

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

**SISTEMA DE INFORMAÇÃO VIA WEB PARA O COMITÊ**  
**PARA DEMOCRATIZAÇÃO DA INFORMÁTICA**

**JAIME STÄHELIN JUNIOR**

**BLUMENAU**  
**2003**

**2003/2-21**

**JAIME STÄHELIN JUNIOR**

**SISTEMA DE INFORMAÇÃO VIA WEB PARA O COMITÊ  
PARA DEMOCRATIZAÇÃO DA INFORMÁTICA**

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

Prof. Wilson Pedro Carli - Orientador

**BLUMENAU  
2003**

**2003/2-21**

**SISTEMA DE INFORMAÇÃO VIA WEB PARA O COMITÊ  
PARA DEMOCRATIZAÇÃO DA INFORMÁTICA**

Por

**JAIME STÄHELIN JUNIOR**

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

Presidente: \_\_\_\_\_  
Prof. Wilson Pedro Carli – Orientador, FURB

Membro: \_\_\_\_\_  
Prof. Paulo Roberto Dias, FURB

Membro: \_\_\_\_\_  
Prof. Everaldo Artur Grahl, FURB

Blumenau, 05 de novembro de 2003.

Dedico este trabalho a meus pais e minhas irmãs por acreditarem e me acompanharem em todos os passos de minha vida. À minha namorada (Vanessa Tatiana Peixer) pela paciência e apoio durante todos os momentos de desenvolvimento deste trabalho.

Nunca desista de seus sonhos, pois se eles se  
forem você continuará vivendo, mas terá  
deixado de existir.

Anônimo

## AGRADECIMENTOS

A Deus, o onipotente e a razão pela qual tive forças para enfrentar todos os obstáculos que nesta caminhada encontrei.

Aos meus pais, Jaime Stähelin e Nilza Gonçalves Stähelin, principais responsáveis por esta conquista. Verdadeiros exemplos de vida que procurei seguir para chegar até aqui.

Ao meu orientador, Wilson Pedro Carli, pela paciência e pelo grande apoio que me deu durante o desenvolvimento do trabalho.

Agradeço à minha namorada, Vanessa Tatiana Peixer, pelo carinho e compreensão nos momentos difíceis que passamos juntos.

A meus familiares por tudo que me ensinaram, em especial, a memória de meu tio Salézio Stähelin. Sem ele esse sonho não teria começado.

A todos os meus colegas de trabalho e de faculdade que contribuíram para o meu crescimento e tornaram este período de faculdade inesquecível.

A Senior Sistemas, pelo apoio e incentivo que tem mostrado em relação aos estudos ao longo de toda a sua trajetória.

Agradeço em especial, ao grande amigo e companheiro de faculdade, Jeanderson Grippa, pelo apoio e lealdade. Um verdadeiro escudeiro que sempre esteve ao meu lado por todos esse anos.

Finalmente, agradeço a todos que de alguma forma contribuíram para a elaboração deste trabalho.

## RESUMO

O presente trabalho descreve o desenvolvimento de um sistema via *Web* para o Comitê para Democratização da Informática (CDI) aplicando as metodologias de desenvolvimento de um *Data Warehouse* segundo Inmon (1997) e utilizando tecnologia ASP com acesso a banco de dados. A intenção de desenvolver este sistema é de integrar as informações de todas as Escolas de Informática e Cidadania (EIC), facilitando o desenvolvimento dos programas educacionais promovidos pelo CDI no Brasil e no exterior.

Palavras chaves: Sistema *Web*; *Data Warehouse*; ASP;

## **ABSTRACT**

The present work describes the development of a Web system for the Committee for Democratization of Informática (CDI) applying the methodologies of development of one Data Warehouse according to Inmon (1997) and using technology ASP with access the data base. The intention to develop this system is to integrate the information of all the Schools of Informatics and Cidadania (SIC), facilitating the development of the educational programs promoted by the CDI in Brazil and the exterior.

Words keys: Web System; Data Warehouse; ASP;



## LISTA DE ILUSTRAÇÕES

|   |    |
|---|----|
| QUADRO 1 – Dados estatísticos sobre o CDI.....  | 14 |
| FIGURA 1 – Dados baseados em assuntos/negócios.....   | 20 |
| FIGURA 2 – Integração do <i>Data Warehouse</i> .....  | 20 |
| FIGURA 3 – Não volatilidade do <i>Data Warehouse</i> .....                                      | 21 |
| FIGURA 4 – Variação em relação ao tempo.....  | 22 |
| FIGURA 5 – Diferentes níveis de detalhes do <i>Data Warehouse</i> .....                         | 22 |
| FIGURA 6 – Processo de construção do <i>Data Warehouse</i> .....                                | 24 |
| FIGURA 7 – Diagrama de uma arquitetura genérica de um <i>Data Warehouse</i> .....               | 25 |
| FIGURA 8 – Granularidade do <i>Data Warehouse</i> .....   | 26 |
| QUADRO 2 – Exemplo de código ASP.....   | 32 |
| FIGURA 9 – Processo que o ASP realiza para enviar uma página <i>Web</i> ao usuário .....        | 32 |
| FIGURA 10 – A hierarquia de acesso de dados.....  | 35 |
| FIGURA 11 – Relacionamento entre os objetos do ADO .....  | 36 |
| QUADRO 4 – Criação de uma conexão a banco de dados utilizando o objeto <i>connection</i> .....  | 37 |
| QUADRO 5 – Exemplo de como criar e fechar uma conexão a banco de dados .....                    | 37 |
| QUADRO 6 – Conexão direta ao banco de dados sem utilizar sistema DSN.....                       | 37 |
| QUADRO 7 – Conjunto de argumentos do método <i>open</i> do objeto <i>recordset</i> .....        | 38 |
| QUADRO 8 – Utilização mais comum do método <i>open</i> .....                                    | 39 |
| QUADRO 9 – Leitura da tabela “Alunos” utilizando os métodos <i>open</i> e <i>MoveNext</i> ..... | 39 |
| QUADRO 10 – Utilização de comando SQL para acesso direto ao banco de dados.....                 | 40 |
| QUADRO 11 – Utilização dos comandos SQL Insert, Update e Delete. ....                           | 40 |
| QUADRO 12 – Utilização dos métodos AddNew, Update e Delete do <i>recordset</i> .....            | 40 |
| QUADRO 13 – Lista de Eventos .....  | 43 |
| FIGURA 12 – Diagrama de Contexto .....  | 43 |
| FIGURA 13 – Modelo Entidade Relacionamento (MER) .....  | 44 |
| FIGURA 14 - DFD da carga de dados .....   | 45 |
| FIGURA 15 - DFD da geração do <i>Data Warehouse</i> .....                                       | 46 |
| FIGURA 16 - DFD da geração de relatórios.....   | 46 |
| QUADRO 14 – Rotina de geração das informações do DW .....                                       | 51 |
| FIGURA 17 – Enterprise Manager do MSDE.....   | 54 |
| FIGURA 18 – Fluxo de navegação do sistema .....   | 55 |
| FIGURA 19 – Tela principal do sistema.....  | 55 |
| FIGURA 20 – Tela de Cadastro de Situações .....   | 56 |
| FIGURA 21 – Acesso ao link de geração do DW .....   | 56 |
| FIGURA 22 – Tela de opções para a geração de gráficos.....                                      | 57 |
| FIGURA 23 – Gráfico estatístico sobre a situação dos alunos.....                                | 57 |
| FIGURA 24 – Outro exemplo de gráfico estatístico .....  | 58 |
| FIGURA 25 – Exemplo de relatório Analítico.....   | 58 |
| QUADRO 15 – Gráfico: geração das barras .....   | 59 |
| QUADRO 16 – Gráfico: geração das informações do eixo horizontal .....                           | 59 |
| QUADRO 17 – Gráfico: geração da legenda .....   | 59 |

## SUMÁRIO

|                                 |           |
|---------------------------------|-----------|
| <b>1 INTRODUÇÃO.....</b>        | <b>11</b> |
| 1.1 OBJETIVOS DO TRABALHO ..... | 11        |

|          |  |                                      |
|----------|--|--------------------------------------|
| 1.2      | ESTRUTURA DO TRABALHO .....                                  | 12                                   |
| <b>2</b> | <b>COMITÊ PARA DEMOCRATIZAÇÃO DA INFORMÁTICA (CDI) .....</b> | <b>13</b>                            |
| 2.1      | HISTÓRICO DO CDI REGIONAL DE BLUMENAU.....                   | 13                                   |
| <b>3</b> | <b>SISTEMAS DE INFORMAÇÕES .....</b>                         | <b>15</b>                            |
| 3.1      | TIPOS DE SISTEMAS DE INFORMAÇÃO .....                        | 16                                   |
| 3.1.1    | SISTEMA DE INFORMAÇÃO GERENCIAL.....                         | 17                                   |
| <b>4</b> | <b>TECNOLOGIAS E FERRAMENTAS .....</b>                       | <b>19</b>                            |
| 4.1      | DATA WAREHOUSE.....  | 19                                   |
| 4.1.1    | CARACTERÍSTICAS .....  | 19                                   |
| 4.1.2    | ESTRUTURA .....  | 22                                   |
| 4.1.3    | ARQUITETURA .....  | 24                                   |
| 4.1.4    | FUNCIONALIDADE .....   | 26                                   |
| 4.1.5    | GRANULARIDADE .....  | 26                                   |
| 4.1.6    | METODOLOGIA DE DESENVOLVIMENTO .....                         | 27                                   |
| 4.1.6.1  | CONSIDERAÇÕES INICIAIS .....                                 | 27                                   |
| 4.1.6.2  | DESENVOLVIMENTO .....  | 27                                   |
| 4.1.6.3  | ERROS NA IMPLEMENTAÇÃO DE UM DATA WAREHOUSE .....            | 29                                   |
| 4.2      | INTERNET.....  | 30                                   |
| 4.3      | ACTIVE SERVER PAGES (ASP) .....                              | 31                                   |
| 4.3.1    | OBJETOS PRÉ-DEFINIDOS DE ASP .....                           | 33                                   |
| 4.3.2    | ASP E BANCO DE DADOS.....                                    | 33                                   |
| 4.3.2.1  | BANCO DE DADOS RELACIONAIS.....                              | 34                                   |
| 4.3.2.2  | COMUNICANDO-SE COM UM BANCO DE DADOS UTILIZANDO ADO .....    | 35                                   |
| 4.3.2.3  | CONECTANDO-SE A UM BANCO DE DADOS .....                      | 36                                   |
| 4.3.2.4  | LENDO E ALTERANDO DADOS DE UM BANCO DE DADOS .....           | 38                                   |
| 4.3.3    | INTERNET INFORMATION SERVER .....                            | 40                                   |
| <b>5</b> | <b>DESENVOLVIMENTO DO TRABALHO .....</b>                     | <b>42</b>                            |
| 5.1      | ESPECIFICAÇÃO .....  | <b>ERRO! INDICADOR NÃO DEFINIDO.</b> |
| 5.1.1    | LISTA DE EVENTOS .....                                       | 42                                   |
| 5.1.2    | DIAGRAMA DE CONTEXTO.....                                    | 43                                   |
| 5.1.3    | MODELO ENTIDADE RELACIONAMENTO (MER).....                    | 43                                   |
| 5.1.4    | DIAGRAMA DE FLUXO DE DADOS (DFD) .....                       | 45                                   |
| 5.1.5    | DICIONÁRIO DE DADOS.....                                     | 47                                   |

|  |           |
|--|-----------|
| 5.2 IMPLEMENTAÇÃO .....                                | 47        |
| 5.2.1 ANÁLISE DO MODELO DE DADOS.....                  | 47        |
| 5.2.2 DIMENSIONAMENTO .....                            | 48        |
| 5.2.3 AVALIAÇÃO TÉCNICA.....                           | 48        |
| 5.2.4 PREPARAÇÃO DO AMBIENTE TÉCNICO .....             | 48        |
| 5.2.5 ANÁLISE DAS ÁREAS DE INTERESSE.....              | 49        |
| 5.2.6 PROJETO DO DATA WAREHOUSE.....                   | 49        |
| 5.2.7 ANÁLISE DO SISTEMA FONTE.....                    | 50        |
| 5.2.8 ESPECIFICAÇÃO DE PROGRAMAS.....                  | 50        |
| 5.2.9 ELABORAÇÃO DE PROGRAMAS .....                    | 50        |
| 5.2.10 POVOAMENTO.....                                 | 51        |
| 5.2.11 UTILIZAÇÃO DOS DADOS.....                       | 51        |
| 5.3 TÉCNICAS E FERRAMENTAS UTILIZADAS .....            | 52        |
| 5.3.1 DREAMWEAVER MX.....                              | 52        |
| 5.3.2 INTERNET EXPLORER .....                          | 52        |
| 5.3.3 BANCO DE DADOS MICROSOFT DATA ENGINE (MSDE)..... | 52        |
| 5.3.4 FERRAMENTA CASE POWER DESIGNER.....              | 54        |
| 5.4 OPERACIONALIDADE DA IMPLEMENTAÇÃO.....             | 54        |
| 5.4.1 APRESENTAÇÃO DO SISTEMA .....                    | 55        |
| <b>6 CONCLUSÕES.....</b>                               | <b>60</b> |
| 6.1 EXTENSÕES .....                                    | 60        |
| <b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>                | <b>61</b> |
| ANEXO A – DICIONÁRIO DE DADOS .....                    | 63        |

## 1 INTRODUÇÃO

Através de suas Escolas de Informática e Cidadania, o Comitê para Democratização da Informática (CDI) implementa programas educacionais no Brasil e no exterior, com o objetivo de mobilizar os segmentos excluídos da sociedade para transformação de sua realidade. Trabalha-se em parceria com comunidades de baixa renda e públicos com necessidades especiais, tais como: deficientes físicos, visuais, usuários psiquiátricos, jovens em situação de rua, presidiários, população indígena, entre outros. O domínio das novas tecnologias não só abre oportunidades de trabalho e de geração de renda, como também possibilita o acesso a fontes de informação e espaços de sociabilidade (CDI, 2003).

Para que o CDI possa ter um acompanhamento melhor dos resultados deste projeto na região de Blumenau, existe a necessidade de um controle dos percentuais de frequência e de aproveitamento escolar, do número de alunos, dos equipamentos disponíveis e da situação dos mesmos. É necessário também obter as informações de ordem administrativas referente aos educadores e estrutura física das Escolas de Informática e Cidadania (EIC's).

Desta forma, visando dar continuidade no Trabalho de Conclusão de Curso (TCC) desenvolvido no primeiro semestre de 2003, pelo acadêmico Jean Carlos Fidelis com o título "Elaboração de um Sistema de Informação para o Comitê para Democratização da Informática" (Fidelis, 2003) este trabalho mostra o desenvolvimento de um Sistema de Informação via *Web* utilizando o conceito de granularidade de um *Data Warehouse* (DW) e tecnologia *Active Server Pages* (ASP). Com base na modelagem de dados do TCC desenvolvido por Fidelis (2003), será implementada a base de dados com informações sobre a estrutura física das EIC's (equipamentos), possibilitando a atualização de todas as informações através da *Internet*, a geração do DW sobre os dados históricos da base operacional e a geração de informações estatísticas.

### 1.1 OBJETIVOS DO TRABALHO

O objetivo do trabalho é criar um DW a fim de gerar uma base dados com todos os dados relevante das EICs para ajudar nas tomadas de decisões. Com o uso da *Internet* e da tecnologia ASP, tais informações serão disponibilizadas para todo o Comitê Regional de Blumenau de modo a centralizar todas as informações inerentes ao mesmo.

Os objetivos específicos do trabalho são:

- a) utilização da tecnologia ASP para o desenvolvimento de um *site* via *Web* para o CDI com acesso a banco de dados;
- b) gerar uma base de dados com todos os dados relevantes das EIC's utilizando a metodologia de desenvolvimento de um DW proposta por Inmon (1997);
- c) disponibilizar a consulta dos dados através de gráficos estatísticos a fim de facilitar o gerenciamento das informações e as tomadas de decisões.

## 1.2 ESTRUTURA DO TRABALHO

O presente trabalho está estruturado em 6 capítulos.

No capítulo 2 é apresentada uma visão geral sobre o Comitê para Democratização da Informática (CDI), suas características, estrutura, objetivos e público alvo.

No capítulo 3 são descritas as bases teóricas sobre SI, definições, alguns conceitos, tipos de SI e outras características.

O capítulo 4 descreve as tecnologias e ferramentas utilizadas. As características de um DW, sua estrutura, arquitetura e funcionalidade, granularidade, toda a metodologia de desenvolvimento, as funções e os erros mais comuns na implementação de um DW. Abrange também assuntos referentes à *Internet* como: *www*, *url*, *http* e *html*. Finalizando este capítulo, é feito um estudo sobre a tecnologia ASP. O que são, seus objetos predefinidos, conexão a banco de dados e manipulação de dados. São descritos também todos os passos para a instalação de um servidor profissional para desenvolvedores *Web*.

O capítulo 5 apresenta o desenvolvimento do trabalho, a especificação, implementação, técnicas e ferramentas utilizadas, operacionalidade e resultados obtidos.

No capítulo 6 são apresentadas as principais conclusões obtidas com o desenvolvimento deste trabalho, os conhecimentos adquiridos e sugestões para novas pesquisas.

## 2 COMITÊ PARA DEMOCRATIZAÇÃO DA INFORMÁTICA (CDI)

O CDI é uma organização não-governamental sem fins lucrativos que, desde 1995, desenvolve o trabalho pioneiro de promover a inclusão social utilizando a tecnologia da informação (TI) como um instrumento para a construção e o exercício da cidadania. O CDI investe na capacidade das próprias comunidades e/ou organizações nelas constituídas de realizar os seus empreendimentos sócio-educacionais. Atualmente, 789 Escolas de Informática e Cidadania (EIC), autônomas e auto-sustentáveis, desenvolvem atividades. Para a formação da EIC, o CDI oferece gratuitamente computadores, softwares, metodologia para aplicação da proposta político-pedagógica baseada em princípios de Paulo Freire, sistematização das estratégias pedagógicas, formação contínua de educadores da própria comunidade e apoio técnico de gestão. Encontros periódicos com os coordenadores e educadores das EICs permitem o acompanhamento de seu desenvolvimento, identificando demandas, dificuldades e favorecendo a troca de experiências (CDI, 2003).

### 2.1 HISTÓRICO DO CDI REGIONAL DE BLUMENAU

Em entrevista com o coordenador regional e com a coordenadora pedagógica do CDI Blumenau, obteve-se as informações do histórico da criação do CDI na cidade de Blumenau.

