor então vale-se do metadados para processar a consulta para em seguida enviar o resultado à aplicação cliente requisitante. No caso do servidor OLAP, as requisições de dados são processadas junto ao cubo do servidor ou junto ao *data warehouse* em caso de necessidade (Tanler, 1998) e (Kimball, 1998b).

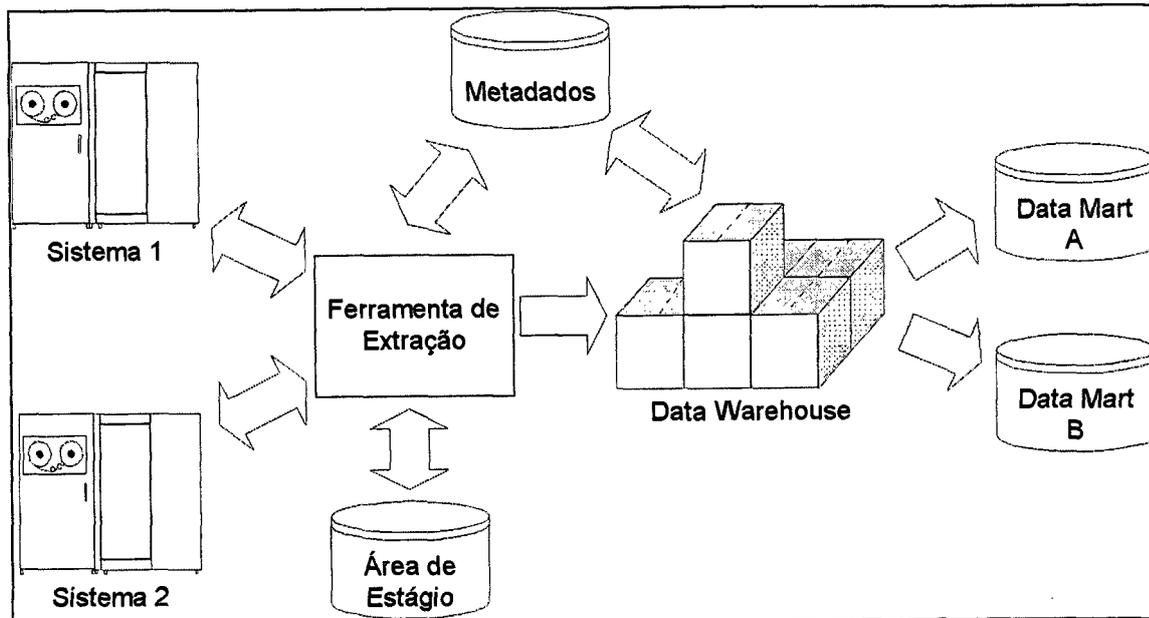
4.2.1 Modelos de Arquiteturas

Várias são as arquiteturas descritas na literatura e propostas por empresas de consultoria especializada. As principais diferenças entre as arquiteturas baseiam-se na forma de organização da área de armazenamento e da área de apresentação, conforme apresentado anteriormente. Entre as principais arquiteturas figuram a *Top-Down*, a *Bottom-Up* e a *BUS*.

A arquitetura *Top-Down* foi introduzida por Inmon (1997) e é caracterizada pela existência de um *data warehouse* centralizado que reúne todos os dados relativos à corporação e uma série de *data marts* derivados do *data warehouse*, conforme ilustra Figura 15. O modelo adotado no *data warehouse* normalmente é o entidade-relacionamento, enquanto que nos *data marts* é o dimensional. A sua área de extração é composta por uma única área de estágio e por um único mecanismo de extração.

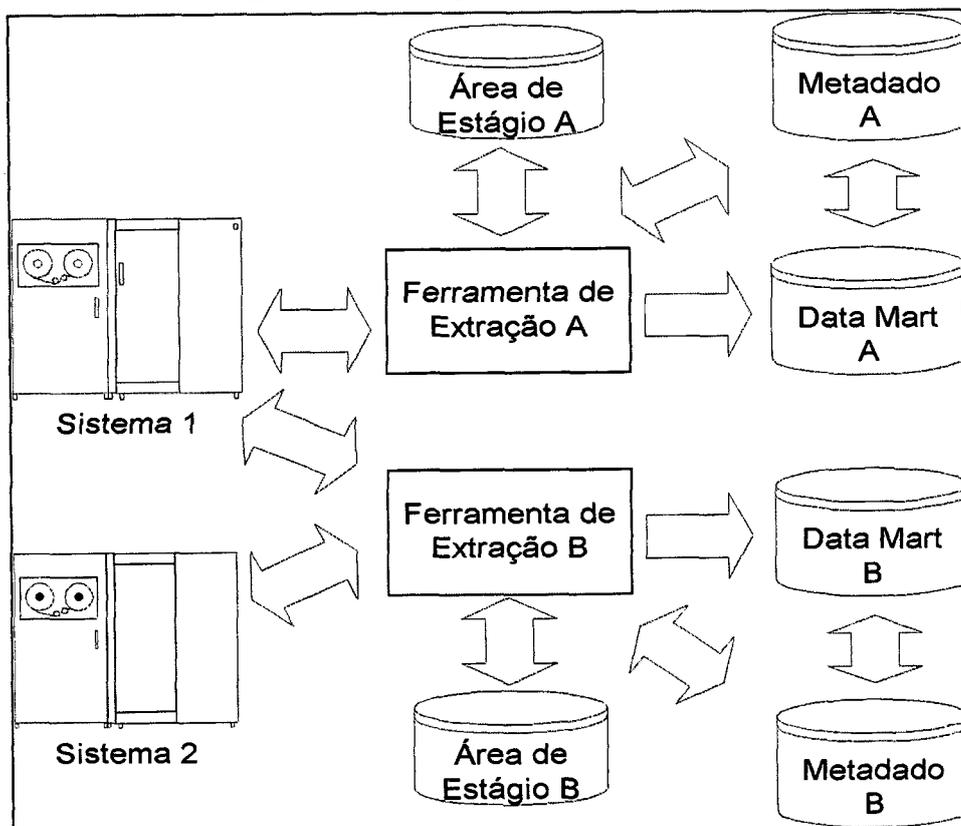
A arquitetura *Bottom-Up* é caracterizada pela estrutura de armazenamento e extração compostas pela criação incremental de vários *data marts* independentes, cada qual relacionado ao seu metadado e à sua área de extração, os quais serão as fontes de dados do *data warehouse*. A sua área de extração é composta por áreas de estágio e mecanismos de extração distintos para cada *data mart*. Esta arquitetura é conhecida por *legamart*, devido à coleção de *data marts* não integrados. A arquitetura é ilustrada na Figura 16 (Vasconcelos, 1999) e (Firestone, 2000).

Figura 15 - A Arquitetura *Top-Down*



Fonte: Adaptado de Firestone (2000)

Figura 16 - A Arquitetura *Bottom-Up*

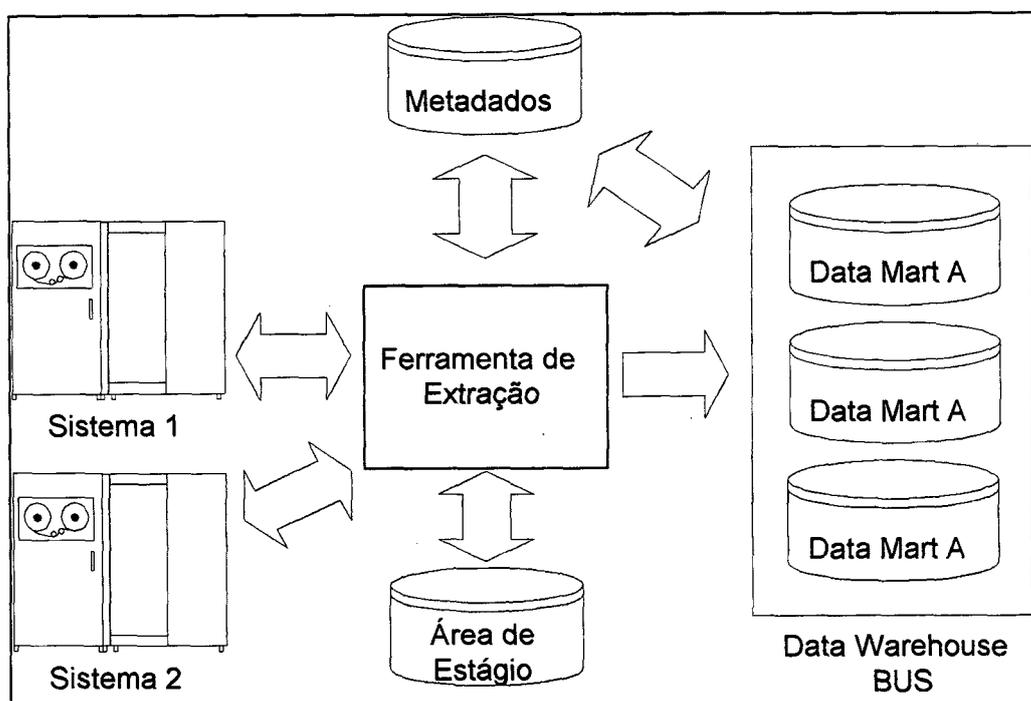


Fonte: Adaptado de Firestone (2000)

A arquitetura *BUS* foi introduzida por Kimball (1998b) e é caracterizada pela estrutura de armazenamento composta por vários *data marts* planejados e integrados através do metadados e de tabelas de fatos e dimensões padronizados. Antes de iniciar a construção do *data warehouse*, são definidos os *data marts* que serão construídos e quais as dimensões e fatos comuns. Cada *data mart* é construído respeitando a pré-estruturação dos fatos e dimensões comuns. O *data warehouse* é composto pela união dos *data marts* coordenado pelo metadados, conforme ilustra a Figura 17. A área de extração é construída também incrementalmente, sendo composta por uma área de estágio e processo de extração única. Algumas variações desta arquitetura são discutidas em Firestone (2000), sendo que as mesmas apresentam áreas de extração independentes e esquema de metadados diferentes ao proposto pela arquitetura *BUS*.

Existem ainda variações das arquiteturas citadas, relacionando o *ODS* (*Operational Data Store*, ou repositório de integração dos dados operacionais) na área de extração, ou organizações de metadados e área de armazenamento com poucas diferenças em relação às apresentadas (Chuck, Dirk et al.,1998), (Firestone, 2000), (Tanler, 1998) e (Kimball, 1998b).

