

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO E SISTEMAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE
PRODUÇÃO

IMPLEMENTAÇÃO DE UM DATA MART PARA O SETOR DE
PRODUÇÃO DE UMA INDÚSTRIA DE EMBALAGENS FLEXÍVEIS

MARCELO SARTOR

Dissertação apresentada ao programa de pós-graduação em Engenharia de Produção da Universidade Federal de Santa Catarina para obtenção do grau de mestre em Engenharia

Florianópolis, 2004

MARCELO SARTOR

**IMPLEMENTAÇÃO DE UM DATA MART PARA O SETOR DE
PRODUÇÃO DE UMA INDÚSTRIA DE EMBALAGENS
FLEXÍVEIS**

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

Florianópolis, 2004.

Prof. Edson Pacheco Paladini, Dr. Eng.
Coordenador do Curso

BANCA EXAMINADORA

Prof. Aran Bey Tcholakian Morales, Dr. Eng.
Orientador

Prof. Oscar Ciro López, Dr. Eng

Prof. Ricardo Villarroel Dávalos, Dr. Eng

Florianópolis, 2004

AGRADECIMENTOS

No decorrer deste trabalho, além da dificuldade natural de conciliar o estudo, o trabalho e a família, surgiram também outras dificuldades, de saúde e troca de emprego. Por isso, se hoje esse trabalho esta sendo apresentado, além do meu empenho pessoal, é graças a minha família que sempre esteve ao meu lado, nos momentos alegres e nos momentos difíceis, abdicando de inúmeros finais de semana e feriados.

Ao meu mestre e orientador Aran que no decorrer desta, teve paciência e compreensão aos meus problemas particulares me dando toda orientação que só um verdadeiro mestre possui. E a receptividade da Plasc – Indústria de Plásticos, que abriu as portas e seu banco de dados para que este trabalho fosse realizado.

A todos meu sincero obrigado.

RESUMO

Em uma indústria de embalagens flexíveis, assim como em qualquer organização, o setor produtivo é o que fornece o maior número de informações para a tomada de decisões, nas diversas áreas da empresa, sendo que nos dias atuais a eficiência e eficácia nas tomadas de decisões são fatores chave de sucesso. Neste sentido, a tecnologia da informação tem evoluído mais e mais a cada dia, avançando em todas as camadas hierárquicas da pirâmide organizacional. E é na camada decisória desta pirâmide que se situa o *Data Warehouse*. Este trabalho propõe a criação de um sistema de apoio à decisão baseado na tecnologia de *Data Warehouse*, para os níveis intermediários e superiores da pirâmide organizacional, do setor de produção de uma indústria de embalagens flexíveis. Tratam-se de um estudo de caso, cuja literatura aborda sistemas de informação, embalagens, além de apresentar, sucintamente o panorama de mercado e uma descrição do setor produtivo de uma indústria de embalagens flexíveis.

Palavras Chaves Sistemas de Informação, *Data Warehouse*, Apoio à Decisão, Embalagens Flexíveis.

Abstract

In a flexible packaging industry, as well as in any other industry, the productive sector is the one which supplies the largest amount of information for making decisions. Nowadays, the efficiency and efficacy in making decisions are important points to be taken into account for success. The information technology has evolved more and more each day. It has being spread in all over the hierarchical layers of the organizational pyramid of a company. The Data Warehouse is located in the decision layer of the pyramid. The present work proposes the creation of a support decision system based on Data Warehouse technology for the intermediate levels and superior ones of the organizational pyramid of the production sector in a flexible packaging industry. It is a case study, in which the general theoretical framework was based on system information, on packaging approach as well as on a brief overview of the market, and a description of the productive sector of a flexible packaging industry.

Key words: Information Systems, Date Warehouse, Decision Support, Flexible Packaging.

LISTA DE FIGURAS

Fig. 1: Metodologia de Pesquisa Utilizada.....	3
Fig. 2: Funções de um Sistema de Informação.....	6
Fig. 3: Componentes de um Sistema de Informação Computadorizado.....	7
Fig. 4 Pirâmide Organizacional – Os Diferentes Níveis das Organizações.....	8
Fig. 5 Pirâmide Organizacional – Os Diferentes Níveis das Organizações.....	8
Fig. 6: Arquitetura da Informação em uma Empresa.....	9
Fig. 7: Transação de Folha de Pagamento.....	10
Fig. 8: Diagrama de um Sistema de Informação Gerencial.....	11
Fig. 9: Elementos Essenciais de um SSD.....	12
Fig. 10: Pirâmide de Problemas.....	13
Fig. 11: Arquitetura Global.....	23
Fig. 12: Arquitetura de Data Marts Integrados.....	25
Fig. 13: Implementação Top_Down.....	27
Fig. 14: Implementação Buttom_Up.....	29
Fig. 15: Implementação Combinada.....	30

Fig. 16: Exemplo de uma Tabela de Fatos.....	32
Fig. 17: Star Schema.....	33
Fig. 18: Elementos Básicos de um Data Warehous.....	34
Fig. 19: Exemplo de um Cubo.....	39
Fig. 20: Crescimento e Tendências do Mercado de Embalagens.....	45
Fig. 21: Distribuição do Mercado de Embalagens em 2002.....	46
Fig. 22: Crescimento das Embalagens Plásticas no Brasil.....	47
Fig. 23: Organograma da PLASC.....	48
Fig. 24: Estrutura de Produção.....	49
Fig. 25: Estrutura de TI da PLASC.....	50
Fig. 26: Diagrama do Ciclo de Vida Dimensional do Negócio.....	51
Fig. 27: Fato Perda.....	63
Fig. 28: Fato Tempo de Parada.....	64
Fig. 29: Fato Produção.....	65
Fig. 30: Processo de ETL.....	67
Fig. 31: Esquema de Drill Down da Análise de Produção.....	70
Fig. 32: Cubo Produção Realizada.....	71

Fig. 33: Gráfico Produção Realizada – Nível 0 Setor de Impressão, Período	71
Fig 34: Gráfico Produção Realizada – Nível 1 Setor de Impressão, Período 07/2004, Classe de Produto.....	72
Fig. 35 Gráfico Produção Realizada – Nível 2 Setor de Impressão, Período 07/2004, Classe de Produto Pães, Operador.....	72
Fig. 36: Esquema de Drill Down da Análise de Tempo de Produção Parada.....	73
Fig. 37: Cubo Tempo de Produção Parada.....	74
Fig. 38. Gráfico Tempo de Produção Parada – Nível 0 Ano 2004, Período	74
Fig. 39: Gráfico Tempo de Produção Parada – Nível 1 Ano 2004, Período 07/2004, Classe de Produto	75
Fig.40 Gráfico Tempo de Produção Parada – Nível 2 Ano 2004, Período 07/2004, Classe de Produto Pães, Setor	75
Fig. 41: Gráfico Tempo de Produção Parada – Nível 3 Ano 2004, Período 07/2004, Classe de Produto Pães, Setor Impressão, Motivo.....	76
Fig. 42: Gráfico Tempo de Produção Parada – Nível 4 Ano 2004, Período 07/2004, Classe de Produto Pães, Setor Impressão, Motivo Aguardando Liberação, Recurso.....	76

Fig. 43: Gráfico Tempo de Produção Parada – Nível 5 Ano 2004, Período 07/2004, Classe de Produto Pães, Setor Impressão, Motivo Aguardando Liberação, Recurso Impressora 09, Turno77

Fig. 44: Gráfico Tempo de Produção Parada – Nível 6 Ano 2004, Período 07/2004, Classe de Produto Pães, Setor Impressão, Motivo Aguardando Liberação, Recurso Impressora 09, Turno Terceiro, Operador77

LISTA DE TABELAS

Tabela 1: Comparação entre Banco de Dados Operacionais e Data Warehouse.....	19
Tabela 2: Demonstração do Roll Up.....	40
Tabela 3: Demonstração do Drill Down.....	40
Tabela 4: Demonstração do Slice.....	41
Tabela 5: Demonstração do Dice.....	42
Tabela 6: Plano de Comunicação.....	56
Tabela 7: Matriz de Barramento do Data Warehouse.....	62
Tabela 8: Relação de Análise.....	69

SUMÁRIO

Resumo.....	III
Abstract.....	IV
Lista de Figuras.....	V
Lista de Tabelas.....	IX
INTRODUÇÃO	
1.1 Apresentação do Problema.....	1
1.2 Justificativa.....	1
1.3 Objetivos.....	2
1.3.1 Objetivo Geral.....	2
1.3.2 Objetivos Específicos.....	3
1.4 Metodologia.....	3
1.5 Estrutura do Trabalho.....	4
CAPÍTULO II SISTEMAS DE INFORMAÇÃO	
2.1 Introdução.....	5
2.2 Definição.....	5
2.2.1 Classificação.....	7
2.2.1.1 Sistema de Informação Transacional.....	9
2.2.1.2 Sistemas de Informação Gerencial – SIG.....	10
2.2.1.3 Sistema de Suporte à Decisão.....	11
2.2.2 Abordagem dos Problemas.....	12
2.2.3 Benefícios.....	13
2.3 Conclusão.....	14

CAPÍTULO III – DATA WAREHOUSE

3.1 Introdução.....	15
3.2 Data Warehouse.....	15
3.2.1Característica do Data Warehouse.....	16
3.2.2 Justificativa de um Data Warehouse.....	19
3.2.3 Requisitos de um Data Warehouse.....	20
3.2.4 Arquitetura de um Data Warehouse.....	22
3.2.4.1 Arquitetura Global.....	22
3.2.4.2 Arquitetura de Data Mart Independente.....	23
3.2.4.3 Arquitetura de Data Marts Integrados.....	24
3.2.5 Implementações.....	25
3.2.5.1 Implementação Top_Down.....	25
3.2.5.2 Implementação Bottom-Up.....	27
3.2.5.3 Implementação Combinada.....	29
3.3 O Modelo Dimensional.....	30
3.3.1 Fatos.....	31
3.3.2 Medidas.....	32
3.3.3 Dimensão.....	32
3.3.4 Star Schema.....	33
3.4 Componentes de um Data Warehouse.....	34
3.4.1 Sistemas Operacionais de Origem.....	34
3.4.2 Data Staging Área.....	35
3.4.2.1 Extração.....	35
3.4.2.2 Transformação.....	35

3.4.2.3 Carga.....	36
3.4.3 Apresentação dos Dados.....	36
3.4.4 Ferramentas de Acesso a Dados.....	38
3.4.4.1 O Cubo.....	38
3.4.4.2 Operações Básicas de OLAP.....	39
3.4.4.2.1 Drill Down e Roll Up.....	39
3.4.4.2.2 Slice and Dice.....	41
3.5 Conclusão.....	42

CAPÍTULO IV - ESTUDO DE CASO – O SETOR PRODUTIVO DE UMA INDÚSTRIA DE EMBALAGENS FLEXÍVEIS

4.1 Introdução.....	44
4.2 A Indústria de Embalagens.....	44
4.2.1 Embalagens Plásticas Flexíveis.....	47
4.3 Conhecendo a PLASC.....	48
4.3.1 Estrutura de Produção.....	49
4.3.2 Estrutura de TI.....	49
4.4 Metodologia de Desenvolvimento.....	51
4.4.1. Planejamento e Gerenciamento do Projeto.....	52
4.4.1.1. Avaliação da Preparação.....	52
4.4.1.2 Escopo.....	54
4.4.1.3 Justificativa.....	54
4.4.1.4 Trabalho em Equipe.....	55
4.4.1.5 Desenvolvimento e Manutenção.....	56
4.4.2 Definição dos Requisitos de Negócio.....	56

4.4.3 Projeto da Arquitetura Técnica.....	59
4.4.4 Seleção e Instalação do Produto.....	60
4.4.5 Modelagem Dimensional.....	61
4.4.6 Projeto Físico.....	65
4.4.7 Projeto e Desenvolvimento do ETL.....	66
4.4.8 Especificação da Aplicação Analítica.....	68
4.4.9 Desenvolvimento de Aplicações Analíticas.....	69
4.10 Implantação.....	78
4.11 Manutenção e Crescimento.....	79
4.5 Conclusão.....	80
CONCLUSÃO	
5.1 Conclusões.....	81
5.2 Trabalhos Futuros.....	83
REFERÊNCIAS	85
Apêndice A Modelo Dimensional.....	89
Apêndice B – Dicionário de Dados.....	91
Apêndice C – ETL (Extract, Transform and Load).....	94
Apêndice D – Abreviaturas.....	116

INTRODUÇÃO

1.1 Apresentação do Problema

As indústrias estão apostando nas novas tecnologias para auxiliá-las a atuar em um mercado cada vez mais competitivo e globalizado, onde a necessidade de tomar decisões de forma rápida, efetiva e eficiente se torna um imperativo.

Fazendo uma breve análise observa-se que os dados dos sistemas transacionais são os que determinam o rumo da empresa, neles constam os pedidos de compra, pedidos de venda, estoque, financeiro e todas as outras informações que faz o dia a dia da empresa. Esses dados são na sua grande maioria, incluídos, manipulados e gerenciados através de sistemas ditos transacionais, operacionais ou simplesmente OLTPs, que atuam muitas vezes como fontes de informação para o auxílio ao processo decisório.

Como o próprio nome já sugere os OLTPs atuam no processo operacional de uma empresa. Esse fato, está levando as empresas a procurarem algo que esteja além do OLTP, algo que lhes propiciem um apoio à decisão mais rápido e eficiente que os atuais sistemas.

1.2 Justificativa

Esse algo a mais que as empresas buscam, diz respeito a sistemas de informação que auxiliem no processo gerencial e decisório. Ou seja, as informações devem ser apresentadas aos gerentes e diretores de forma rápida e precisa, de tal forma que auxiliem os mesmos a observar o rumo que o setor ou a empresa estão seguindo ou devem seguir.

Os gerentes e diretores observam tendências e traçam perfis, praticamente não lidam com um único dado por vez, muito pelo contrário, lidam com um número muito grande de dados, sempre os analisando em referência a uma pergunta a ser respondida.

Analisando o problema desta forma, é percebida a necessidade de um repositório que armazene os dados de forma a agilizar e simplificar o acesso às informações e conseqüentemente, a tomada de decisão. E o que mais se enquadra neste perfil é um *Data Warehouse*.

Direcionando o foco deste estudo de caso ao setor de produção de uma indústria de embalagens flexíveis, com o auxílio de um *Data Warehouse*, o decisor poderá obter uma visão sintetizada da área produtiva, confrontando produtos em relação às máquinas disponíveis. Analisando a melhor máquina, ou as melhores máquinas que melhor produzem o referido produto. Ao mencionar a melhor máquina, destaca-se a máquina disponível que terá a melhor relação entre: o menor tempo de *setup* X maior Produção Hora X menor perda produção.

Vale lembrar que um *Data Warehouse* é um meio para se chegar a um fim. O *Data Warehouse* não vai chegar a nenhuma conclusão, ele é uma ferramenta destinada a apoiar os gerentes e diretores o tomarem as decisões com maior eficiência e competência.

1.3 Objetivos

1.3.1 Objetivo Geral

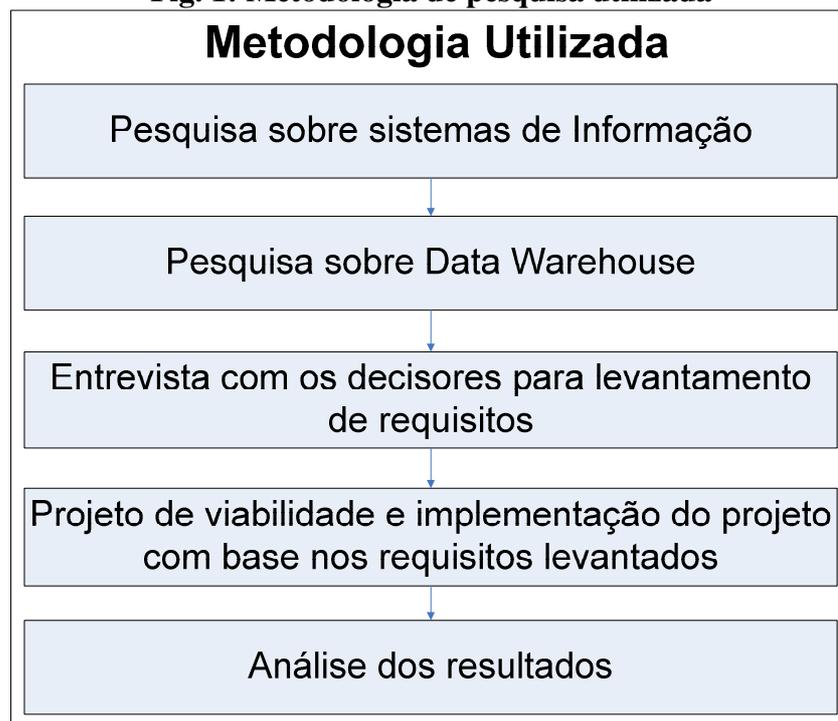
Este trabalho visa conceber a arquitetura hierárquica dos sistemas de informações, a fim de dar suporte aos processos de tomada de decisão dos diversos setores produtivos de uma indústria de embalagens flexíveis, por meio do projeto de um *Data Warehouse*.

1.3.2 Objetivos Específicos

- Apresentar e fazer uso em um estudo de caso, dos elementos teóricos relacionados ao projeto e desenvolvimento da técnica de *Data Warehouse*;
- Propor, definir e desenvolver uma arquitetura de informações baseadas em técnicas de *data warehouse* para auxiliar os gerentes e diretores de produção, contendo informações relevantes para a tomada de decisões;
- Validar o ciclo de vida dimensional de negócio de Ralph Kimball para uma indústria de embalagens flexíveis.

1.4 Metodologia

Fig. 1: Metodologia de pesquisa utilizada



Fonte: Autor

A primeira parte da metodologia consiste numa pesquisa na documentação sobre Sistemas de Informação e sua distribuição na pirâmide hierárquica.

A segunda parte consiste numa pesquisa na documentação sobre *Data Warehouse*, onde se relatam as nomenclaturas, os conceitos, às arquiteturas disponíveis, e tudo mais relacionado ao *DW*.

A terceira parte consiste em entrevistas com os decisores e gerentes de Ti, a fim de entender a estrutura da empresa em questão e conhecer suas necessidades e requisitos.

A quarta parte consiste em montar um projeto de *Data Warehouse* baseado nas necessidades e requisitos levantados nas entrevistas. Desenvolvendo o projeto, implementando o ETL, e criando as análises que auxiliaram os decisores a chegarem as devidas conclusões.

A quinta e última parte focaliza as conclusões obtidas através das análises efetuadas no decorrer do trabalho.

1.5 Estrutura do Trabalho

O primeiro capítulo traz uma visão geral do trabalho, seus objetivos, justificativa, metodologia e a sua estrutura.

O segundo capítulo apresenta um estudo sobre o que são sistemas de informação.

O terceiro capítulo apresenta um estudo sobre a tecnologia de *data warehouse*.

O quarto capítulo demonstra todo o projeto de implementação de um *Data Warehouse* voltado à indústria de embalagens flexíveis, segundo o ciclo de vida dimensional de negócio proposto Ralph Kimball.

O quinto capítulo demonstra as conclusões obtidas através do trabalho de pesquisa e direciona as possibilidades de continuidade com trabalhos futuros.

CAPÍTULO II

SISTEMAS DE INFORMAÇÃO

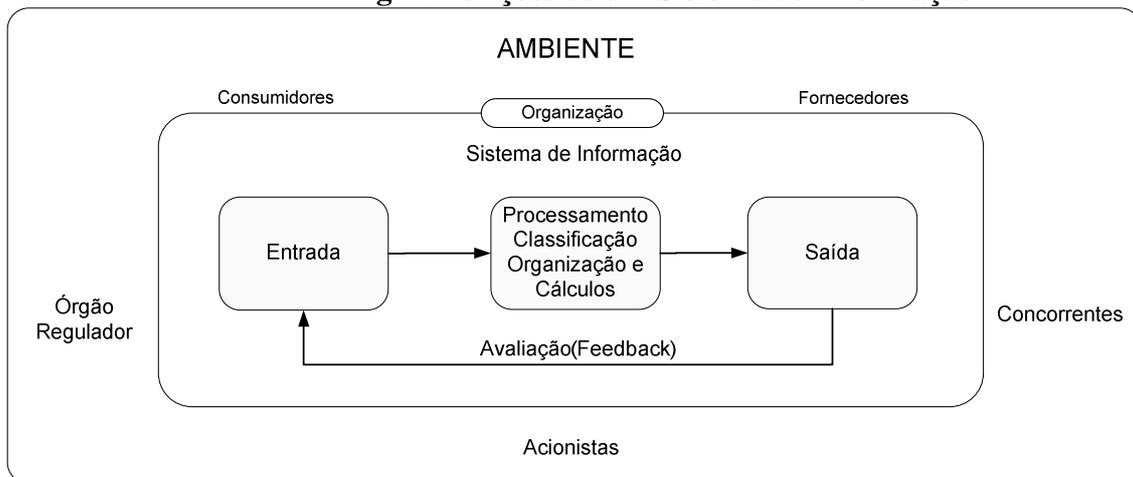
2.1 Introdução

Com o amadurecimento da tecnologia da informação, através dos anos, cada vez mais surgem novos conceitos, novas técnicas, novas segmentações. Isso está fazendo com que a velocidade em que as coisas aconteçam neste novo mundo globalizado, seja cada vez maior. E com essa velocidade aumenta também a complexidade dos problemas, conforme vai se avançando nas camadas hierárquicas de uma empresa.

O objetivo deste capítulo é conceituar de forma sucinta, o que são sistemas de informação, posicionando-os na pirâmide de camadas, de forma a evidenciar a sua divisão em áreas e a interdependência destas áreas entre as camadas da pirâmide.