O CDI Regional de Blumenau foi lançado oficialmente em 28 de setembro de 1999 através do trabalho voluntário do professor da Fundação Universidade Regional de Blumenau (FURB), Wilson Pedro Carli, que desde 1997 vinha exercendo a atividade de coordenação no projeto de Extensão do Departamento de Informática, denominado Escola Móvel de Informática. Este projeto itinerante visava levar a informática para entidades e associações de moradores no município de Blumenau e no Vale do Itajaí, sendo que os computadores e instrutores deslocavam-se durante o período de treinamento. O que era observado no período de acompanhamento e encerramento das atividades em cada local, foi a insatisfação dos moradores quando da saída dos equipamentos, pois não havia uma continuidade dos trabalhos. Os mais carentes, sem computador em seus domicílios, não tinham oportunidade de treinar e aumentar seus conhecimentos.

Desta forma, o professor Wilson Pedro Carli, no final de 1998, procurando outros meios de beneficiar estas comunidades, descobriu o *site* do CDI do Rio de Janeiro, hoje CDI-Matriz. Após este contato inicial, obteve autorização para iniciar o CDI-Blumenau.

Em contato com o presidente da Associação de Moradores do Médio Garcia (AmorMega), associação com uma infra estrutura e diretoria receptiva, começou-se os estudos para a implantação de uma EIC, conforme consta do Plano Político Pedagógico do CDI. Através de contatos com alguns empresários e colaboradores, no final de 1999, o CDI-Blumenau recebeu através de doações, dois micro computadores usados e vários equipamentos (*cpu's*, teclados, *mouses*). Com este material, o trabalho voluntário de algumas pessoas e a compra dos equipamentos faltantes, a primeira EIC do CDI de Blumenau foi inaugurada em março de 2000 na AmorMega. Depois disso, foram inauguradas mais onze EICs, sendo seis em Blumenau e as demais nos municípios de Camboriú, Gaspar, Indaial, Luís Alves e São João Batista.

Desde então, a coordenação regional e a coordenação pedagógica trabalha para manter em funcionamento as 12 EIC's, a maioria delas através de voluntariado. Em 2003, o CDI-Blumenau foi contemplado com convênios do Centro de Integração Empresa Escola (CIEE) para auxiliar na contratação de estagiários, do Serviço Nacional do Comércio (SENAC) para a manutenção de equipamentos, do Instituto Souza Cruz (ISC) para a manutenção da coordenadora pedagógica e sustentabilidade das 5 EIC's nos outros municípios e da *Accenture Foundation* para a contratação de 3 assessores, sustentabilidade do CDI-regional e das 7 EIC's de Blumenau.

Os dados estatísticos atualizados sobre o CDI em 2003 podem ser vistos a seguir no quadro 1.

| Dados estatísticos do CDI                 |           | Dados estatísticos do CDI - Blumenau      |        |
|---|-----------|---|--------|
| Número total de EIC's                     | : 833     | EIC's em Blumenau                         | : 7    |
| EIC's no Brasil                           | : 749     | EIC's em outros municípios                | : 5    |
| EIC's no exterior                         | : 84      | Número de educandos formados em 2003      | : 700  |
| Número de educandos formados em 2003      | : 214.620 | Número de educandos formados desde 1999   | : 1000 |
| Número de educandos formados desde 1995   | : 536.900 | Número de educadores                      | : 30   |
| Número de educadores                      | : 1.666   | Número de computadores alocados nas EIC's | : 60   |
| Número de computadores alocados nas EIC's | : 4.165   | Número de voluntários                     | : 10   |
| Número de voluntários                     | : 1.000   |   |        |

QUADRO 1 – Dados estatísticos sobre o CDI

### 3 SISTEMAS DE INFORMAÇÕES

Os executivos das empresas necessitam de informações para a tomada de decisão, premissa básica para o seu sucesso e o das organizações. Para isto utilizam-se de dados transformados, que podem ser classificados como informação. Informação é uma coleção de dados inter-relacionados, que permite ao executivo tomar decisões (Furlan, 1994).

Segundo Dalfovo (2000), a não utilização das informações como recursos estratégicos leva o executivo a maioria das vezes a administrar impulsivamente ou baseado em modismos. A utilização de um SI pode vir a facilitar o processo decisório com a obtenção de dados, estrategicamente escolhidos e de conteúdos relevantes para qualquer nível e tamanho de empresa.

O SI “é o requisito básico para a decisão automatizada, pois o processo decisório apóia-se na malha de SI na empresa” (Binder, 1994). Ainda, SI é um método organizado de prover informações passadas, presentes e futuras, relacionadas com as operações internas e o serviço de inteligência externa. Serve de suporte para as funções de planejamento, controle e operação de uma empresa através do fornecimento de informações no padrão de tempo apropriado para assistir o tomador de decisão.

Segundo Oliveira (1992), um SI é um tipo especializado de sistema e pode ser definido de inúmeros modos. Um modo é dizer que SI são conjuntos de elementos ou componentes inter-relacionados que coletam (entrada), manipulam e armazenam (processo), disseminam (saída) os dados e informações e fornecem um mecanismo de *feedback*. A entrada é a atividade de captar e reunir novos dados, o processamento envolve a conversão ou transformação dos dados em saídas úteis, e a saída envolve a produção de informação útil. O *feedback* é a saída que é usada para fazer ajustes ou modificações nas atividades de entrada ou processamento.

Segundo Prates (1994), os SI são formados pela combinação estruturada de vários elementos, organizados da melhor maneira possível, visando atingir os objetivos da organização. São integrantes dos sistemas de informações:

- a) informação: dados formatados, textos livres, imagens e sons;
- b) recursos humanos: pessoas que coletam, armazenam, recuperam, processam, disseminam e utilizam as informações;



- c) tecnologias de informação: *hardware* e *software* usados no suporte aos sistemas de informação;
- d) práticas de trabalho: métodos utilizados pelas pessoas no desempenho de suas atividades.

Os executivos recebem tantas informações que se tornam incapazes de processá-las a tempo; e a perda de agilidade nas decisões é causada principalmente pela falta de possibilidade de manipular informações. Os SI foram criados justamente para dar suporte aos executivos na tomada de decisões. Ninguém vive isoladamente; dessa forma, o sistema deve possibilitar a comunicação e a troca de informações entre executivos para a tomada conjunta de decisões (Furlan, 1994).

### 3.1 TIPOS DE SISTEMAS DE INFORMAÇÃO

Segundo Furlan (1994), a área de gerenciamento de SI é bastante abrangente, onde se encontra uma grande quantidade de termos que caracterizam e classificam os diversos tipos de SI. Os Sistemas de Informações se classificam em:

- a) Sistema de Automação de Escritório (SAE): auxilia no processamento de documentos e fornece ferramentas que tornam o trabalho no escritório mais eficiente e eficaz. Definir a forma e o método para a execução de tarefas diárias dificilmente afeta as informações em si. Editores de texto, planilhas de cálculos, *softwares* para correio eletrônico e outros, são exemplos deste tipo de sistema. Usuários com a função de redigir textos, enviar mensagens, criar apresentações, etc., estão diretamente envolvidos com um SAE;
- b) Sistema de Processamento de Transações (SPT): é o sistema que coleta e armazena dados sobre transações e às vezes controlam decisões que são executadas como parte de uma transação. Uma transação é um evento empresarial que pode gerar ou modificar dados armazenados num SI que surgiu e é frequentemente encontrado. No pagamento de uma conta com cartão de crédito, o SPT efetua a transação com a central e valida o cartão. Enfim, o SPT assegura que as informações estão consistentes e disponíveis;
- c) Sistema de Informação Gerencial (SIG): converte os dados coletados num SPT em informação para gerenciar a organização e monitorar o seu desempenho no intuito de efetuar comparações visando suas metas. Os gerentes são os profissionais que

mais utilizam um SIG. Os relatórios de acompanhamento do faturamento de uma empresa emitidos diariamente são exemplos de um SIG;

- d) Sistema de Informação Executiva (SIE): de forma rápida e acessível, fornece informações aos executivos sem forçar os mesmos a pedir ajuda a especialistas em análises de informações. O SIE é utilizado para estruturar o planejamento da organização e o controle de processos, podendo ser utilizado eventualmente para monitorar o desempenho da empresa. Um bom exemplo são os sistemas que fornecem comparativos simples e fáceis de Vendas x Estoque x Produção;
- e) Sistema de Informação Executivo para o Gerenciamento Operacional (SIEGO): fornece aos executivos uma visão das operações regulares da organização, de modo que possam controlar, organizar e planejar mais eficaz e eficientemente. Tem o seu foco na participação e envolvimento de toda organização, desde a alta administração até o piso de fábrica. O principal objetivo do SIEGO é realizar melhorias operacionais em relação a custo, tempo e qualidade;
- f) Sistema de Apoio a Decisão (SAD): auxilia na tomada de decisões, fornecendo informações, padrões ou ferramentas para análise de informações. O SAD pode prover métodos e formatos para porções de um processo de decisão. Analistas, gerentes e outros profissionais são os maiores usuários deste sistema. Uns exemplos de SAD são os sistemas que disponibilizam gráficos 3D para comparativos;
- g) Sistemas Especialistas (SE): auxilia na resolução de problemas em áreas onde o conhecimento de especialistas é necessário. Um exemplo de SE pode ser um sistema onde os médicos dizem os sintomas e são pesquisados em uma base de conhecimentos com possíveis diagnósticos.

Neste trabalho utiliza-se o SIG, pelo fato do sistema *Web* transformar as informações operacionais em gerenciais através do DW.

### 3.1.1 SISTEMA DE INFORMAÇÃO GERENCIAL

SIG é o processo de transformação de dados em informações que são utilizadas na estrutura decisória da empresa, bem como proporcionam a sustentação administrativa para otimizar os resultados esperados. Esse sistema é normalmente composto de diversos subsistemas de natureza conceitual idêntica à daquele que integram, mas com características

específicas quanto à sua finalidade e justificação, quanto ao tipo de tecnologias utilizadas e quanto ao nível dos processos ou natureza das pessoas que envolvem (Oliveira, 1992).

O SIG é desenvolvido para garantir a administração eficiente a qualquer tipo de empresa. É esse sistema que vai garantir que os dados operacionais utilizados para manter a empresa produzindo serão traduzidos em informações passando a todos que tiverem funções executivas (Cruz, 1998).

## 4 TECNOLOGIAS E FERRAMENTAS

A seguir são descritas as tecnologias e ferramentas utilizadas no desenvolvimento do trabalho.

### 4.1 DATA WAREHOUSE

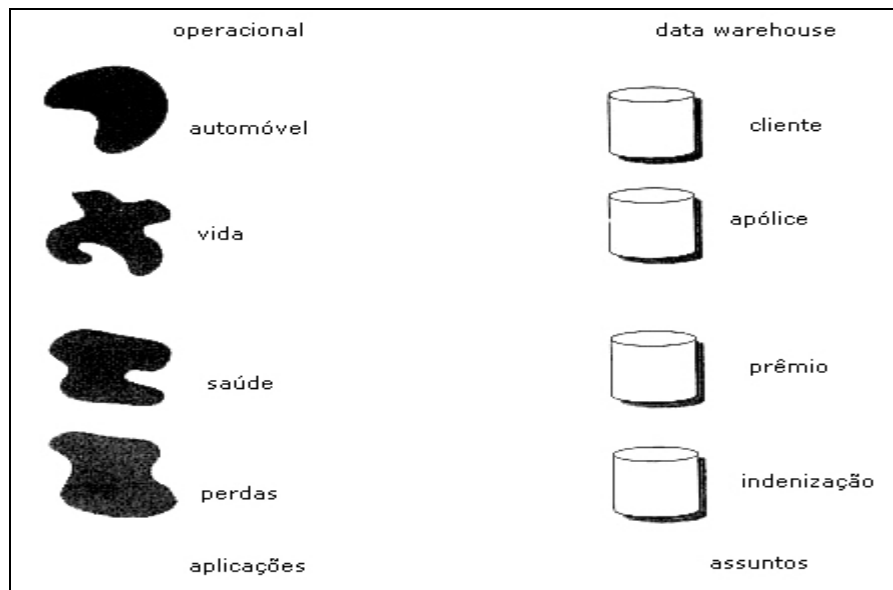
O ambiente de dados para suporte aos processos de gerência e tomada de decisão é fundamentalmente diferente do ambiente convencional de processamento de transações. No coração deste ambiente está a idéia do *Data Warehouse* (DW), integrando e consolidando dados disponíveis em diferentes acervos para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

Nos últimos anos, o conceito de DW evoluiu rapidamente de um considerável conjunto de idéias relacionadas para uma arquitetura voltada para a extração de informação especializada e derivada a partir dos dados operacionais da empresa. O estudo de uma arquitetura descrevendo o ambiente de DW permite compreender melhor a estrutura geral de armazenamento, integração, comunicação, processamento e apresentação dos dados que servirão para subsidiar o processo de tomada de decisão nas empresas.

Inmon (1997) define um DW como sendo um banco de dados especializado, o qual integra e gerencia o fluxo de informações a partir do banco de dados corporativos e fontes de dados externas à empresa. Na construção do DW são filtrados e normalizados os dados de vários bancos de dados dos sistemas transacionais, formando uma base de dados com todos os dados relevantes da empresa ou de uma área específica. Com o cruzamento desses dados extrai-se informações que os sistemas de informações transacionais não conseguem identificar.

#### 4.1.1 CARACTERÍSTICAS

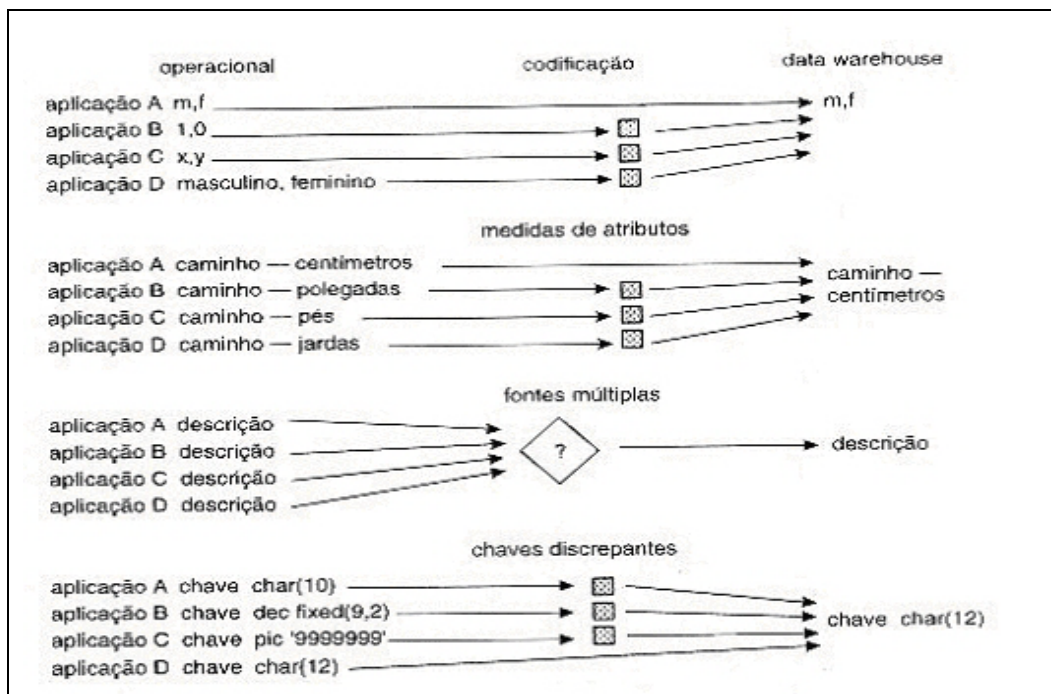
Segundo Inmon (1997), um DW é um conjunto de dados baseado em assuntos, integrado, não-volátil e variável em relação ao tempo, para o apoio às decisões gerenciais. O fato de o DW ser baseado em assuntos/negócios pode ser visto na fig. 1



Fonte: Inmon (1997, p. 34)

FIGURA 1 – Dados baseados em assuntos/negócios

A segunda e mais marcante característica de um DW é o fato de que ele é integrado. A fig. 2 ilustra a integração que ocorre quando os dados passam do ambiente operacional baseado em aplicações para o DW.



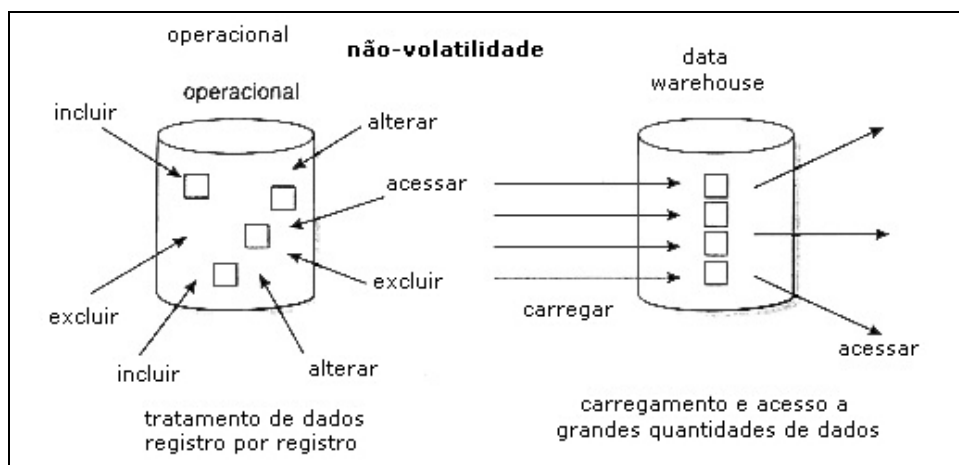
Fonte: Inmon (1997, p. 35)

FIGURA 2 – Integração do *Data Warehouse*

Essa característica contempla todas as distorções e deformidades que os dados podem passar no ambiente transacional e os transforma para um estado único e uniforme antes da

carga definitiva (Baptista, 1998). O exemplo clássico desta anomalia, segundo Inmon (1997), é o caso da identificação do atributo “sexo”, que pode receber as mais diferentes codificações nos sistemas aplicativos (M/F, 1/0, H/M), mas que são convertidas para uma forma única ao passar para o DW.