Figura 17 - A Arquitetura *BUS*



Fonte: Adaptado de Firestone (2000)

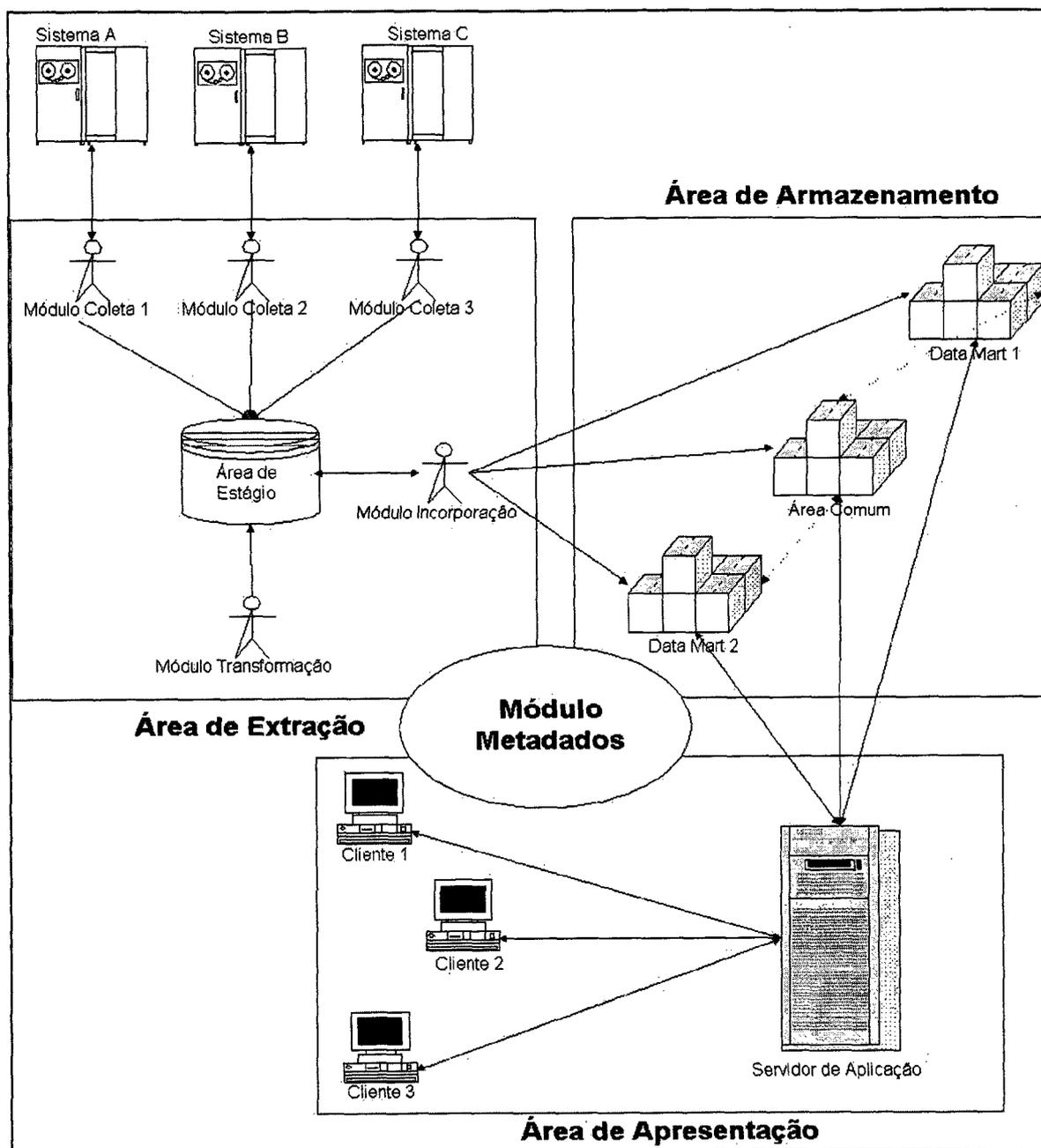
4.3 A ARQUITETURA PROPOSTA

Conforme apresentado na introdução, o principal fator para o fracasso de muitos sistemas é o enfoque orientado a “projeto”. Com a utilização de uma metodologia de desenvolvimento baseada em *data warehouse* orientado a “processo”, torna-se possível o desenvolvimento de sistemas de informações flexíveis, que possam facilmente sofrer ajustes de acordo com as necessidades dos usuários.

Para a implementação de um *data warehouse* na forma de processo, verifica-se que o modelo de *data mart* incremental é o mais indicado, devido ao seu enfoque incremental e integrado. De acordo com o surgimento de novas necessidades, novos *data marts* podem ser implementados ou ajustes podem ser realizados nos existentes. No entanto, de nada adianta utilizar-se da metodologia de desenvolvimento baseada em *data marts* incrementais se a arquitetura do sistema não for flexível o suficiente para incorporar novas fontes de dados e modificações. Desta forma, propõe-se uma distribuição dos componentes da arquitetura em módulos para proporcionar a flexibilidade necessária. A arquitetura proposta é distribuída conforme Figura 18.

De maneira geral, a interação entre os componentes da arquitetura inicia-se quando do início das atividades dos módulos de coleta dos dados operacionais. Os módulos de coleta recolhem os dados necessários para a área de estágio, dentro do servidor de extração. Em seguida, são acionados os módulos de transformação, que para limpar e estruturar os dados da área de estágio, recorrem ao metadados para obter as regras de transformação. Terminado o processo de transformação, serão acionados então os módulos de incorporação que fazem as inserções dos dados transformados nos *data marts*, também se valendo das informações existentes no metadados.

Figura 18 - Componentes da Arquitetura Proposta



Devido à natureza independente das atividades realizadas pelos módulos existentes na área de extração, sugere-se a implementação de cada módulo utilizando o conceito de agentes⁶.

⁶ Agentes são unidades de processamento que carregam um conjunto de operações para o benefício de um usuário ou outro aplicativo com um certo grau de independência ou autonomia, usando com essa finalidade algum tipo de representação das metas ou desejos do usuário (Wooldridge, Jennings, 2000) e (Franklin, 2000).

O acesso às informações contidas nos *data marts* pelos usuários ocorre através de diversas aplicações baseadas em consultas alfanuméricas e gráficas, *data mining* e outras, por intermédio do servidor de aplicações. O servidor de aplicações é o componente responsável por receber os pedidos de dados das aplicações cliente, acessar o metadados para saber em quais *data marts* se encontram as informações, recuperar as informações e, então, enviar a aplicação ao cliente solicitante.

A seguir, serão detalhados os componentes da arquitetura agrupados por área de extração, onde serão ilustrados os componentes do processo de extração; área de armazenamento, onde serão ilustradas as estruturações dos *data marts*; metadados, onde são descritos os conteúdos e gerenciamento do metadados; e finalmente, a área de apresentação, onde se descreve o servidor de aplicação e as aplicações clientes.

4.3.1 Extração

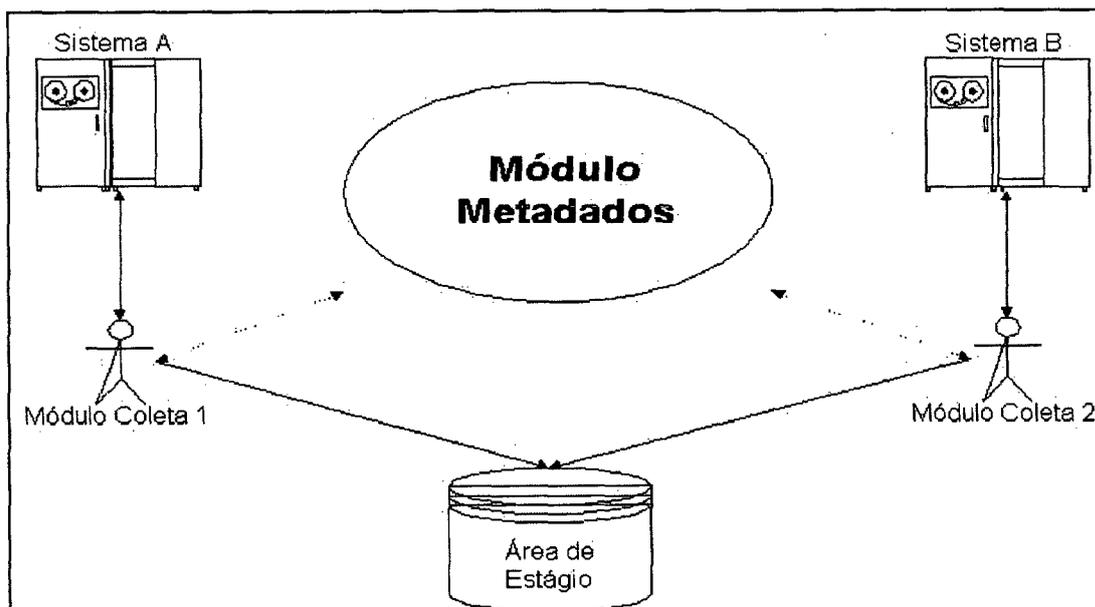
O processo de extração na arquitetura proposta dá-se através da colaboração entre módulos independentes. Cada módulo é uma unidade de programa com um comportamento determinado por parâmetros.

A principal vantagem da utilização de módulos é o isolamento das atividades de extração em processos distintos e bem definidos, o que facilita as manutenções que se façam necessárias e a perfeita adaptação e aproveitamento das características de cada ambiente. A Figura 18 ilustrou a interação dos módulos com os outros componentes no processo de extração.

Os módulos de coleta são responsáveis pelo acesso aos dados necessários nos diversos sistemas operacionais. Possuem uma grande interação com o metadados, pois no metadados encontram-se as informações que relacionam os dados a serem obtidos e sua localização e quais as tabelas e campos destino na área de estágio.

Para cada fonte de dados operacional, existe um módulo de coleta responsável pela busca dos dados necessários, conforme ilustra a Figura 19. Com a utilização de um módulo para cada sistema operacional, o processo de extração torna-se mais flexível de ser mantido, pois qualquer mudança que se faça necessária no acesso a um determinado sistema não afetará aos demais, além de possibilitar uma implementação personalizada de maneira a aproveitar as melhores características de cada ambiente.

Figura 19 – Os Módulos de Coleta



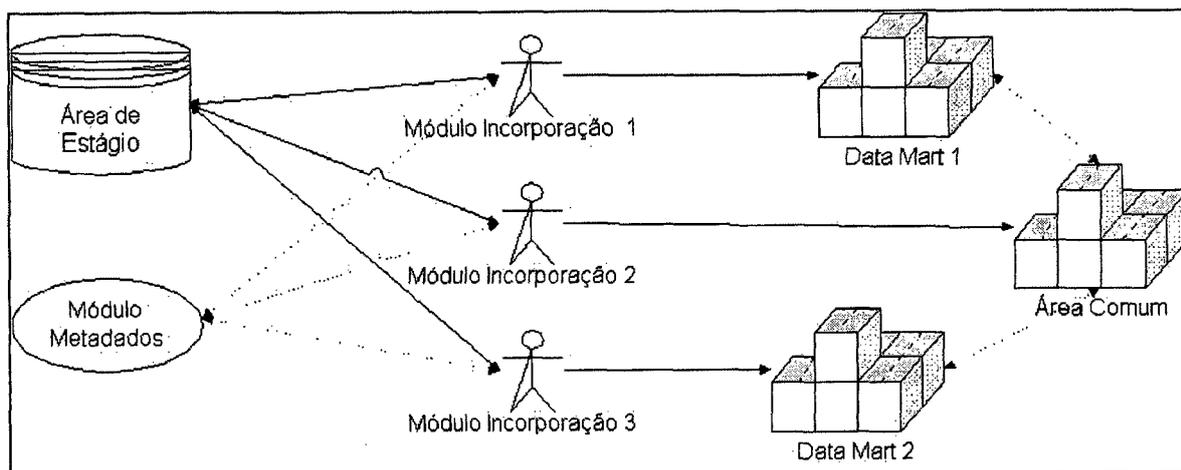
O módulo de transformação é responsável pela limpeza e adequação dos dados da área de estágio de acordo com o modelo dos *data marts*. Assim como os módulos de coleta, o módulo de transformação também possui uma grande interação com o metadados, pois no metadados encontram-se as regras do processo de transformação.