2.2 Definição

Um sistema de informação (*SI*) é um conjunto de elementos ou componentes inter-relacionados que coletam (*entrada*), manipulam (*processamento*) e disseminam (*saída*) os dados e a informação, fornecendo um mecanismo de *feedback*, (avaliação) a fim de atender a um objetivo, como demonstrado na figura 2. (STAIR,2002, LAUDON,1999 e 2001).

Fig. 2: Funções de um Sistema de Informação

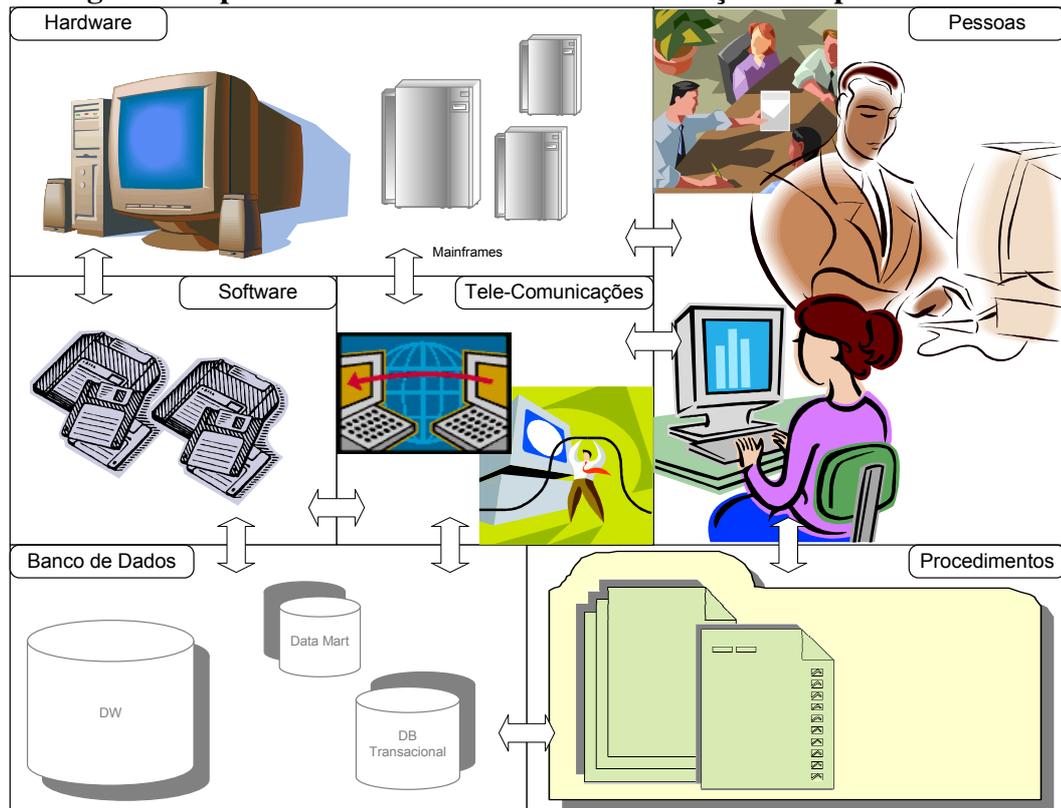
Fonte: STAIR, 2002, LAUDON, 1999 e 2001.

Segundo Fernando de Souza Meireles citado em (OLIVEIRA, 2000), é possível identificar três eras evolucionárias nos sistemas de informações:

- Entre 1960 a 1970, chamada de era transacional, onde o enfoque era contábil / financeiro;
- Entre 1970 a 1989, chamada de era informacional, onde surgiram as centrais de processamentos de dados, os CPDs;
- A partir dos anos 90, a era do conhecimento, onde o termo informação é substituído por Tecnologia da Informação, e a informação passa a ser encarada como ativo, tornando-se um diferencial competitivo.

Para Stair (2002), um sistema de informação pode ser manual ou computadorizado. Um sistema de informação computadorizado é composto de *Hardwares*, *Softwares*, Banco de dados, Telecomunicações, Pessoas e Procedimentos, que são configurados para coletar manipular, armazenar e processar os dados em informações. Todos esses componentes podem ser visualizados na figura 3.

Fig. 3: Componentes de um Sistema de Informação Computadorizado

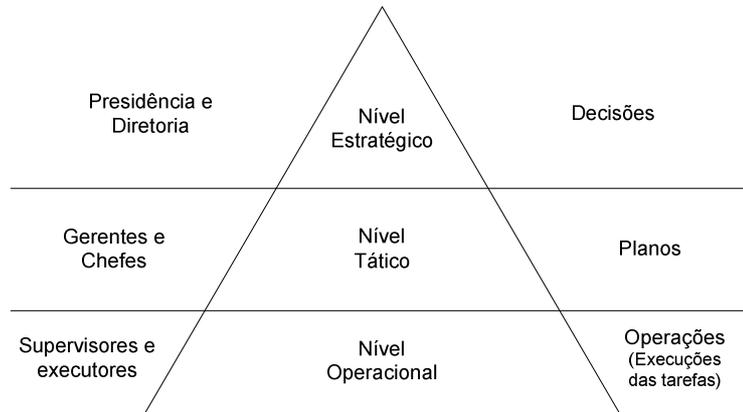


Fonte Adaptado de Stair, 2002.

2.2.1 Classificação

Segundo Rezende (2003), na prática não existe uma classificação rígida de sistemas de informações. A ênfase é dada em relação aos níveis hierárquicos das empresas. Sendo mais comumente divididas em: sistema de informação operacional ou de transações, sistema de informação gerencial e sistemas de informação estratégica, podendo se ter uma visão da estrutura na figura 4. (OLIVEIRA, 2000, 1998, NAZÁRIO, 2003, KEEN, 1978).

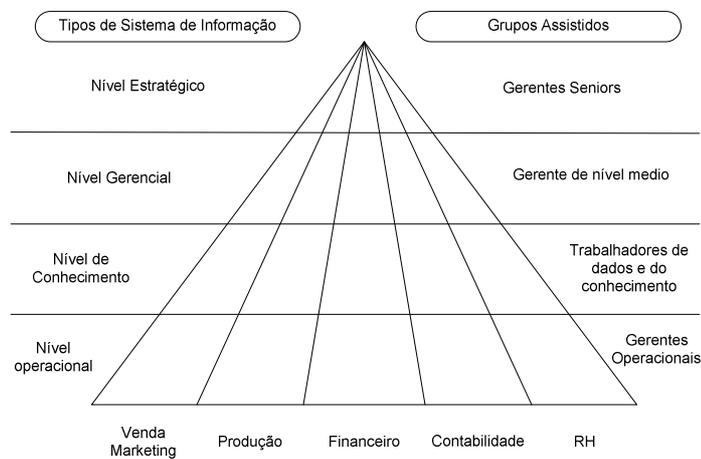
Fig. 4: Pirâmide Organizacional – Os diferentes níveis das organizações



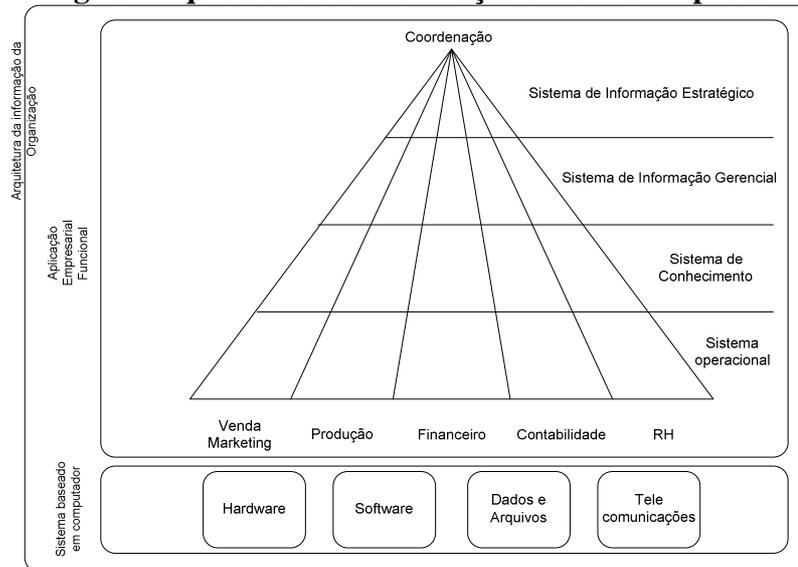
Fonte: Oliveira, 2000.

Os Laudons (2001) acrescentam mais um nível nesta classificação que é o nível de conhecimento, demonstrando que um nível serve de base aos próximos, como visto na figura 5. Com base nessa estrutura, montam a arquitetura da informação em uma empresa, demonstrado na figura 6.

Fig. 5: Pirâmide Organizacional – Os diferentes níveis das organizações



Fonte: Laudon, 2001.

Fig. 6: Arquitetura da Informação em uma Empresa

Fonte: Laudon, 2001.

Stair (2002) também classifica em termos corporativos em 4 níveis: nível de processamento de transação e comércio eletrônico, nível de informações gerenciais, nível de suporte a decisão e nível de inteligência artificial.

Em síntese, todos os autores até aqui citados mencionam os três níveis básicos de classificação referenciada acima por Rezende (2003).

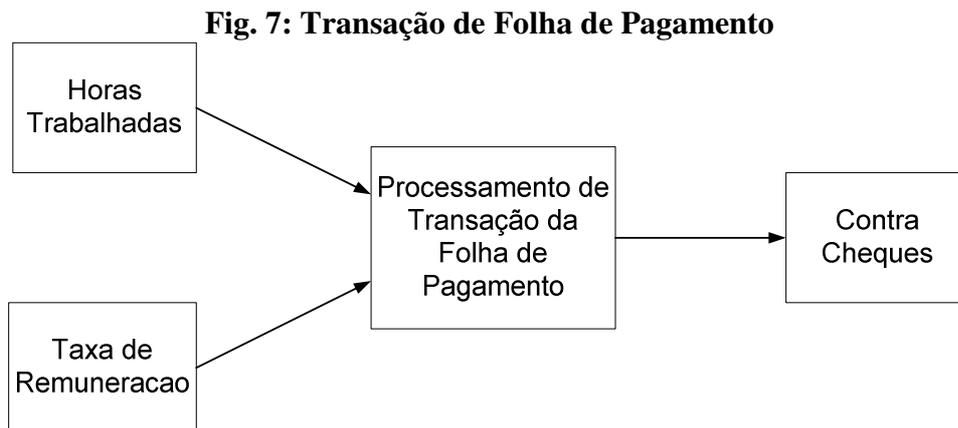
2.2.1.1 Sistemas de Informação Transacional – SIT

Uma transação equivale a qualquer troca relacionada ao negócio, como venda à cliente, pagamento a fornecedor, pagamento de empregados, entre outras.

Um Sistema de Informação Transacional é uma coleção organizada de pessoas, procedimentos, *softwares*, banco de dados e dispositivos, com a finalidade de registrar as transações empresariais realizadas.

Para entender um sistema de transações, faz-se necessário entender as operações e as funções básicas da empresa.

Os Sistemas de Informação Transacionais representam a aplicação nos conceitos de informação e tecnologia para as transações rotineiras e repetitivas. Geralmente, são transações empresariais comuns, mas críticas para as funções diárias da empresa. A figura 7 demonstra um exemplo de transação. (STAIR, 2002, LAUDON, 2001, 1999).



Fonte: Stair, 2002.

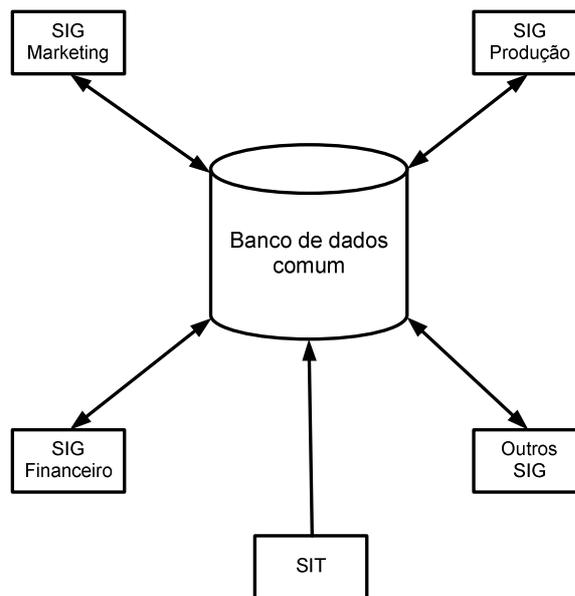
2.2.1.2 Sistemas de Informação Gerencial – SIG

Os benefícios gerados por um sistema de informação transacional são tangíveis e podem ser utilizados para justificar seus custos. Mas logo ficou claro que os dados armazenados nos SITs poderiam auxiliar os gerentes na tomada de decisão em suas respectivas áreas de negócios. Pois, satisfazer as necessidades dos gerentes e dos tomadores de decisão continua a ser o principal fator no desenvolvimento dos sistemas de informação.

Um Sistema de Informação Gerencial (SIG) abrange uma coleção organizada de pessoas, procedimentos, *software*, banco de dados e dispositivos que fornecem informações rotineiras aos gerentes e aos tomadores de decisões. O foco de um SIG é principalmente a eficiência operacional.

Os sistemas de informações gerenciais basicamente fornecem relatórios padronizados com base nos dados e nas informações de sistemas de informação transacional (SIT). Um diagrama da estrutura de um SIG é apresentado na figura 8. (STAIR, 2002, LAUDON, 1999, 2001).

Fig. 8: Diagrama de um Sistema de Informação Gerencial



Fonte: Stair, 2002.

2.2.1.3 Sistemas de Suporte à Decisão – SSD

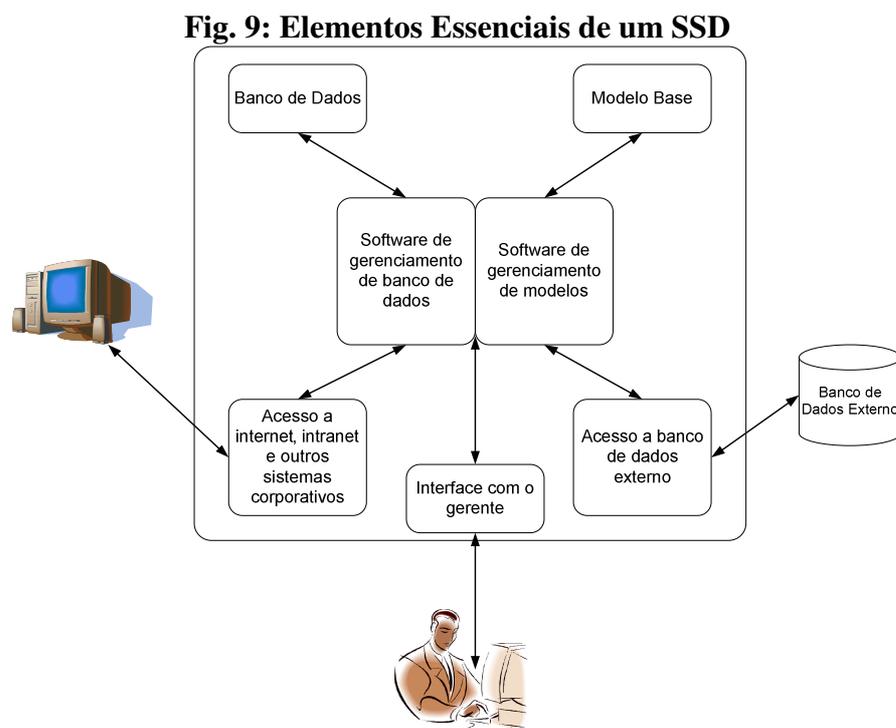
Um Sistema de Suporte a Decisão (SSD) é uma coleção organizada de pessoas, procedimentos e *software*, banco de dados e dispositivos usados para dar suporte a um problema específico na tomada de decisão.

O foco de um SSD incide sobre a eficácia na tomada de decisão. Enquanto um SIG ajuda a organização a fazer “as coisas certas”, um SSD ajuda o gerente ou decisor a fazer “a coisa certa”, naquele momento. Um SSD vai além de um SIG tradicional, pois pode fornecer

assistência imediata na solução de problemas complexos não suportados por um SIG tradicional.

Os Sistemas de Suporte a Decisão são utilizados onde o problema é complexo e a informação, necessária para a melhor decisão, é difícil de ser obtida e usada. Sendo que os gerentes frequentemente desempenham um papel ativo no desenvolvimento e na implementação dos SSDs.

A ênfase geral do SSD é dar suporte, em vez de substituir, a tomada de decisão gerencial. Na figura 9 demonstra-se os elementos essenciais de um Sistema de Suporte a Decisão.

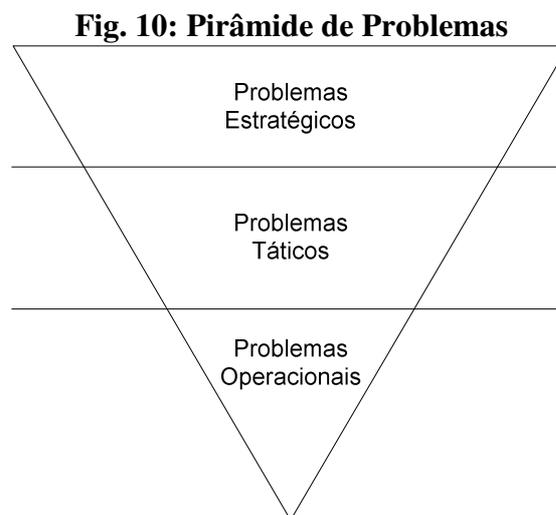


Fonte: Stair, 2002.

2.2.2 Abordagem dos Problemas

Como vimos nos itens 2.3.1, 2.3.2, 2.3.3, os problemas aumentam o nível de complexidade conforme se avança na estrutura hierárquica das empresas.

Segundo Chiavenato (2000), para as empresa são apresentados inúmeros problemas e os mais complexos e relevantes estão concentrados no topo da pirâmide hierárquica. Esses problemas podem referir-se a manutenção de clientes, análise de concorrentes, influências governamentais, influências do mercado em que a empresa atua, entre outros. Uma visão de complexidade desses problemas podem ser visualizados através de uma pirâmide inversa, demonstrada aqui através da figura 10.



Fonte: Chiavenato, 2000.

Os Sistemas de informações poderão contribuir significativamente para a solução dos problemas. Assim, o esforço das empresas deve-se concentrar nos níveis superiores dos SI. Ou seja, nos sistemas de informação gerencial e de suporte a decisão.

2.2.3 Benefícios

Um sistema de informação eficiente pode ter um grande impacto na estratégia corporativa e no sucesso da empresa (STAIR, 1998; OLIVEIRA, 1988).

Rezende (2003) cita como benefícios:

- Suporte a tomada de decisão profícua.

- Valor agregado ao produto.
- Melhores serviços e vantagens competitivas.
- Produto de melhor qualidade
- Oportunidade de negócios e aumento da rentabilidade.
- Mais segurança na informação, menos erros e mais precisão.
- Aperfeiçoamento nos sistemas, eficiência e eficácia, efetividade e produtividade.
- Carga de trabalho reduzida.
- Redução de custos e desperdícios.
- Controle das operações

2.3 Conclusão

Neste capítulo, foi possível ver e compreender o que é, como funciona e como estão classificados os sistemas de Informação.

Observa-se que os problemas estão crescendo com uma velocidade muito grande e que os mesmos se concentram nos níveis mais altos das camadas hierárquicas. Como visto, os sistemas transacionais servem de base para os sistemas subsequentes (SIG e SSD), sendo os mesmos dados tratados e apresentados de forma diferente da transacional.

CAPÍTULO III

DATA WAREHOUSE

3.1 Introdução

Na era da informação, o bem mais precioso de uma empresa são suas informações. Em teoria Kimball (2002) admite existir duas formas de armazenar essas informações. O sistema transacional, onde os dados são originalmente armazenados e o *data warehouse*.

As informações do *data warehouse* auxiliam os tomadores de decisões a observar o rumo que a empresa esta seguindo, ou deve seguir.

Neste capítulo são expostos os conceitos, a arquitetura, os passos necessários para criação de um *Data Warehouse*, bem como suas ferramentas para análise.

3.2 Data Warehouse

O ambiente de *Data Warehouse*, esta baseado na tecnologia chamada de *Data Warehousing*, considerada por muitos como a evolução natural do ambiente a apoio à decisão (MACHADO, 2000; KIMBALL; INMON, 2004d; DW BRASI, 2003b, 2003c}.

Segundo Kimball (2002; DWBRASIL.2003c), *Data Warehousing* é um conjunto de ferramentas e técnicas de projeto, que quando aplicadas às necessidades específicas dos usuários e aos bancos de dados específicos permitirá o planejamento e a construção de um *data warehouse*.

Conforme Batini (DWBRASIL.2003c), "Um banco de dados é uma coleção de dados operacionais armazenados e utilizados pelo sistema de aplicações de uma empresa

específica". Os dados mantidos por uma empresa são chamados de "operacionais" ou "primitivos". Batini (DWBRASIL.2003c) refere-se aos dados do banco de dados como "dados operacionais", distinguindo-se de dados de entrada, dados de saída e outros tipos de dados.

Baseado na definição acima, DW é na verdade, uma coleção de dados derivados dos dados operacionais para sistemas de suporte à decisão. Estes dados derivados são, muitas vezes, referidos como dados "gerenciais", "informativos" ou "analíticos". (KIMBALL, 2002; MACHADO, 2000; DWBRASIL.2003b, 2003c; VIEIRA, 2003).

Para Inmon, citado em (SELL, 2000) "*Data Warehouse* é um conjunto de dados baseado em assuntos, integrado, não volátil e variável em relação ao tempo, de apoio às decisões gerenciais".