A não-volatilidade é a terceira característica importante de um DW. A fig. 3 mostra que os dados operacionais são regularmente acessados e tratados um registro por vez. Geralmente a atualização dos dados ocorre no ambiente transacional, enquanto que no ambiente de DW são apenas carregados e acessados.



Fonte: Inmon (1997, p. 36)

FIGURA 3 – Não volatilidade do *Data Warehouse*

A última característica significativa do DW, segundo Inmon (1997), diz respeito ao fato de ele ser variável em relação ao tempo. Os diversos modos, ilustrados na fig. 4, pelos quais a variação em relação ao tempo se manifesta são:

- O horizonte de tempo válido para o DW é significativamente maior do que o dos sistemas operacionais. Um horizonte de tempo de 60 a 90 dias é o normal para os sistemas operacionais, já um horizonte de 5 a 10 anos de dados é o normal para o DW;
- banco de dados operacional possui dados com valor corrente, cuja exatidão é válida no momento do acesso, podendo ser atualizado; já dados do DW não passam de uma série sofisticada de instantâneos, capturados num determinado momento;
- a estrutura chave de dados operacionais pode conter, ou não, elementos de tempo como dia, mês e ano; já a estrutura do DW sempre contém algum elemento de tempo.

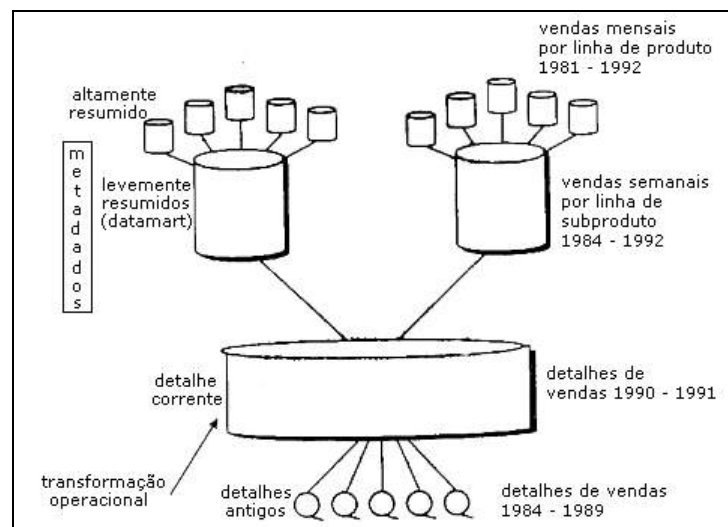


Fonte: Inmon (1997, p. 37)

FIGURA 4 – Variação em relação ao tempo

#### 4.1.2 ESTRUTURA

Segundo Inmon (1997), existem diferentes níveis de detalhes no DW. Há um nível de detalhe mais antigo (geralmente residente em um armazenamento alternativo, de massa), um nível corrente de detalhe, um nível de dados levemente resumidos (o *data-mart*) e um nível de dados altamente resumidos. A fig. 5 mostra estes diferentes níveis de detalhe.



Fonte: Inmon (1997, p. 38)

FIGURA 5 – Diferentes níveis de detalhes do Data Warehouse

Os dados fluem do ambiente transacional para o DW. Na passagem do nível operacional para o nível do DW, geralmente ocorre uma quantidade significativa de transformações sobre os dados.

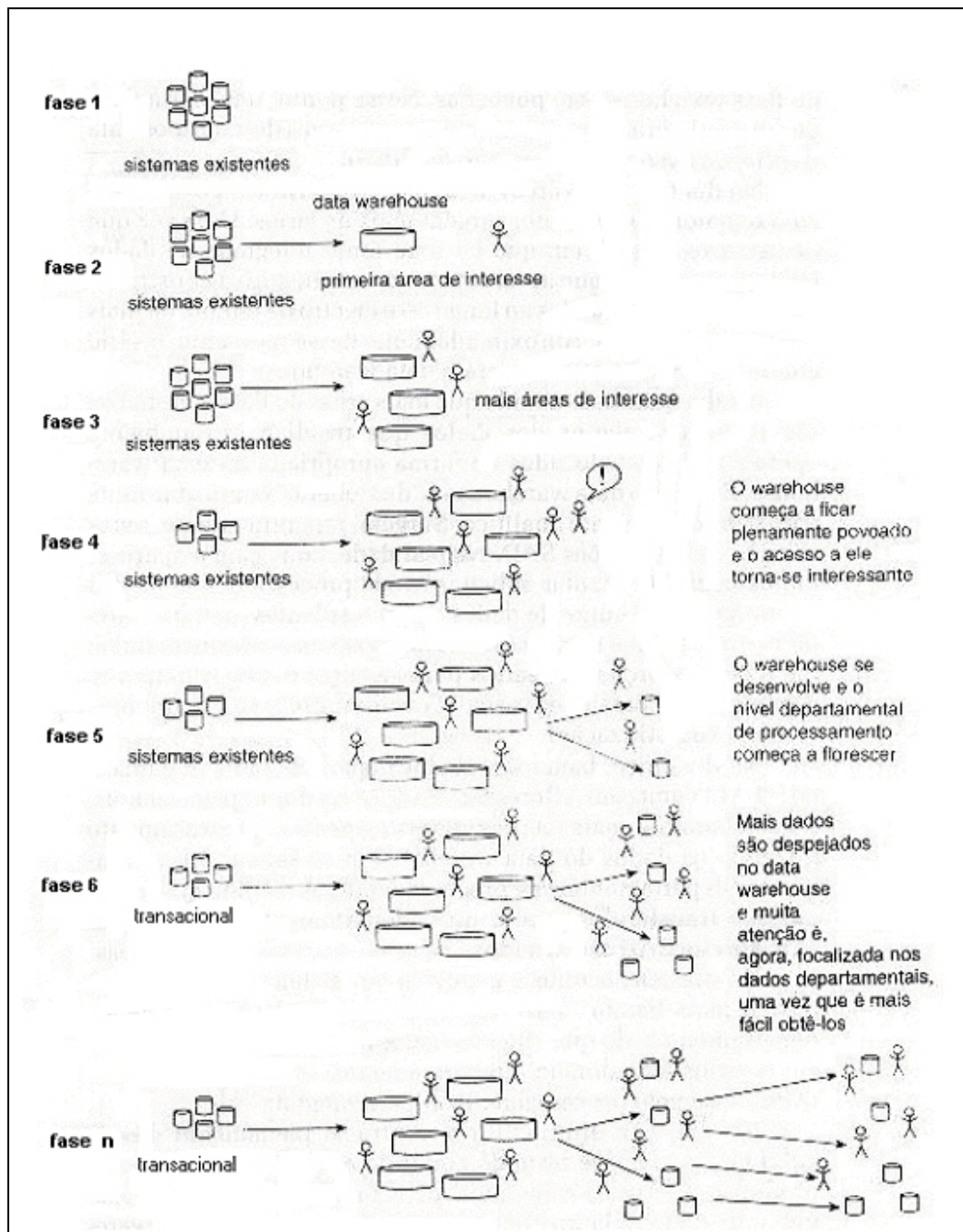
Quando os dados são considerados antigos, passam do detalhe corrente para o detalhe mais antigo. À medida que os dados são resumidos, passam do detalhe corrente para os dados

levemente resumidos e, a seguir, dos dados levemente resumidos para os dados altamente resumidos (Inmon, 1997).

O processo típico de construção de um DW pode ser visto na fig. 6. Na fase um há um grupo de sistemas efetuando o processo transacional. Na fase dois, as poucas primeiras tabelas da primeira área de interesse do DW são povoadas. Na fase três, mais áreas do DW são povoadas, onde novos usuários surgem com o maior povoamento. Na fase quatro, à medida que mais áreas são povoadas, alguns dos dados que residiam no ambiente operacional são colocados de forma apropriada no DW, aparecendo cada vez mais e mais usuários. A competição para chegar ao DW começa a tornar-se um obstáculo à sua utilização. Na fase cinco, bancos de dados departamentais (*data-mart* ou OLAP) começam a florescer. Na opinião dos departamentos é mais barato e mais fácil efetuar o respectivo processamento trazendo os dados do DW para seus próprios ambientes departamentais de processamento.

Na fase seis, acontece à corrida aos sistemas departamentais. É mais rápido e fácil obter os dados desses sistemas do que do DW. Logo, os usuários abandonam, gradativamente, os detalhes do DW pelo processamento departamental. Na fase n, a arquitetura encontra-se plenamente desenvolvida, restando do conjunto original de sistemas de produção apenas o processamento transacional. O DW está pleno de dados. Este trabalho irá alcançar apenas até a fase dois.



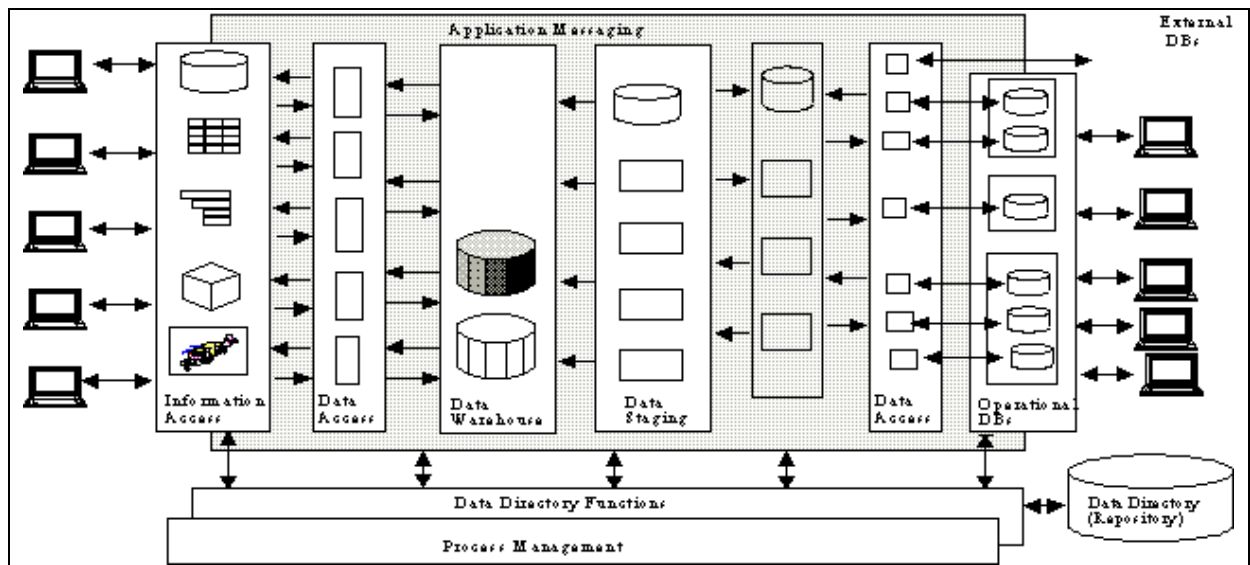


Fonte: Inmon (1997, p. 44)

FIGURA 6 – Processo de construção do Data Warehouse

#### 4.1.3 ARQUITETURA

Orr (2001) propõe conforme a fig. 7 o diagrama de uma arquitetura genérica, composta por camadas interconectadas, onde diversos elementos do DW são sistematizados de forma abrangente, permitindo que outras concepções do tema possam ser adaptadas neste modelo.



Fonte: Orr (2001)

FIGURA 7 – Diagrama de uma arquitetura genérica de um *Data Warehouse*

A seguir são descritas as camadas do diagrama proposto por Orr (2001):

- a) *External Db's*: camada de banco de dados transacionais e de fontes externas;
- b) *Information Access*: camada de acesso à informação;
- c) *Data Access*: camada de acesso aos dados;
- d) Metadados - *repository*: camada dicionário de dados. Pode-se citar como informações contidas nos metadados:
  - quando os dados foram atualizados pela última vez;
  - que regras de transformação foram aplicadas no processamento dos dados;
  - qual é o esquema do banco de dados;
  - quais são as regras de transformação dos dados;
  - que agregações existem no DW;
- e) *Process Management*: camada de gerenciamento de processo;
- f) *Middleware – Application Messaging*: camada de transporte;
- g) camada do DW: o DW propriamente dito corresponde aos dados usados para fins "informativos";
- h) *Data Staging*: camada de representação dos dados.

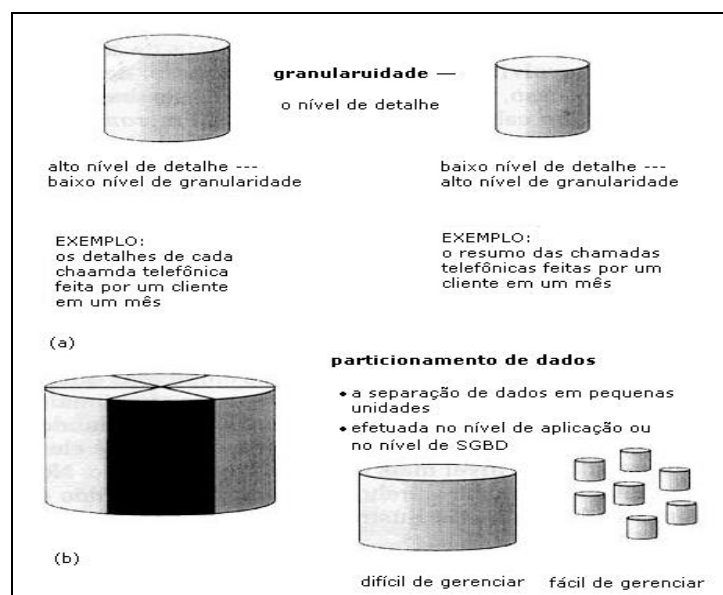
#### 4.1.4 FUNCIONALIDADE

Um DW é constituído por três componentes funcionais diferentes, sendo que cada um dos quais precisa ser adequadamente customizado para atender às necessidades da empresa (Mengarda, 1998):

- aquisição dos dados: abrange a aquisição dos dados nos sistemas de origem e fontes externas;
- armazenamento dos dados: este componente está relacionado com o armazenamento dos dados, considerando a forma como os dados irão estar modelados e como serão acessados;
- acesso dos dados: compreende a etapa final do DW, isto é, a visualização dos dados pelo usuário através de produtos para análise multidimensional, redes neurais, ferramentas de recuperação de dados ou de análise de dados.

#### 4.1.5 GRANULARIDADE

Granularidade se refere ao nível de detalhe em que as unidades de dados são mantidas no DW. Quanto maior o nível de detalhes, menor o nível de granularidade. O volume de dados que é contido no DW é balanceado de acordo com o nível de detalhe de uma consulta. A fig. 8 mostra as questões de granularidade.



Fonte: Inmon (1997, p. 46)

FIGURA 8 – Granularidade do *Data Warehouse*

Segundo Inmon (1997), espaço é um problema óbvio em um DW. Com um nível mais alto de granularidade utiliza-se não só, menos bytes, como menos índices também.

#### 4.1.6 METODOLOGIA DE DESENVOLVIMENTO

Deve estar claro para a equipe de desenvolvimento que para inserir os dados operacionais no DW é necessário ter um método eficiente e que satisfaça as condições do DW da empresa. Para aumentar a performance do DW existem algumas técnicas que podem ser aplicadas durante o seu desenvolvimento como a redundância dos dados e o agrupamento de registros, também é discutido o intervalo que deve existir entre as atualizações dos dados no DW.

Dependendo do tamanho do projeto é possível desenvolver um *data mart* que poderá ser definitivo, se o escopo do projeto for pequeno ou poderá ser agrupado com outros *data marts* para formar um único DW.

##### 4.1.6.1 CONSIDERAÇÕES INICIAIS

O início basicamente consiste na extração dos dados das fontes internas, incluindo a transformação e a limpeza dos dados. Um dos itens mais importantes é o repositório dos metadados, responsável pela documentação de cada registro realizado na base de dados. Os metadados vão proporcionar a segurança sobre a qualidade das informações obtidas (Villanova, 2000).

Muitas empresas iniciam o processo a partir de uma área específica, que normalmente é uma área carente de informação e cujo trabalho seja relevante para os negócios da empresa, criando *Data Marts*, para depois ir crescendo aos poucos, seguindo uma estratégia *botton-up* ou assunto-por-assunto.

##### 4.1.6.2 DESENVOLVIMENTO

A mais difundida metodologia de desenvolvimento de um DW dentre as várias existentes, e que será seguida durante o desenvolvimento do protótipo é a chamada Metodologia Baseada em Dados. Segundo Inmon (1997), tal metodologia é de uso geral, para aplicação em projetos baseado em dados, caso típico dos ambientes DW.

Nesta metodologia a primeira preocupação é com os dados e depois com os requisitos do DW a ser desenvolvido, o que contraria o ciclo de vida normal do desenvolvimento de um sistema, que inicia com os requisitos e termina com o código fonte.

A Metodologia Baseada em Dados se preocupa com os resultados das atividades a serem efetuadas, seus parâmetros de sucesso e com a ordem natural em que devem ocorrer, incumbindo o desenvolvedor do DW da responsabilidade de como as atividades devem ser feitas. A seguir serão descritas as 11 atividades que compõem esta metodologia, de acordo com Inmon (1997).

A primeira atividade se refere à Análise do Modelo de Dados, atividade esta onde é feita uma análise com profundidade no modelo de dados da empresa e se for preciso, modificá-lo e até construí-lo de uma forma que se tenha uma idéia clara dos seguintes pontos:

- a) principais áreas de interesse envolvidas;
- b) definição clara das fronteiras (escopo de integração);
- c) diferenciação dos dados – primitivos e derivados;
- d) especificação de chaves, agrupamentos, relacionamentos, atributos, redundância e tipos de dados.

Com este processo procura-se obter um modelo de dados sólido, que atenda os requisitos especificados no desenvolvimento e possibilite a identificação correta dos elementos de um DW.

Na segunda atividade, o Dimensionamento, faz-se uma estimativa do volume de dados projetados para o DW, levando em consideração o nível de detalhamento requerido para cada assunto e definir possíveis níveis de granularidade requeridos.

- a) A terceira atividade é a Avaliação Técnica. Nela procura-se avaliar criteriosamente os requisitos básicos das tecnologias de suporte de um ambiente de DW e determinar as que melhor atenderão às necessidades do projeto.

A próxima atividade é a Preparação do Ambiente Técnico. Essa parte é de suma importância no desenvolvimento de um DW, pois é nela que serão definidas as ferramentas que irão interagir com o DW. Quanto maior o grau de eficiência das ferramentas de suporte, maior será o desempenho obtido pelo DW implantado.