A área de estágio é o local onde serão armazenados os dados carregados pelos módulos de coleta e posteriormente transformados pelos módulos de transformação. É constituída por uma coleção de tabelas organizadas de modo a possibilitar um reflexo dos dados no ambiente operacional e nos *data marts*.

O modelo de dados da área de estágio segue a linha do modelo entidade-relacionamento. O modelo entidade-relacionamento facilitará a incorporação e a transformação, devido à sua estrutura baseada na normalização.

Os módulos de incorporação são responsáveis pela incorporação em cada *data mart* dos dados transformados existentes na área de estágio. Cada módulo de incorporação populará um *data mart* e utilizará as regras existentes no metadados, conforme ilustra a Figura 20, as quais determinam as tabelas e atributos na área de estágio e tabelas e atributos do *data mart* destino, além de campos calculados e de status a serem atualizados.

Figura 20 – Os Módulos de Incorporação



4.3.2 Área de Armazenamento

O módulo de armazenamento da arquitetura segue ao proposto pela metodologia de *data mart* incremental, o qual é baseado em modelos dimensionais mantidos em diversos *data marts*, contendo tabelas de dimensões e de fatos previamente estruturadas para manter a integração entre os dados dos diversos *data marts*.

Para manter a integração entre os *data marts*, a arquitetura vale-se da estruturação dos *data marts* seguindo a idéia da arquitetura *BUS* proposta por Kimball (1998b), onde após um levantamento dos requisitos e das fontes de dados, é elaborada uma matriz para identificação das interseções dos *data marts*. Deste modo, a definição do modelo de dados segue os seguintes passos:

1. listar os *data marts*;
2. listar as dimensões de cada *data mart*;
3. marcar as interseções entre os *data marts*.

No passo 1 é listada uma série de *data marts* que possuam uma única fonte de dados inicialmente. A tendência é que após o levantamento destes, surjam possibilidades de reunir em um só, vários *data marts*, contendo fontes de dados distintas.

No passo 2 são identificadas as possíveis dimensões de cada *data mart* originado do passo 1, para então no passo 3 marcar as interseções entre os *data marts* e Dimensões, formando então uma matriz, como ilustrado na Figura 21.

Figura 21 – Matriz de Data Marts e Dimensões

	Tempo	Local	Doença	Procedimento
Mortalidade	X	X	X	
Internações	X	X	X	
Atendimentos Ambulatoriais	X	X		X

A partir da elaboração da matriz, será possível realizar definições padronizadas das dimensões do modelo. O próximo passo será a construção sequencial de cada *data mart*, seguindo os seguintes procedimentos (Kimball, 1998a) e (Kimball, 1998b):

- escolha do *data mart*;
- definição da granularidade;

- escolha das dimensões;
- definição dos fatos.

No passo 1 é definido o *data mart* a ser construído a partir da matriz. Deve ser considerado para a escolha um *data mart* que represente um negócio de impacto para a corporação.

No passo 2 é declarada a granularidade da tabela de fatos do *data mart* selecionado. Esta representa o registro da tabela de fatos, como por exemplo, o total mensal de atendimentos de um hospital.

No passo 3 são verificadas quais dimensões estão relacionadas à tabela de fatos. Uma boa identificação da granularidade já sugere as dimensões. No exemplo anterior claramente identificam-se as dimensões tempo, hospital e doença.

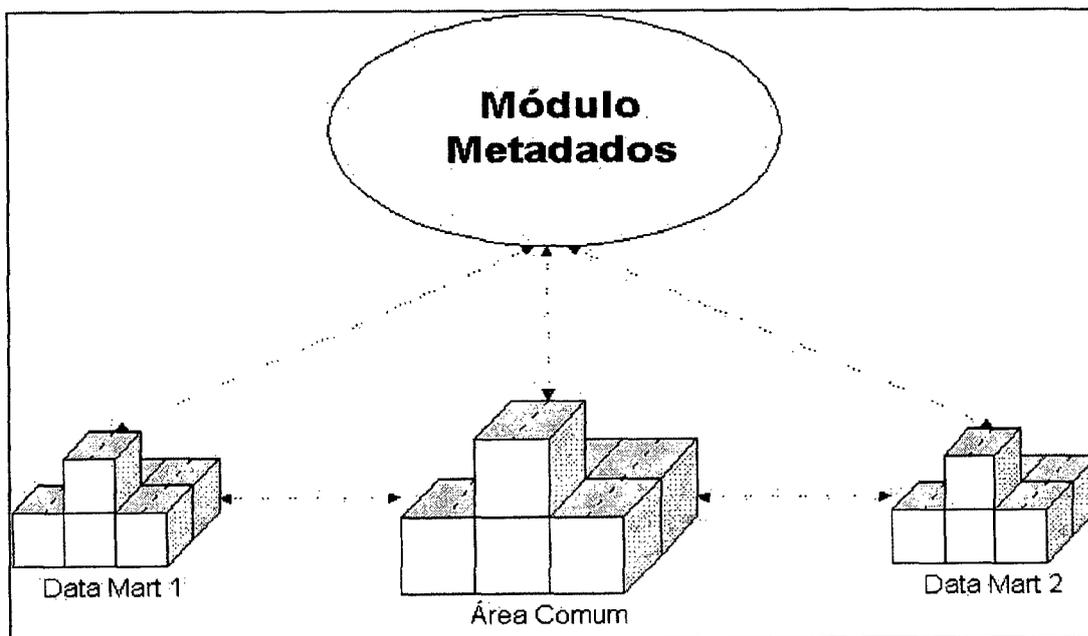
Cada dimensão sugerida deve ser verificada se é compatível com a granularidade definida. Caso não seja, deverá ser verificada a mudança da granularidade ou modificação/exclusão da dimensão.

E por fim, no passo 4 são definidos os fatos que compõem a tabela de fatos. Como nas dimensões, devem ser analisados a compatibilidade de cada fato com a granularidade definida.

Seguindo este modelo de implementação, existe a possibilidade de haver sobreposição de fatos entre as tabelas de fatos dos *data marts*, além de replicação de dimensões.

Para resolver este problema, o modelo desta arquitetura prevê ainda o acréscimo de uma área comum, onde constam as dimensões e os fatos que são utilizados por vários *data marts*, além de agregados que integram e resumizam dados utilizados em conjunto, desde que haja viabilidade técnica. A Figura 22 ilustra a área de armazenamento proposta pela arquitetura.

Figura 22 – Área de Armazenamento



Com a definição da área comum, são reduzidas de forma considerável as replicações de dados entre os *data marts*, o que incidirá em:

- ganho em performance e consistência no processo de incorporação;
- simplificação da gerência dos metadados, devido à integração dos *data marts* e armazenamento dos agregados serem realizados em uma única área;
- simplificação do mapeamento das pesquisas das aplicações clientes.

4.3.3 Metadados

No metadados residem informações variadas que são utilizadas e alimentadas por vários componentes da arquitetura. As informações contidas neste repositório são essenciais para o pleno funcionamento dos componentes e para a sua manutenção.

A estruturação dos dados reunidos no repositório é feita através de regras para o componente de extração, modelos de dados e tabelas descritivas para o componente de armazenamento e tabelas descritivas para as aplicações cliente.

As regras utilizadas para o componente de extração são apresentadas no Quadro 8 e podem ser estruturadas para serem integradas a um sistema especialista⁷, para facilitar a sua manutenção. Além das regras, são mantidas também descrições sobre periodicidade de carga e modelo de dados da área de estágio e dos sistemas operacionais.

Quadro 8 – Regras para Extração Contidas no Metadados

Regra	Definição	Estrutura
Coleta	Define a origem dos dados e local de armazenamento na área de estágio. Podem ser feitas operações simples no momento da coleta.	(BD,Tabela1,Campo1 [, Operador1, BDn,Tabelan,Campon]) -> (Tabela,Campo)
Transformação	Define o código responsável pela limpeza, integração e transformação, entre outros, dos dados existentes na área de estágio. Representada inicialmente por codificação SQL relacionada a uma tabela da área de estágio.	(Tabela1,codificação-da_transformação),(Tabelan,codificação-da-transformação)
Incorporação	Define a origem dos dados da área de estágio e tabelas dos <i>data marts</i> destino.	(Tabela) -> (Data Mart1, Tabela), [(Data Mart2, Tabela)]

⁷ Sistemas especialistas são sistemas computacionais que permitem simular um especialista humano na resolução de problemas. Seu processamento é baseado em uma base de conhecimento levantada junto a especialistas, uma máquina de inferência que é responsável por acionar a base de conhecimento e pela memória de trabalho, onde serão armazenados as conclusões e dados inferidos (Liebowitz, 1988) e (Durkin, 1994).

Os dados mantidos no metadados relativos à área de armazenamento e de apresentação serão apresentados nos Quadros 9 e 10 respectivamente.

Quadro 9 – Metadados da Área de Armazenamento

Metadados da Área de Armazenamento
Dados relativos ao projeto lógico, tais como, modelo de dados dos <i>data marts</i> , da área de controle e dos agregados, bem como histórico de incorporações e manutenções no modelo, principalmente as que dizem respeito à criação de agregados, como motivo e resultados obtidos;
Dados relativos ao projeto físico, tais como esquema de particionamento, índices e outros;
Dados relativos à manutenção, tais como, procedimento de backup, restauração e outros.

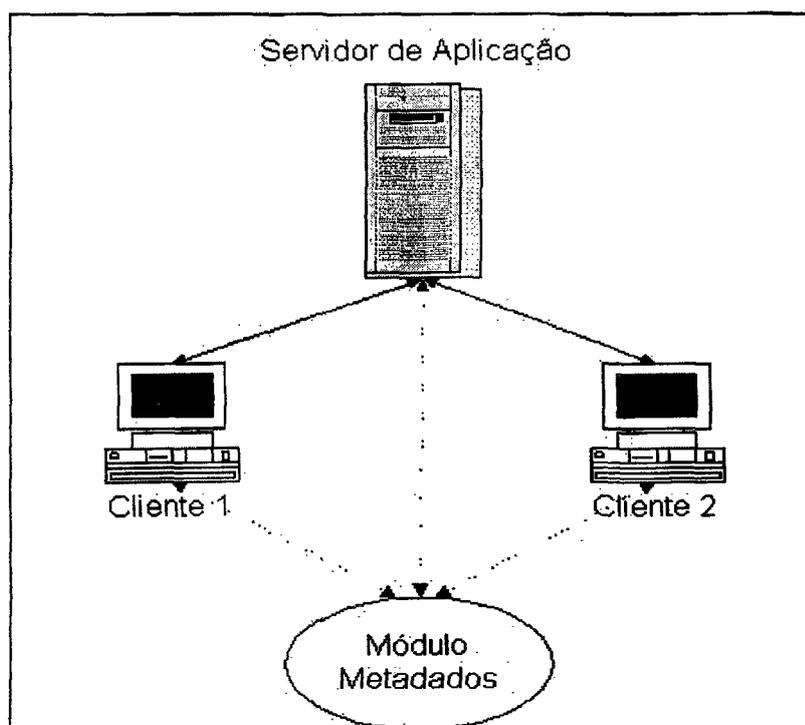
Quadro 10 – Metadados da Área de Apresentação

Metadados da Área de Apresentação
Dados relativos aos usuários, tais como, detalhes de um valor calculado, dados disponíveis e última carga, entre outros;
Os dados mantidos para utilização do re-direcionamento das consultas, tais como, hierarquia nas dimensões, conteúdo das tabelas, tempo de resposta, número de registros, descrição dos agregados existentes, entre outros;
O metadados de apresentação possui uma pequena quantidade de dados que é armazenada na área cliente. Este repositório contém variados parâmetros que serão selecionados na formatação de consultas pelo usuário. A utilização deste metadados será descrita a seguir, na descrição da área de apresentação.