3.2.1 Características de um Data Warehouse

Segundo Inmon (2004d; DWBRASIL, 2003b, 2003c; SELL, 2000), um DW possui as seguintes características:

Orientação por Assunto

Em um *data warehouse*, toda modelagem é voltada para os principais assuntos da empresa. Já os sistemas transacionais são voltados para processos e aplicações específicas. (INMON,2004d; DWBRASIL.2003b,.2003c; SELL, 2000).

É possível definir que assuntos: são os conjuntos de informações relativas à determinada área estratégica de uma empresa. Os assuntos é que darão origem as tabelas de fato. (INMON,2004d; DWBRASIL.2003b,.2003c; SELL, 2000).

Integração

Através da integração, padroniza-se uma representação única para os dados de todos os sistemas que formarão a base de dados do *Data Warehouse*. Desta forma, a maior parte do trabalho de construção de um *DW* está na análise dos sistemas transacionais e dos dados que eles contêm. (INMON,2004d; DWBRASIL.2003b,.2003c; SELL, 2000).

O processo de integração é necessário porque geralmente os dados encontram-se armazenados em vários padrões de codificação, isso se deve aos inúmeros sistemas existentes nas empresas, podendo assim, os mesmos dados estarem em formatos diferentes.

Variação no Tempo

Segundo Inmon (2004d; DWBRASIL,2003b,.2003c; SELL, 2000), *Warehouses* são variáveis em relação ao tempo. Deste modo é possível dizer que isso é manter o histórico dos dados, durante um período de tempo superior ao dos sistemas transacionais.

Num *Data Warehouse* mantém-se um intervalo de tempo superior ao dos sistemas transacionais. Isso porque num sistema transacional a finalidade é fornecer as informações no momento exato, já no *Data Warehouse*, o objetivo é analisar o comportamento das informações durante um período de tempo maior. Fundamentados nessa variação, os gerentes tomam as decisões em cima de fatos e não de intuições.

Os dados nos sistemas transacionais estão sendo atualizados constantemente, com exatidão válida somente para o momento de acesso. Os dados existentes num *Data Warehouse* são como fotografias que refletem os mesmos num determinado momento do tempo. Essas fotografias são chamadas de *snapshots*.

A dimensão data, sempre estará presente em qualquer fato de um *DW*, isso ocorre porque sempre os dados refletirão num determinado momento de tempo, e obrigatoriamente deverão conter uma chave de data para expressar o tempo em que os dados foram extraídos. Desta forma, afirma-se que os dados armazenados corretamente no *Data Warehouse* não serão mais atualizados tendo-se assim uma imagem fiel da época em que foram gerados.

Os metadados, assim como os dados também possuem elementos temporais, porque mantêm um histórico das mudanças nas regras de negócio da empresa. Os metadados são responsáveis pelas informações referentes ao caminho do dado dentro do *Data Warehouse*.

Não Volatilidade

Num *Data Warehouse* existe somente duas operações, a carga dos dados (inicial e incremental) e o acesso aos dados em modo de leitura.

A maneira como os dados são carregados e tratados é completamente diferente dos sistemas transacionais, enquanto nesses sistemas têm-se vários controles e *updates* de registros, no *Data Warehouse* tem-se somente *inserts e selects* de dados.

Os dados sempre passam por filtros antes de serem inseridos no *DW*, com isso muitos dados jamais saem do ambiente transacional.

Segundo Inmon citado em (DWBRASI, 2003b, 2003c), a maior parte dos dados é física e radicalmente alterada quando passa a fazer parte do *data warehouse*. Os dados não são mais os mesmos do ambiente operacional. Desta forma, a redundância de dados entre os dois ambientes raramente ocorre, resultando em menos de 1 por cento de duplicações. Uma comparação destas características podem ser visualizadas na tabela 1

Tabela 1: Comparação entre Banco de Dados Operacionais e Data Warehouse

Características	Bancos de dados Operacionais	Data Warehouse
Objetivo	Operações diárias do negócio	Analisar o negócio
Uso	Operacional	Informativo
Tipo de processamento	OLTP	OLAP
Unidade de trabalho	Inclusão, alteração, exclusão	Carga e consulta
Número de usuários	Milhares	Centenas
Tipo de usuário	Operadores	Comunidade gerencial
Interação do usuário	Somente pré-definida	Pré-definida e <i>ad-hoc</i>
Condições dos dados	Dados operacionais	Dados Analíticos
Volume	Megabytes – gigabytes	Gigabytes – terabytes
Histórico	60 a 90 dias	5 a 10 anos
Granularidade	Detalhados	Detalhados e resumidos
Redundância	Não ocorre	Ocorre
Estrutura	Estática	Variável
Manutenção desejada	Mínima	Constante
Acesso a registros	Dezenas	Milhares
Atualização	Contínua (tempo real)	Periódica (em <i>batch</i>)
Integridade	Transação	A cada atualização
Número de índices	Poucos/simples	Muitos/complexos
Intenção dos índices	Localizar um registro	Aperfeiçoar consultas

Fonte: Herinia, 2003.

3.2.2 Justificativa de um Data Warehouse

A principal justificativa para aplicação de tecnologia de *Data Warehouse* em uma organização é a existência na empresa de: (KIMPALL, 2002, MACHADO, 2000; DWBRASIL, 2003).

- Várias plataformas de hardware e de software;
- Constantes alterações nos sistemas transacionais corporativos;
- Dificuldade acentuada na recuperação de dados históricos em períodos superiores ao ano atual de operações;
- Existência de sistemas pacotes de fornecedores diferentes;
- Falta de padronização e integração dos dados existentes nos diversos sistemas;
- Carência de documentação e segurança no armazenamento dos dados;
- Dificuldade de aplicação de sistemas EIS ou SSD devido a dependências múltiplas de sistemas corporativos.

3.2.3 Requisitos de um Data Warehouse

As informações do *data warehouse* auxiliam os tomadores de decisão a observar o rumo que a empresa esta seguindo ou deve seguir.

Os usuários de um *DW* observam tendências e traçam perfis. Praticamente, não lidam com uma única linha por vez, muito pelo contrário, lidam com um número muito grande de registros. Sempre analisando os dados em referência a uma pergunta.

Ao entrevistar gerentes e tomadores de decisões depara-se com respostas do tipo (KIMBALL,2002):

- “Dispomos de um enorme volume de dados na empresa, mas não conseguimos acessá-los”.
- “Temos que combinar os dados a partir de fontes diversas”.
- “É preciso tornar mais fácil para os profissionais acessarem diretamente os dados”.
- “Apenas necessito ver o que realmente interessa”.

- “É imperdoável quando duas pessoas me mostram o mesmo indicador com dados diferentes”.
- “Queremos que as decisões sejam tomadas com o uso de informações, baseando-se em fatos”.

Segundo Kimball (2002), as preocupações mencionadas acima são universais e são transformadas em requisitos para criação de um *Data Warehouse*:

- O *DW* deve fazer com que as informações de uma empresa possam ser facilmente acessadas;
- O *DW* deve apresentar as informações da empresa de modo consistente;
- O *DW* deve ser adaptável e flexível às mudanças;
- O *DW* deve ser um baluarte seguro que protege as informações;
- O *DW* deve funcionar como a base para uma melhor tomada de decisão;
- Para que um *DW* seja considerado de sucesso a comunidade de negócio deve aceitá-lo.

Para implantar um *Data Warehouse* de sucesso não basta apenas ser um DBA, mas sim um híbrido entre TI e a gerência do negócio, os comportamentos devem ser como DBA/MBA, além de alguns cuidados. De acordo com Kimball (2003f; PERENTE, 2003) para levar um *DW* ao fracasso basta seguir os passos abaixo:

1. Não descubra quem são os verdadeiros conhecedores do negócio;
2. Não entenda os objetivos e as demandas dos usuários finais;
3. Faça com que tudo fique muito, muito, complicado;
4. Faça com que tudo fique extremamente e irritantemente lento;
5. Arraste o projeto por mais de um ano;
6. Faça com que seu *DW* não se adapte a novos tempos e regras de negócio.

3.2.4 Arquitetura de um Data Warehouse

As literaturas mencionam vários tipos de arquiteturas. Segundo Machado (2000), o que realmente define a arquitetura é onde o *Data Warehouse* ou os *Data Marts* estão residindo e como são implementados.

É possível definir três arquiteturas básicas: (MACHADO, 2000; KIMBALL, 2002), a arquitetura Global, arquitetura de *Data Mart* Independente e arquitetura de *Data Marts Integrados*.

Defini-se também três tipos de implementações básicas: a implementação *Top_Down*, a implementação *Bottom_Up* e a implementação combinada.

3.2.4.1 Arquitetura Global

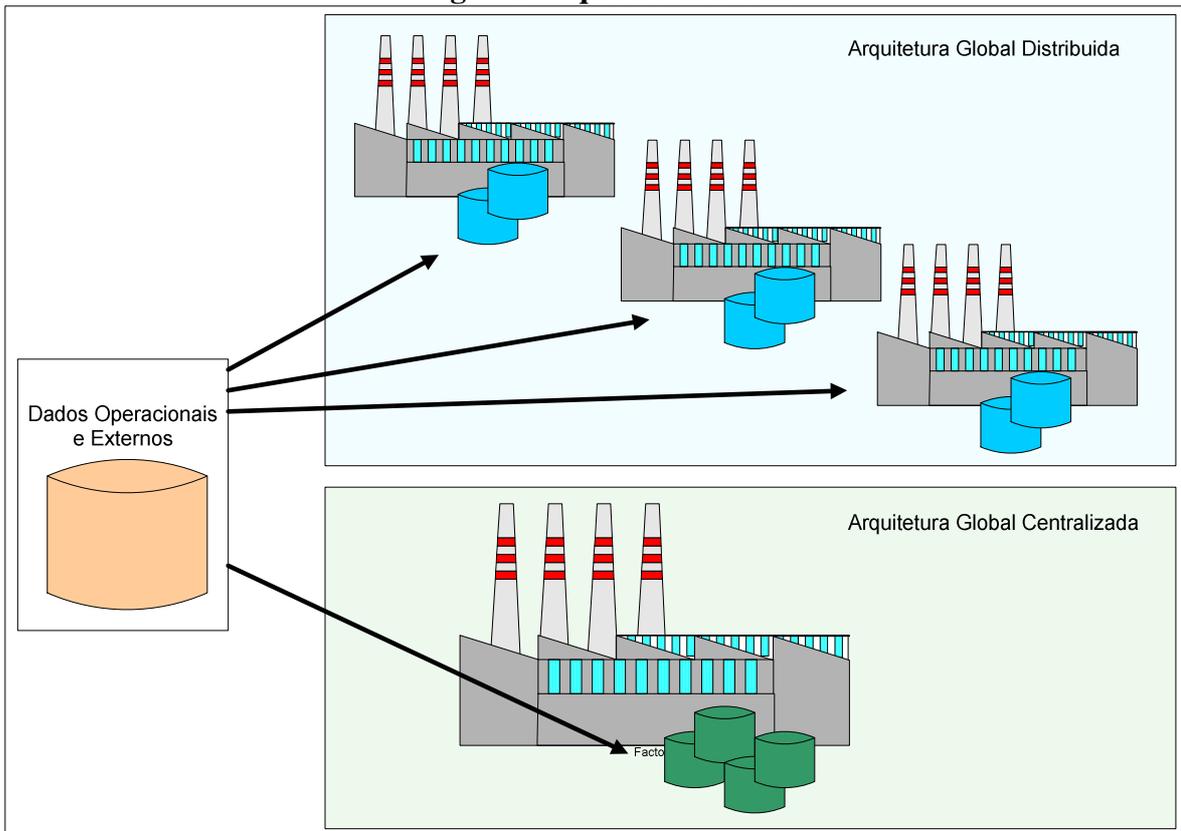
Na arquitetura global, o *Data Warehouse* é projetado e construído baseado nas necessidades da empresa como um todo. O repositório de dados está disponível para toda empresa como base para suporte a decisão.

Segundo Machado (2000), o termo global transmite que o *Data Warehouse* é centralizado. O termo também reflete o escopo de utilização de *Data Warehouse*, que nesta arquitetura é para toda empresa.

Os dados podem estar fisicamente centralizados ou distribuídos em várias unidades da empresa.

Vale ressaltar que a arquitetura global habilita o usuário a utilizar visões corporativas, visualizando a empresa como um todo. Mas, seu desenvolvimento é muito lento e de custo muito elevado. Na figura 11 tem-se uma visão da arquitetura global.

Fig. 11: Arquitetura Global



Fonte: Machado, 2000.

3.2.4.2 Arquitetura de *Data Mart* Independente

Essa é a arquitetura preferida dos fornecedores de *software* para consulta de informações de *Data Warehouse*, pois é isolada e o resultado encanta aos olhos dos usuários (KIMBALL, 2002; MACHADO, 2000).

A arquitetura independente implica em *Data Marts stand-alone* controlados por um grupo específico de usuários e que atende somente às suas necessidades específicas e departamentais, sem foco cooperativo algum. Isso faz com que não exista nenhuma conectividade desses *Data Marts* com outros *Data Marts* de outros departamentos ou área de negócios.

A arquitetura de *Data Marts* independentes requer os mesmos perfis técnicos para a implementação, porém os recursos, e pessoal operacional podem ser administrados pelo grupo de trabalho ou departamento.

Sua restrição está em possuir um mínimo de integração corporativa e não permite nenhuma visão global da empresa.

3.2.4.3 Arquitetura de *Data Marts* Integrados

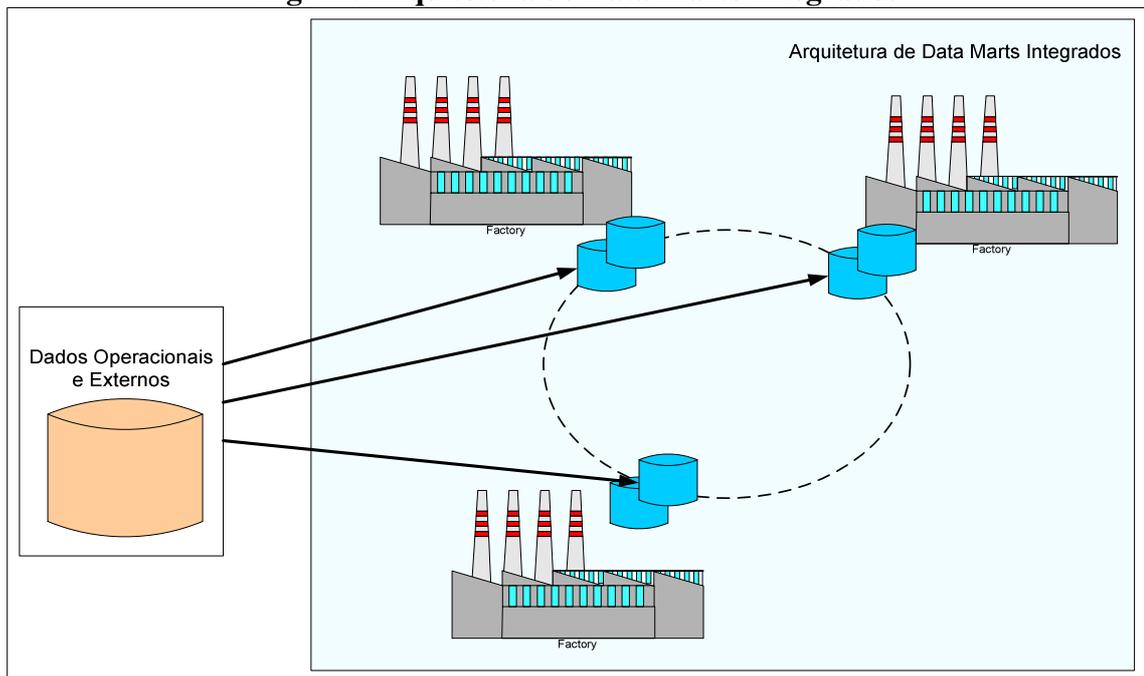
A arquitetura de *Data Marts* integrados é basicamente uma distribuição de implementação. Mesmo os *Data Marts* sendo desenvolvidos e implementados separadamente, eles são integrados e interconectados dando assim uma visão corporativa. Segundo Machado (2000), o autônível de integração é similar ao da arquitetura global.

A arquitetura de *Data Marts* integrados permite muitas outras funções, além de apresentar capacidade de prover informações em relação à arquitetura independente. Entretanto, isso aumenta a complexidade de requisitos.

Essa arquitetura contém dados comuns a outros *Data Marts*, sendo assim é possível compartilhar os dados, o que conseqüentemente aumenta a administração do ambiente, em contrapartida observa-se um aumento considerável de qualidade da visão corporativa das informações.

Nesta arquitetura a atuação do setor de tecnologia da informação deve ser muito maior, ficando sob sua responsabilidade a administração dos *Data Marts*. Na figura 12 destaca-se uma visão do que vem a ser a arquitetura de *Data Marts* integrados:

Fig. 12: Arquitetura de Data Marts Integrados



Fonte: Machado, 2000.

3.2.5 Implementações

3.2.5.1 Implementação *Top_Down*

A implementação *Top_Down* é conhecida como padrão inicial do conceito de *Data Warehouse*. Ela requer um maior planejamento e trabalho de definições conceituais de tecnologia, antes de iniciar-se o projeto de *data warehouse* propriamente dito (MACHADO, 2000).

Todo pessoal envolvido no processo deve participar das reuniões que irão definir o projeto. Nessas reuniões devem ser tomadas as seguintes decisões: identificação das fontes de dados que serão utilizadas; segurança; estrutura de dados; qualidade e padrões dos dados.

Nessa arquitetura o processo se inicia com a extração, a transformação e a integração das informações dos sistemas transacionais para área de estagiamento, em seguida os dados e metadados são transferidos para o *Data Warehouse*.

A partir do *Data Warehouse* os dados e metadados são transferidos para os *Data Marts*.

O lado positivo desta implementação esta no fato de a empresa ter de definir regras de negócio de forma corporativa antes de iniciar o projeto. Uma visão da Implementação *Top Down* pode ser obtida na figura 13.

Vantagens: (MACHADO, 2000):

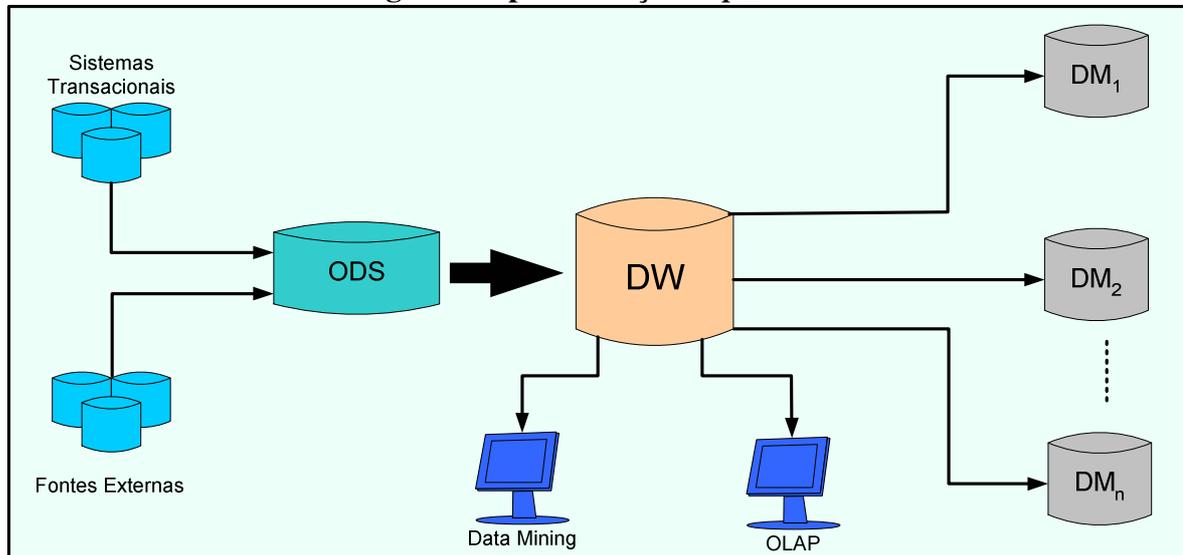
- Herança de arquitetura – Todos os *Data Marts* são originados do *Data Warehouse* permitindo uma fácil manutenção;
- Visão de empreendimento – O *Data Warehouse* concentra todos os dados da empresa;
- Repositório de metadados centralizados – O *Data Warehouse* provê um repositório de metadados central para o sistema, isso permite manutenção mais simples;
- Controle e centralização de regras – Um único conjunto de aplicações para extração, limpeza e integração de dados e processos centralizados de manutenção e monitoração.

Desvantagens: (MACHADO, 2000):

- Implementações muito longas – São necessárias em média 15 ou mais meses para que a primeira área de assunto entre em produção, dificultando o apoio político e orçamentário;
- Alta taxa de risco – Não existem garantias para o investimento neste tipo de ambiente;
- Herança de cruzamentos funcionais – É necessária uma equipe de desenvolvimento e usuários altamente capacitados que saibam avaliar as informações e consultas que garantam à empresa habilidade de sobreviver e prosperar.

- Expectativas relacionadas ao ambiente – A demora de projeto e a falta de retorno podem induzir falsas expectativas nos usuários.

Fig. 13: Implementação *Top_Down*



Fonte: Machado, 2000.

3.2.5.2 Implementação *Bottom-Up*

Essa implementação permite que o planejamento e o desenho dos *Data Marts* possam ser realizados, sem esperar que seja definida uma infra-estrutura corporativa para o *Data Warehouse* na empresa. Essa infra-estrutura não deixará de existir, só que ela é implementada de forma incremental conforme venham sendo implementados os *Data Marts*. (KIMBALL, 2002; MACHADO, 2000).