Como quinta atividade tem-se a Análise das Áreas de Interesse. Aqui é selecionada a área de interesse que será enfocada no DW usando critérios de tamanho e complexidade, estabelecendo assim o limite de atuação do mesmo. O assunto selecionado deve estar coerente com as necessidades do projeto.

Na sexta atividade é realizado o projeto físico do DW (modelo de dados). A partir do modelo de dados, o projeto é elaborado. Procura-se montar o modelo de dados de forma a deixar o DW em condições de proporcionar gerenciamento de um volume de dados que possam ser carregados, acessados, indexados e pesquisados de forma eficiente.

A sétima atividade envolve a análise do sistema fonte. É feito o mapeamento dos dados do ambiente transacional para o DW. Isso busca atender um padrão de conformidade aceitável entre os dados das fontes e os projetados, dentro das necessidades da estrutura do DW.

A especificação de programas é a oitava atividade onde é feita a descrição dos programas específicos para efetuar a migração dos dados transacionais para o DW. Esta elaboração das definições dos programas deve ser concisa e condizente com as necessidades do DW.

A fase de Elaboração de Programas é a nona atividade. Nela ocorre o desenvolvimento de programas para extração e integração de dados no DW. Isso visa produzir programas que estejam em plenas condições de execução, devidamente testados e documentados.

A penúltima atividade, o Povoamento, é determinada pela entrada em produção dos programas que compõem o projeto. Isso ocorre devido ao DW estar povoado com dados de qualidade e plenamente apto para uso. Em decorrência disso o nível de utilização do DW cresce, são proporcionadas facilidades de acesso e as expectativas dos usuários são atendidas.

E por fim, a última atividade do processo de desenvolvimento do DW: a utilização dos dados. Aqui ocorre a utilização satisfatória do DW para atendimento de questões estratégicas e gerenciais. As necessidades dos usuários são atendidas através da geração de relatórios ou consultas.

#### 4.1.6.3 ERROS NA IMPLEMENTAÇÃO DE UM DATA WAREHOUSE

Campos (2000), aponta os dez erros mais comuns na implantação de um DW:

- b) começar o projeto com o tipo errado de "patrocínio";
- c) gerar expectativas que não podem ser satisfeitas, frustrando os executivos quando da utilização do DW;
- d) dizer: “Isto vai ajudar os gerentes a tomar decisões melhores” e outras afirmações politicamente ingênuas;
- e) carregar o DW com informações só "porque estavam disponíveis";
- f) falhar no objetivo de acrescentar valor aos dados através de mecanismos de desnormalização, categorização e navegação assistida;
- g) focalizar o DW em dados tradicionais internos orientados a registro e ignorar o valor potencial de dados textuais, imagens, som, vídeo e dados externos;
- h) fornecer dados com definições confusas e sobrepostas;
- i) acreditar nas promessas de desempenho, capacidade e escalabilidade dos vendedores de produtos para DW;
- j) usar DW como uma justificativa para modelagem de dados e uso de ferramentas case.

## 4.2 INTERNET

A *Internet* é o conjunto de diversas redes de computadores que se comunicam através dos protocolos TCP/IP. Quando se conecta a um *site* está-se conectando à *Internet*. A *World Wide Web* (www ou a *web*) é uma rede de computadores na *Internet* que fornece informação em forma de hipertexto ou ainda, é um sistema de informações organizado de maneira a englobar todos os outros sistemas de informação disponíveis na *Internet*. Para ver a informação, utiliza-se um *software* chamado navegador onde são descarregadas informações (chamadas “documentos” ou “páginas”) de servidores de *Internet* ou *sites* e mostrá-los na tela do usuário. O usuário pode então, através dos *links*, acessar outros documentos ou mesmo enviar informações de volta para o servidor de modo a interagir com ele. O ato de acessar os *links* é comumente chamado de “surfear” na *Web*.

Segundo ICMC (2003), a *World Wide Web* implementa três ferramentas importantes:

- a) HTTP: o *Hypertext Transfer Protocol* é um protocolo de transferência de hipertexto usado para a transmissão de dados no sistema *WWW*. Cada vez que o usuário aciona um *link*, o *browser* realiza uma comunicação com um servidor da *Web* através deste protocolo;

- b) URL: o *Uniform Resource Locator* (URL) é um localizador uniforme de recursos que permite o usuário acionar programas (*scripts*) e enviar parâmetros através deles, etc. No endereço “http://www.icmc.usp.br/ensino/material/html/url.html” o URL identifica:
- o protocolo de acesso ao recurso desejado (http);
  - a máquina a ser conectada (www.icmc.usp.br);
  - o caminho de diretórios (ensino/material/html/);
  - o recurso (arquivo) a ser obtido (url.html).
- c) HTML: o *Hypertext Markup Language* é uma linguagem de marcação de hipertexto usada para transmitir documentos formatados através da rede. Os *browsers* são os responsáveis por identificar as marcações em HTML e apresentar os documentos conforme o que foi especificado por essas marcações.

A *World Wide Web* revolucionou a comunicação entre pessoas no mundo todo. Essa mídia global tomou proporções gigantescas trazendo consigo um enorme e variado volume de informações. Tais informações variam dos mais irrelevantes até os mais importantes assuntos a nível mundial. Tornou-se comum visitar novos lugares, conhecer novas pessoas sem sair de casa, apenas com alguns “*clicks*”.

#### 4.3 ACTIVE SERVER PAGES (ASP)

Segundo Mitchell (2000), as *Active Server Pages* (ASP) são a solução da *Microsoft* para criar páginas dinâmicas da *Web*. Os primeiros *sites* da *Internet* consistiam basicamente de páginas fáceis de ler com imagens gráficas, todas estáticas, ou seja, o seu conteúdo era sempre o mesmo a cada visita. Com a evolução da *Internet* surgiu a necessidade de criar *sites* interativos e mais programáveis.

Nos últimos anos, o ASP tornou-se uma ferramenta amplamente utilizada para criar páginas dinâmicas da *Web*. Uma das principais razões para esse crescimento é o fato de que o ASP pode utilizar uma coleção de objetos da *Microsoft* para se conectar a muitos bancos de dados. O que torna o ASP uma ferramenta inestimável é o fato de ser capaz de interfacear facilmente com uma página da *Web* com qualquer banco de dados compatível com o *Open Database Connectivity* (ODBC).



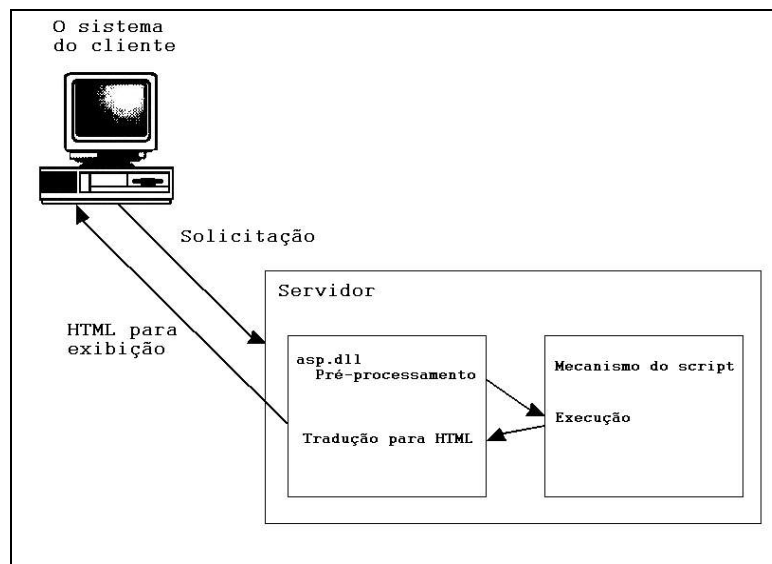
O ASP é um ambiente de programação e não uma linguagem de programação, onde se pode combinar *Hypertext Markup Language* (HTML), *scripts* e outros componentes, para se criar aplicações dinâmicas e poderosas no servidor *Web*. Este ambiente foi criado pela *Microsoft* para o *Internet Information Server* (IIS). Os arquivos criados com ASP têm a extensão “.asp” e devem ser executados em um servidor *Web* compatível com ASP.

Basicamente o ASP se divide em duas partes: o código programático e o HTML, sendo que o código programático pode ser escrito em linguagens de *script*, como *VBScript* e *JScript*. O conteúdo programático do código ASP fica delimitado pelas *tags* `<%` e `%>`, que é designado como código ASP. Tal conteúdo é o que o servidor processa antes de enviar a página para o navegador da *Web*. O quadro 2 mostra um exemplo de código ASP.

```
<%Response.Write "Este é um código ASP" %>
```

QUADRO 2 – Exemplo de código ASP

O processo que o ASP realiza para enviar uma página *Web* ao usuário, pode ser visto na fig. 9.



Fonte: Mitchell (2000, p. 49)

FIGURA 9 – Processo que o ASP realiza para enviar uma página *Web* ao usuário

A seguir são descritas as etapas do processo ASP mostradas na fig. 9:

- a) solicitação: o navegador da *Web* entra em contato com o servidor e diz que página ele quer ver;
- b) pré-processamento: o arquivo “asp.dll” faz algum processamento inicial no *script* solicitado;

- c) execução: o mecanismo de *script* executa as instruções no *script*;
- d) tradução: o ASP traduz os resultados da execução para HTML;
- e) exibição: a HTML é enviada de volta para o navegador da *Web*, que processa os *tags* e exibe a página.

O ASP é uma tecnologia do lado servidor, o que significa que deve estar concluída no momento em que a página é descarregada. O ASP faz todo o seu trabalho então gera uma página de HTML comum que é enviada para o navegador. Ou seja, o processo ASP termina quando a página é enviada para o usuário.

A maioria das páginas ASP são criadas utilizando *VBScript*. *VBScript* tem a *sintaxe* mais semelhante ao inglês das quatro linguagens de *script* e é semelhante à *sintaxe* do *Visual Basic*, com que muitos desenvolvedores da *Web* têm experiência.

#### 4.3.1 OBJETOS PRÉ-DEFINIDOS DE ASP

Segundo Mitchell (2000), o ASP possui seis objetos pré-definidos:

- a) *response*: é utilizado para enviar saída.;
- b) *request*: é utilizado para recuperar dados do cliente;
- c) *application*: é utilizado para compartilhar informações entre vários clientes que visitam o mesmo grupo de página.;
- d) *session*: uma sessão, por outro lado, refere-se a um único cliente que acessa um aplicativo.;
- e) *server*: esse objeto fornece algumas propriedades e métodos básicos;
- f) *ObjectContext*: é utilizado para vincular ASP e o *Microsoft Transaction Server* (MTS);
- g) *ASPErrors*: é novo no ASP. Ele permite obter as informações sobre erros de *script* nas páginas *Web*.

#### 4.3.2 ASP E BANCO DE DADOS

Segundo Michell (2000), o mais útil recurso das *Active Server Pages* é a capacidade de uma página ASP facilmente interagir com um banco de dados. Através de uma coleção de objetos *Activex Data Object* (ADO) da *Microsoft*, o ASP dispõe de recursos para ler e modificar o conteúdo de um banco de dados.

#### 4.3.2.1 BANCO DE DADOS RELACIONAIS

Um banco de dados é uma coleção de informações que pode ser facilmente consultada e modificada (Wille, 1999). Um banco de dados possui as seguintes características:

- a) é uma coleção lógica coerente de dados com um significado inerente; uma disposição desordenada dos dados não pode ser referenciada como um banco de dados;
- b) é projetado e construído com dados com um propósito específico, possuindo um conjunto pré-definido de usuários e aplicações;
- c) representa algum aspecto do mundo real, o qual é chamado de “mini-mundo”; qualquer alteração efetuada no mini-mundo é automaticamente refletida no banco de dados.

Ao utilizar um banco de dados, é possível fazer apenas quatro coisas: recuperar, inserir, consultar ou excluir dados existentes. O principal propósito de um banco de dados é armazenar informações. Embora cada sistema de banco de dados possa ter pequenas diferenças entre si, cada banco de dados utiliza tabelas para armazenar informações.

Os bancos de dados constituem-se de várias tabelas e cada tabela representa um único objeto. Mas, se um objeto contiver instâncias de tipos de objetos diferentes? Por exemplo, se uma das colunas de uma tabela constituir de um objeto, pode-se criar uma nova tabela com as características do novo objeto e fazer o relacionamento entre as tabelas. Um banco de dados relacional representa cada objeto, relacionado ou não como sua própria tabela de banco de dados (Wille, 1999).

Segundo Mitchell (2000), muitos sistemas comerciais de banco de dados relacional estão disponíveis no mercado. Entre os mais populares encontra-se o *Microsoft SQL-Server*, *Microsoft Access*, *Oracle* e *Informix*. Com o ASP pode-se acessar qualquer banco de dados compatível com um padrão de acesso a banco de dados criado pela *Microsoft*, denominado *Open Database Connectivity* (ODBC). O ODBC fornece uma interface comum para um banco de dados de modo que o mesmo trecho de código funcionará independentemente do banco de dados que se está consultando.

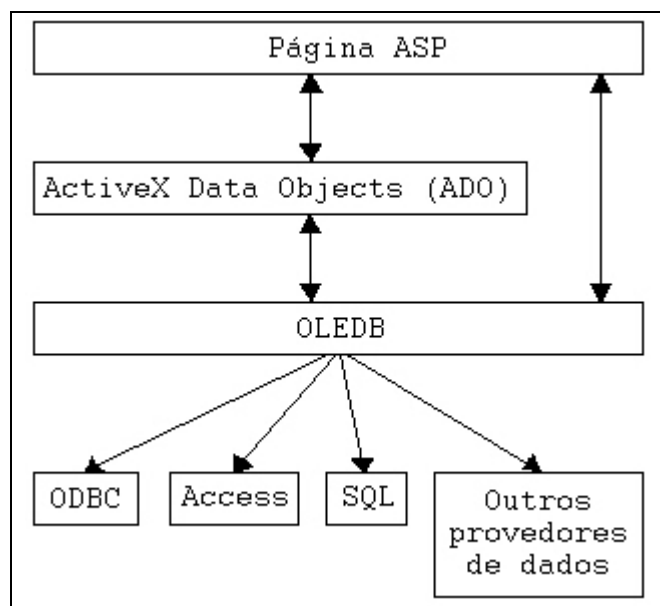
Projetados para simplificar o processo de armazenar e recuperar informações, os bancos de dados fornecem o melhor desempenho para armazenar informações longas e heterogêneas.

Por outro lado, a conexão com o banco de dados transporta com ela um alto custo de desempenho. Os bancos de dados devem ser utilizados somente para armazenar grandes quantidades de informações sobre períodos indefinidos de tempo.

#### 4.3.2.2 COMUNICANDO-SE COM UM BANCO DE DADOS UTILIZANDO ADO

O modelo ADO trabalha com qualquer fonte de *Object Linking and Embedding Database* (OLEDB), que inclui fontes compatíveis com ODBC. O OLEDB é um conjunto de interfaces de programação no nível de sistema que fornece acesso a diversas informações e fontes de dados. Como o OLEDB está no nível de sistema o programador tem controle completo sobre todas as interfaces (Wille, 1999).

Na fig. 10, Mitchell (2000) exemplifica o funcionamento do OLEDB com suas fontes de dados.



Fonte: Mitchell (2000, p. 458)

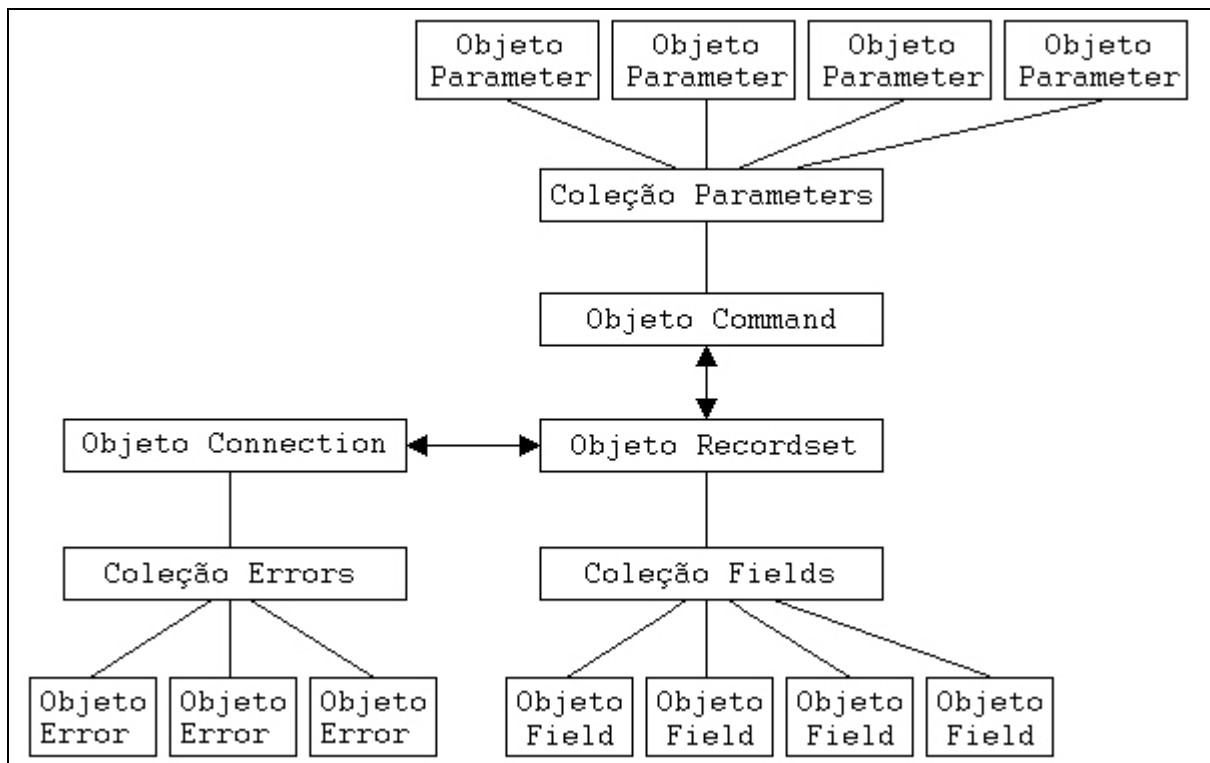
FIGURA 10 – A hierarquia de acesso de dados

O ADO é um objeto de acesso de dados no nível de aplicativo que é fácil de usar, neutro quanto à linguagem de programação, podendo ser acessado em *VBScript*, *Jscript*, *Visual Basic* e quaisquer outras linguagens que possam tratar dos objetos. A seguir se tem os seis objetos do ADO:

- a) *connection*: estabelece a conexão com uma fonte de dados;
- b) *recordset*: contém um conjunto de linhas de uma tabela retornados de uma consulta de banco de dados;

- c) *command*: fornece a possibilidade de definir comandos específicos que se pretende executar em um banco de dados várias vezes com parâmetros diferentes. Combina o objeto *recordset* e o objeto *connection*;
- d) *property*: contém informações dinâmicas sobre um objeto ADO, fornecido pelo provedor básico;
- e) *field*: contém dados de uma única coluna e informações sobre esses dados. O objeto *recordset* fornece a coleção *fields*, que contém todos os objetos *field* de um *recordset*;
- f) *error*: contém informações estendidas de erro retornadas pelo provedor.