4.3.4 Área de Apresentação

Na área de apresentação figuram os componentes responsáveis pela busca de dados nos *data marts* e disponibilização de informações ao usuário. Os componentes, conforme ilustrado na Figura 23, são constituídos basicamente pelo servidor de aplicação ou servidor OLAP e aplicações clientes.

Figura 23 – Área de Apresentação



O acesso aos dados por parte das aplicações clientes se dá através do servidor de aplicações devido aos seguintes fatores:

- manutenções no modelo de dados dos *data marts* existentes incorrerão em ajustes apenas no servidor de aplicação, quando necessário, ou somente no metadados;
- possibilidade de utilização de mecanismos de re-direcionamento das consultas para agregados;

- possibilidade de apresentação de informações em clientes magros como sites, pois toda a lógica da pesquisa se encontra no servidor de aplicações;
- facilidade e redução do custo de manutenção do conteúdo das aplicações clientes baseadas em consultas pré-formatadas, como relatórios, gráficos e outros baseados em temas, pois a lógica da consulta reside no servidor de aplicações.

O servidor de aplicação conta com um mecanismo de re-direcionamento de consultas baseado no metadados. A idéia originalmente apresentada por Kimball (1998a), consiste em montar o melhor plano de acesso às tabelas dos *data marts* para uma consulta, valendo-se dos agregados existentes e de informações reunidas no metadados descritas anteriormente.

As aplicações clientes poderão ser construídas utilizando qualquer tecnologia ou formato de apresentação, desde que mantenham o esquema de comunicação com a base de dados através do servidor de aplicação.

Para o aproveitamento dos benefícios da arquitetura da área de apresentação, apresenta-se uma nova forma de efetuar-se as consultas, baseando-se em uma camada de metadados na área cliente.

Os dados contidos no metadados da área cliente conterão uma relação das consultas e parâmetros relacionados existentes no servidor de aplicação. Estes dados auxiliarão o usuário a formatar a pesquisa desejada. Seguindo este esquema, uma consulta é processada através dos seguintes passos:

1. o usuário seleciona uma pesquisa. Ao selecionar a pesquisa, todos os parâmetros possíveis da consulta serão buscados do metadados local e relacionados na interface. O usuário então determina os parâmetros e submete a pesquisa;

2. uma mensagem é enviada ao servidor de aplicação contendo um identificador da consulta e conjuntos de parâmetros selecionados;
3. o servidor de aplicação então monta o código SQL a partir da codificação da consulta armazenada no metadados e dos parâmetros recebidos e, então, submete ao re-direcionador de SQL;
4. o SQL resultante do re-direcionador é processado na área de armazenamento, junto aos *data marts*, e o resultado é enviado ao servidor de aplicação;
5. o servidor de aplicação envia o resultado à aplicação cliente solicitante, para a apresentação ao usuário.

A utilização desta metodologia de pesquisa possibilita que qualquer tipo de interface de visualização seja adotada, devido ao isolamento do processamento da consulta. Assim, os resultados podem ser visualizados em poderosas aplicações gráficas, ou ainda servir de entrada de dados para ferramentas de *data mining* ou outras.

4.4 CONCLUSÃO

Na implementação de um *data warehouse*, a escolha da metodologia de desenvolvimento e a definição da arquitetura de implementação constituem tarefas cruciais para o sucesso do projeto e afetam diretamente variáveis como prazo, o retorno do investimento, satisfação do usuário, quantidade de implementação a ser refeita, requisitos de recursos e flexibilidade para adequações (Chuck, Dirk et al., 1998).

Neste capítulo, foram apresentadas algumas metodologias de desenvolvimento. Foram discutidas suas vantagens e desvantagens, além dos seus ciclos de vida. Dentre as metodologias apresentadas, aponta-se que a mais adequada nos tempos atuais é a de desenvolvimento baseado em *data marts* incrementais, por reunir vantagens das metodologias *data warehouse* global e *data marts* isolados.

A arquitetura apresentada neste capítulo foi desenvolvida para aplicação em conjunto com a metodologia de *data marts* incrementais. Cada componente da arquitetura foi projetado para proporcionar máxima flexibilidade de adaptação a mudanças, fator crucial para o sucesso de um projeto de *data warehouse*.

Os principais diferenciais da arquitetura, segundo os seus componentes são:

- o processo de extração dos dados dos sistemas operacionais é baseado na colaboração de módulos. Cada módulo possui uma tarefa bem definida, e qualquer mudança que se faça necessária no comportamento de um módulo não afetará os outros. Além disso, a interação com cada sistema operacional pode ser implementada de maneira personalizada, de modo a obter os maiores benefícios possíveis de cada ambiente. Recomenda-se a implementação dos módulos utilizando-se o conceito de agentes;
- no metadados foi incorporada uma estrutura de regras para o processo de extração. Novas regras podem ser adicionadas e excluídas e modificações nas regras existentes podem ser efetuadas, sem acarretar em mudanças nos módulos de extração;
- na área de armazenamento além dos *data marts*, figura uma área de controle onde residem as tabelas de dimensões comuns aos *data marts*, além dos agregados, o que facilita o processo de extração e a gerência do metadados, por haver menos tabelas a serem carregadas e menor preocupação quanto à incoerência entre tabelas com a mesma definição, além de facilitar o processamento das consultas, devido à boa parte das tabelas e agregados serem mantidas em um único lugar;
- na área de apresentação, adota-se a distribuição do processamento das consultas, seguindo a arquitetura cliente-servidor 3 camadas. Os pedidos de dados provenientes das aplicações clientes serão processadas por um redirecionador de consultas no servidor de aplicação. Com o isolamento das

aplicações de consulta dos *data marts* e a utilização do re-direcionador de consultas, a realização de modificações nos modelos dos *data marts* ou incorporação de novos *data marts* e agregados não incidirá em mudanças nas aplicações de consulta;

- uma nova estrutura para as aplicações clientes foi apresentada. Nesta estrutura é incorporada uma camada de metadados responsável pela relação de consultas e parâmetros relacionados existentes no servidor de aplicação, o que possibilita a implementação de aplicações distribuídas com grande flexibilidade.

No próximo capítulo será apresentado um estudo de caso onde foi empregada a arquitetura apresentada neste capítulo.

5

APLICAÇÃO: SISTEMA DE APOIO À DECISÃO PARA ONCOLOGIA

5.1 INTRODUÇÃO

Segundo levantamento descrito em SBOC (2000), o câncer é a terceira causa de morte no Brasil e é responsável por grande parte do faturamento do SUS, por conta de procedimentos ambulatoriais e internações.

Como ocorre com qualquer outra especialidade, a oncologia não dispõe de informações detalhadas que supram as necessidades de avaliação da situação do câncer no Brasil, tal como eficácia de tratamentos, áreas de incidência, entre outros.

Neste capítulo será apresentado como a arquitetura discutida no capítulo anterior foi aplicada com sucesso na construção de um sistema de informações para a Sociedade Brasileira de Oncologia Clínica (SBOC), de forma a reverter as dificuldades no estudo do câncer no Brasil, existentes até então.

Primeiramente, serão relacionados os aspectos abordados na análise da evolução do câncer e diagnóstico da situação atual, além dos fatores considerados no processo de tomada de decisão do oncologista e outros. Em seguida, serão apresentados detalhes da implementação dos componentes da arquitetura.

5.2 ABORDAGEM DO PROBLEMA

A Sociedade Brasileira de Oncologia Clínica (SBOC) é constituída de profissionais ligados ao tratamento e prevenção de câncer no Brasil. O principal

problema relatado pelos associados da SBOC era a dificuldade em localizar e utilizar dados que suprissem a necessidade de informações analíticas.

A partir de uma identificação inicial dos requisitos dos associados, foi dado início a uma investigação de possíveis fontes de dados para a concepção de um sistema de informações.

Nesta investigação, identificaram-se as informações importantes que poderiam ser obtidas através dos dados do faturamento do SUS (Sistema Unificado de Saúde), relativos às internações hospitalares, procedimentos ambulatoriais, fichas de óbito e outros documentos. Estes dados são gerados por todos os hospitais e ambulatórios credenciados a atender a população através do SUS.

Apesar de existir uma grande diversidade de dados atrelados ao faturamento de serviços junto ao SUS e de estimativas realizadas por entidades oficiais, o acesso a estes dados de forma rápida e lógica não é possível devido à grande quantidade e diversidade de informações e às dificuldades de agregá-las de acordo com as necessidades.

Os requisitos dos associados foram então levantados de forma detalhada, através de questionários apresentados em sucessivas reuniões, onde participavam especialistas da área médica e analistas do corpo técnico da equipe.