O propósito desta implementação é a construção de um *Data Warehouse* incremental a partir da construção de *Data Marts* independentes.

O processo começa com a extração, transformação e a integração dos dados para um ou mais *Data Marts*, sendo esses baseados normalmente num modelo dimensional.

Um dos problemas desta implementação é a falta de um gerenciador que garanta padrões únicos de metadados. Mesmo com a independência dos *Data Marts*. A dificuldade

em garantir essa padronização é a responsável pela falha na elaboração incremental do *Data Warehouse*, pois pode, ocorrer redundâncias de dados e inconsistências entre os *Data Marts*.

Com a estrutura de múltiplos *Data Marts* a extração dos dados pode tornar-se crítico no que diz respeito à interferência nos sistemas transacionais. Na figura 14 é possível ter uma visão da implementação *Bottom - Up*

Vantagens: (KIMBALL, 2002; MACHADO, 2000):

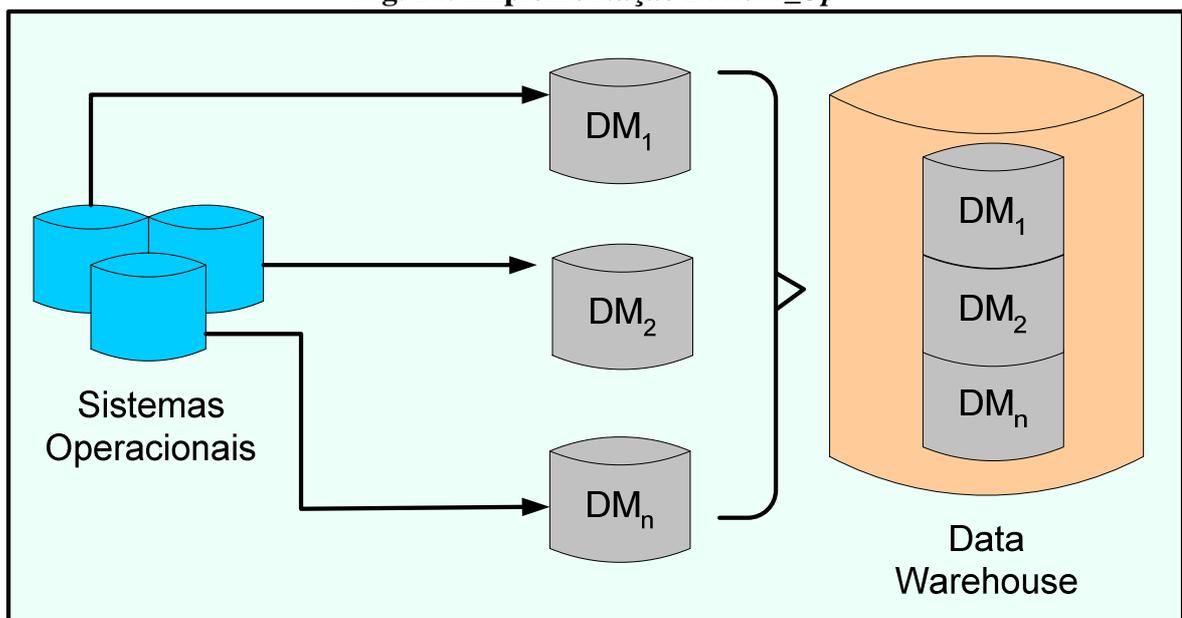
- Implementação rápida – A construção de um *Data Mart* é altamente direcionada, podendo estar em produção em seis a nove meses;
- Retorno Rápido – Esse tipo de implementação mostra rapidamente seu valor, garantindo a sua base orçamentária;
- Manutenção do enfoque da equipe – *Data Marts* incrementais permitem que os principais negócios sejam enfocados inicialmente, sem que haja gastos no desenvolvimento de áreas que não sejam essenciais à empresa;
- Herança incremental – *Data Marts* incrementais obriga a entrega de recursos de informação, passo a passo. Isso permite à equipe crescer e aprender, reduzindo riscos.

Desvantagens: (KIMBALL, 2002; MACHADO, 2000)

- Perigo de *legamarts* – Um dos maiores perigos é a criação de *Data Marts* independentes. Quando as soluções não consideram a arquitetura de forma global, transformam-se em *Data Marts* legados ou *legamarts*. Quando os *Data Marts* inviabilizam futuras implementações eles tornam-se parte do problema e não da solução;

- Desafio de possuir a visão de empreendimento - Durante a construção dos *Data Marts* incrementais é necessário um rígido controle para manter a visão global. Esse controle requer um maior trabalho do que utilizar um *Data Warehouse*;
- Administrar e coordenar múltiplas equipes e iniciativas – Esse tipo de implementação emprega o desenvolvimento de *Data Marts* em paralelo. Isso conduz a uma rígida administração para manter a visão global do projeto;
- A maldição do sucesso – Os usuários que já estão trabalhando nos seus *Data Marts* encontram-se satisfeitos. Enquanto os usuários que ainda aguardam seus *Data Marts*, colocam a equipe de desenvolvimento sobre uma pressão constante.

Fig. 14: Implementação *Bottom-Up*



Fonte: Machado, 2000.

3.2.5.3 Implementação Combinada

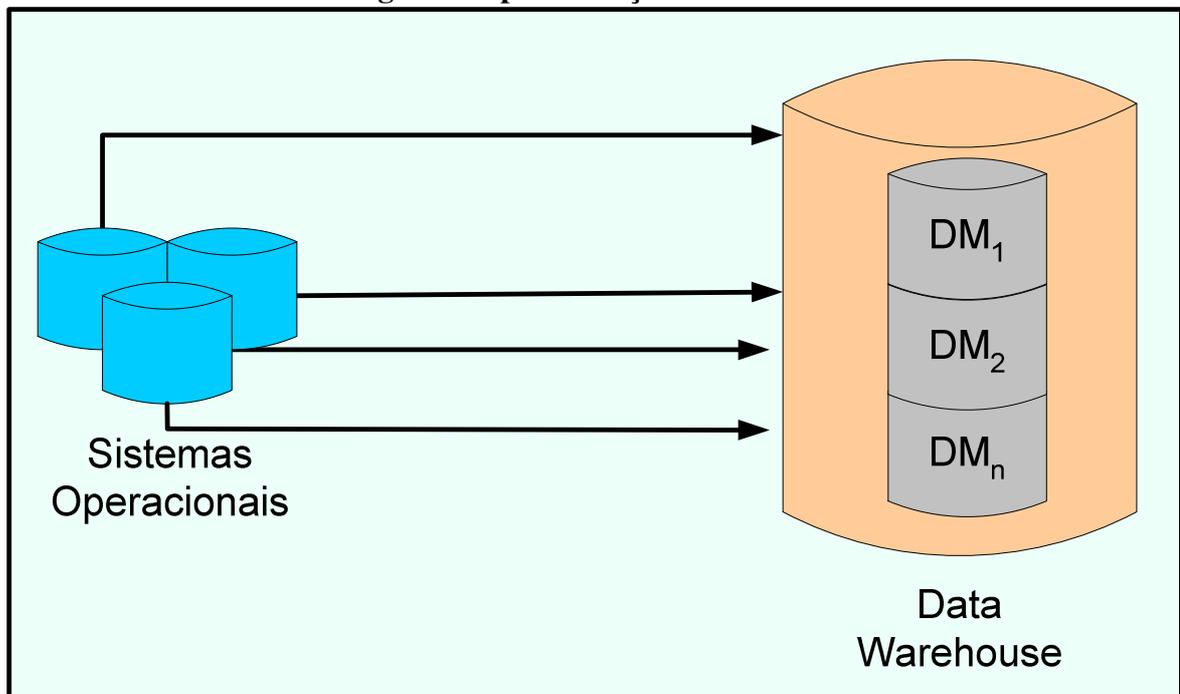
Essa implementação como o nome já sugere, é a combinação da implementação *Top-Down* e *Bottom-Up*.

Nesta abordagem efetua-se a modelagem de dados do *Data Warehouse* de visão macro, sendo o passo seguinte à implementação de partes deste modelo, aos quais são escolhidas as áreas de interesse aos quais irão constituir-se os *Data Marts*.

Cada *Data Mart* gerado a partir do macro-modelo é integrado ao modelo físico do *Data Warehouse*.

A principal vantagem desta implementação é a garantia da consistência dos dados. Isso se dá porque o modelo dos *Data Marts* é único, possibilitando realizar o mapeamento e o controle dos dados. Uma visão da implementação combinada é dada na figura 15.

Fig. 15: Implementação Combinada



Fonte: Machado, 2000.

3.3 O Modelo Dimensional

A modelagem multidimensional é uma técnica de concepção e visualização de um modelo de dados, de um conjunto de medidas que descrevem aspectos comuns de negócios.

Utiliza-se esse modelo para sumarizar e reestruturar dados e apresentá-los em visões que suportem a análise dos valores desses dados.(MACHADO, 2000; KIMBALL, 2002, 2003a)

O modelo dimensional está baseado em três elementos:

- Fatos;
- Dimensões;
- Medidas

Os termos dimensões e fatos tiveram origem em um projeto de pesquisa em conjunto realizado pela *General Mills* e pela *Dartmouth University* na década de 60. Na década de 70, AC Nielsen e IRI, usaram esses termos de modo consistente para descrever suas ofertas de dados corporativos, que poderiam ser descritos precisamente hoje em dia como *data marts* dimensionais para dados de vendas no varejo (KIMBALL, 2002).

3.3.1 Fatos

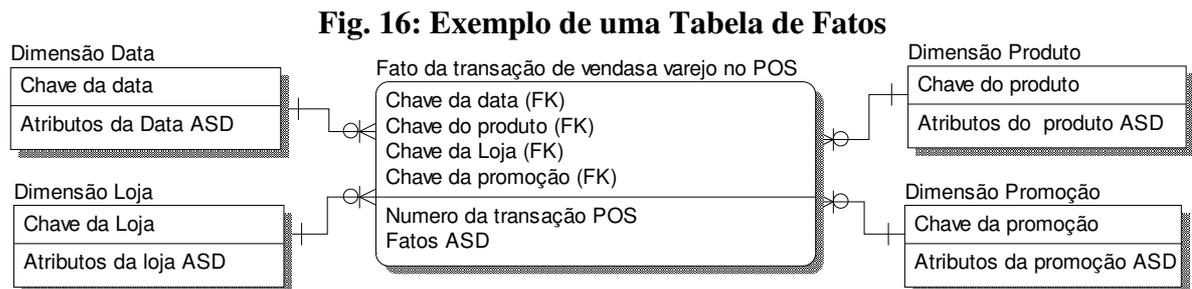
Fato é uma coleção de itens de dados compostos, de dados de medidas e de contexto; representa um item de negócio, uma transação de negócio ou um evento de negócio. É utilizado para analisar o processo de negócio de uma empresa (MACHADO, 2000).

Em um *Data Warehouse* o fato é representado por uma tabela onde são armazenadas as medições de negócio.

A tabela de fatos é a principal tabela de um modelo dimensional. É onde as medições numéricas de desempenho da empresa estão armazenadas.

A capacidade de adição é fundamental em uma tabela de fatos, isso porque um *Data Warehouse* raramente recupera uma única linha, geralmente recupera muitas linhas por vez, podendo chegar aos milhares ou até milhões.

Além dos fatos aditivos, existem os fatos semi-aditivos e os não aditivos, os semi-aditivos podem ser adicionados a algumas dimensões os não aditivos simplesmente não podem. Mas, os fatos numéricos e aditivos são os mais úteis e conseqüentemente os mais utilizados em um *DW*. Como exemplo de fatos aditivos é possível citar o volume de vendas de um supermercado. Na figura 16 observa-se um exemplo de uma tabela de fato.



Fonte: Kimball, 2002.

3.3.2 Medidas

Medidas são os atributos numéricos que representam um fato. Elas representam a performance de um indicador de negócios relativo às dimensões que participam deste fato. Esses números são denominados de variáveis e estão localizados como atributos de um fato. (KIMBALL, 2002; MACHADO, 2000).

3.3.3 Dimensão

Dimensões são conceitualmente os elementos que participam de um fato.

As dimensões determinam o contexto de um assunto de negócio, como por exemplo: o fato que analisa as vendas de um produto, como visto na figura 16.

Dimensões normalmente não possuem atributos numéricos, pois são somente descritivas e classificatórias dos elementos que participam de um fato. (KIMBALL, 2002; MACHADO, 2000).

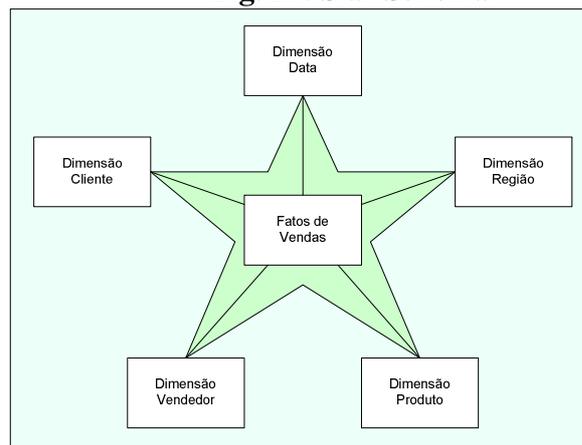
3.3.4 Star Schema

O modelo estrela (*Star Schema*) (KIMBALL, 2002; MACHADO, 2000) é a designação mais comum de uma tabela de fatos em um modelo dimensional.

Sua composição típica possui uma grande entidade central denominada fato (tabela de fato) e um conjunto de entidades menores denominadas dimensões (tabela de dimensões), arranjadas ao redor desta entidade central, formando uma estrela, como mostra a figura 17. (KIMBALL, 2002b; MACHADO, 2000).

Na figura 17 o centro da estrela é o fato vendas e ao seu redor estão dispostas as dimensões que estão ligadas aos fatos: Vendedor, Cliente, Produto, Região e Data..

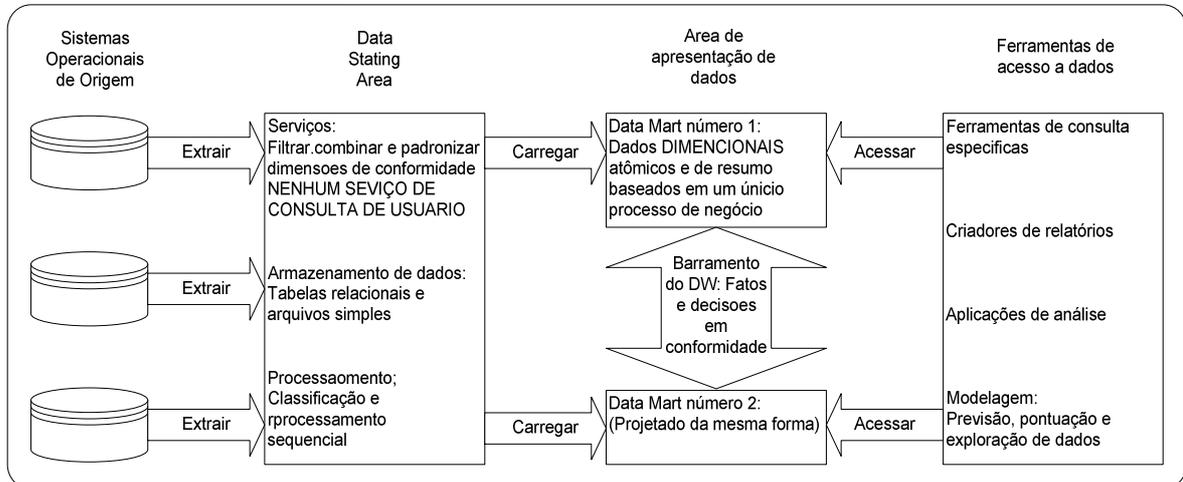
Fig. 17: Star Schema



Fonte: Machado, 2000.

3.4 Componentes de um *Data Warehouse*

Fig. 18: Elementos Básicos de um *Data Warehouse*



Fon: Kimball, 2002.

Na figura 18 identifica-se quatro componentes: sistemas operacionais de origem, *data staging area*, área de apresentação de dados e ferramentas de acesso a dados.

3.4.1 Sistemas Operacionais de Origem

Os sistemas operacionais são responsáveis por capturar as transações diárias da empresa.

As principais prioridades dos sistemas de origem são o desempenho e a disponibilidade de processamento. As consultas feitas nos sistemas de origem são consultas limitadas, feitas em um registro por vez, que fazem parte do fluxo normal de transações e com uma demanda no sistema operacional extremamente limitada, eles não são consultados de forma ampla e inesperada.

Os sistemas de origem mantêm um volume pequeno de dados históricos e quando se estiver com um *Data Warehouse* adequado, a responsabilidade de os sistemas operacionais representarem o passado será significativamente reduzido.

3.4.2 Data Staging Area

A *data staging area* do *Data Warehouse* é a área de armazenamento com um conjunto de processos que, normalmente denomina-se *ETL* (*Extract-Transformation-Load*) (KIMBALL,2002; CIELO, 2003c; DWBRASIL, 2003g) . A *data staging area* abrange os sistemas operacionais de origem e a área de apresentação dos dados. Na verdade é onde os dados operacionais brutos são transformados para o formato dimensional.

A *data staging area* só pode ser acessada por profissionais qualificados. O principal requisito é que ela não esteja acessível aos usuários e que não forneça serviços de consulta nem de apresentação.

3.4.2.1 Extração

A extração é a primeira etapa do processo de obtenção de dados. O processo envolve a leitura e a compreensão dos dados de origem, copiando os dados necessários ao *Data Warehouse* para que sejam manipulados posteriormente.

3.4.2.2 Transformação

Após os dados já estarem na *data staging área*, ocorrem às transformações como filtragem dos dados (correções de erros de digitação, solução de conflitos de domínio, tratamento de elementos ausentes ou a divisão em formatos padrão), combinação de dados de várias origens, cancelamento de dados duplicados e atribuição de chaves de *Warehouse*.

A *data staging area* é dominada pelas atividades simples de classificação e processamento seqüencial. Algumas equipes de projeto de *data warehouse* falharam, e muito, porque concentraram todos os esforços e recursos na construção das estruturas normalizadas e não na alocação de tempo para o desenvolvimento de uma área de apresentação que suportasse uma melhor tomada de decisões de negócio.

3.4.2.3 Carga

A última etapa de um processo de *ETL* é carregar os dados. O processo de carga no ambiente do *Data Warehouse*, normalmente assume a forma de um processo de apresentação de tabelas dimensionais, garantidas por controle de qualidade para os vários recursos de carga de cada *data mart*.

A publicação inclui a comunicação da natureza de qualquer alteração que tenha ocorrido nas dimensões subjacentes e novas conclusões que tenham sido introduzidas nos fatos medidos ou calculadas.

3.4.3 Apresentação dos dados

A área de apresentação dos dados é o local em que eles ficam organizados, armazenados e tornam-se disponíveis para serem consultados diretamente pelos usuários, por criadores de relatórios e por outras aplicações de análise. Essa área é tudo o que a comunidade de negócio vê e acessa através das ferramentas de acesso a dados. (KIMBALL, 2002; DWBRASIL, 2003f).

Normalmente, à área de apresentação refere-se a uma série de *Data Marts* integrados. Um *Data Mart* é uma parte do todo que compõe a área de apresentação. Em sua forma mais simples um *Data Mart* representa os dados de um único processo de negócio.

Com o amadurecimento do setor de *Data Warehouse* concluiu-se que a modelagem dimensional é a técnica mais viável para entregar dados a usuários. Modelagem dimensional é um novo nome para uma técnica antiga que permitia tornar os bancos de dados fáceis e compreensíveis.

A maioria das pessoas acha intuitivo considerar esse ramo de negócio como um cubo de dados com as faces identificadas como produto, mercado e tempo. É possível imaginar a aplicação do recurso de “separação e combinação” (*slicing and dicing*) em cada uma dessas dimensões. Os pontos dentro do cubo são o local em que as medições dessa combinação de produto, mercado e tempo, estão armazenados. A capacidade de visualizar algo tão abstrato como um conjunto de dados de forma concreta e tangível é o segredo da capacidade de compreensão. Um modelo de dados que começa sendo simples tem uma grande chance de continuar simples até o fim. Já um projeto que começa sendo complicado, certamente tende a ficar bem mais complicado no fim. Modelos extremamente complicados serão executados lentamente e certamente serão rejeitados pelos usuários finais. (KIMBALL, 2002; DWBRASIL, 2003f; MACHADO, 2000).

Um modelo dimensional contém as mesmas informações que um modelo normalizado, mas empacota os dados em um formato cujos objetivos de design são a capacidade de ser compreendido pelo usuário, o desempenho de consulta e a flexibilidade de adaptação a mudanças.

Os *Data Marts*, precisam conter dados detalhados e atômicos. Os dados atômicos são necessários para suportar às investidas de consultas de usuários específicas e inesperadas.

Todos os *Data Marts* devem ser construídos usando dimensões e fatos diferentes, isto é, em conformidade. Sem dimensões e fatos compartilhados e em conformidade, um *Data Mart* é uma aplicação independente. (KIMBALL, 2002; DWBRASIL, 2004).

Os dados na área de apresentação que podem ser consultados no *Data Warehouse* precisam ser dimensionais, atômicos e obedecerem à arquitetura de barramento do *Data Warehouse*.

3.4.4 Ferramentas de Acesso a Dados

O último componente é as ferramentas de acesso a dados. Pois, a consulta é o ponto principal no uso do *Data Warehouse*.

Uma ferramenta de acesso a dados pode ser tão simples como uma ferramenta de consulta específica, a exemplo do *Microsoft Excel*, ou tão complexa quanto uma aplicação sofisticada de modelagem ou exploração de dados. (KIMBALL, 2002; DWBRASIL, 2003f).