A fig. 11 mostra um diagrama de como esses objetos se relacionam entre si:



Fonte: Mitchell (2000, p. 459)

FIGURA 11 – Relacionamento entre os objetos do ADO

#### 4.3.2.3 CONECTANDO-SE A UM BANCO DE DADOS

Antes de fazer qualquer coisa com o banco de dados é necessário conectar-se a ele. Entra em cena o objeto *connection* do ADO que é utilizado para armazenar informações sobre o armazenamento de dados que se deseja acessar. Ele é criado da mesma forma que seus

outros componentes, utilizando o *Server.CreateObject*. O quadro 4 mostra uma conexão sendo criada:

```
set conexao = Server.CreateObject("ADODB.Connection")
```

Fonte: Mitchell (2000, p. 460)

QUADRO 4 – Criação de uma conexão a banco de dados utilizando o objeto *connection*

Um meio ao qual pode-se conectar facilmente a qualquer tipo de banco de dados compatível com ODBC é um sistema DSN (*Data Source Names*). O DSN é um arquivo que contém informações sobre o banco de dados tais como onde ele está e que tipo de banco de dados é.

Para estabelecer a conexão com o banco deve-se utilizar o método *Open* do objeto *Connection*. Para fechar utiliza-se o método *Close* e para liberar memória deve-se configurar o objeto como *Nothing*. O quadro 5 mostra esses passos.

```
<%
set conexao = Server.CreateObject("ADODB.Connection")
conexao.ConnectionString = "DSN=TCC"
conexao.Open
...
conexao.Close
set conexao = Nothing
%>
```

QUADRO 5 – Exemplo de como criar e fechar uma conexão a banco de dados

Outra alternativa, que é a mais relevante para este trabalho, é a conexão direta ao banco sem utilizar um sistema DSN. Um exemplo de uma conexão sem DSN é mostrado no quadro 6.

```
Conn.Open "Driver={SQL Server};Server=MyServerName;" & _
          "Database=myDatabaseName;" & _
          "Uid=myUsername;Pwd=myPassword"
```

QUADRO 6 – Conexão direta ao banco de dados sem utilizar sistema DSN

O parâmetro “Driver=” informa ao objeto *connection* o tipo de banco de dados ao qual ele deve se conectar. O parâmetro “Server=” indica o nome do servidor, como se trata de um banco local, informa-se o nome da máquina. O parâmetro “Database=” indica o nome do arquivo que corresponde à base de dados. Como se deseja conectar a um banco de dados de

SQL, é preciso um nome de usuário e da senha para acessá-lo. Estas informações são passadas através dos parâmetros “Uin” e “Pwd”.

#### 4.3.2.4 LENDO E ALTERANDO DADOS DE UM BANCO DE DADOS

Após estabelecer a conexão com o banco de dados, torna-se possível manipular todas as informações nele contidas.

Para isso, é necessário utilizar outros objetos do ASP. Segundo Mitchell (2000) o objeto *recordset* é simplesmente um conjunto de registros e que pode ser utilizado para conter um subconjunto dos registros em uma tabela ou mesmo todos os registros da tabela.

Criando a instância do objeto *recordset* não o preenche com quaisquer dados. Para fazer isso necessário utilizar o método *open*. O método *open* pode aceitar muitos conjuntos diferentes de argumentos e pode ser utilizado de várias maneiras diferentes. No quadro 7 é mostrada a sua *sintaxe*.

```
recordset.Open origem, conexão, tipodecursos, tipodebloqueio, tipodecomando
```

Fonte: Mitchell (2000, p. 465)

#### QUADRO 7 – Conjunto de argumentos do método *open* do objeto *recordset*

Os argumentos do método *open* são descritos a seguir:

- a) origem: é uma *string* contendo um comando reconhecido;
- b) conexão: é um objeto *connection* ou uma *string* contendo as informações de conexão. Se a *string* é utilizada, ela deve conter todas as informações que entrariam na propriedade *ConnectionString* do objeto *connection*;
- c) tipodecursor: indica a maneira como se deseja mover pelo *recordset*. Seu valor padrão é a constante de ADO “adOpenForwardOnly”, que permite acessar somente o próximo registro do *recordset*. Este argumento pode ser substituído pelo valor zero, que equivale a constante “adOpenForwardOnly”. Outro tipo de cursor utilizado é o “adOpenDynamic” que permite o movimento através do conjunto de registros tanto para o próximo quanto para o anterior;
- d) tipodebloqueio: determina se é possível gravar na tabela, caso sim, como;
- e) tipodecomando: indica como o parâmetro origem deve ser avaliado.

O quadro 8 mostra um exemplo da utilização mais comum do método *open*.

```
recordset.Open = "nome_tabela", conexao,,, adCmdTable
```

Fonte: Mitchell (2000, p. 466)

#### QUADRO 8 – Utilização mais comum do método *open*

No quadro 9 é mostrado um exemplo completo de leitura na tabela “Alunos” utilizando o objeto *recordset* e seus métodos *Open* e *MoveNext*, sendo este último utilizado para acessar o próximo registro do *recordset*.

```
<%@ Language=VBScript%>
<!--#include virtual="/adovbs.inc"-->
<%
Dim conexao, recordset_Alunos

'=== ABRE A CONEXÃO COM O BANCO DE DADOS
set conexao = Server.CreateObject("ADODB.Connection")
conexao.ConnectionString = "DSN=TCC"
conexao.Open

'=== CRIA UMA INSTÂNCIA DO OBJETO RECORDSET
'=== CARREGA AS INFORMAÇÕES DA TABELA "ALUNOS"
set recordset_Alunos = Server.CreateObject("ADODB.Recordset")
recordset_Alunos.Open = "Alunos", conexao,,, adCmdTable

'=== LISTA O CONTEÚDO DA TABELA "ALUNOS"
Do While Not (recordset_Alunos.EOF)
    Response.Write recordset_Alunos("NOME") & "<br>"
    recordset_Alunos.MoveNext
Loop

'=== FECHA E LIMPA OS OBJETOS ADO
recordset_Alunos.Close
set recordset_Alunos = Nothing

conexao.Close
set conexao = Nothing
%>
```

#### QUADRO 9 – Leitura da tabela “Alunos” utilizando os métodos *open* e *MoveNext*

A linguagem mais usada para acessar os registros de uma tabela em um banco de dados é a *Structured Query Language* (SQL). A SQL é usada para consultar, atualizar e gerenciar os dados do banco de dados relacionais. Usando a SQL pode-se recuperar, filtrar e classificar dados específicos de um banco de dados (Wille, 1999).



A forma mais direta para se acessar um *recordset* utilizando um comando SQL é mostrado no quadro 10.

```
set recordset_Alunos = conexao.Execute(sql_comando)
```

QUADRO 10 – Utilização de comando SQL para acesso direto ao banco de dados

No parâmetro “sql\_comando” do método *execute* deve ser utilizado, por exemplo, o comando “Select \* from Alunos”.

Para inserir, modificar ou excluir dados de uma tabela, existem duas maneiras. Através dos comandos SQL “Insert”, “Update” ou “Delete” que pode ser visto no quadro 11, ou através dos métodos encontrados no *recordset*: “ADDNEW”, “UPDATE” ou “DELETE” mostrados no quadro 12.

```
'=== CRIANDO UM NOVO REGISTRO UTILIZANDO COMANDO SQL
recordset_Alunos.Execute("INSERT INTO ALUNOS (NOME) VALUES ('João da Silva')")

'=== ALTERANDO O REGISTRO ATUAL UTILIZANDO COMANDO SQL
recordset_Alunos.Execute("UPDATE ALUNOS SET NOME = 'João de Souza' WHERE CODALU = 1")

'=== EXCLUINDO O REGISTRO ATUAL UTILIZANDO COMANDO SQL
recordset_Alunos.Execute("DELETE ALUNOS WHERE CODALU = 1")
```

QUADRO 11 – Utilização dos comandos SQL Insert, Update e Delete.

```
'=== CRIANDO UM NOVO REGISTRO UTILIZANDO COMANDO SQL
recordset_Alunos.Execute("INSERT INTO ALUNOS (NOME) VALUES ('João da Silva')")

'=== ALTERANDO O REGISTRO ATUAL UTILIZANDO COMANDO SQL
recordset_Alunos.Execute("UPDATE ALUNOS SET NOME = 'João de Souza' WHERE CODALU = 1")

'=== EXCLUINDO O REGISTRO ATUAL UTILIZANDO COMANDO SQL
recordset_Alunos.Execute("DELETE ALUNOS WHERE CODALU = 1")
```

QUADRO 12 – Utilização dos métodos AddNew, UpDate e Delete do recordset.

### 4.3.3 INTERNET INFORMATION SERVER

Desde 1996, com o lançamento do *Windows NT 4.0* com o *Internet Information Server* (IIS) que a utilização deste produto vem se familiarizado. O IIS é o servidor da *Web* profissional da *Microsoft* que foi projetado para desenvolvedores *Web* para ser usado em casa ou nas empresas. Substituto do PWS (*Persional Web Server*), o IIS só pode servir 10 conexões de clientes simultâneas e não aproveita todos os recursos das versões do servidor. Por isso, não é recomendada a disponibilização de *sites* na *Internet* ou *Intranet* através deste servidor, devido ao limite de conexões simultâneas e para uso em um ambiente maior (ainda

mais se for a *Internet*) várias operações de segurança devem ser realizadas no servidor *Web* e no sistema operacional.

O IIS suporta:

- a) ASP;
- b) PHP;
- c) *Server Side Includes* (SSI);
- d) controles *ActiveX*;
- e) roda *scripts* “*Internet Server API* (ISAPI) ” e “*Common Gateway Interface* (CGI)”;
- f) acesso à banco de dados via ODBC;
- g) *Secure Sockets Layer* (SSL).

De maneira semelhante ao PWS, antes de instalar o IIS, é necessário instalar o protocolo TCP/IP do *Windows* e os utilitários de conectividade. O IIS é distribuído gratuitamente junto com a instalação do *Windows* NT/2000/XP.

## 5 DESENVOLVIMENTO DO TRABALHO

Neste capítulo são apresentadas a especificação, implementação e as técnicas e ferramentas utilizadas e a operacionalidade do modelo proposto neste trabalho.

### 5.1 REQUISITOS DO *SOFTWARE* A SER DESENVOLVIDO

Não importa quão bem projetado ou codificado seja, um programa mal analisado e especificado desapontará o usuário e trará aborrecimentos ao programador. Segundo Pressman (1995), uma compreensão completa dos requisitos de *software* é fundamental para um bem-sucedido desenvolvimento de *software*.

Os requisitos funcionais estão ligados diretamente à funcionalidade do sistema. Os requisitos funcionais do sistema desenvolvido são:

- a) o cadastro de aluno, educador, escola, equipamento, cidade ou situação deve ter um identificador único;
- b) a geração dos dados do *Data Warehouse* será feita mensalmente a partir da informações das tabelas de Alunos, Educadores, Escolas e Equipamentos;
- c) relatórios gráficos e analíticos mostrando a situação dos Alunos, Educadores, Escolas e Equipamentos.

Os requisitos não funcionais refletem os requisitos que expressam as qualidades específicas que o aplicativo deve possuir, que são:

- a) fácil manutenção no *software*;
- b) integração dos dados de todas as EIC's do CDI;
- c) agilidade no processo de tomada de decisão;
- d) acesso às informações em qualquer micro com *internet*;

#### 5.1.1 LISTA DE EVENTOS

A Lista de Eventos é uma lista textual dos "eventos", ou estímulos no ambiente externo, aos quais o sistema deve responder e uma indicação da pessoa ou sistema que inicia o evento. Na lista de eventos pode-se encontrar os acontecimentos (eventos), que ocorrem para que o sistema possa ser criado a partir deles. O quadro 13 mostra a lista de eventos do sistema.

|   |
|---|
| 1 - Usuário mantém cadastro de Alunos           |
| 2 - Usuário mantém cadastro de Educadores       |
| 3 - Usuário mantém cadastro de Escolas          |
| 4 - Usuário mantém cadastro de Equipamentos     |
| 5 - Usuário mantém cadastro de Cidades          |
| 6 - Usuário mantém cadastro de Situações        |
| 7 - Usuário atualiza Data Warehouse             |
| 8 - Coordenador solicita situação Alunos        |
| 9 - Coordenador solicita situação Educadores    |
| 10 - Coordenador solicita situação Escolas      |
| 11 - Coordenador solicita situação Equipamentos |

QUADRO 13 – Lista de Eventos

### 5.1.2 DIAGRAMA DE CONTEXTO

No diagrama de contexto são apresentados os relacionamentos com as entidades externas, conforme demonstrado na fig. 12.

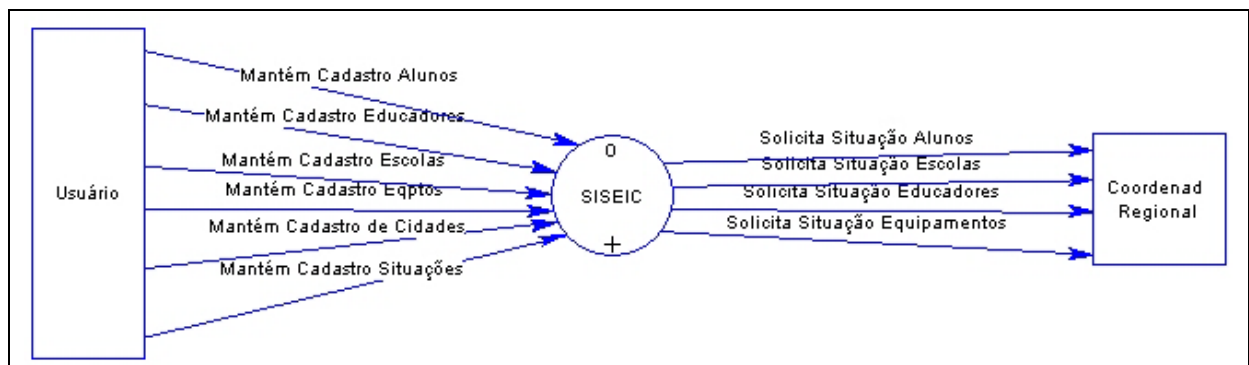


FIGURA 12 – Diagrama de Contexto

### 5.1.3 MODELO ENTIDADE RELACIONAMENTO (MER)

O modelo entidade relacionamento é um modelo baseado na percepção do mundo real, que consiste em um conjunto de objetos básicos (entidades) e nos relacionamentos entre esses objetos. A fig. 13 mostra o MER do sistema desenvolvido.

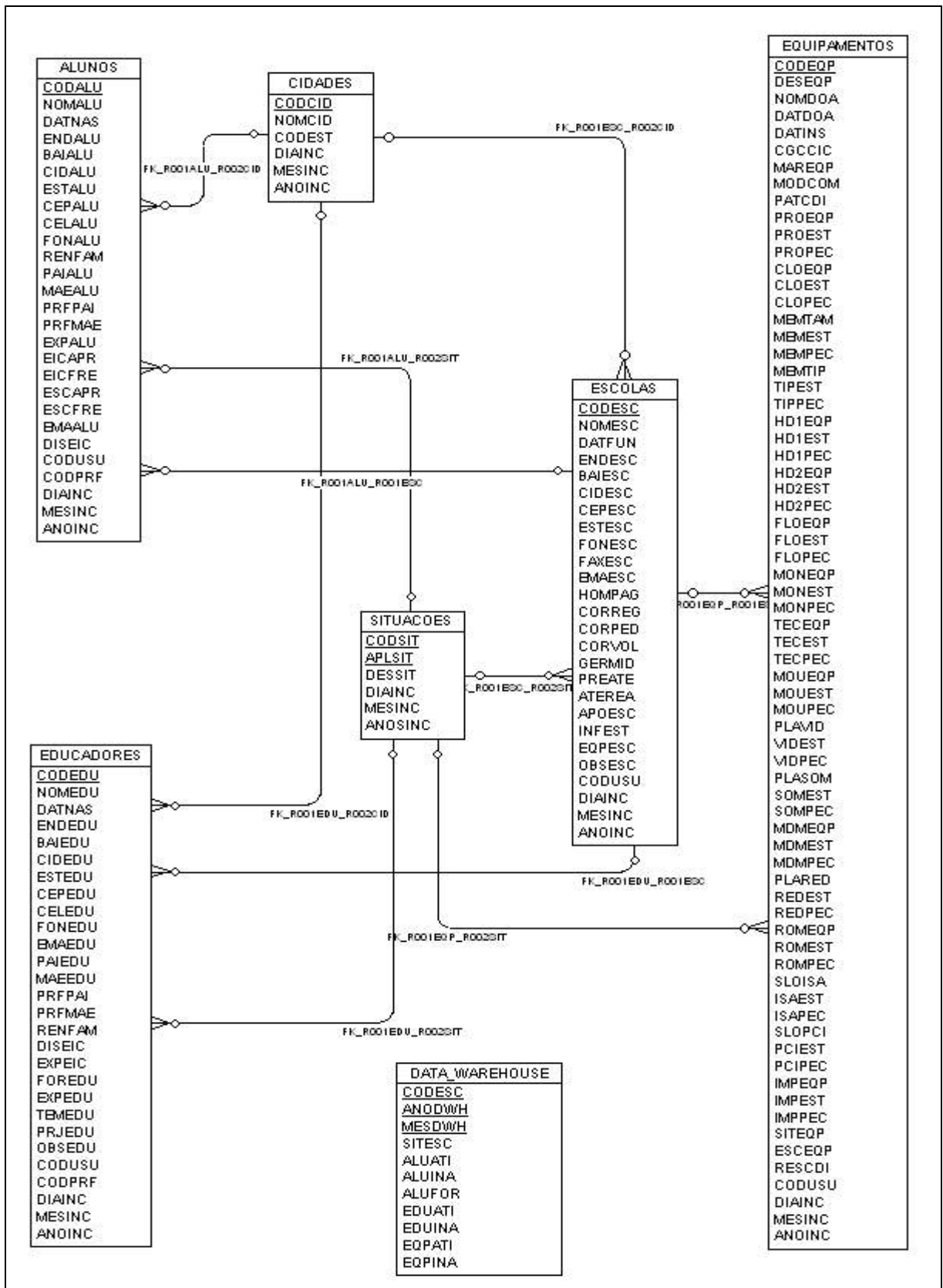


FIGURA 13 – Modelo Entidade Relacionamento (MER)

#### 5.1.4 DIAGRAMA DE FLUXO DE DADOS (DFD)

Segundo Pressman (1995), um diagrama de fluxo de dados (DFD) é uma técnica gráfica que descreve o fluxo de informação e as transformações que são aplicadas à medida que os dados se movimentam da entrada para a saída. As figuras 14, 15 e 16 mostram o DFD do sistema desenvolvido.