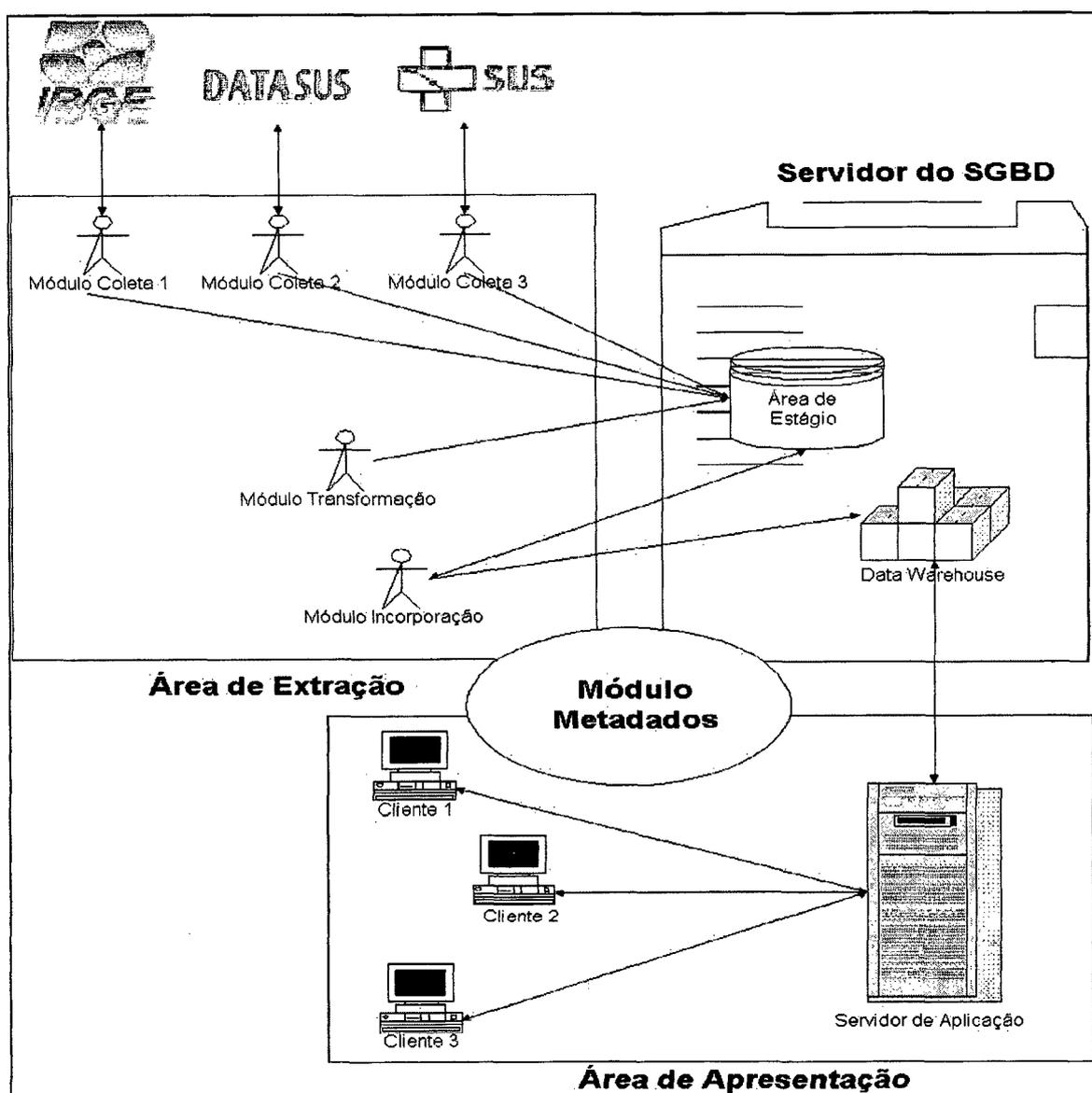
Com o objetivo de atender as necessidades da SBOC e de pesquisadores, foi construído um ambiente para a disponibilização de informações analíticas relacionadas ao câncer no Brasil. Denominado *Câncer On-Line*, o ambiente integra os dados publicados pelo SUS, relativos ao Sistema de Informação sobre a Mortalidade (SIM), Movimento de Atendimento Ambulatorial (SIA-SUS), Autorização de Procedimentos de Alta Complexidade/Custo (APAC) e Movimento de Autorização de Internação Hospitalar (SIH-SUS), além dos dados obtidos junto ao SINASC e IBGE.

A seguir, serão apresentados os detalhes envolvidos no desenvolvimento do ambiente proposto, analisando as questões relacionadas com a arquitetura do sistema.

5.3 A ARQUITETURA DO SISTEMA

A arquitetura do sistema Câncer On-Line foi definida segundo o modelo apresentado no capítulo anterior. Desta forma, o sistema foi concebido a partir de módulos, conforme ilustra a Figura 24, cada qual construído de forma a permitir a realização de mudanças que se façam necessárias sem causar impacto sobre a toda a estrutura.

Figura 24 – A Arquitetura Geral do Sistema



O sistema possui um módulo responsável pela extração dos dados relativos ao domínio da oncologia, um módulo responsável pelo armazenamento do *data warehouse*, uma área de apresentação composta pelo servidor de aplicação e pelo aplicativo de consultas, e um módulo responsável pelo gerenciamento do metadados.

A seguir, serão apresentados detalhes da concepção e integração dos componentes da arquitetura.

5.4 A EXTRAÇÃO DOS DADOS PARA O AMBIENTE

A principal fonte de dados do sistema é composta por dados do faturamento do SUS (Sistema Unificado de Saúde). Informações muito importantes podem ser obtidas através do registro de internações hospitalares, procedimentos ambulatoriais, boletins de óbito e outros documentos.

Estes dados são enviados por todos os hospitais e ambulatórios credenciados para a entidade responsável pelo faturamento do SUS, a Datasus. Parte destes dados é reunida e disponibilizada mensalmente através da Internet, BBS, e CDs, em arquivos separados por estado.

As outras fontes de dados utilizadas pelo sistema são o IBGE e SINASC. Ambos disponibilizam mecanismos de consultas na Internet. Os dados obtidos junto ao IBGE são relativos a censos, relacionados ao número de habitantes de cada município, dado que é utilizado em consultas que cruzam incidência com o número de habitantes. Quanto ao SINASC, são obtidas estatísticas de crianças nascidas vivas e mortalidade infantil, dados que são utilizados em consultas relacionadas à mortalidade infantil.

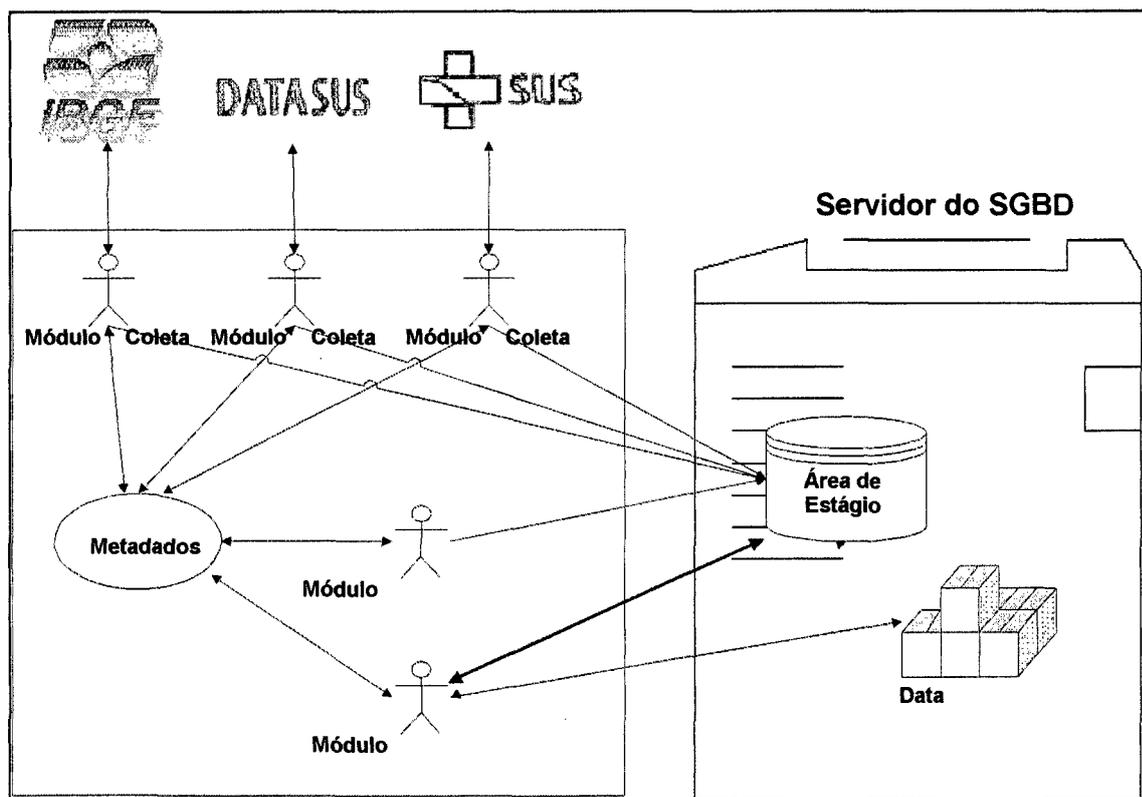
Como mencionado anteriormente, o principal objetivo do módulo de extração é obter, tratar e incorporar os dados operacionais ao *data warehouse*. O módulo de extração foi concebido de acordo com a arquitetura proposta, dividindo-se as principais tarefas em módulos, como ilustra a Figura 25.

Com a utilização de módulos independentes de coleta de dados, foi possível desenvolver aplicações específicas para cada fonte de dados. Desta forma, a equipe de desenvolvimento pôde projetar os aplicativos aproveitando as características de cada ambiente, o que incidiu em processos de coleta mais rápidos e seguros.

Para a coleta dos dados do IBGE e de alguns dados do SUS, foram construídos aplicativos com a linguagem *Inprise Delphi®*, que utilizam o metadados para a designação da origem, na área de coleta, e do destino na área de estágio.

Para a maior parte dos dados coletados junto a Datasus, utilizou-se o aplicativo *Inprise Database Desktop®* e do aplicativo *SQL Loader®* da *Oracle®*, devido à performance superior aos programas originalmente construídos em *Delphi®*. Além da performance tais aplicativos proporcionam uma melhor integração com o metadados de coleta.

Figura 25 – Iteração dos Componentes de Extração do Sistema



O módulo de transformação é responsável pela limpeza e adequação dos dados da área de estágio de acordo com o modelo dos *data marts*. A transformação foi implementada através de chamadas aos metadados de transformação, o qual é constituído basicamente de um catálogo de *stored-procedures* construídas com a linguagem *PL-SQL* da *Oracle*, solução que se mostrou bastante eficiente em termos de performance e flexibilidade.

A área de estágio é o local onde serão armazenados os dados coletados e posteriormente transformados. O modelo de dados da área de estágio segue a linha do modelo entidade-relacionamento, devido à sua estrutura baseada na normalização, evitando a redundância de dados.

A área de estágio foi construída sobre o banco de dados *Oracle 8i*, devido à robustez deste gerenciador de banco de dados. O metadados da área de estágio é constituído pela documentação completa do modelo, mantida através da ferramenta *Case Er-Win*. Até o momento, a área de estágio permanece no mesmo servidor onde se encontra o *data warehouse*. A previsão é que a área de estágio seja alocada em outro servidor para não influenciar na performance das consultas sobre o *data warehouse*.

Os módulos de incorporação previstos na arquitetura apresentada são responsáveis pela incorporação em cada *data mart* dos dados transformados existentes na área de estágio. O módulo de incorporação implementado vale-se das regras existentes no metadados, as quais foram implementadas também na linguagem *PL-SQL* da *Oracle*.

Uma das grandes vantagens da implementação da área de extração deste sistema seguindo a arquitetura proposta, foi a flexibilidade obtida para que adequações sejam feitas sem influenciar o conjunto de operações realizadas. A principal razão disto é a divisão do processo pelos módulos citados e pela integração de cada qual com o metadados.

5.5 O ARMAZENAMENTO

O repositório de dados constitui o núcleo do ambiente do *data warehouse*. Nele estarão representados todos os dados extraídos dos sistemas operacionais, necessários para o processo de tomada de decisão.

Existem duas fases no projeto do repositório de dados: o projeto lógico e o projeto físico. O projeto lógico tem por objetivo modelar os aspectos relacionados ao contexto do empreendimento. O projeto físico parte das informações provenientes do projeto lógico e faz considerações sobre a implementação física do modelo, considerando uma tecnologia (SGBD) em particular.