3.4.4.1 O Cubo

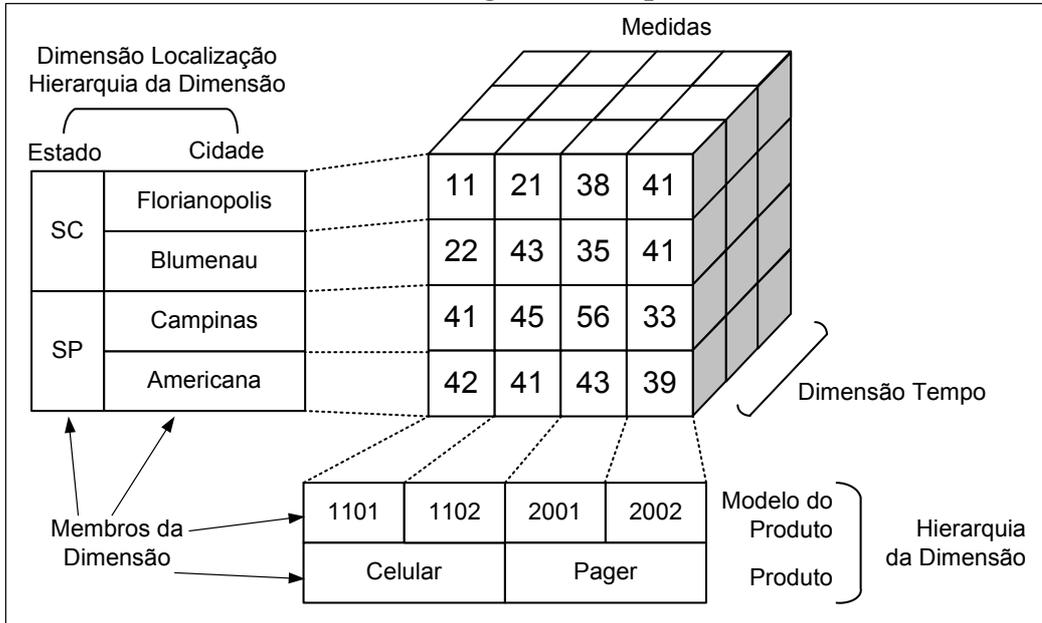
O meio mais popular de visualização de um modelo dimensional é a visão de um cubo.

Em um cubo é possível visualizar até três dimensões, mas em um modelo dimensional, geralmente utiliza mais de três, sendo assim definido como hipercubo. É muito difícil visualizar graficamente um hipercubo, ressalta-se que se utiliza a referência de cubo para qualquer modelo dimensional.

A metáfora denominada CUBO é apenas uma aproximação da forma como os dados estão organizados, e não a expressão de uma realidade (MACHADO, 2000).

A razão de utilizar o cubo é transmitir a idéia de múltiplas dimensões. Na figura 19 é possível ter uma visão detalhada de um cubo:

Fig. 19: Exemplo de um Cubo



Fonte: Machado, 2000.

3.4.4.2 Operações Básicas de OLAP

O modelo multidimensional possui quatro operações básicas. Utilizando a granularidade tem-se operações de *drill down* e *roll up* e navegando pelas dimensões tem-se as operações de *slice and dice*. (THOMSEN, 2002; MACHADO, 2000; DWBRASIL, 2003f; CIELO, 2003d; KIMBALL, 2003cde).

3.4.4.2.1 Drill Down e Roll Up

O *Drill Down* e o *Roll Up* são operações para movimentar a visão dos dados ao longo dos níveis hierárquicos de uma dimensão.

O *Drill Down* faz o usuário navegar do nível mais alto até o nível mais baixo, como visto na tabela 2. O *Roll Up* faz o usuário navegar do nível mais baixo ao mais alto, como visto na tabela 3.

Tabela 2: Demonstração do *Roll Up*

Volume de Produção (Em milhares)		1999			
		1º Trimestre	5º Trimestre	3º Trimestre	4º Trimestre
Região	RGS	78	67	22	56
Sul	SC	90	67	88	99



 Roll UP
 Dimensão Tempo

Volume de Produção (Em milhares)		1º Trimestre		
		Janeiro	Fevereiro	Março
Região	RGS	30	26	22
Sul	SC	28	30	32

Fonte: Machado, 2000.

Tabela 3: Demonstração do *Drill Down*

Volume de Produção (Em milhares)		1999			
		1º Trimestre	5º Trimestre	3º Trimestre	4º Trimestre
Região	RGS	78	67	22	56
Sul	SC	90	67	88	99



 Drill Down
 Dimensão Tempo

Volume de Produção (Em milhares)		1º Trimestre		
		Janeiro	Fevereiro	Março
Região	RGS	30	26	22
Sul	SC	28	30	32

Fonte: Machado, 2000.

3.4.4.2.2 *Slice and Dice*

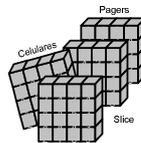
Slice and Dice são operações para realizar navegação pelo meio dos dados na visualização de um cubo.

Slice é a operação que corta o cubo, mas mantém a mesma perspectiva de visualização dos dados, como visto na tabela 4.

Dice é a mudança de perspectiva da visão, como visto na tabela 5.

Tabela 4: Demonstração do *Slice*

Volume de Produção (Em milhares)		Celular e <i>Pagers</i>		
		Janeiro	Fevereiro	Março
Região	RGS	30	26	22
Sul	SC	28	30	32

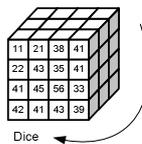


Volume de Produção (Em milhares)		Celular		
		Janeiro	Fevereiro	Março
Região	RGS	22	18	18
Sul	SC	19	27	25

Fonte: Machado, 2000.

Tabela 5: Demonstração do *Dice*

Volume de Produção (Em milhares)		1999			
		Telefone Celular		Pagers	
		1001	1002	2001	2002
RGS	Canoas	13	4	2	5
	Porto Alegre	20	8	6	7



Volume de Produção (Em milhares)		1999	
		RGS	
		Canoas	Porto Alegre
Telefone	1001	13	2
Celular	1002	4	5
<i>Pagers</i>	2001	2	6
	2002	5	7

Fonte: Machado, 2000.

3.5 Conclusão

Este capítulo focou o que vem a ser um *Data Warehouse*. O seu conceito, suas diversas arquiteturas comparando umas com as outras, observando as vantagens e as desvantagens de cada uma delas.

Demonstrou-se como o banco de dados é modelado e como os dados são inseridos, além da forma como os dados são trabalhados a fim de se obter as informações desejadas no apoio à decisão.

De posse de todas essas informações sobre o *Data Warehouse* é possível definir qual a melhor arquitetura e como direcionar o projeto de sua implementação.

CAPÍTULO IV

ESTUDO DE CASO – O SETOR PRODUTIVO DE UMA INDÚSTRIA DE EMBALAGENS FLEXÍVEIS

4.1 Introdução

Este capítulo apresenta o processo de produção de uma indústria de plásticos flexíveis e sua atual estrutura, no que se refere a sistemas de informação.

Este também enfatiza detalhes da metodologia de desenvolvimento de um *Data Warehouse*, passando por cada fase do desenvolvimento do mesmo. A metodologia aqui apresentada é uma proposta de Ralph Kimball.

O setor produtivo, referente ao chão de fábrica, está bastante estruturado e automatizado. O que será demonstrado neste capítulo, com a implantação de *Data Warehouse*, é um avanço na hierarquia da pirâmide de tecnologia da informação (ver figura 6) preenchendo um espaço vazio referente ao tratamento das informações.

4.2 A Indústria de Embalagens

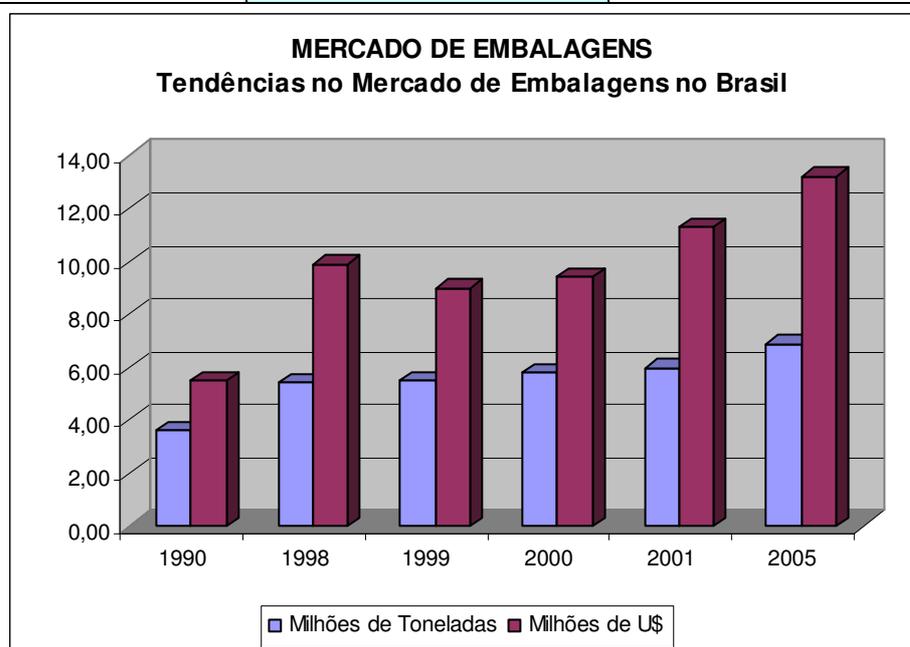
A definição de embalagem em Filosofia de marketing é: “vender o que protege e proteger o que vende” ou “a embalagem é a arte, a ciência e a técnica de acondicionar o produto, para que ele seja transportado, vendido e consumido” (GONZALEZ, 2004).

A embalagem se destaca entre as atividades industriais, pelo volume que representa no consumo de diversas matérias-primas e por sinalizar rapidamente qualquer alteração no mercado de consumo.

Segundo a Datamark (2004), a indústria brasileira de embalagens vem crescendo a cada ano, tanto em peso como em valores, fato que pode ser visto na figura 20.

Fig. 20: Crescimento e Tendências do Mercado de Embalagens

	Milhões de Toneladas	Milhões de U\$
1990	3,60	5,50
1998	5,40	9,90
1999	5,50	9,00
2000	5,80	9,40
2001	6,00	11,30
2005	6,90	13,20



Fonte: Datamark.

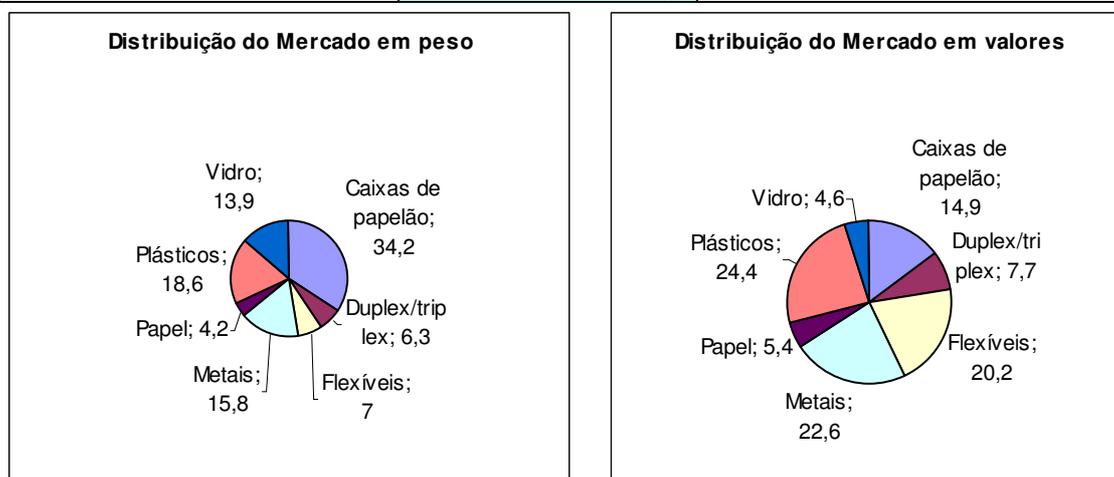
Na verdade o mercado cresceu mais do que o gráfico mostra. Há um constante esforço para diminuir os custos através da diminuição do peso, por isso a tonelagem não cresce, além do fato da desvalorização ter distorcido o valor do mercado (em 1999 o Real teve uma desvalorização de 40% em relação ao dólar e os preços de plásticos se relacionam com o dólar).

Ainda segundo o Datamark (2004), a indústria brasileira de embalagens foi avaliada em US\$ 8.182 milhões (6,3 milhões de toneladas no total) em 2002, equivalente a 1,8% do

produto nacional bruto. A distribuição entre os vários materiais pode ser vista na figura 21, onde se demonstra que o segmento de plásticos está em segundo na quantidade e em primeiro no valor (DATAMARK, 2004).

Fig. 21: Distribuição do Mercado de Embalagens em 2002

	PESO %	VALOR %
Caixas de papelão	34,2	14,9
Duplex/triplex	6,3	7,7
Flexíveis	7	20,2
Metais	15,8	22,6
Papel	4,2	5,4
Plásticos	18,6	24,4
Vidro	13,9	4,6



Fonte: Datamark.

Segundo Gonzáles (2004), no período de 1995 a 2001 (7 anos), alguns materiais concentraram um forte crescimento. É o caso do PET, que teve alta de 150%, do alumínio com 137% e da embalagem multicamada, com 81%.

De um modo geral, os plásticos tiveram um bom desempenho, com alta de 54%, com exceção do PVC, que teve queda de 46%. Os desempenhos negativos ficaram por conta do papel monolúcido (-24%), tambores de aço (-5%) e vidro (-10%)

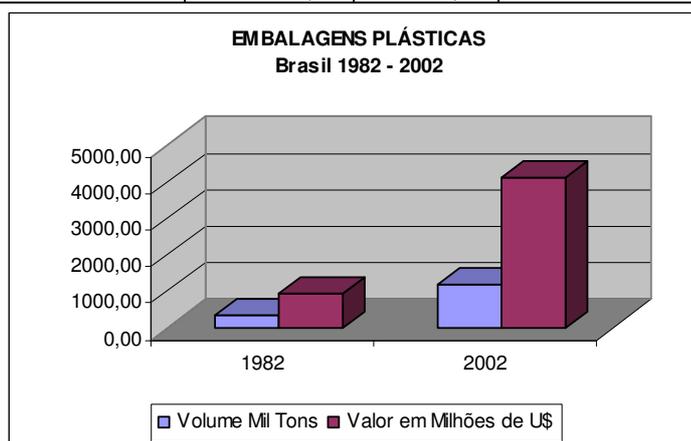
4.2.1 Embalagens Plásticas Flexíveis

Embalagem flexível é um envoltório maleável de fácil manuseio, no qual podem ser acondicionados sólidos ou líquidos, em diversos volumes, formatos e dimensões, através da estrutura confeccionada, utilizando diversos tipos de materiais, associados entre si ou não (GONÇALEZ, 2004).

As embalagens flexíveis em geral têm um dos mais expressivos crescimentos entre as embalagens, de 154% no período de 1990 a 1998. Espera-se que este mercado cresça em 42% até 2005. Em uma estatística feita nos últimos 20 anos (de 1982 a 2002), pelo Datamark (2000), mostra um crescimento de 378,78% em volume e 445,38% em toneladas como pode ser visto na figura 22.

Fig. 22: Crescimento das Embalagens Plásticas no Brasil

	1982	2002	Crescimento
Volume Milhões Tons	311,00	1178,00	378,78%
Valor em Milhões de U\$	930,00	4142,00	445,38%



Fonte: Datamark.

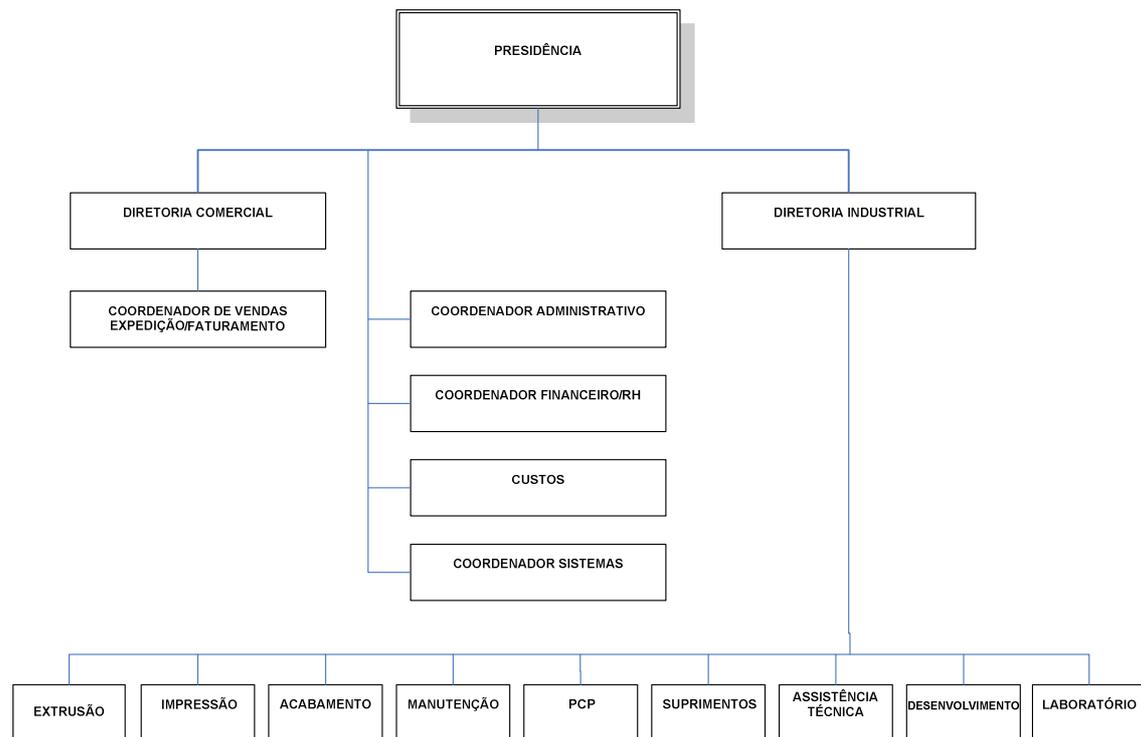
4.3 Conhecendo a PLASC

Localizada no sul do Brasil, na grande Florianópolis, a PLASC (Plásticos Santa Catarina Ltda), atua no ramo industrial de plásticos flexíveis e tem como missão, desenvolver e suprir o mercado, com produtos plásticos e embalagens. Tem como base os princípios da competitividade global e a preservação do meio ambiente, superando as expectativas e necessidades dos clientes, empregados e acionistas.

Na figura 23 é possível ter uma visão do atual organograma da empresa, demonstrando como a mesma esta organizada.

Fig. 23: Organograma da PLASC

Organograma PLASC - Indústria de Plásticos Santa Catarina Ltda

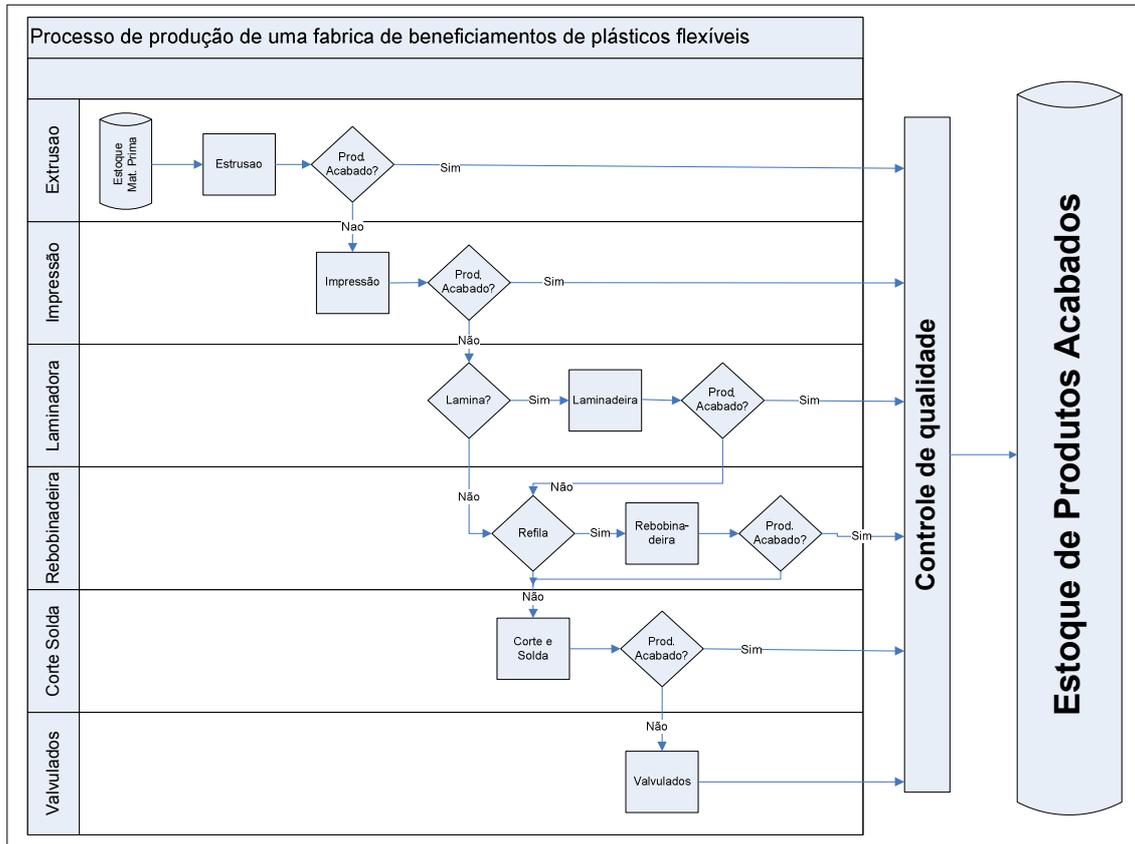


Fonte: Plasc.