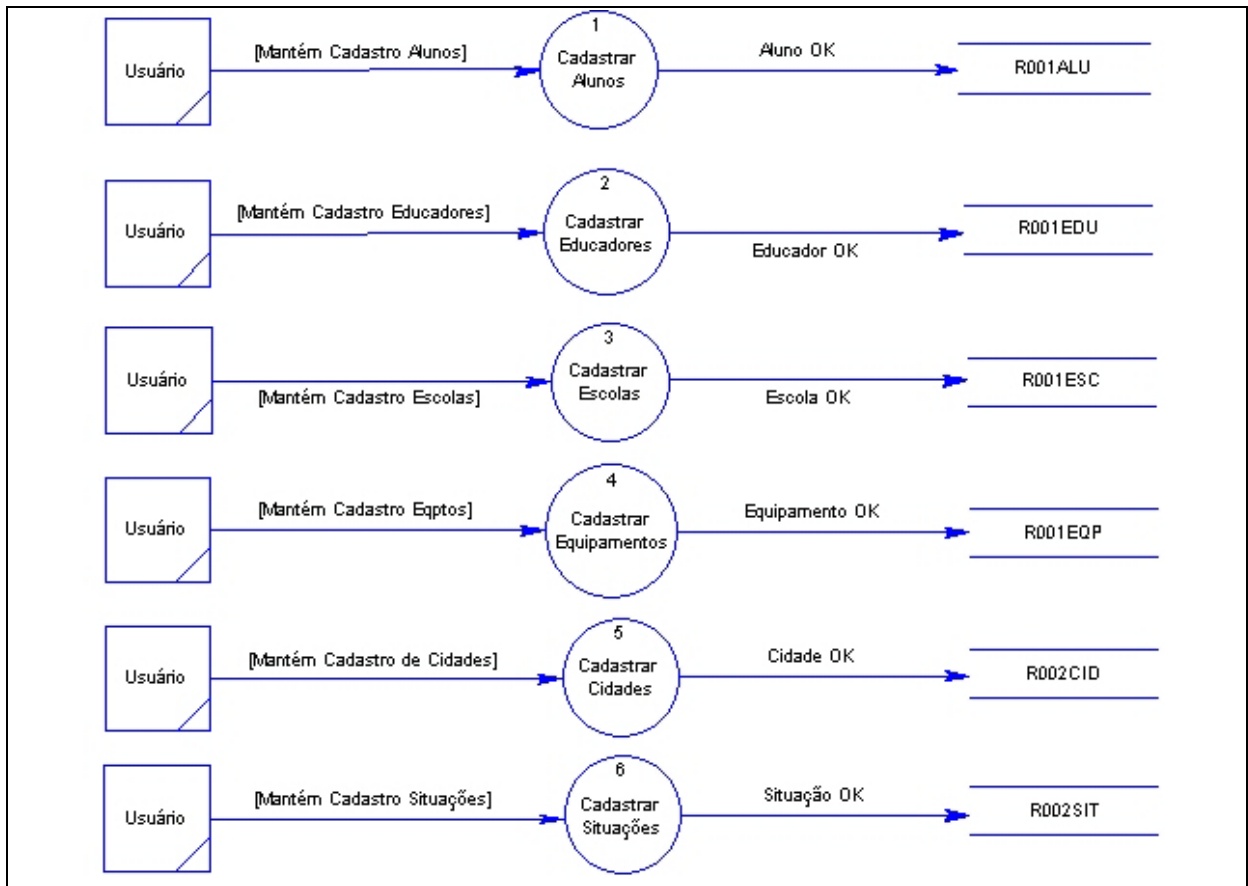


FIGURA 14 - DFD da carga de dados

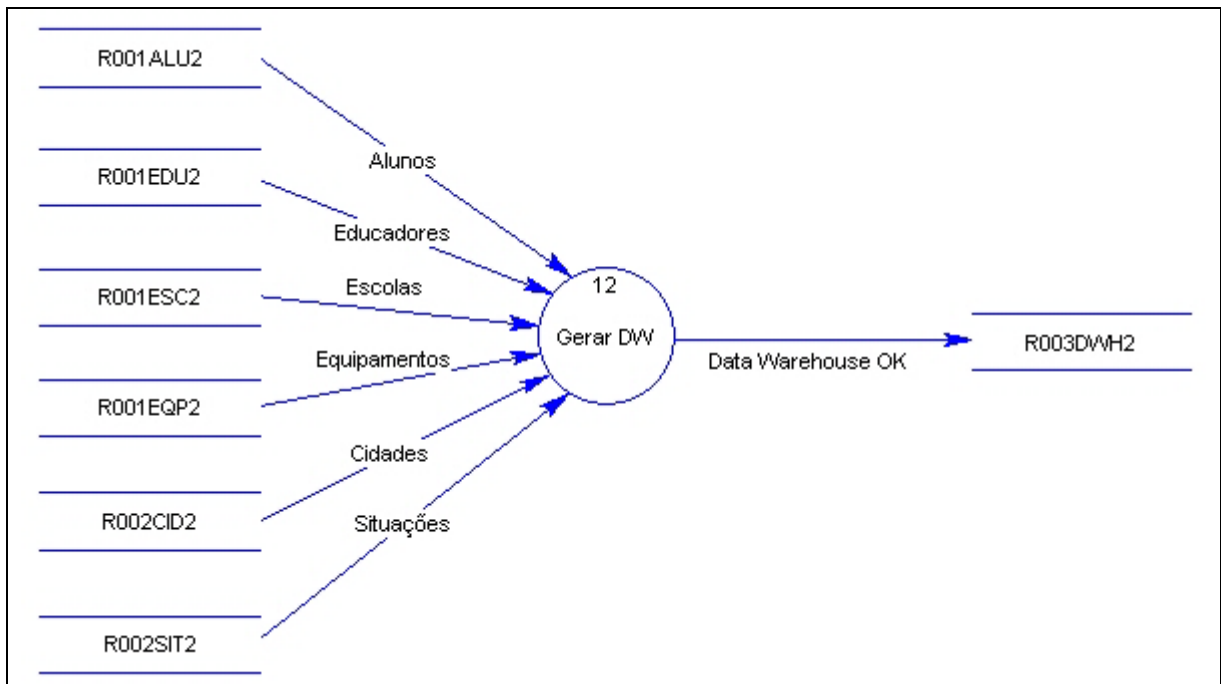


FIGURA 15 - DFD da geração do *Data Warehouse*

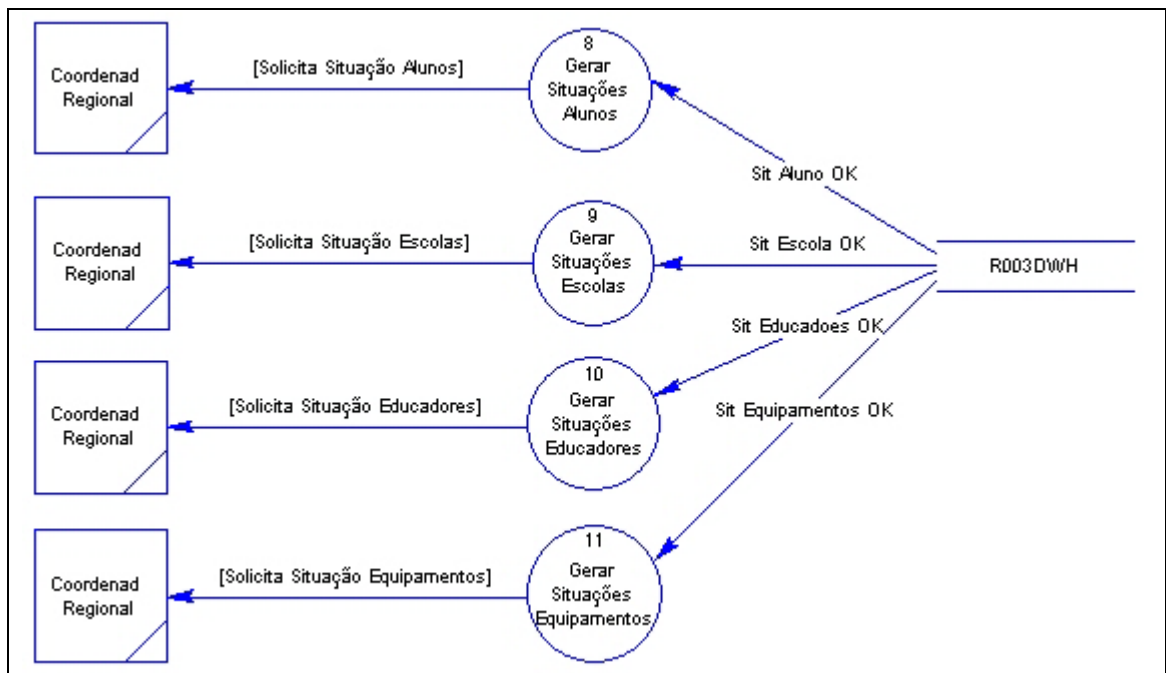


FIGURA 16 - DFD da geração de relatórios

### 5.1.5 DICIONÁRIO DE DADOS

No anexo A é mostrado o dicionário de dados, contendo as estruturas das tabelas do sistema, que são:

- a) Alunos: nesta tabela é mantido o cadastro de alunos;
- b) Educadores: nesta tabela é mantido o cadastro de educadores;
- c) Escolas: nesta tabela é mantido o cadastro de escolas;
- d) Equipamentos: nesta tabela é mantido o cadastro de equipamentos;
- e) Cidades: nesta tabela é mantido o cadastro de cidades;
- f) Situações: nesta tabela é mantido o cadastro de situações;
- g) Data\_Warehouse: nesta tabela são mantidos os dados do DW.

O dicionário de dados é composto pelas seguintes colunas:

- a) “Name”: indica o nome do campo (coluna) definido na tabela;
- b) “Type”: indica o tipo do campo definido na tabela;
- c) “I”: indica se o campo é um índice da tabela ou não;
- d) “M”: indica se o campo é mandatório (obrigatório) na tabela ou não.

## 5.2 IMPLEMENTAÇÃO

O desenvolvimento deste trabalho é baseado na Metodologia Baseada em Dados, proposta por Inmon (1997), para o desenvolvimento de um DW. Sendo assim, serão seguidas e analisadas as 11 atividades propostas por esta metodologia.

### 5.2.1 ANÁLISE DO MODELO DE DADOS

A análise do modelo de dados existente é o primeiro passo antes da construção do DW.

O modelo de dados do sistema pode ser considerado de fácil administração. O sistema possui as tabelas de cadastramento Alunos, Educadores, Escolas, Equipamentos, Cidades, Situações e uma tabela para manter as informações do DW. O usuário mantém a base operacional e o Coordenador Regional do CDI mantém o DW atualizado.



### 5.2.2 DIMENSIONAMENTO

O volume de dados do sistema é grande, por se tratar de um sistema via *Internet* onde o CDI cadastra seus alunos, escolas, educadores e equipamentos.

Segundo Inmon (1997), a pior postura que pode ser adotada é a de tentar projetar antecipadamente todos os níveis de granularidade do DW. É necessária a criação de uma pequena parte do DW e deixar o usuário acessar os dados. A seguir, ouvir o mesmo colhendo as informações para, só a partir daí, ajustar o nível de granularidade.

Como a granularidade está relacionada ao nível de detalhe ou de resumo dos dados contidos no DW, será feita uma granularidade alta do DW. Os dados terão um baixo nível de resumo, isso devido ao caráter acadêmico do trabalho, onde será avaliada a geração do DW e não a análise dos dados produzidos pela mesma.

### 5.2.3 AVALIAÇÃO TÉCNICA

Ao verificar qual a melhor tecnologia a ser usada na geração de um DW, deve-se analisar vários aspectos relacionados ao mesmo que são apresentados pelos bancos de dados. Os bancos de dados compatíveis com o sistema desenvolvido são quaisquer bancos compatíveis com o padrão de acesso ODBC. Mesmo que a maioria dos bancos utilizados pelo sistema possuir ferramentas para construção e migração para o DW e, além disso, todos preencherem os requisitos básicos necessários para dar suporte ao mesmo, não será utilizada nenhuma ferramenta para construção do DW. Isso evita a vinculação a tecnologias específicas e contempla um dos objetivos do trabalho que é a geração do DW.

### 5.2.4 PREPARAÇÃO DO AMBIENTE TÉCNICO

Para o desenvolvimento do DW proposto no trabalho, torna-se necessário os seguintes equipamentos:

- a) banco de dados compatível com o padrão de acesso ODBC para suportar a tecnologia ASP;
- b) servidor com banco de dados conforme acima mencionado, IIS, suporte para ASP e *Windows* NT ou *Windows* 2000;
- c) ambiente de rede.

Para suportar o volume de informações trafegáveis no servidor, o mesmo deve possuir no mínimo a seguinte configuração:

- a) processador Pentium III – 500;
- b) memória de 128 Mb (*Mega Bytes*);
- c) suporte a rede;
- d) disco rígido de 20 Gb (*Giga Bytes*).

### 5.2.5 ANÁLISE DAS ÁREAS DE INTERESSE

Uma grande necessidade do CDI é controlar o índice de evasão dos alunos, manutenção dos equipamentos, alocação de educadores e situação das escolas. Sendo assim, serão consolidadas informações referentes às escolas, com base na situação de alunos, educadores, equipamentos e até mesmo das escolas.

O CDI definiu as seguintes informações como sendo de grande importância para satisfazer as suas necessidades anteriormente citadas:

- a) percentual de alunos por situação: índice que retrata as situações “Ativo”, “Inativo” e “Formado”;
- b) percentual de educadores por situação: índice que retrata as situações “Ativo” e “Inativo”;
- c) percentual de equipamentos por situação: índice que retrata as situações “Ativo” e “Inativo”;
- d) percentual de escolas por situação: índice que retrata as situações “Ativo” e “Inativo”.

### 5.2.6 PROJETO DO DATA WAREHOUSE

Apesar de ser teoricamente possível tentar construir um ambiente DW sem um modelo de dados, isto é como tentar navegar sem uma carta de navegação (Inmon, 1997).

Sendo assim, é possível perceber que o modelo de dados ou projeto físico do DW deve ser desenvolvido levando em consideração as áreas de interesse da empresa. O dicionário de dados desenvolvido pode ser visto no Anexo A.

As entidades definidas para o modelo foram:

- a) Aluno: cadastro de alunos;

- b) Educador: cadastro de educadores;
- c) Escola: cadastro de escolas;
- d) Equipamento: cadastro de equipamentos;
- e) Cidade: cadastro de cidades;
- f) Situação: cadastro de situações;
- g) Data\_Warehouse: geração do DW.

### 5.2.7 ANÁLISE DO SISTEMA FONTE

Fazendo a análise das áreas de interesse do sistema transacional, de onde os dados a serem extraídos são provenientes, e a base do DW, pode-se notar que não será necessária uma grande transformação nos dados a serem extraídos, sendo apenas realizadas operações aritméticas básicas como somas e multiplicações. Conforme verificado na definição do modelo de dados do DW foi inserido a variável tempo para determinar o período a que pertencem os dados.

### 5.2.8 ESPECIFICAÇÃO DE PROGRAMAS

Não foi utilizada nem desenvolvida alguma ferramenta específica para a geração do DW. Apenas foram utilizadas as seguintes ferramentas de apoio do *SQL Server*

- a) *Enterprise Manager*: ferramenta utilizada para a criação de tabelas, definição dos campos (seus tipos e propriedades), criação dos relacionamentos entre as tabelas e manutenção de toda a estrutura das tabelas;
- b) *Query Analyzer*: editor de sql utilizado para a manutenção da estrutura das tabelas contidas na base de dados e para executar ações de inserção, exclusão e alteração de dados da mesma.

### 5.2.9 ELABORAÇÃO DE PROGRAMAS

Para a geração do DW foi desenvolvida uma rotina que tem como objetivo criar a base de dados do DW e posteriormente consultá-la. O quadro 14 mostra o trecho principal da rotina de geração das informações do DW.

```

'===== ALUMOS
SQL_Alu = "SELECT COUNT(CODALU) QTDALU, CODSIT FROM R001ALU WHERE CODESC = " & _
obj_Recordset_Esc("CodEsc") & " GROUP BY CODSIT ORDER BY CODSIT"
obj_Recordset_Alu.Open SQL_Alu, conexao
If NOT(obj_Recordset_Alu.EOF) Then
  Do While NOT(obj_Recordset_Alu.EOF)
    if (obj_Recordset_Alu("CodSit") = "1") Then
      xqtd_aluati = obj_Recordset_Alu("QtdAlu")
    elseif (obj_Recordset_Alu("CodSit") = "99") Then
      xqtd_alufor = obj_Recordset_Alu("QtdAlu")
    else
      xqtd_aluina = xqtd_aluina + obj_Recordset_Alu("QtdAlu")
    End If
    obj_Recordset_Alu.MoveNext
  Loop
End If
obj_Recordset_Alu.Close

...

'===== INSERÇÃO
obj_Recordset_Dwh.Open "R003DWH", conexao, , 3, 2
obj_Recordset_Dwh.AddNew
obj_Recordset_Dwh("CODESC") = obj_Recordset_Esc("CodEsc")
obj_Recordset_Dwh("ANODWH") = xano
obj_Recordset_Dwh("MESDWH") = xmes
obj_Recordset_Dwh("SITESC") = obj_Recordset_Esc("CodSit")
obj_Recordset_Dwh("ALUATI") = xqtd_aluati
obj_Recordset_Dwh("ALUINA") = xqtd_aluina
obj_Recordset_Dwh("ALUFOR") = xqtd_alufor
obj_Recordset_Dwh("EDUATI") = xqtd_eduati
obj_Recordset_Dwh("EDUINA") = xqtd_eduina
obj_Recordset_Dwh("EQPATI") = xqtd_eqpati
obj_Recordset_Dwh("EQPINA") = xqtd_eqpina

obj_Recordset_Dwh.Update
obj_Recordset_Dwh.Close

```

QUADRO 14 – Rotina de geração das informações do DW

Essa rotina deverá ser executada pelo Coordenador Regional do CDI ou pelo administrador do DW mensalmente, sendo que a mesma possuirá dois processos básicos:

- a) criação da base de dados do DW;
- b) atualização da base de dados baseada na situação atual do momento em que se inicia o processo de atualização.