Os requisitos analisados junto a SBOC possibilitaram o detalhamento preciso das informações relevantes e, em consequência, o desenvolvimento de um projeto lógico conciso e coerente. A partir do cruzamento de informações do modelo lógico é possível suprir as informações que poderão dar suporte ao processo de tomada de decisão.

Em síntese, o projeto dimensional lógico do ambiente *Câncer On-Line* modela as informações relacionadas no Quadro 11.

O *data warehouse* foi construído de forma incremental, seguindo a metodologia orientada a *data marts* incrementais. Desta forma, a construção do modelo de dados do sistema seguiu os seguintes passos:

- levantamento dos *data marts*;
- definição básica da granularidade das tabelas de fatos;
- levantamento das dimensões de todo o modelo;
- padronização das dimensões de acordo com o método da matriz (interseções entre todas as dimensões e fatos do modelo) (Kimball, 1998b);

- definição lógica e física de *data mart* e ajustes no processo de extração, de forma incremental.

Quadro 11 – Escopo das Informações Contidas no *Data Warehouse*

<i>Data Mart</i>	Conteúdo
Mortalidade	Informações relacionadas à mortalidade por localidade, índices de mortalidade considerando causas (grupos e subgrupos de tipos de câncer), séries históricas de mortalidade considerando faixa etária, entre outras.
Atendimento Ambulatorial	Informações relacionadas ao número de atendimentos ambulatoriais de acordo com o tipo de prestador (se público, privado, filantrópico, etc), índices de gastos ambulatoriais por tipo de prestador, número de procedimentos por localidade, séries históricas relativas com atendimentos ambulatoriais, entre outras.
Internação Hospitalar	Informações relacionadas ao cadastro de hospitais por tipo de prestador, internações por tipo de prestador, internações por causa considerando faixa etária e sexo, tempo médio de permanência das internações, gastos por tipo de prestador considerando causa, entre outras.
Procedimentos de Alta Complexidade e Custo	Informações relacionadas a tratamento radioterápico e quimioterápico, número de pacientes em tratamento, finalidades do tratamento, sobrevivência do paciente segundo o tipo de câncer, motivo do tratamento, estágio, entre outras.

Deste modo, dentre as tabelas de fatos projetadas figuram os fatos de Óbitos, Internações, Procedimentos Ambulatoriais, Procedimentos Ambulatoriais de Alto Custo/Complexidade, entre outras. Entre as dimensões figuram Local, Doença, Procedimentos Ambulatoriais, Ambulatório, Hospital, entre outras.

Devido à utilização de um único repositório de dados, onde figuram todos os *data marts*, foram simplificados o gerenciamento dos metadados do processo de incorporação e do servidor de aplicação, a manutenção do SGBD e a própria manutenção do modelo da área de armazenamento, não havendo necessidades da aplicação da área comum constante na área de armazenamento da arquitetura proposta.

Considerando o grande volume de dados a ser gerenciado no contexto do ambiente desenvolvido, volume este que tem uma relação direta com a técnica de *data*

warehouse utilizada, optou-se pelo SGBD Oracle 8i para o gerenciamento do repositório de dados.

Como visto no capítulo 3, um aspecto crítico dentro do projeto físico de *data warehouse* é a granularidade. A escolha adequada da granularidade deve levar em consideração os requisitos do contexto modelado, o volume de informações a serem gerenciadas e as informações disponíveis no modelo operacional.

Baseado nestes aspectos, na concepção do sistema optou-se por utilizar uma granularidade dual. A medida em que novos dados vão sendo incorporados, vão sendo gerados registros com grão mensal, enquanto que os mais velhos vão sendo sumarizados em registros de grão anual.

O emprego de uma modelagem dimensional no contexto deste sistema se mostrou eficiente, possibilitando a extração rápida de conhecimento adequada aos anseios da instituição, no que se refere ao processo de tomada de decisões, e principalmente uma grande flexibilidade nas mudanças do modelo.

5.6 A ÁREA DE APRESENTAÇÃO

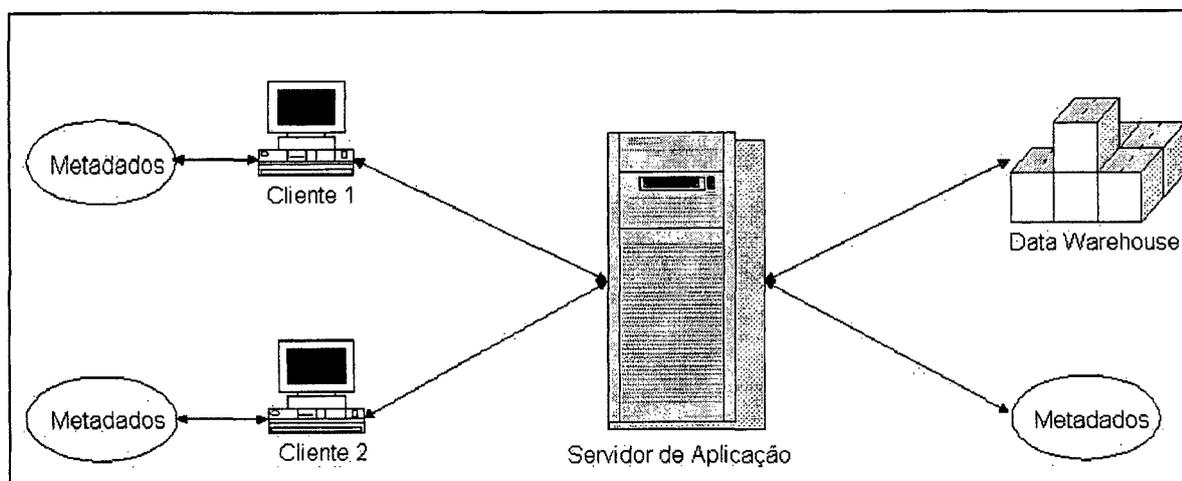
O principal objetivo de um *data warehouse* é fornecer informações analíticas que possam auxiliar nas decisões estratégicas. Para alcançar este objetivo são necessárias ferramentas que auxiliem os usuários a interagirem com o ambiente do *data warehouse* projetado.

O conjunto de ferramentas para geração de consultas, análises estatísticas e *data mining* deve ser cuidadosamente projetado ajustando-se às tendências e características dos profissionais do domínio considerado. Seguindo esta orientação, realizou-se uma análise junto aos profissionais da Sociedade Brasileira de Oncologia Clínica, e então se desenvolveu uma interface gráfica que possibilita a disponibilização de informações concentradas na área de armazenamento.

A interface disponibiliza aos usuários um conjunto de consultas agrupadas em: Mortalidade, Atendimentos Ambulatoriais, Procedimentos de Alta Complexidade/Custo e Internações Hospitalares.

Todas as consultas possuem sua implementação catalogadas no metadados do servidor de aplicação e uma referência no metadados da aplicação cliente, conforme descrito na apresentação da arquitetura proposta. Cada consulta é configurável a partir de um conjunto de filtros associados, também armazenados no metadados da aplicação. A Figura 26 ilustra a estrutura da área de apresentação do sistema.

Figura 26 – Área de Apresentação do Sistema



A aplicação cliente foi concebida para que, ao se formular uma consulta, envie ao servidor de aplicação através da internet, o identificador da consulta e os filtros selecionados. A consulta é então processada junto ao *data warehouse*, a partir do metadados contendo a implementação da consulta e após ser avaliada pelo re-direcionador de consultas. O re-direcionamento foi implementado através de mecanismos existentes no SGBD adotado.

A aplicação cliente e o servidor de aplicação foram implementados com *Inprise Delphi 4.0*. A decisão que sustentou a escolha da ferramenta foi fundamentada nas funcionalidades para a implementação de código cliente-servidor e performance, entre outros.

Um aspecto desejável no desenvolvimento de *software* e que auxilia em eventuais expansões necessárias é o conceito de independência lógica. A independência lógica possibilita a inclusão de novas funções em um sistema sem a necessidade de alterações na interface (a interface é projetada prevendo eventuais expansões).

A aplicação cliente do sistema foi implementada observando-se este conceito. Novas consultas podem ser adicionadas ao sistema sem a necessidade de ajustes na interface gráfica desenvolvida, bastando-se registrar no metadados a implementação da nova consulta e associar-se os filtros relacionados.

O processo de tomada de decisão geralmente envolve a análise de informações sumarizadas, as quais propiciam uma visão geral do negócio. A medida em que possíveis problemas são detectados (considerando as informações sumarizadas), é desejável uma análise mais detalhada da situação, sendo necessário a recuperação de dados cujo nível de detalhe é maior.

A aplicação cliente implementa a navegação entre diferentes níveis de detalhe, através da definição dos filtros da consulta, por exemplo, de estado para município, ou mesmo utilização de dados anuais a mensais.

Além de apresentar os resultados de forma tabular, foram disponibilizadas na aplicação ferramentas de análise estatística, visualização de gráficos, séries históricas e visualização de dados em mapas, entre outras. Permite-se ainda a exportação de dados para outras ferramentas como o *Microsoft Excel*. No anexo B seguem telas do sistema retratando a definição do nível de detalhe e visualização da consulta de número de interações segundo causa em tabela, gráfico e mapa.

5.7 CONCLUSÃO

Neste capítulo, apresentou-se os fatores considerados na interpretação da situação do Câncer no Brasil, bem como a problemática na realização deste processo devido à dificuldade em obter-se informações.

A disponibilização de informações que permitam uma análise sobre a situação do câncer no país é de fundamental importância para a SBOC, auxiliando as decisões que envolvem distribuição de recursos, acompanhamento da evolução dos diversos tipos de câncer, entre outros.

Neste contexto o emprego de um *data warehouse* se mostrou a solução adequada. A partir de uma base de dados disponibilizada com informações relacionadas aos atendimentos efetuados na especialidade de câncer, pôde-se agrupar os dados considerando o aspecto temporal necessário e outros dados disponibilizados por fontes externas (como o IBGE), o que possibilitou a extração de informações analíticas indispensáveis ao processo de tomada de decisão.

A aplicação da arquitetura proposta se mostrou bastante eficiente. No decorrer do projeto, mudanças nos requisitos dos usuários, nas fontes de dados e na estrutura física foram constantes. Tais mudanças foram assimiladas com facilidade, pois eram absorvidas com ajustes isolados nos componentes da aplicação, sem afetar o conjunto.