4.3.1 Estrutura de Produção

A Estrutura de produção da PLASC pode ser compreendida através da figura 24, onde todo o processo produtivo da empresa é demonstrado com detalhes.

Fig. 24: Estrutura de Produção



Fonte: Autor.

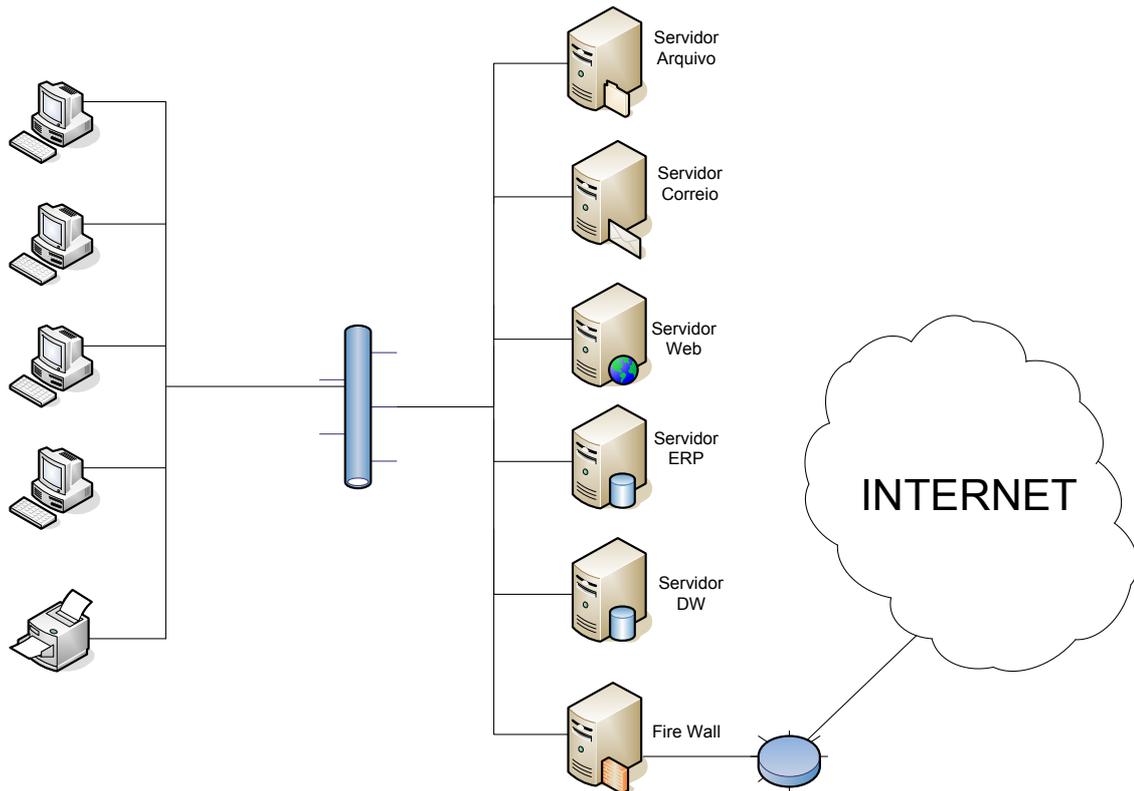
4.3.2 Estrutura de TI

É possível ter uma visão ampla da estrutura de TI da PLASC através da figura 25. Ela demonstra como estão distribuídos os serviços internos e externos, oferecidos aos usuários da empresa.

A estrutura de TI está baseada no sistema de ERP da Microsiga, o *Advanced Protheus*. O ERP está bem consolidado na empresa contando atualmente com a versão *Máster* do

produto. O mesmo roda na plataforma *MS-Windows 2000* com o banco de dados *MS-SQL Server 2000*, usufruindo, de toda estrutura de rede, com o servidor de correio para disparar *e-mails* automáticos através da ferramenta de *workflow* e o servidor *Web* para disponibilizar acesso ao módulo de representantes.

Fig. 25: Estrutura de TI da PLASC



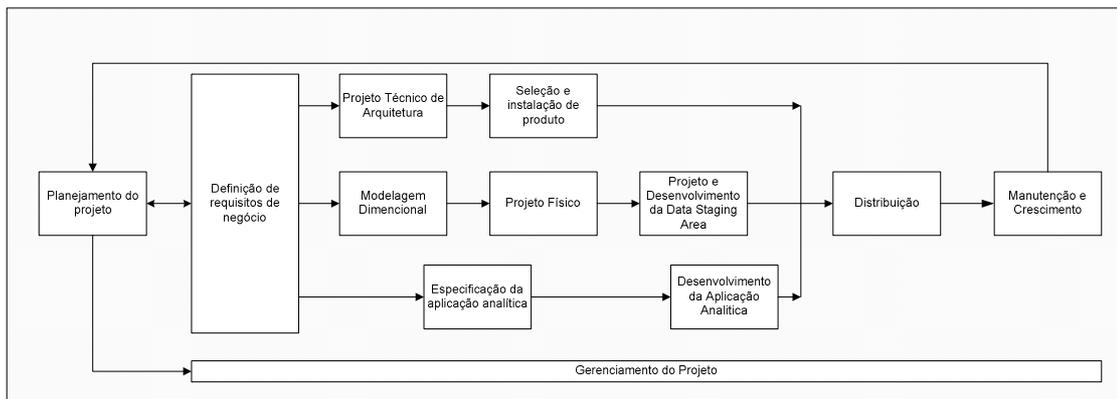
Fonte: Autor.

Agora com a implantação do *Data Warehouse*, a empresa busca agilidade nas tomadas de decisões, com um acesso mais rápido e fácil às informações. Procurando assim, preencher o espaço vazio na pirâmide organizacional (ver figura 5), começando pelo setor de produção.

4.4 Metodologia de Desenvolvimento

A metodologia de desenvolvimento do projeto está baseada no diagrama do ciclo de vida dimensional de negócio, desenvolvido por Ralph Kimball (2002), representado aqui pelo diagrama da figura 26.

Fig. 26: Diagrama do Ciclo de Vida Dimensional do Negócio



Fonte: Kimball, 2002.

Esse diagrama inicia-se com o planejamento do projeto. A partir deste ponto, o gerenciamento ocorre de forma contínua durante todo o mesmo.

A segunda parte consiste em levantar e definir os requisitos de negócio. A partir daí, todo o restante do projeto pode ser dividido em três trilhas principais tendo impacto direto desta parte.

A terceira parte como mencionado acima, pode ser visto em três trilhas principais: a trilha tecnológica, trilha de dados e trilha de aplicações analíticas.

- **Trilha Tecnológica:** define a arquitetura técnica e a seleção e instalação dos produtos;
- **Trilha de Dados:** transforma os requisitos de negócio em um modelo dimensional que se tornará um modelo físico. É também, nessa trilha que se dá o processo de ETL e tudo o que se refere a tratamento e manipulação dos dados;

- **Trilha de Aplicações analíticas:** parte do DW onde efetivamente aparecem os resultados de todo o trabalho. É onde se projeta e desenvolve as aplicações que enfocam os requisitos analíticos dos usuários.

4.4.1 Planejamento e Gerenciamento do Projeto

O *Data Warehouse* é uma série de atividades de planejamentos. Algumas vezes, denominadas tarefas *marshmallow*, pois são maleáveis, pegajosas e podem danificar seriamente um projeto de *data warehouse*.

O planejamento é um fator chave de sucesso do projeto, e deve envolver as lideranças influentes na organização, os decisores departamentais, o pessoal de TI e todos os que são relevantes ao sucesso e continuidade do projeto.

Segundo Kimball (2002), o planejamento e gerenciamento do projeto divide-se em: avaliação da preparação, escopo, justificativa, trabalho em equipe, desenvolvimento e manutenção do projeto

4.4.1.1 Avaliação da Preparação

Segundo Kimball (2002), é prudente dedicar um tempo à avaliação da preparação da empresa para o prosseguimento. O mesmo Kimball identifica cinco fatores que diferenciam projetos que foram de navegação suave, aos que exigem esforço constante.

- O fator mais importante, para um *Data Warehouse* bem-sucedido, é ter um patrocinador forte na área de negócio que é cliente do *Data Warehouse*. Mas, deve se

tomar cuidado com patrocinadores isolados, pois os mesmos tendem a seguir adiante interna e externamente causando estagnação do *DW*;

1. O segundo fator é ter motivação forte e convincente para a construção de um *Data Warehouse*, baseada nas necessidades do negócio. Esse fator muitas vezes anda de mãos dadas com o patrocínio.
2. Um projeto de *DW* precisa solucionar problemas fundamentais do negócio, e uma motivação irrefutável normalmente cria um senso de urgência.
3. O terceiro fator é a viabilidade. Há vários aspectos de viabilidade técnica ou recursos, mas a viabilidade dos dados é a mais importante.
4. O quarto fator se concentra no relacionamento entre o negócio e a organização de TI. Na empresa a organização de TI compreende e respeita o negócio? De forma inversa, o negócio compreende e respeita a organização de TI?
5. O quinto e último aspecto é a cultura analítica atual dentro da empresa. Os analistas de negócio tomam decisões com base nos fatos e nos valores, ou as suas decisões são baseadas na intuição, na evidência de informações e nas reações instintivas?

O planejamento do projeto iniciou-se em outubro de 2003 com a escolha dos patrocinadores que são: Diretor Comercial, Diretor Industrial e Coordenador de Sistemas (ver organograma da empresa na figura 23) .

4.4.1.2 Escopo

O Escopo requer a entrada conjunta do gerenciamento da organização de TI e da empresa. Quando iniciar, deve-se concentrar nos dados de um único processo de negócio. Salve os projetos mais desafiadores e de processo cruzado para uma fase posterior.

Após algumas reuniões dentro do mesmo mês de outubro, foram levantadas as necessidades básicas da empresa, onde por consenso, definiu-se o início pelo setor produtivo, com a finalidade de fornecer informações administrativas. A princípio, ao diretor industrial e, posteriormente a toda empresa.

Uma das visões necessárias é o controle e acompanhamento de perdas, verificando o relacionamento com: Tipo de Perda, Produto, Máquina e Turno.

Outra análise necessária é o controle e acompanhamento do tempo de parada da máquina, verificando o relacionamento com: Motivo da Parada, Produto e Turno.

Um dos motivos mais importantes de parada, é a parada por Setup da máquina. Requerendo uma visão mais criteriosa.

4.4.1.3 Justificativa

A justificativa requer uma estimativa das vantagens e dos custos associados a um *Data Warehouse*; é de se esperar que as vantagens antecipadas compensem os custos.

No caso de uma indústria de embalagens, que sinaliza rapidamente qualquer alteração no mercado de consumo, a velocidade associada com a clareza das informações, é um fator chave de sucesso para se tomar decisões rápidas e certas.

É com a obtenção rápida de índices atualizados, onde mostram o comportamento da empresa que trará um diferencial competitivo real, na forma de como proceder, neste mercado tão competitivo e mutável.

4.4.1.4 Trabalho em Equipe

Os projetos de *Data Warehouse* requerem a integração de equipes funcionais, cruzadas com recursos da empresa e das comunidades de TI. É comum a mesma pessoa ocupar mais uma função (papel), especialmente quando o custo de entrada para o *Data Warehouse* diminui.

Da área de negócios da casa

Patrocinador da área de negócios, controlador da área de negócio, líder da área de negócio e usuários da área de negócio. No projeto em questão, essas áreas são acumuladas pelo diretor industrial (ver organograma na figura 23) .

Novas áreas

Analista de sistema de negócios, perito em negócios, desenvolvedor de aplicações analíticas, instrutor de data warehouse, organizações de TI, gerente de projeto, arquiteto técnico, especialista em suporte técnico, modelador de dados, administrador de banco de dados, coordenador de metadados, organizador de dados, projetista de data staging área, desenvolvedor da data staging área e suporte de data warehouse. Todas essas funções foram acumuladas pelo redator deste trabalho, auxiliado pelo coordenador de sistemas da empresa.

4.4.1.5 Desenvolvimento e Manutenção

O projeto deve identificar um ponto de verificação, de aceitação do usuário, segundo todo marco principal e que pode ser transmitido para garantir que o projeto mantenha o caminho certo e que o negócio continue intimamente envolvido.

O projeto de *data warehouse* demanda ampla comunicação. Para tal foi estabelecida uma matriz de comunicação representada aqui pela tabela 6, adaptada da matriz sugerida por Kimball (2002).

Tabela 6: Plano de Comunicação

PÚBLICO	FREQÜÊNCIA	FORUM	MENSAGENS PRINCIPAIS
Patrocinadores (Diretor Industrial)	Uma vez por mês (Segundo Kimball [2002] o ideal seria duas vezes por mês)	Relato pessoal dos fatos	Resolução de problemas e pendências, gerenciamento de expectativas, financiamento.
Comunidade de Negócio (Diretor de Industrial e Coordenador de Sistemas)	Quinzenal	Reunião Informal (Segundo Kimball [2002], o ideal um meio mais formal e de fácil acesso. Exemplo: um Web site)	Envolvimento de requisitos, gerenciamento de expectativas, dados fundamentais.
Equipe de Projeto (Mestrando e Orientador)	Quinzenal (Segundo Kimball [2002]o ideal seria uma vez por semana)	Reunião de status	Andamento, identificação e resolução de problemas e pendências.
Colegas de TI (Mestrando e Coordenador de Sistemas)	Quinzenal	Reunião de equipes de TI existentes	Gerenciamento de expectativas, necessidades de recursos.

Fonte: do autor, adaptado de Kimball (2002)

4.4.2 Definição dos Requisitos de Negócio

É essencial entender os requisitos dos usuários e se adequar as suas necessidades para obter um *Data Warehouse* bem-sucedido. As técnicas básicas para a realização desta tarefa são:

- Escolha do fórum.

- Identificação e preparação da equipe de requisitos.
- Seleção, programação e preparação dos representantes de negócio.
- Coleta dos requisitos de negócio.
- Abertura da entrevista.
- Fluxo da entrevista.
- Encerramento.
- Condução de entrevistas centradas nos dados.
- Documentação pós-coleta e seguimento.
- Priorização e consenso.

No dia a dia do processo produtivo, a empresa conta com o sistema de ERP da Microsiga para organizar e controlar cada setor do processo produtivo (extrusão, impressão, acabamento, manutenção, pcp, suprimentos, assistência técnica, desenvolvimento e laboratório). Contudo, o diretor industrial, necessita de informações relevantes e em tempo hábil para ter uma visão do todo e ter o suporte necessário à tomada de decisão. Sendo assim necessita das seguintes visões:

Relação 1 - Perda

- Qual o maior motivo de perda?
- Qual relação Perda X Produto?
- Qual relação Perda X Produto X Operador?
- Qual relação Perda X Operador?
- Qual relação Perda X Máquina?
- Qual relação Perda X Produto X Máquina?
- Qual relação Perda X Produto X Máquina X Operador?

Relação 2 - Tempo de Parada

- Qual o maior motivo de parada?
- Qual relação Parada X Operador?
- Qual relação Parada X Produto?
- Qual relação Parada X Produto X Operador?
- Qual relação Parada X Máquina?
- Qual relação Parada X Máquina X Operador?
- Qual relação Parada X Produto X Máquina?
- Qual relação Parada X Produto X Máquina X Operador?

Relação 3 - Setup

- Qual relação Parada *Setup* X Operador?
- Qual relação Parada *Setup* X Produto?
- Qual relação Parada *Setup* X Produto X Operador?
- Qual relação Parada *Setup* X Máquina?
- Qual relação Parada *Setup* X Máquina X Operador?
- Qual relação Parada *Setup* X Produto X Máquina?
- Qual relação Parada *Setup* X Produto X Máquina X Operador?

Relação 4 - Produção

- Qual a produção da Fábrica por Período?
- Qual a quantidade produzida por período, segmentados por produto e suas subclassificações (Processo, Formato, Segmento, Classe e Composição)?
- Qual a relação da Quantidade Produzida X Perda?
- Qual a relação do Tempo de Produção X Tempo de Parada.?
- Qual a relação da Quantidade Produzida X Perda X Tempo de Produção X Tempo de Parada?

4.4.3 Projeto da Arquitetura Técnica

A arquitetura técnica é um projeto dos serviços e elementos técnicos do *warehouse*, servindo para suportar a integração de tecnologias. Segundo Kimball (2002), um projeto de arquitetura técnica consiste em uma série de modelos que se aprofundam em cada detalhe dos componentes mais importantes. São identificáveis oito etapas de processos:

- Estabelecimento de uma força tarefa de arquitetura.
- Coleta de requisitos relacionados à arquitetura.
- Requisitos arquitetônicos do documento.
- Desenvolvimento de um modelo arquitetônico de alto nível.
- Projeto e especificação dos subsistemas.
- Determinação de fases da implementação arquitetônica.
- Documentação da arquitetura técnica.
- Revisão e finalização da arquitetura técnica.

Conforme demonstrado na figura 25, a arquitetura técnica foi definida pelo coordenador de sistemas da empresa da seguinte forma:

- Um Servidor de Aplicação ERP (*Advanced Protheus da Microsiga*) rodando em *MS Windows 2000 Server*.
- Um Servidor de Banco de Dados Transacional *MS-SQL Server 2000* rodando atualmente na mesma máquina que o ERP.
- Um Servidor de *Web* rodando em Linux.
- Um Servidor de *Login* rodando em Linux.
- Um Servidor de OLAP Server denominado DW rodando em *MS Windows 2000 Server*.

- Um Servidor de banco de dados dimensional *MS-SQL Server 2000* rodando atualmente na mesma máquina do servidor OLAP.
- Um Servidor de correio rodando em Linux.
- Um Servidor de *Firewall* rodando em Linux.

4.4.4 Seleção e Instalação do Produto

O Plano arquitetônico é como uma lista de compras. São selecionados os produtos que se encaixam no contexto do plano arquitetônico do projeto a fim de se obter a funcionalidade necessária.

As tarefas associadas à seleção dos produtos são:

- Entender o processo corporativo de compras.
- Desenvolver uma matriz de avaliação dos produtos.
- Condução de pesquisa de mercado.
- Opções limitadas a uma pequena lista e execução de avaliação detalhada.
- Condução de protótipo, se necessário.
- Seleção de produto, instalação em teste e negociação.

No projeto em questão, foram utilizadas as ferramentas disponíveis na empresa.

Ficando a estrutura assim:

- O *Data Warehouse* está rodando em um servidor de banco de dados *MS SQL 2000 Server*, que por sua vez roda sobre a plataforma *MS Windows 2000 Server*.
- A *Data Stagim Área* (Área de Estagiamento) está rodando em um servidor de banco de dados *MS SQL 2000 Server*, que por sua vez roda sobre a plataforma *MS Windows 2000 Server*. A mesma máquina que o DW.

- O processo de ETL está rodando diariamente através de *scripts SQL*, complementado pelo ETL da ferramenta de DW da Microsiga.
- Para visualização dos cubos estão sendo utilizadas as ferramentas já disponíveis no ERP da Microsiga. Sendo elas:
- O E.I.S. (*Executive Information System*) – Ferramenta Olap utilizada pelos coordenadores de produção da empresa.
- O *WorkFlow* – Ferramenta de automatização do Fluxo de trabalho, onde dispara relatórios gerenciais e estratégicos.
- O DW – Ferramenta destinada à implementação de um DW, onde se faz grande parte o E.T.L. e as visualizações dos Cubos em forma de tabela e gráfico.

4.4.5 Modelagem Dimensional

Seguindo a metodologia de Kimball (2002), após a definição dos requisitos de negócio, devemos traçar ou revisar a matriz de barramento do *Data Warehouse*. Essa matriz é composta pelos fatos relacionados com suas respectivas dimensões, como pode ser visto através da tabela 7.

Tabela 7: Matriz de Barramento do *Data Warehouse*

	Dimensão Operador	Dimensão Turno	Dimensão Tipo de Perda	Dimensão Tipo de Parada	Dimensão Setor	Dimensão Recurso	Dimensão Processo	Dimensão Formato	Dimensão Segmento	Dimensão Classe	Dimensão Composição	Dimensão Produto
Fato Perda	X	X	X		X	X	X	X	X	X	X	X
Fato Tempo de Parada	X	X		X	X	X	X	X	X	X	X	X
Fato Produção	X	X			X	X	X	X	X	X	X	X

Fonte do Autor

Na etapa de priorização final das atividades dos requisitos de negócio, é identificado o processo de negócio específico que será tratado em cada tabela de fato. É necessário chegar ao âmago de cada processo, a fim de identificar a granularidade, a consistência histórica, os valores válidos e a disponibilidade de atributos.

Concluído a análise dos dados, deve-se fazer *workshops* de projeto para criar o esquema dimensional. Uma vez concluído, deve ser comunicado e validado com todos os envolvidos no projeto.

A documentação do modelo validado deve identificar os nomes de tabelas e de coluna, as definições e as regras de cálculo para fatos, ou então, as regras de dimensões que mudam lentamente para atributos de dimensão.