#### 5.2.10 POVOAMENTO

Esta atividade é determinada pela entrada em produção dos programas que compõem o projeto. Isso ocorre devido ao DW estar povoado com dados de qualidade e plenamente apto para uso. Com isso o nível de utilização do DW cresce e conseqüentemente são proporcionadas facilidades de acesso e as expectativas dos usuários são atendidas.

#### 5.2.11 UTILIZAÇÃO DOS DADOS

Nesta última fase, as necessidades dos usuários são atendidas com a geração de relatórios gráficos e consultas. Ocorre a utilização satisfatória do DW para atender as questões estratégicas e gerenciais.

Para ter-se um melhor entendimento da implementação do sistema, a seguir é mostrado as técnicas e ferramentas utilizadas e sua operacionalidade.

### 5.3 TÉCNICAS E FERRAMENTAS UTILIZADAS

Visando uma melhor compreensão da pesquisa como um todo, a seguir são apresentadas as técnicas e ferramentas utilizadas durante o desenvolvimento do trabalho.

#### 5.3.1 DREAMWEAVER MX

Para o desenvolvimento do sistema *Web* foi utilizado o *Dreamweaver* MX da *Macromedia*. Essa poderosa ferramenta com código direcionado ao mundo da HTML, trás uma interface de apontar e clicar para a codificação complexa do *javascript*. O *Dreamweaver* é um programa profissional, para desenvolvimento de sites da *Web* (Lowery, 2001). O *Dreamweaver* possui também suporte para comunicação com banco de dados através da tecnologia ASP.

#### 5.3.2 INTERNET EXPLORER

Todos os testes e validações do sistema *Web* desenvolvido foram feitos através do *Internet Explorer* 6.0. O mais utilizado *browser* de navegação na *Internet* foi desenvolvido pela *Microsoft Corporation*.

#### 5.3.3 BANCO DE DADOS MICROSOFT DATA ENGINE (MSDE)

A base contendo as informações necessárias para a geração do DW foi criada no banco de dados *Microsoft Data Engine* (MSDE). O MSDE é uma versão reduzida do MS-SQL *Server* 7. Suas vantagens em relação ao MS-SQL *Server* é que, desde que seja desenvolvido com alguma ferramenta do *Visual Studio* 6.0, ele pode ser distribuído gratuitamente junto com a aplicação. Quando a aplicação exige mais do banco de dados (mais usuários concorrentes, mais espaço de armazenamento), então é necessário fazer um *upgrade* para o MS-SQL *Server*, sem que isso exija sequer uma recompilação da aplicação (Senior, 2002).

As principais características do MSDE são:

- a) totalmente compatível com o MS-SQL *Server*7 (possui o mesmo *kernel* do MS-SQL *Server*7);

- b) pode ser distribuído gratuitamente para aplicações desenvolvidas com MSDE for *Visual Studio 6.0* (edições *Professional* e *Enterprise*) ou *MS-Office 2000 Developer*;
- c) compatível com plataformas *MS-Windows 95, 98, NT e 2000*;
- d) possui arquitetura cliente-servidor (as operações ocorrem no servidor e não no cliente);
- e) pode ser executado como um serviço (ou seja, um processo *background*).

Limitações:

- a) não inclui uma interface de usuário para desenvolvimento em banco de dados. Em vez disso, outras aplicações como o *MS-Access 2000* ou aplicações em *Visual Basic* fornecem essa interface;
- b) as bases estão limitadas a um máximo de 2 *Gigabytes* (Gb) de dados. Esse limite é por base, e não por servidor, sendo que um servidor MSDE pode suportar múltiplas bases MSDE, cada uma contendo até 2 Gb;
- c) é otimizado para um máximo de 5 usuários concorrentes (não há nada que impeça a sua utilização com um número maior de usuários, exceto pela diminuição da performance).

Foi utilizado o banco MSDE neste trabalho por ser um banco distribuído gratuitamente e que utiliza base local. Nada impede que o sistema *Web* se comunique com outros bancos, desde que estes sejam compatíveis com o padrão de acesso a banco de dados ODBC.

Para criar tabelas e fazer manutenção da base foi usado o *Enterprise Manager*, do MSDE, conforme mostra a fig. 17. Para consultar os dados na base foi utilizado o *Query Analyzer* do MSDE.

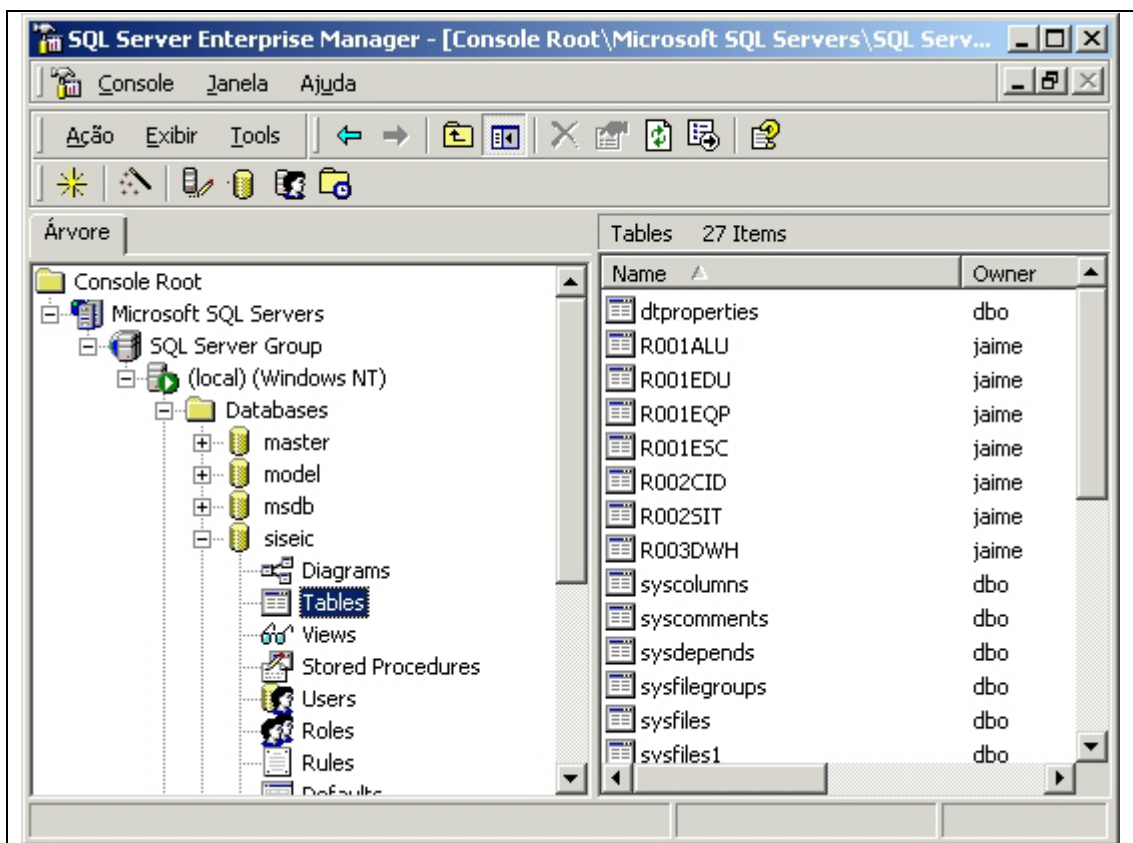


FIGURA 17 – Enterprise Manager do MSDE

#### 5.3.4 FERRAMENTA CASE POWER DESIGNER

A ferramenta *CASE PowerDesigner - Data Architect / Process Analyst* é usada para a especificação, visualização, documentação e construção do sistema, onde irá gerar todas as tabelas para o banco de dados. (Grahl, 1998). A modelagem de dados com base na especificação estruturada foi desenvolvida através desta ferramenta.

#### 5.4 OPERACIONALIDADE DA IMPLEMENTAÇÃO

Neste item, será abordado o sistema *Web* que é o objetivo geral deste trabalho, mostrando a sua interface e o seu funcionamento. A fig. 18 representa o fluxo de navegação do sistema desenvolvido.

| SISTEMA                    |                            |
|----------------------------|----------------------------|
| Usuário                    | Coordenador                |
| Cadastra Alunos            | Cadastra Alunos            |
| Cadastra Educadores        | Cadastra Educadores        |
| Cadastra Escolas           | Cadastra Escolas           |
| Cadastra Equipamentos      | Cadastra Equipamentos      |
| Gera Relatórios Gráficos   | Cadastra Cidades           |
| Gera Relatórios Analíticos | Cadastra Situações         |
|                            | Gera Data Warehouse        |
|                            | Gera Relatórios Gráficos   |
|                            | Gera Relatórios Analíticos |

FIGURA 18 – Fluxo de navegação do sistema

#### 5.4.1 APRESENTAÇÃO DO SISTEMA

A tela inicial do sistema contém o menu de acesso a todas as operações do sistema no lado esquerdo. No lado direito serão mostradas as telas de cadastros e processamentos. A tela pode ser vista conforme fig. 19.

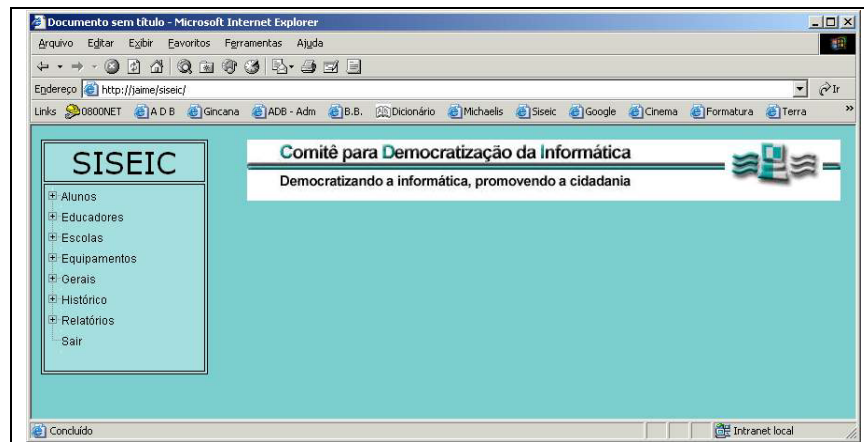


FIGURA 19 – Tela principal do sistema

Através do menu *treeview* da tela principal é possível acessar todas as telas de cadastro e processamentos do sistema.



A fig. 20 mostra um exemplo, a tela de Cadastro de Situações, acessada através do menu.

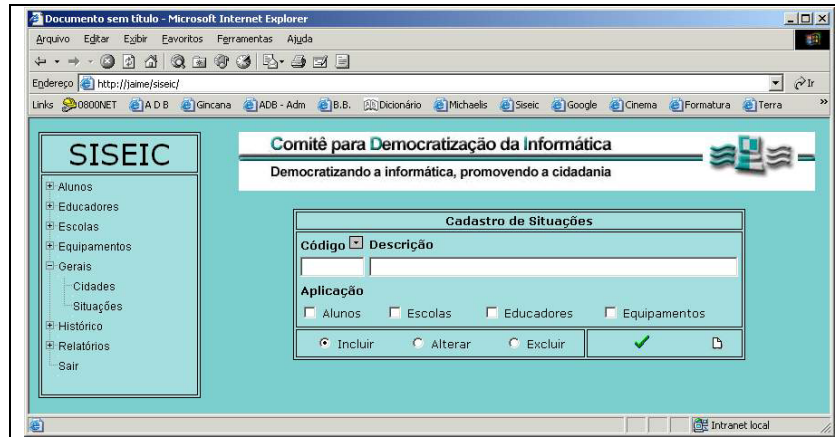


FIGURA 20 – Tela de Cadastro de Situações

Conforme pode ser visto na fig. 21, através do menu “Histórico”, “Gerar” são disponibilizados dois *links*:

- a) visualizar histórico: lista um resumo de todas as gerações feitas;
- b) gerar: executa a rotina que gera informações para a base de dados do DW baseada na data atual da geração.



FIGURA 21 – Acesso ao link de geração do DW

No link “Relatórios”, “Gerar” é feito o acesso a geração dos relatórios gráficos. A fig. 22 mostra a tela de opções do usuário, onde é possível escolher o tipo de relatório e o período que se deseja visualizar.

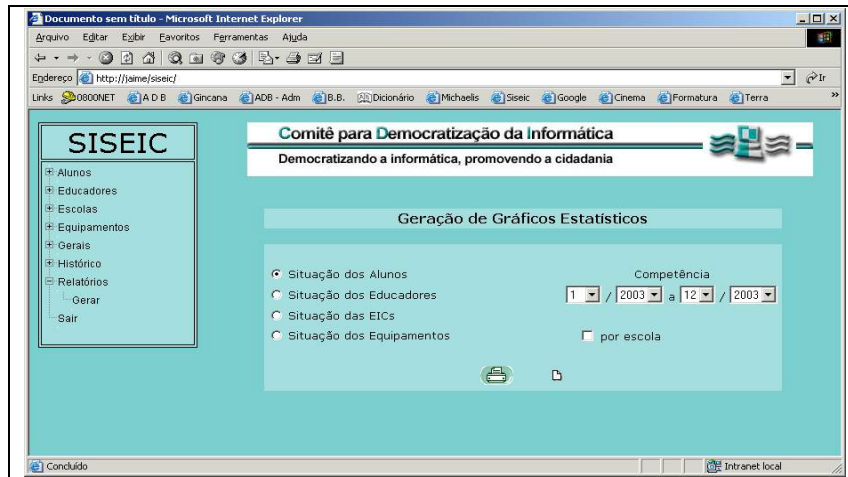


FIGURA 22 – Tela de opções para a geração de gráficos

Após selecionadas as opções do usuário, é gerado o gráfico com os percentuais de acordo com o período informado, conforme pode ser visto na fig. 23.

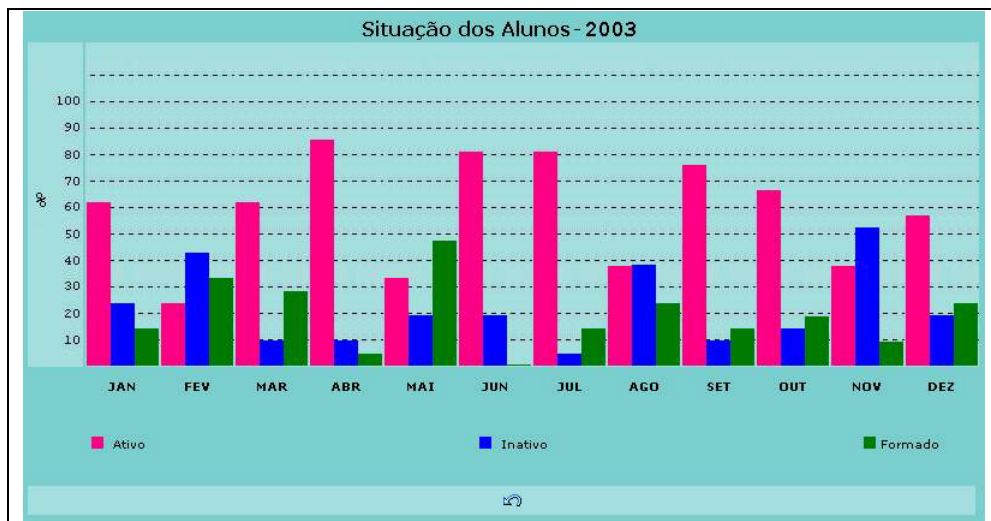


FIGURA 23 – Gráfico estatístico sobre a situação dos alunos

A figura 24 mostra a geração de outro gráfico, com um período menor informado.