De forma geral, a arquitetura proposta foi aplicada da seguinte forma no sistema:

- seguindo a filosofia de desenvolvimento de vários módulos de coleta independentes, foram implementadas aplicações e utilizados aplicativos que melhor se adaptavam ao ambiente de cada fonte de dados. Desta forma, se obteve uma melhoria em termos de performance e segurança. Sem este esquema (como é feito normalmente), a implementação da extração perderia em termos de compatibilidade, performance e flexibilidade, devido às limitações imposta por uma única tecnologia;

- todos os módulos que compõem a área de extração (módulos de coleta, módulos de extração e módulos de incorporação), são guiados através do metadados de extração. Desta forma, qualquer modificação que se faça no processo de coleta (novas fontes de dados, mudanças na estrutura das fontes de dados, etc), ou qualquer modificação no processo de transformação (mudança de granularidade, inserção de novas regras ou modificações nas existentes, etc), ou qualquer modificação no processo de inserção (mudança no particionamento dos *data marts*, mudanças na estrutura de dados, etc), é facilmente assimilada, dada a independência dos módulos em relação às regras mencionadas;
- com o isolamento dos módulos, a manutenção dos aplicativos torna-se mais simples e segura, pois todo acerto é feito no módulo correspondente, sem afetar os demais módulos. Durante o desenvolvimento do sistema, esta técnica se mostrou muito eficiente, devido aos constantes ajustes que ocorreram;
- a utilização do metadados como um módulo totalmente independente não foi totalmente adotada no desenvolvimento do sistema. Parte do metadados foi incorporado em alguns módulos para agilizar a performance. O próximo passo será implementar um sistema especialista para manter todo o metadados separado dos módulos, otimizado de forma a oferecer a performance necessária e dar a flexibilidade desejada na manutenção das regras;
- a adoção de uma estrutura de dados baseada em *data marts* integrados e de uma área comum se mostrou bastante eficiente. Desta forma, o *data warehouse* foi construído de forma incremental, permitindo uma maior aproximação do usuário o que facilitou a validação;
- a estruturação do servidor e dos clientes da área de apresentação seguindo a estrutura cliente-servidor e integrados com os metadados se mostrou uma

ótima solução, devido a grande flexibilidade para mudanças, como definição de novas consultas e manutenções nas existentes, ou mudança na estrutura dos dados e no particionamento, entre outros.

6

CONCLUSÃO E TRABALHOS FUTUROS

O objetivo principal deste trabalho é o de apresentar uma arquitetura diferenciada para distribuição dos componentes de um *data warehouse*. Desta forma, foram divididos nos capítulos constantes deste trabalho, os elementos teóricos necessários e a discussão sobre a organização da arquitetura e sua implementação.

No segundo capítulo foram introduzidos conceitos envolvidos em projetos de sistemas de informações, para o qual se destina a arquitetura. No terceiro capítulo, foi apresentada uma breve fundamentação sobre a técnica *data warehousing*, a qual vem sendo empregada com sucesso em projetos de sistemas de informações. No quarto capítulo foi detalhada a arquitetura proposta, seus módulos e relação com os conceitos abordados no segundo e terceiro capítulos. No quinto capítulo foram apresentados os resultados do emprego da arquitetura no projeto de um sistema de apoio à decisão para os oncologistas brasileiros.

Conforme mencionado, um *data warehouse* é indicado quando o projeto de um sistema de informações necessita agregar dados de vários sistemas operacionais. Em diversas situações, a performance e a acurácia das informações obtidas a partir dos sistemas operacionais podem ser comprometidas, devido a fatores, tais como, diversidade de plataformas de sistemas operacionais e sistemas gerenciadores de banco de dados, informações duplicadas, representações diferentes dos mesmos dados, entre outras.

Além disto, a amplitude temporal (informações de meses e anos anteriores) e os dados externos desejados pelos usuários podem não estar disponíveis nos sistemas

operacionais, visto que tais sistemas foram planejados para atender ao processamento das transações do dia a dia, e não para fins analíticos.

Na implementação de um *data warehouse*, a escolha da metodologia de desenvolvimento e a definição da arquitetura de implementação, constituem tarefa de suma importância para o sucesso do projeto. Dentre as metodologias apresentadas no quarto capítulo, aponta-se como mais adequada a baseada em *data marts* incrementais, por reunir vantagens das metodologias baseadas em *data warehouse* global e *data marts* isolados.

A arquitetura proposta foi desenvolvida para aplicação em conjunto com a metodologia de *data marts* incrementais. Cada componente da arquitetura foi projetado para proporcionar máxima flexibilidade de adaptação a mudanças, fator crucial para o sucesso de um projeto de *data warehouse*. Assim sendo, foram criadas áreas destinadas ao tratamento das diversas operações envolvidas em um *data warehouse*. Cada área engloba módulos independentes com tarefas específicas.

A eficácia da arquitetura foi comprovada quando do desenvolvimento de um sistema de informações para os associados da Sociedade Brasileira de Oncologia Clínica, atendendo assim, a demanda de informações dos associados. A utilização da arquitetura permitiu que várias modificações oriundas de mudanças nos requisitos dos usuários fossem implementadas de forma rápida, sem comprometer o funcionamento do sistema.

Como seqüência deste trabalho, pretende-se especificar interfaces *IDL* para cada módulo da área de extração, para sugerir ao órgão padronizador da especificação *CORBA*, *OMG*, um modelo de especificação para componentes da área de extração do *data warehouse*.

Na seqüência do trabalho pretende-se ainda especificar e implementar um sistema especialista para gerenciamento do metadados. Com a utilização de um sistema especialista, com regras definidas para a extração e apresentação dos dados, a

manutenção do metadados se tornaria mais fácil e haveria uma maior flexibilidade para adaptações às mudanças.

Outra possibilidade em estudo é a construção de componentes baseados em técnicas de *data mining* (Berry, Linoff, 1997), para integração na arquitetura.

7

REFERÊNCIAS BIBLIOGRÁFICAS

- ARMSTRONG, R., *Data Warehousing: Clearing the Confusion*. Disponível em:
<<http://www.ncr.com>>. Acesso em 10 jun. 2000.
- BERRY, M. J. A., LINOFF, G. *Data Mining Techniques*. John Wiley & Sons Inc.,
New York, 1997.
- BERSON, A., *Data Warehousing, Data Mining & OLAP*. McGraw-Hill, USA, 1997.
- BORT, J., *The Wiser, Gentler Data Warehouse*. Disponível em:
<http://www.sunworld.com/unixinsideronline/swol-05-1997/swol-05-datawarehouse_p.html>. Acesso em: 26 fev. 2001.
- BRACKETT, M. H. *The Data Warehouse Challenge. Taming Data Chaos*. John
Wiley & Sons Inc., New York, 1996.
- CHUCK, B., DIRK, H., DON, S., et al. *Data Modeling Techniques for Data
Warehousing*. 1998.
- COREY, M. J., ABBEY, M., IAN, A., et al, *Oracle 8 Data Warehousing*.
Osborne/McGraw-Hill, Berkeley, 1998.
- CRUZ, T., *Sistemas de Informações Gerenciais*. Atlas, São Paulo, 1996.
- DATA KNOWLEDGE, *Managing the Data Warehouse Throughout Its Lifecycle*.
Disponível em:
<<http://china3i.freewebsites.com/whitepaper/datawarehouse/WPManagingDW.htm>>.
Acesso em: 21 jan. 2001.

- DEBEVOISE, N. T., *The Data Warehouse Method*. Prentice Hall, New Jersey, 1999.
- DONALD, B., *High Performance Oracle Data Warehousing*. The Coriolis Group, USA, 1997.
- DURKIN, J., *Expert Systems: Design and Development*. Prentice Hall, New Jersey, 1994.
- FELICIANO NETO, A., SHIMIZU, T., *Sistemas Flexíveis de Informações*. Makron Books, São Paulo, 1996.
- FIRESTONE, J. M., *Architectural Evolution in Data Warehousing and Distributed Knowledge Management Architecture*. Disponível em: <<http://www.dkms.com/>>. Acesso em: 20 out. 2000.
- FRANKLIN, S. G., *Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents*. Disponível em: <<http://www.mscl.memphis.edu/~franklin/AgentProg.html>>. Acesso em 7 jan. 2000.
- GIOVINAZZO, W. A., *Object-Oriented Data Warehouse Design - Building a Star Schema*. Prentice Hall, New Jersey, 2000.
- HARRISON, T. H., *Intranet Data Warehouse*. McGraw-Hill, USA, 1997.
- INMON, W. H., *Como Construir o Data Warehouse*. Campus, Rio de Janeiro, 1997.
- KELLY, T. J., *Dimensional Data Modeling*. Disponível em: <<http://www.sybase.com>>. Acesso em: 10 jun. 2000.
- KIMBALL, R., *Data Warehouse Toolkit*. Makron Books, São Paulo, 1998a.
- KIMBALL, R., *The Data Warehouse Lifecycle Toolkit*. John Wiley & Sons Inc., New York, 1998b.
- LAUDON, K. C. & LAUDON, J. P., *Management Information Systems. New Approaches to Organization & Technology*. Prentice Hall, New Jersey, 1998.

- LIEBOWITZ, J., *Introduction to Expert Systems*. Mitchell Publishing Inc., USA, 1988.
- MARTIN, J., *Engenharia da Informação: Introdução*. Campus, Rio de Janeiro, 1991.
- MECHELN, P. J. V., *SALLPI-GI - Sistema de Apoio ao Planejamento no Processo de Tomada de Decisão do Jogo de Empresas GI-EPS*. Florianópolis, 1997.
Dissertação (Mestrado em Engenharia de Produção) – Engenharia de Produção e Sistemas, UFSC.
- NEIL, B., SCHRADER, M., DAKIN, J., *Oracle Data Warehousing Unleashed*. Publishing, Inc., 1997.
- PALMA, S., Os Componentes Funcionais de um Data Warehouse. In: *Developers' Magazine*, Rio de Janeiro: Axcel Books, 1998, n. 18, 66 p.p.18-20.
- SBOC. *Anuário Brasileiro do Câncer*. Florianópolis: Ed. da UFSC, 2000. /no prelo/
- TAIT, T. F. C., *Um Modelo de Arquitetura de Sistemas de Informação para Setor Público: Estudo em Empresas Estatais Prestadoras de Serviços de Informática*. Florianópolis, 2000. Tese (Doutorado em Engenharia de Produção) - Engenharia de Produção e Sistemas, UFSC.
- TANLER, R., *Intranet Data Warehouse*, Infobook, Rio de Janeiro, 1998.
- VASCONCELOS, J. M., Implementando um Data Warehouse Incremental. In: *Developers' Magazine*, Rio de Janeiro: Axcel Books, 1999, n.32, 66 p.p.18-20.
- VERSTRAET, A. A., *Systems Definition*. Disponível em:
<<http://www.smeal.psu.edu/misweb/systems/sycodef.html>>. Acesso em: 10 ago. 2000.
- WOOLDRIDGE, M., JENNINGS, N., *Intelligent Agents: Theory and Practice*. Disponível em: <<http://www.doc.mmu.ac.uk/STAFF/mike/ker95/ker95.html>>. Acesso em: 7 jan. 2000.

ZIMMER, H., *Data Warehousing: Are You on a Path to Success or Failure?*.

Disponível em: <<http://www.dw-institute.com/whatworks10/lessons/lessons.html>>.

Acesso em: 21/01/2001.