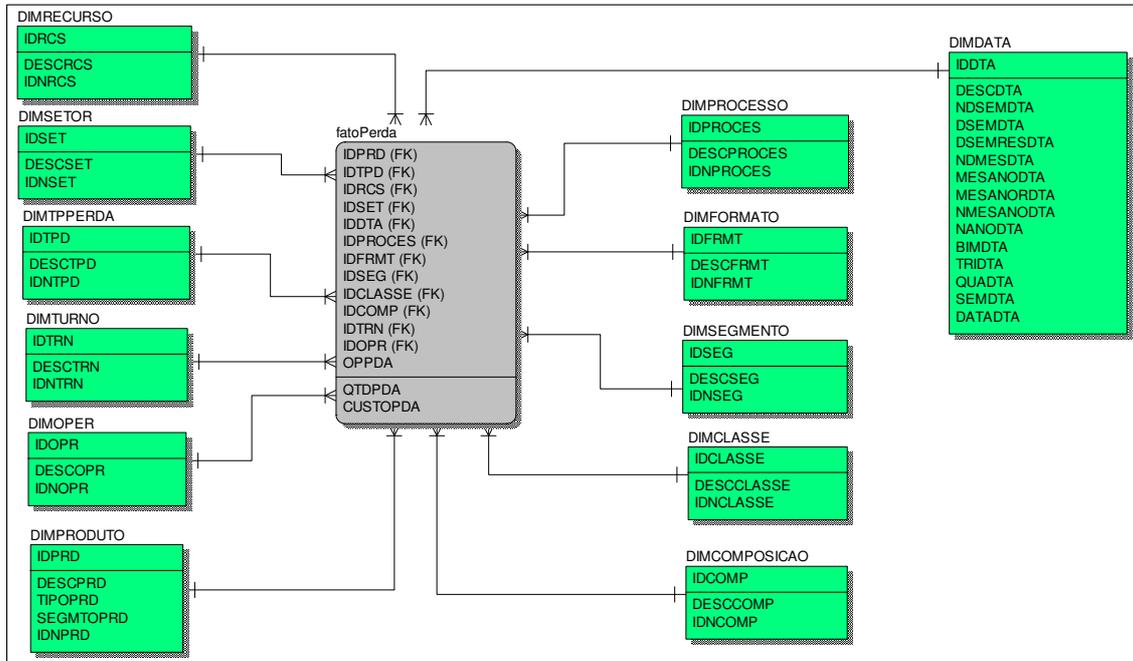
A documentação do esquema é suplementada pela inclusão do sistema de origem específico, dos campos e das regras de transformação para derivar o mapeamento de origem para destino, juntamente com a equipe de ETL.

No caso do projeto em questão foram levantados os fatos, Perda, Tempo de Parada e Produção. Levando em conta os requisitos de negócios definidos na tabela 6.

Para a modelagem, foi utilizada a ferramenta *case* ErWin, obtendo-se os diagramas abaixo.

- Fato Perda

Fig. 27: Fato Perda

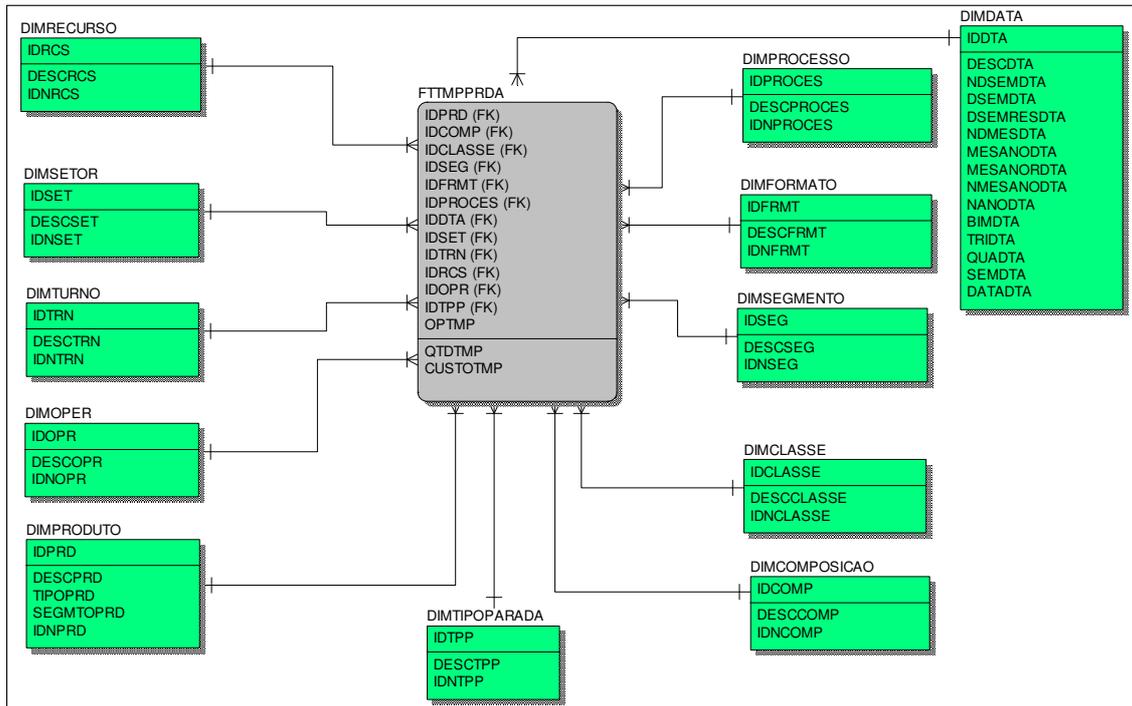


Fonte do Autor

Através do Fato Perda, pode-se obter todo o cenário das perdas de produção em Kg (Kilograma) da empresa, classificada por: período, produto e suas subclassificações (processo, formato, segmento, classe e composição), setor, recurso, operador, turno, motivo da perda e chegando até a ordem de produção.

- Fato Tempo de Parada

Fig. 28: Fato Tempo de Parada



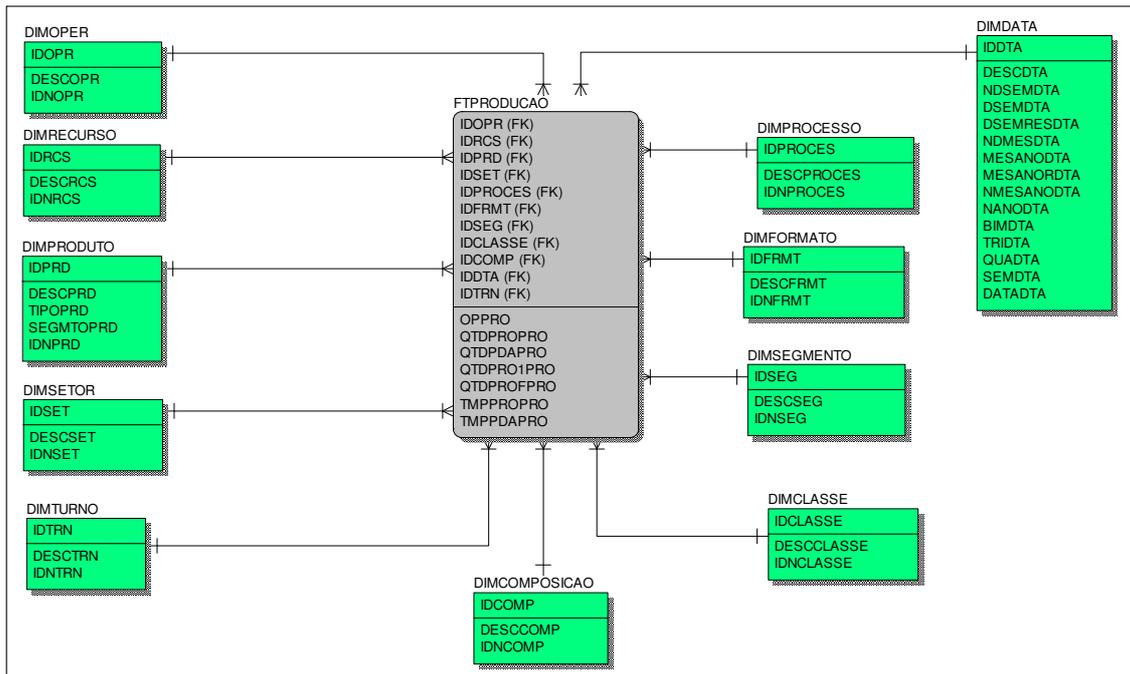
Fonte do Autor

Através do Fato Tempo de Parada, pode-se obter todo o cenário das paradas de máquinas (Recursos) na produção em minutos, classificadas por: período, produto e suas subclassificações (processo, formato, segmento, classe e composição), setor, recurso, operador, turno, motivo da parada e chegando até a ordem de produção.

Filtrando o Motivo da Parada por “*Setup*”, pode-se obter toda a relação acima descrita para o *Setup* da máquina, visto que este motivo de parada é tão relevante e inevitável no processo produtivo.

- Fato Produção

Fig. 29: Fato Produção



Fonte do Autor

Através do Fato Produção, pode-se obter todo o cenário do setor produtivo da empresa. Verificando a Produção em Kg (Kilograma), classificadas por: período, produto e suas subclassificações (processo, formato, segmento, classe e composição), setor, recurso, operador, turno, chegando até a Ordem de Produção. Pode-se verificar também a relação de Perda por Produção e Tempo de Parada por Tempo de Produção.

4.4.6 Projeto Físico

Os modelos dimensionais desenvolvidos precisam ser traduzidos em um projeto físico. Na modelagem dimensional, os projetos físicos e lógicos são semelhantes. O modelo físico diferirá do modelo lógico em termos de detalhes específicos do banco de dados a ser utilizado. O projeto físico também enfrenta atividades de elementos básicos como ajuste de desempenho, particionamento e *layout* dos arquivos.

Com relação ao desempenho criar índices e agregar tabelas são alternativas de melhor relação custo/benefício, ao invés de simplesmente melhorar o *hardware* (KIMBALL, 2002).

Para o projeto em questão, foi utilizado o banco de dados MS-SQL *Server* 2000 da *Microsoft* e como o projeto lógico foi modelado através do *ErWin*, bastou conectar a ferramenta ao banco e gerar os diagramas.

4.4.7 Projeto e Desenvolvimento do ETL

A atividade final na trilha de dados é o projeto e o desenvolvimento do sistema da *data staging área* ou ETL. O ETL toma os dados brutos dos sistemas transacionais (OLTP) e os prepara para o modelo dimensional na área de apresentação de dados. É um serviço de *backroom*.

Muitas equipes se concentram somente no E e no L do acrônimo ETL. Esquecendo-se que: a combinação dados, questões de qualidade, identificação e atualização dos dados, a gerência de chaves substitutas, as agregações e o tratamento de erros, grande parte ocorre na etapa de transformação(T).

O processo de ETL deve ser distinto para as tabelas de Fato e Dimensões, visto que os dados são tratados de forma diferenciada, devida justamente as suas características diferenciadas (KIMBALL, 2002).

O ETL das tabelas de dimensões envolve as seguintes etapas:

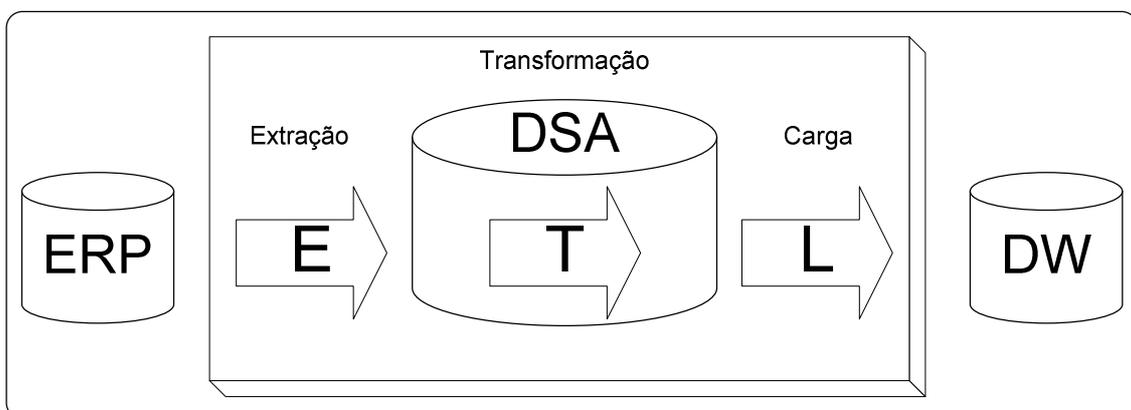
- Extração de dados dimensionais de sistemas operacionais de origem.
- Limpeza de valores de atributo.
- Gerenciamento de atribuições de chaves substitutas.

O ETL das tabelas de Fato envolve as seguintes etapas:

- Extrair os dados de fatos do sistema operacional de origem.
- Receber dimensões atualizadas das autoridades de dimensão.
- Separar os dados de fatos por granularidade conforme necessário.
- Transformar os dados de fatos conforme necessário.
- Trocar as chaves operacionais de origem por chaves substitutas.
- Adicionar chaves para contexto conhecido.
- Garantir a qualidade dos dados da tabela de fatos.
- Construir ou atualizar tabelas de fatos de agregação.
- Fazer uma carga em massa dos dados.
- Alertar os usuários.

Para o projeto em questão foi desenvolvido todo processo de ETL através de *Scripts SQL* (Transact SQL) que podem ser visualizados no apêndice A, seguindo a ordem demonstrada na figura 30. Onde os dados são extraídos do ERP *Advanced Protheus* da Microsiga, transformados na *data staging área* e finalmente carregados no DW.

Fig. 30: Processo de ETL



Fonte do Autor

4.4.8 Especificação da Aplicação Analítica

Neste ponto finalmente usa-se todo o investimento em tecnologia e dados para ajudar os usuários a tomarem decisões mais acertadas. As aplicações fornecem um mecanismo chave para reforçar o relacionamento entre a equipe de projeto e a comunidade de negócio. Elas servem para apresentar o rosto do *Data Warehouse* para seus usuários e elas trazem as necessidades de negócio de volta para a equipe de desenvolvedores de aplicação.

Logo após a definição dos requisitos, deve-se rever as constatações e os relatórios coletados a fim de identificar um conjunto inicial de aplicações analíticas.

Antes do início das primeiras aplicações, é necessário estabelecer padrões para as mesmas. Com base nos padrões, especifica-se cada modelo de aplicação, capturando informações sobre *layout*, variáveis de entrada, cálculos e pausas para que tanto o desenvolvedor de aplicações como os representantes, compartilhem um entendimento comum (KIMBALL, 2002).

Com base nos requisitos de negócios levantados (ver tabela 6) foram elaboradas as seguintes análises que podem ser acessadas pelo decisor:

Tabela 8: Relação de Análise

Produção	<p>Nesta análise o decisor tem uma visão geral de toda a produção da empresa em Kg (Kilogramas) por período e setor, relacionados com o tempo de produção, tempo de produção parada e perda de produção.</p> <p>Utilizando as operações OLAP, pode-se descer ao grão de ordem de produção. Onde se pode passar por todas as dimensões possíveis, a fim de fazer as consultas e comparações relacionadas nos requisitos de negócio.</p>
Tempo de Parada	<p>Nesta análise o decisor tem uma visão geral do tempo de parada da produção em minutos, por período e setor.</p> <p>Utilizando as operações OLAP, pode-se descer ao grão de ordem de produção. Onde se pode passar por todas as dimensões possíveis, a fim de fazer as consultas e comparações relacionadas nos requisitos de negócio.</p>
Perda de Produção	<p>Nesta análise o decisor tem uma visão geral de toda a perda de produção da empresa em Kg (Kilogramas) por período e setor.</p> <p>Utilizando as operações OLAP, pode-se descer ao grão de ordem de produção. Onde se pode passar por todas as dimensões possíveis, a fim de fazer as consultas e comparações relacionadas nos requisitos de negocio.</p>

Fonte do Autor

4.4.9 Desenvolvimento de Aplicações Analíticas

Segundo Kimball (2002}, nesta fase deve-se concentrar nos padrões como:

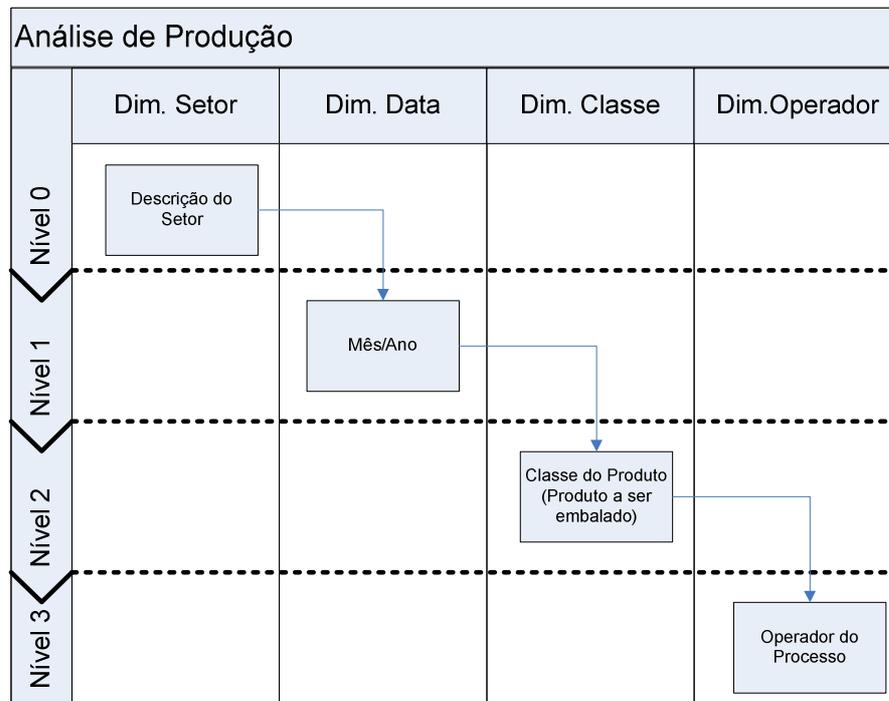
- Padrões para convenção de nomeação.
- Padrões de cálculos.
- Padrões de bibliotecas
- Padrões de codificação

Todos os padrões devem ser bem definidos, de conhecimento de todos para minimizar retrabalho futuro.

No projeto em questão, foi utilizada a ferramenta de *Data Warehouse* que acompanha o sistema de ERP da Microsiga, para visualizar as análises do *DW*. O motivo de se optar por essa ferramenta foi o fato de ela realizar a maioria das operações de OLAP, tanto em forma de tabela (Cubo), como de gráfico, funcionar na *WEB* hoje, fato primordial e o fator custo por já acompanhar o ERP da empresa.

Abaixo segue alguns exemplos de análises sugeridas:

Fig. 31: Esquema de *Drill Down* da Análise de Produção



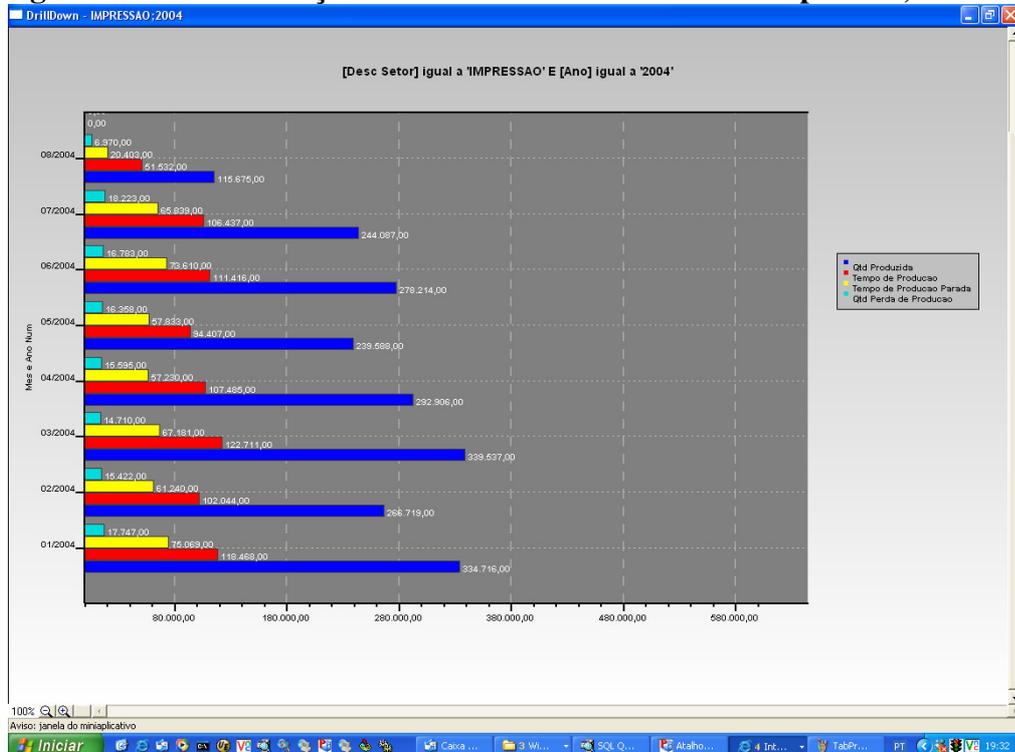
Fonte do Autor

Fig. 32: Cubo Produção Realizada

Ano	Mes e Ano Num	Desc Setor	Qtd Produzida [s]	Tempo de Producao [s]	Qtd Perda de Producao [s]	Tempo de Producao Parada [s]
2004	05/2004	LABORATORIO	18,00	444,00		9,34
		LAMINACAO			4,34	
		REBOBINADEIRA	82.404,14	42.505,00	33,10	11.125,00
		VALVULADO	32.243,88	12.210,00	10,23	
		TOTAL	1.594.448,87	714.288,00	446,11	156.906,00
06/2004	06/2004	CORTE E SOLDA	730.603,77	186.900,00		73,97
		EXTRUSAO	541.717,00	308.709,00	169,52	73.759,00
		IMPRESSAO	278.214,00	111.416,00	167,83	73.610,00
		LABORATORIO			4,65	
		LAMINACAO	9,00	135,00		37
REBOBINADEIRA	305.091,39	36.130,00	26,30	12.899,00		
VALVULADO	32.837,01	13.765,00	7,76			
TOTAL	1.888.472,17	657.055,00	450,40	199.585,00		
07/2004	07/2004	CORTE E SOLDA	196.683,39	170.495,00		44,625,00
		EXTRUSAO	556.192,00	326.673,00	207,72	125.286,00
		IMPRESSAO	244.087,00	106.437,00	182,23	65.839,00
		LABORATORIO			3,61	
		LAMINACAO	3.439,00	698,00		1,72
REBOBINADEIRA	484.935,44	37.570,00	15,40	12.438,00		
VALVULADO	47.665,62	19.340,00	6,36			
TOTAL	1.473.002,45	661.413,00	490,25	248.198,00		
08/2004	08/2004	CORTE E SOLDA	82.829,73	75.520,00		31,88
		EXTRUSAO	263.012,00	154.827,00	88,98	47.274,00
		IMPRESSAO	115.675,00	51.532,00	69,70	20.403,00
		LABORATORIO			4,52	
		LAMINACAO	7.844,00	4.730,00		52
REBOBINADEIRA	48.498,97	20.045,00	10,51	5.697,00		
VALVULADO	18.527,52	5.580,00	2,88			
TOTAL	536.387,22	312.234,00	208,99	89.721,00		
2004			11.541.478,43	5.127.314,00	3.330,35	1.353.090,00