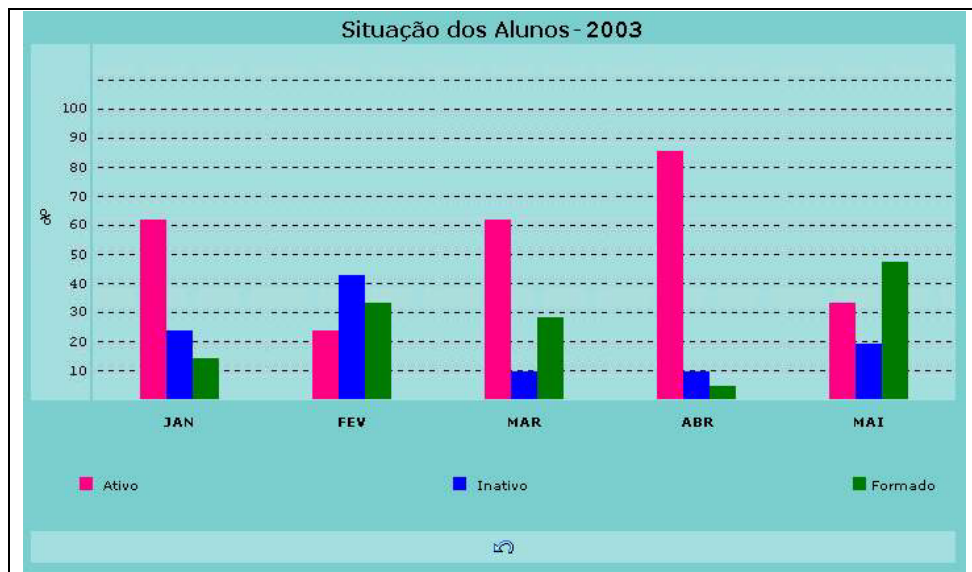


FIGURA 24 – Outro exemplo de gráfico estatístico

Um exemplo de relatório analítico pode ser visto na fig. 25.

| Relatório Analítico de Alunos |                   |                              |
|-------------------------------|-------------------|------------------------------|
| <b>Escola:1</b>               |                   |                              |
| Código                        | Nome              | Situação                     |
| 2                             | Bruno             | Desistente - motivo de saúde |
| 6                             | Felipe            | Ativo                        |
| 15                            | Pualinha          | Ativo                        |
| 20                            | Urda              | Ativo                        |
| <b>Escola:2</b>               |                   |                              |
| Código                        | Nome              | Situação                     |
| 3                             | Carlinhos         | Formado                      |
| 7                             | Guilherme         | Ativo                        |
| 8                             | Hernandes         | Ativo                        |
| 9                             | Ilda              | Ativo                        |
| 16                            | Querubin          | Ativo                        |
| 17                            | Rui               | Ativo                        |
| 22                            | Anderson de Souza | Desistente - arrumou emprego |

FIGURA 25 – Exemplo de relatório Analítico

É importante ressaltar que todos os relatórios gráficos foram gerados com tecnologia ASP, sem a utilização de componentes gráficos ou ferramentas de apoio. Os quadros 15, 16 e 17 mostram as etapas das rotinas de geração destes relatórios.

```
'===== CARREGA AS BARRAS DE PERCENTUAIS
xtamanho_celula = Cint(100/ xqtd_meses)
xind = 1
Do While (xind <= xqtd_meses)
  percentual = xtot_aluati(xind) * 100 / xttotal_geral
  xpercentual_Barra = percentual * 220 / 100
  xind_barra = 1
  Response.Write "<td width='" & Cstr(xtamanho_celula) & "%' height='270'" & _
    "valign='bottom' background='img/fundo_grafico_linhas.JPG'" & _
    "<div align='center'" & _
    "<input name='imgBarAlu' type='image' src='img/Barra'" & xind_barra & _
    ".JPG' align='bottom' width='20' height='" & _
    xpercentual_barra & "' border='1'" & _

  percentual = xtot_aluina(xind) * 100 / xttotal_geral
  xpercentual_Barra = Cint(percentual * 220 / 100)
  xind_barra = 2

  Response.Write "<input name='imgBarAlu' type='image' src='img/Barra'" & xind_barra & _
    ".JPG' align='bottom' width='20' height='" & _
    xpercentual_barra & "' border='1'" & _

  percentual = xtot_alufor(xind) * 100 / xttotal_geral
  xpercentual_Barra = percentual * 220 / 100
  xind_barra = 3

  Response.Write "<input name='imgBarAlu' type='image' src='img/Barra'" & xind_barra & _
    ".JPG' align='bottom' width='20' height='" & _
    xpercentual_barra & "' border='1'" & _

  xind = xind + 1
Loop
```

QUADRO 15 – Gráfico: geração das barras

```
'===== CARREGA OS NOME DOS MESES
xind = 1
Do While (xind <= xqtd_meses)
  Response.Write "<td width='" & Cstr(xtamanho_celula) & "%' height='21'" & _
    "valign='bottom'" & _
    "<div align='center'" & _
    "<strong'" & _
    "<font size='1' face='Verdana, Arial, Helvetica, sans-serif'" & _
    arr_meses(xind) & "</font></strong></div></td>"

  xind = xind + 1
Loop
```

QUADRO 16 – Gráfico: geração das informações do eixo horizontal

```
'===== CARREGA A LEGENDA
xind = 1
Do While (xind <= 3)
  Response.Write "<td width='5%' height='22'" & _
    "<div align='center'" & _
    "<img src='img/Bar'" & Cstr(xind) & ".JPG' width='10' height='10'" & _
    "</div></td><td width='45%" & _
    "<font size='1' face='Verdana, Arial, Helvetica, sans-serif'" & _
    arr_dessit(xind) & "</font> </td>"

  xind = xind + 1
Loop
```

QUADRO 17 – Gráfico: geração da legenda

## 6 CONCLUSÕES

No presente trabalho foram abordados diversos fatores envolvidos no projeto e no desenvolvimento de um sistema de DW. Como se pôde notar, foram descritos diversos aspectos sobre o projeto e o desenvolvimento de um sistema de DW, foram identificadas as funções dos integrantes da equipe de projeto e desenvolvimento. Foram citados também os passos a serem seguidos durante o projeto além das regras a seguir para atingir os objetivos propostos. E ainda, conhecendo alguns dos problemas mais freqüentes da fase de desenvolvimento dos sistemas de apoio à decisão pode-se ter uma idéia do que se irá encontrar na implantação deste tipo de sistema.

A metodologia de desenvolvimento de um DW baseada em dados, conforme proposto por Inmon, é de fácil entendimento e deixa bem claro os passos a serem seguidos para que seus objetivos sejam alcançados.

Outro assunto que, assim como o DW, é de relevância computacional que foi tratado neste trabalho, foi o uso da tecnologia ASP. Foram descritos os diversos passos e pré-requisitos para a utilização desta tecnologia. Foram relatadas suas restrições, técnicas utilizadas, comandos, objetos, enfim, foram abordados os principais assuntos necessários para habilitar o leitor a construir sua página dinâmica com acesso à banco de dados.

A tecnologia ASP é compatível com a maioria dos bancos de dados e sistemas operacionais existentes, e através de seus objetos e métodos se torna simples para desenvolver uma página dinâmica para a *Web*.

Pode-se concluir que, os objetivos do trabalho foram atendidos, pois ao longo de todo o processo de desenvolvimento do mesmo, foi possível desenvolver o sistema via *Web* através do conhecimento adquirido no que diz respeito a um DW e o uso de tecnologia ASP.

### 6.1 EXTENSÕES

Como sugestão para trabalhos futuros, pode-se desenvolver outros recursos para controles administrativos do CDI na área financeira ou na captação de recursos.

## REFERÊNCIAS BIBLIOGRÁFICAS

BAPTISTA, Evaristo. **Alternativas de migração para ambientes Data Warehouse**. 1998. 64 f. Monografia apresentada ao curso de pós-graduação (Tecnologias de Desenvolvimento de Sistemas) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

BINDER, Fabio Vinícios. **Sistemas de apoio à decisão**. São Paulo: Érica, 1994.

CAMPOS, Maria Luiza; FILHO, Arnaldo V. Rocha. **Data warehouse** Rio de Janeiro, [2000?]. Disponível em: <<http://genesis.nce.ufrj.br/dataware/tutorial/indice.html>>. Acesso em: 15 out. 2003.

CDI, Comitê para Democratização da Informática. **Democratizando a informática, promovendo a cidadania**, [S.L.], [2003]. Disponível em: <<http://www.cdi.org.br>>. Acesso em: 17 out. 2003.

CRUZ, Tadeu. **Sistemas de informações gerenciais: tecnologia da informação e a empresa do século XXI**. São Paulo: Atlas, 1998.

DALFOVO, Oscar; AMORIM, Sammy Newton. **Quem tem informação é mais competitivo**. Blumenau: Acadêmica, 2000.

FIDELIS, Jean Carlos. **Sistema de informação para o comitê para democratização da informática**. 2003. 50 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

FURLAN, José Davi; IVO, Ivonildo da Motta; AMARAL, Francisco Piedade. **Sistemas de informações executivas**. São Paulo: Makron Books, 1994. jaimex

GRAHL, Everaldo Artur. **Tutorial do Power Designer- Process Analyst**. Blumenau, [1998?]. Disponível em: <<http://www.inf.furb.br/~egrahl/tutoriais/>>. Acesso em: 27 out 2003.

ICMC, Instituto de Ciências Matemáticas e de Computação. **Tutorial HTML**, São Carlos, [2003?]. Disponível em: <<http://www.icmc.usp.br/ensino/material/html/>>. Acesso em: 28 out. 2003.

INMON, Willian H. **Como construir o Data Warehouse**. 2. ed. Tradução Ana Maria Netto Guz. Rio de Janeiro: Campus, 1997.

LOWERY, Joseph. **Dreamweaver 4, a Bíblia**. 1. ed. Tradução DocWare Traduções Técnicas. Rio de Janeiro: Campus, 2001. 934 p.

MENGARDA, Mariane T. **Definição de um roteiro de preparação para implantação de um Data Warehouse**. 1998. 91 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

MITCHELL, Scott; Atkinson, James. **Aprenda em 21 dias ASP 3.0 (Active Server Pages)**. 1. ed. Tradução DocWare Traduções Técnicas. Rio de Janeiro: Campus, 2000.

OLIVEIRA, Djalma de Pinho Rebouças. **Sistemas de informações gerenciais**. São Paulo: Atlas 1992.

ORR, Ken. **Data Warehouse technologic**, The Ken Orr Institute, 2001. Disponível em: <<http://www.kenorrinst.com/pg%2011%20data%20warehousing.htm>>. Acesso em: 17 set. 2003.

PRATES, Maurício. Conceituação de sistemas de informação do ponto de vista do gerenciamento. **Revista do instituto de Informática. PUC-CAMP**, março/setembro, 1994.

PRESSMAN, Roger S. **Engenharia de Software**. 1. ed. Tradução José Carlos Barbosa dos Santos. São Paulo: Makron Books, 1995.

SENIOR SISTEMAS LTDA. **MSDE (Microsoft Data Engine) for Visual Studio 6.0**. Fabio Fernando dos Santos: Senior Sistemas, 2002.

VILLANOVA, Vanderlei. **Construindo um Data warehouse**, Curitiba, set. 2000. Disponível em: <<http://www.datawarehouse.hpg.ig.com.br>>. Acesso em: 01 nov. 2003.

WILLE, Cristoph. **Aprenda em 24 horas Active Server Pages**. Rio de Janeiro: Campus, 1999.

## ANEXO A – DICIONÁRIO DE DADOS

### TABELA DE ALUNOS - R001ALU

| Name   | Type  | I   | M   |
|--------|-------|-----|-----|
| CODALU | N4    | Yes | Yes |
| NOMALU | VA40  | No  | Yes |
| DATNAS | DT    | No  | No  |
| ENDALU | VA50  | No  | No  |
| BAIALU | VA30  | No  | No  |
| CIDALU | VA40  | No  | No  |
| ESTALU | VA2   | No  | No  |
| CEPALU | VA9   | No  | No  |
| CELALU | VA15  | No  | No  |
| FONALU | VA15  | No  | No  |
| RENFAM | F53   | No  | No  |
| PAIALU | VA40  | No  | No  |
| MAEALU | VA40  | No  | No  |
| PRFPAI | VA50  | No  | No  |
| PRFMAE | VA50  | No  | No  |
| EXPALU | VA200 | No  | No  |
| EICAPR | F53   | No  | No  |
| EICFRE | F53   | No  | No  |
| ESCAPR | F53   | No  | No  |
| ESCFRE | F53   | No  | No  |
| EMAALU | VA40  | No  | No  |
| DISEIC | VA15  | No  | No  |
| CODUSU | N4    | No  | No  |
| CODPRF | N4    | No  | No  |
| DIAINC | N2    | No  | No  |
| MESINC | N2    | No  | No  |
| ANOINC | N4    | No  | No  |

### TABELA DE EDUCADORES - R001EDU

| Name   | Type  | I   | M   |
|--------|-------|-----|-----|
| CODEDU | N4    | Yes | Yes |
| NOMEDU | VA40  | No  | Yes |
| DATNAS | DT    | No  | No  |
| ENDEDU | VA50  | No  | No  |
| BAIEDU | VA30  | No  | No  |
| CIDEDU | VA40  | No  | No  |
| ESTEDU | VA2   | No  | No  |
| CEPEDU | VA9   | No  | No  |
| CELEDU | VA15  | No  | No  |
| FONEDU | VA15  | No  | No  |
| EMAEDU | VA40  | No  | No  |
| PAIEDU | VA40  | No  | No  |
| MAEEDU | VA40  | No  | No  |
| PRFPAI | VA50  | No  | No  |
| PRFMAE | VA50  | No  | No  |
| RENFAM | F53   | No  | No  |
| DISEIC | VA15  | No  | No  |
| EXPEIC | VA250 | No  | No  |
| FOREDU | VA250 | No  | No  |
| EXPEDU | VA250 | No  | No  |
| TEMEDU | VA30  | No  | No  |
| PRJEDU | VA400 | No  | No  |
| OBSEDU | VA200 | No  | No  |



| Name   | Type | I  | M  |
|--------|------|----|----|
| CODUSU | N4   | No | No |
| CODPRF | N4   | No | No |
| DIAINC | N2   | No | No |
| MESINC | N2   | No | No |
| ANOINC | N4   | No | No |

### TABELA DE EQUIPAMENTOS - R001EQP

| Name   | Type | I   | M   |
|--------|------|-----|-----|
| CODEQP | N4   | Yes | Yes |
| DESEQP | VA60 | No  | Yes |
| NOMDOA | VA50 | No  | No  |
| DATDOA | DT   | No  | No  |
| DATINS | DT   | No  | No  |
| CGCCIC | VA20 | No  | No  |
| MAREQP | VA30 | No  | No  |
| MODCOM | VA30 | No  | No  |
| PATCDI | VA30 | No  | No  |
| PROEQP | VA30 | No  | No  |
| PROEST | VA1  | No  | No  |
| PROPEC | VA1  | No  | No  |
| CLOEQP | VA30 | No  | No  |
| CLOEST | VA1  | No  | No  |
| CLOPEC | VA1  | No  | No  |
| MENTAM | VA30 | No  | No  |
| MEMEST | VA1  | No  | No  |
| MEMPEC | VA1  | No  | No  |
| MENTIP | VA30 | No  | No  |
| TIPEST | VA1  | No  | No  |
| TIPPEC | VA1  | No  | No  |
| HD1EQP | VA30 | No  | No  |
| HD1EST | VA1  | No  | No  |
| HD1PEC | VA1  | No  | No  |
| HD2EQP | VA30 | No  | No  |
| HD2EST | VA1  | No  | No  |
| HD2PEC | VA1  | No  | No  |
| FLOEQP | VA30 | No  | No  |
| FLOEST | VA1  | No  | No  |
| FLOPEC | VA1  | No  | No  |
| MONEQP | VA30 | No  | No  |
| MONEST | VA1  | No  | No  |
| MONPEC | VA1  | No  | No  |
| TECEQP | VA30 | No  | No  |
| TECEST | VA1  | No  | No  |
| TECPEC | VA1  | No  | No  |
| MOUEQP | VA30 | No  | No  |
| MOUEST | VA1  | No  | No  |
| MOUPEC | VA1  | No  | No  |
| PLAVID | VA30 | No  | No  |
| VIDEST | VA1  | No  | No  |
| VIDPEC | VA1  | No  | No  |
| PLASOM | VA30 | No  | No  |
| SOMEST | VA1  | No  | No  |
| SOMPEC | VA1  | No  | No  |
| MDMEQP | VA30 | No  | No  |
| MDMEST | VA1  | No  | No  |
| MDMPEC | VA1  | No  | No  |
| PLARED | VA30 | No  | No  |

| Name   | Type | I  | M  |
|--------|------|----|----|
| REDEST | VA1  | No | No |
| REDPEC | VA1  | No | No |
| ROMEQP | VA30 | No | No |
| ROMEST | VA1  | No | No |
| ROMPEC | VA1  | No | No |
| SLOISA | VA30 | No | No |
| ISAEST | VA1  | No | No |
| ISAPEC | VA1  | No | No |
| SLOPCI | VA30 | No | No |
| PCIEST | VA1  | No | No |
| PCIPEC | VA1  | No | No |
| IMPEQP | VA30 | No | No |
| IMPEST | VA1  | No | No |
| IMPPEC | VA1  | No | No |
| SITEQP | VA60 | No | No |
| ESCEQP | VA40 | No | No |
| RESCDI | VA40 | No | No |
| CODUSU | N4   | No | No |
| DIAINC | N2   | No | No |
| MESINC | N2   | No | No |
| ANOINC | N4   | No | No |

### TABELA DE ESCOLAS - R001ESC

| Name   | Type  | I   | M   |
|--------|-------|-----|-----|
| CODESC | N4    | Yes | Yes |
| NOMESC | VA40  | No  | Yes |
| DATFUN | DT    | No  | No  |
| ENDESC | VA50  | No  | No  |
| BAIESC | VA30  | No  | No  |
| CIDESC | VA40  | No  | No  |
| CEPESC | VA9   | No  | No  |
| ESTESC | VA2   | No  | No  |
| FONESC | VA15  | No  | No  |
| FAXESC | VA15  | No  | No  |
| EMAESC | VA40  | No  | No  |
| HOMPAG | VA50  | No  | No  |
| CORREG | VA40  | No  | No  |
| CORPED | VA40  | No  | No  |
| CORVOL | VA40  | No  | No  |
| GERMID | VA200 | No  | No  |
| PREATE | VA40  | No  | No  |
| ATEREA | VA40  | No  | No  |
| APOESC | VA400 | No  | No  |
| INFEST | VA400 | No  | No  |
| EQPESC | VA400 | No  | No  |
| OBSESC | VA40  | No  | No  |
| CODUSU | N4    | No  | No  |
| DIAINC | N2    | No  | No  |
| MESINC | N2    | No  | No  |
| ANOINC | N4    | No  | No  |

### TABELA DE CIDADES - R002CID

| Name   | Type | I   | M   |
|--------|------|-----|-----|
| CODCID | N10  | Yes | Yes |
| NOMCID | VA40 | No  | Yes |
| CODEST | VA2  | No  | Yes |

| Name   | Type | I  | M  |
|--------|------|----|----|
| DIAINC | N2   | No | No |
| MESINC | N2   | No | No |
| ANOINC | N4   | No | No |

### TABELA DE SITUAÇÕES - R002SIT

| Name    | Type | I   | M   |
|---------|------|-----|-----|
| CODSIT  | N4   | Yes | Yes |
| APLSIT  | VA1  | Yes | Yes |
| DESSIT  | VA60 | No  | Yes |
| DIAINC  | N2   | No  | No  |
| MESINC  | N2   | No  | No  |
| ANOSINC | N4   | No  | No  |

### TABELA DO DATA WAREHOUSE - R003DWH

| Name   | Type | I   | M   |
|--------|------|-----|-----|
| CODESC | N4   | Yes | Yes |
| ANODWH | N4   | Yes | Yes |
| MESDWH | N2   | Yes | Yes |
| SITESC | N4   | No  | Yes |
| ALUATI | N6   | No  | No  |
| ALUINA | N6   | No  | No  |
| ALUFOR | N6   | No  | No  |
| EDUATI | N6   | No  | No  |
| EDUINA | N6   | No  | No  |
| EQPATI | N6   | No  | No  |
| EQPINA | N6   | No  | No  |