ANEXO A

Quadro 12 – Papéis em um Projeto de *Data Warehouse*

Papel	Descrição
Gerente de Projeto	Administração das tarefas, atividades e recursos, gerência dos prazos e comunicação do progresso do projeto
Analista da Aplicação cliente	Definição dos requisitos e representação destes na forma de um modelo dimensional a ser desenvolvido
Modelador de Dados	Análise detalhada e construção do modelo dimensional de dados. Participação secundária na análise de requisitos
Analista de Normas	Buscar definições comuns na organização para dimensões e fatos, a fim de publicar tais definições
Administrador da Base de Dados (DBA)	Traduzir o modelo dimensional em tabelas, definir estratégias de indexação, particionamento, integridade dos dados e tarefas ligadas a manutenção a fim de manter o banco de dados disponível para uso
Projetista da Extração dos Dados	Projeto dos processos de extração, transformação, carga dos dados dos sistemas operacionais no <i>data warehouse</i>
Desenvolvedor da Extração dos Dados	Desenvolvimento dos programas de extração em caso de não adoção de ferramentas automatizadas
Desenvolvedor da Aplicação cliente	Desenvolvimento da aplicação que será disponibilizada ao usuário final
Educador do <i>data warehouse</i>	Treinamento sobre as aplicações dos usuários e conteúdo dos dados
Analista de Controle de Qualidade	Verifica a acurácia dos dados carregados no <i>data warehouse</i> e a integridade da aplicação cliente

ANEXO B

Figura 27 - Tela Principal do Sistema

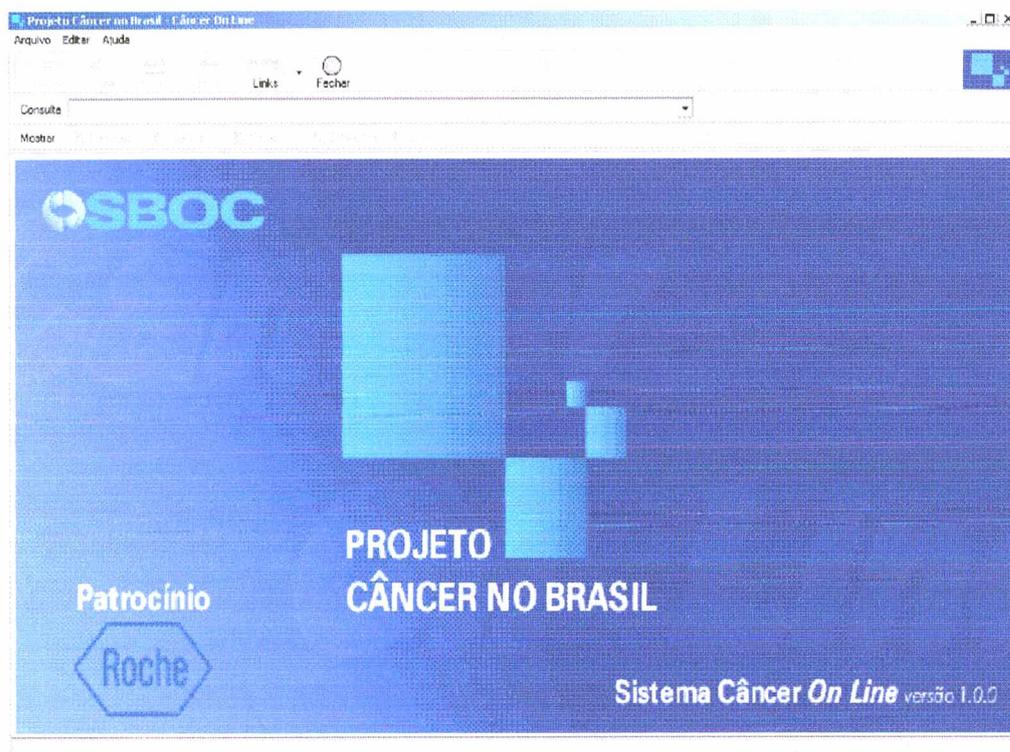


Figura 28 - Seleção de Filtros

Projeto Câncer no Brasil - Câncer On Line

Arquivo Editar Ajuda

Voltar Seguir Início Links Fechar

Consulta: Coeficiente de Mortalidade segundo Causa, Faixa Etária e Sexo

Mostrar: Resultados Gráfico Mapa Estatística Excel

Filtros

- Período
 - Ano
- Localidade
 - Região
 - Estado por Região
 - Estado
 - Município por Estado
 - Município
 - Faixa Etária
 - Sexo
- Causas
 - Grupos de Causas
 - Grupos de Causas
 - Subgrupos de Causas

Conteúdo dos Filtros

Buscar e clicar para: Todos

- ACRE
- ALAGOAS
- AMAPA
- AMAZONAS
- BAHIA
- CEARA
- DISTRITO FEDERAL
- ESPIRITO SANTO
- GOIAS
- MARANHAO
- MATO GROSSO
- MATO GROSSO DO SUL
- MINAS GERAIS
- PAPA
- PARAIBA
- PARANA
- PERNAMBUCO
- PIAUÍ
- RIO DE JANEIRO
- RIO GRANDE DO NORTE
- RIO GRANDE DO SUL
- RODONIA
- RORAIMA
- SANTA CATARINA
- SÃO PAULO
- SERGIPE
- TOCANTINS

Filtros Selecionados

Consulta: Coeficiente de Mortalidade segundo Causa, Faixa Etária e Sexo

Figura 29 - Visualização Através de Tabelas

Projeto Câncer no Brasil - Câncer On Line

Arquivo Editar Ajuda

Voltar Seguir Início Links Fechar

Consulta: Coeficiente de Mortalidade Geral

Mostrar: Resultados Gráfico Mapa Estatística Excel

Localidade **População** **Óbitos** **Óbitos/População**

Localidade	População	Óbitos	Óbitos/População
ACRE	15.294	2272	4,562
ALAGOAS	2657690	14278	5,372
AMAPA	409575	1657	4,136
AMAZONAS	2450927	8285	3,38
BAHIA	12680511	41069	3,230
CEARA	6884718	31160	4,525
DISTRITO FEDERAL	1669644	8405	4,495
ESPIRITO SANTO	2844902	15651	5,571
GOIAS	4621075	23217	5,024
MARANHAO	5282476	13458	2,547
MATO GROSSO	2279944	9158	4,016
MATO GROSSO DO SUL	1958030	11056	5,644
MINAS GERAIS	16664239	97516	5,906
PAPA	5630279	18088	3,212
PARAIBA	3326749	17394	5,228
PARANA	9119746	53473	5,864
PERNAMBUCO	7454361	49532	6,544
PIAUÍ	2691512	7743	2,876
RIO DE JANEIRO	13625067	30651	2,258
RIO GRANDE DO NORTE	2689132	12040	4,552
RIO GRANDE DO SUL	9736865	64312	6,505
RODONIA	1249654	4834	3,888
RORAIMA	253497	926	3,652

Origem dos Dados: Datasus e IBGE.
Os dados em branco representam valores inexistentes em nossa base de dados.

Figura 30 – Visualização Através de Gráficos

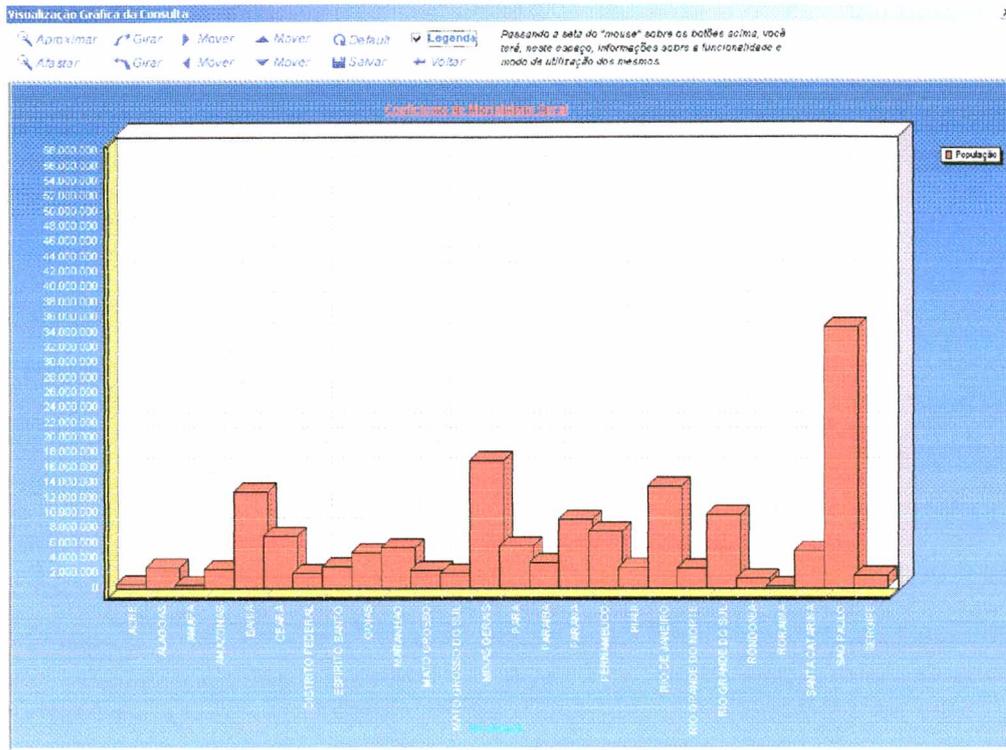
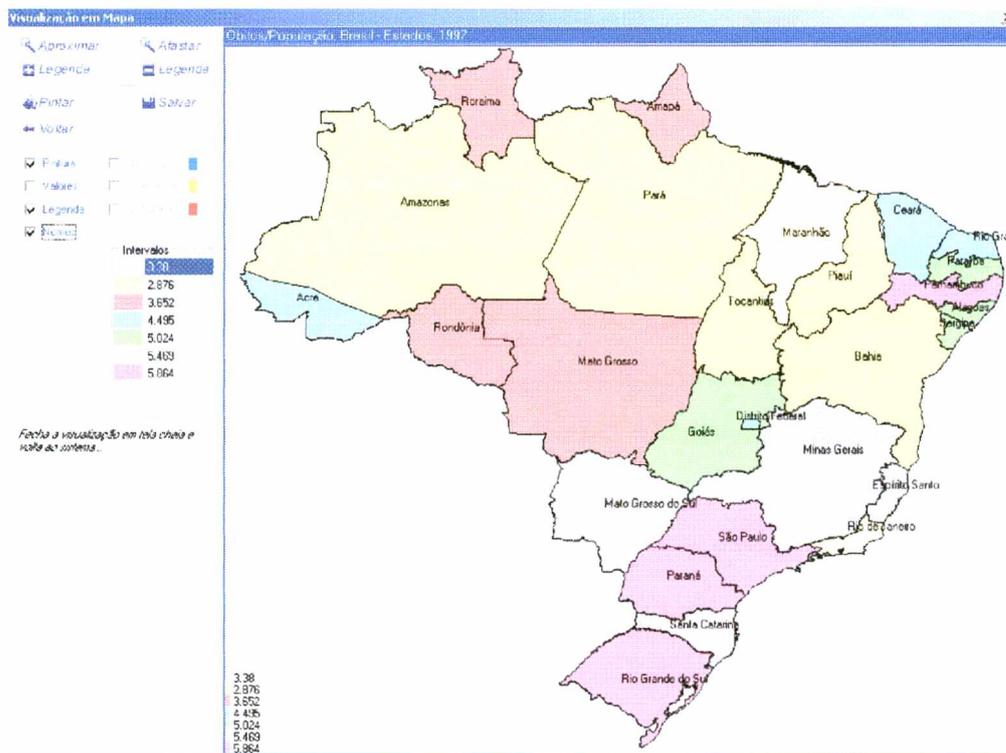


Figura 31 – Visualização Através de Mapas



ANEXO C

Figura 32 - Modelo de Dados do Sistema Câncer On-Line