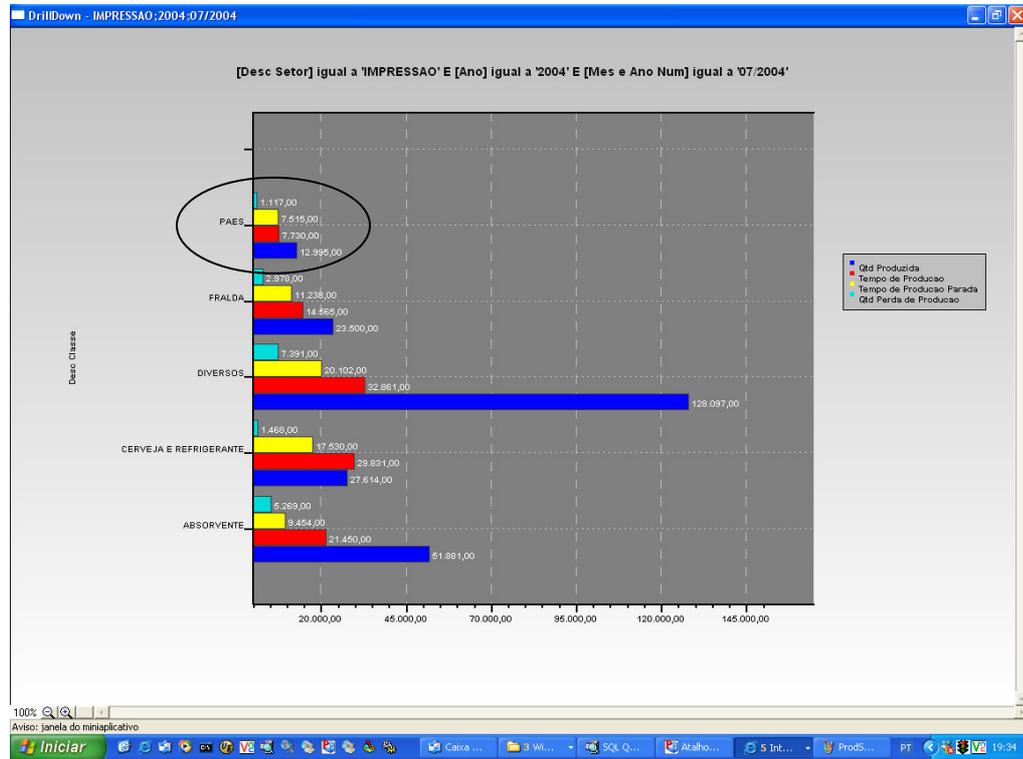
Fonte do Autor

Fig. 33: Gráfico Produção Realizada – Nível 0 Setor de Impressão, Período



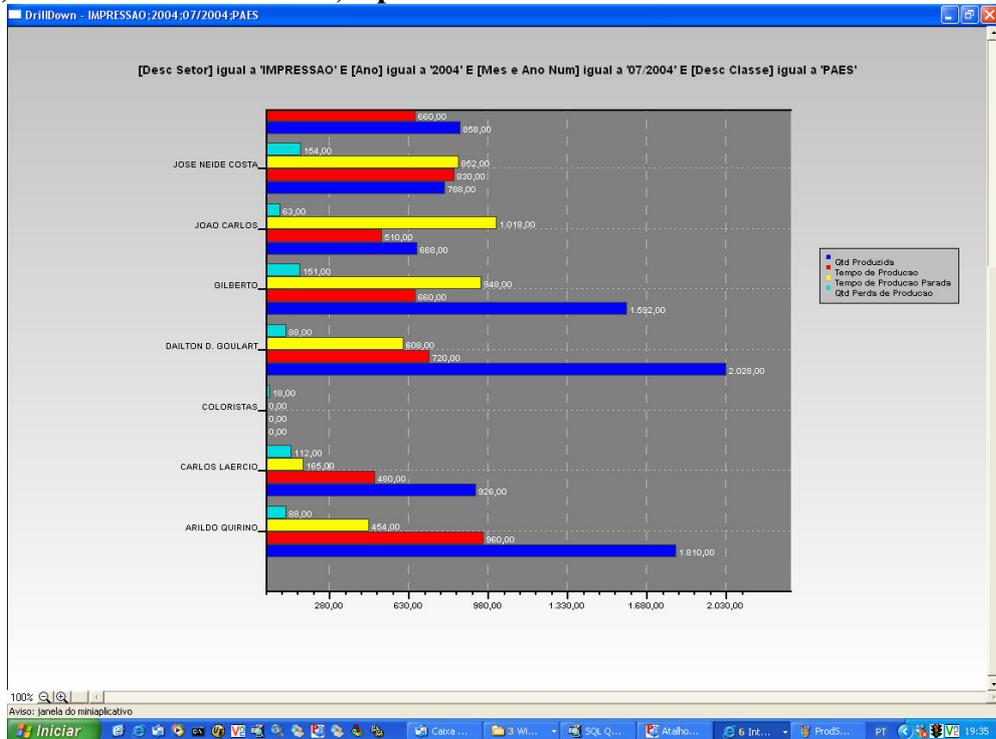
Fonte do Autor

Fig 34: Gráfico Produção Realizada – Nível 1 Setor de Impressão, Período 07/2004, Classe de Produto



Fonte do Autor

Fig. 35 Gráfico Produção Realizada – Nível 2 Setor de Impressão, Período 07/2004, Classe de Produto Pães, Operador

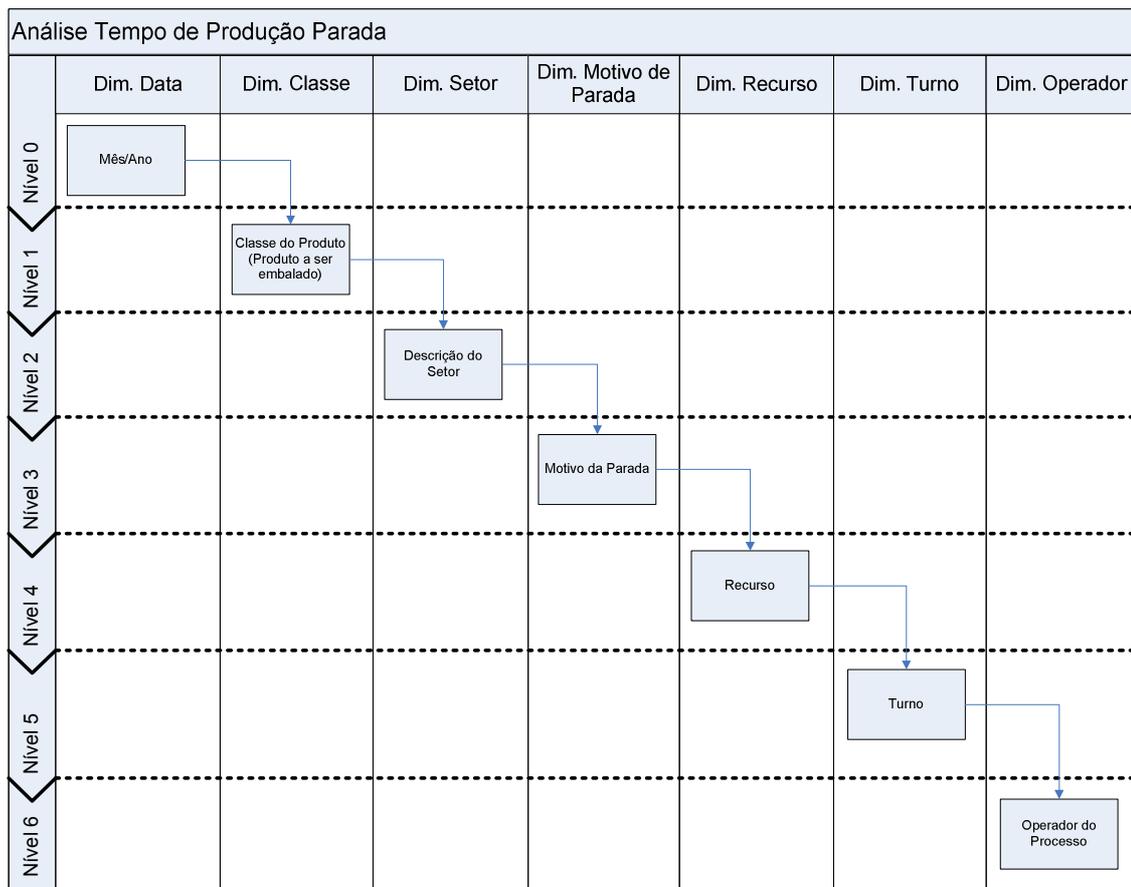


Fonte do Autor

Nesta seqüência de análises o decisor pode comparar como foi a produção mês a mês, comparando o tempo de produção, tempo de parada da máquina e perda.

No *Drill Down* de nível 1, pode-se notar que o tempo de parada da classe de produtos Pães no período 07/2004 é praticamente igual ao tempo de produção. O que automaticamente sugere uma análise mais detalhada no tempo de parada, no segmento em questão.

Fig. 36: Esquema de *Drill Down* da Análise de Tempo de Produção Parada



Fonte do Autor

Fig. 37: Cubo Tempo de Produção Parada

Protheus - Datawarehouse - Microsoft Internet Explorer

DW: vendas Usúrios Logout Ajuda Administrador SigaDW

Efetuar logoff de Administrador SigaDW

Navegação

Resultado da consulta [Tempo de Parada]

Ano	Mes e Ano Num	Desc Setor	Qtd Tempo de Parada[s]
2004	03/2004	CORTE E SOLDA	42.349,00
		EXTRUSAO	31.946,00
		IMPRESSAO	70.685,00
		REBOBINADEIRA	9.538,00
	03/2004		154.518,00
04/2004	04/2004	CORTE E SOLDA	41.903,00
		EXTRUSAO	38.885,00
		IMPRESSAO	58.572,00
		REBOBINADEIRA	10.110,00
	04/2004		149.470,00
05/2004	05/2004	CORTE E SOLDA	39.085,00
		EXTRUSAO	51.202,00
		IMPRESSAO	60.061,00
		REBOBINADEIRA	11.215,00
	05/2004		161.563,00
06/2004	06/2004	CORTE E SOLDA	39.467,00
		EXTRUSAO	74.797,00
		IMPRESSAO	74.444,00
		REBOBINADEIRA	12.899,00
	06/2004		201.607,00
07/2004	07/2004	CORTE E SOLDA	44.795,00
		EXTRUSAO	128.030,00
		IMPRESSAO	66.359,00
		REBOBINADEIRA	12.496,00
	07/2004		251.682,00
08/2004	08/2004	CORTE E SOLDA	16.637,00
		EXTRUSAO	50.416,00
		IMPRESSAO	21.477,00
		REBOBINADEIRA	6.297,00
	08/2004		94.827,00

© Microsiga Intelligence

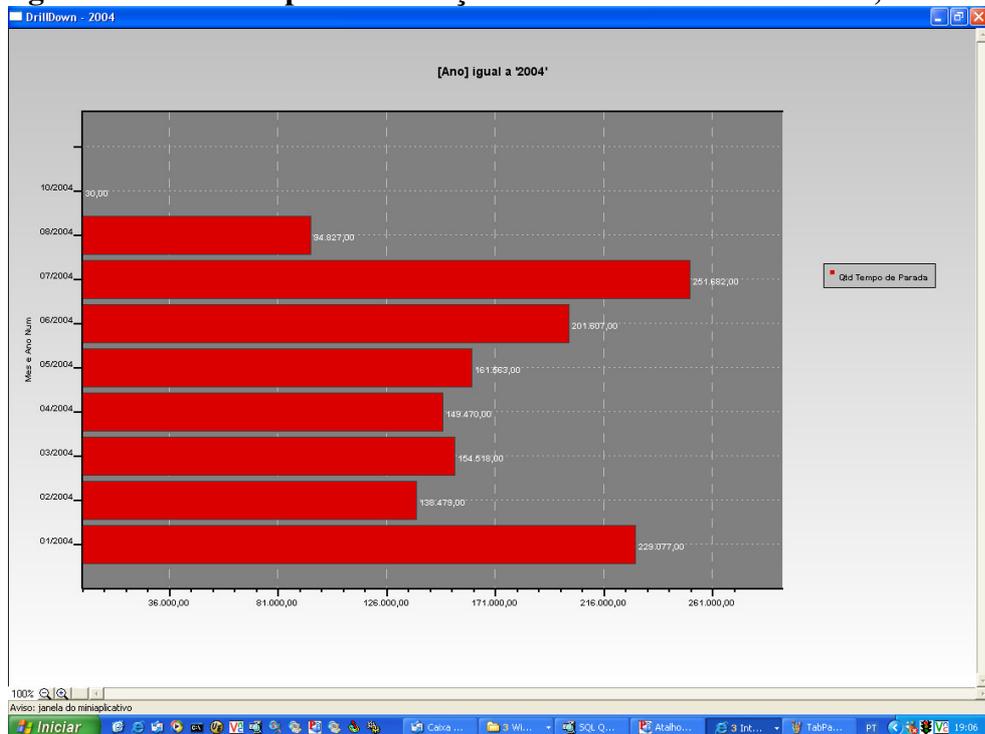
SigaDW 2.01.040407

Intranet local

19:04

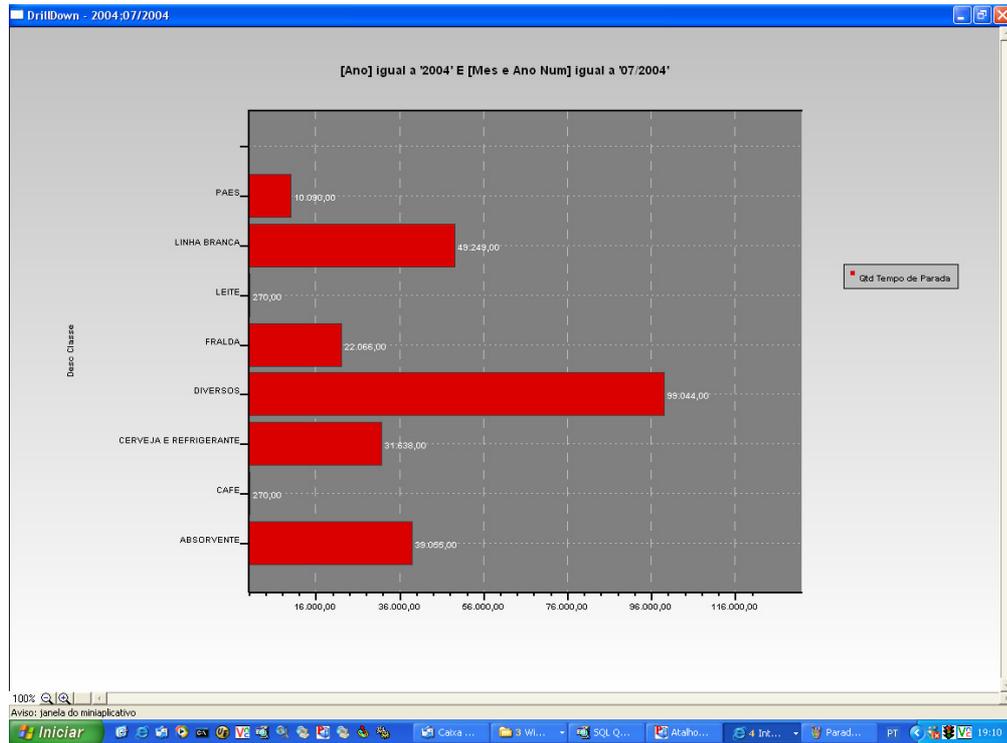
Fonte do Autor

Fig. 38. Gráfico Tempo de Produção Parada – Nível 0 Ano 2004, Período



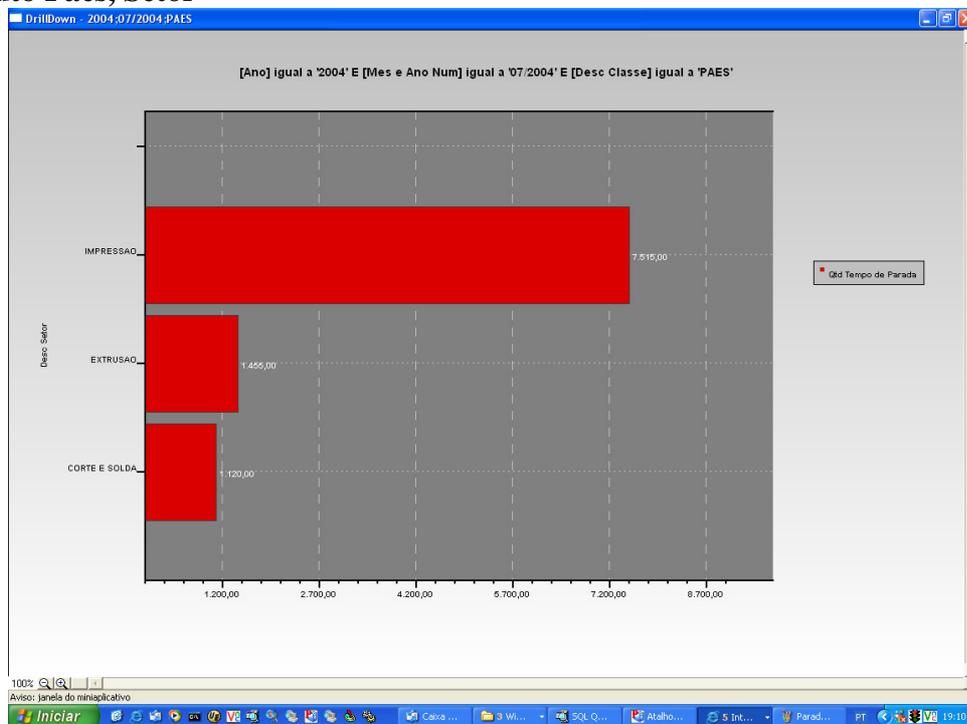
Fonte do Autor

Fig. 39: Gráfico Tempo de Produção Parada – Nível 1 Ano 2004, Período 07/2004, Classe de Produto



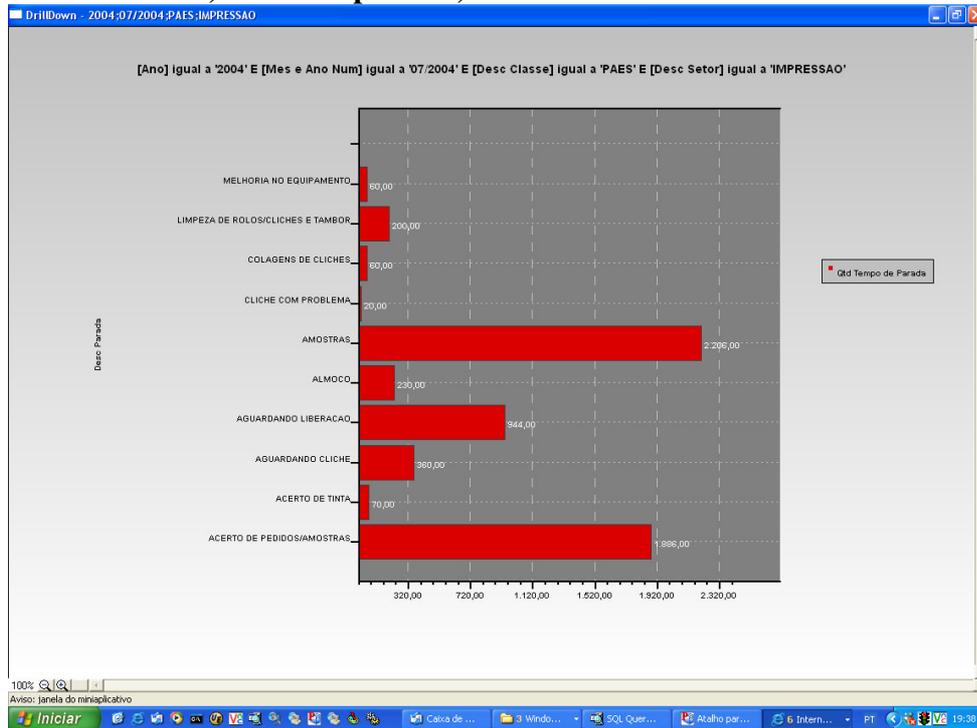
Fonte do Autor

Fig.40 Gráfico Tempo de Produção Parada – Nível 2 Ano 2004, Período 07/2004, Classe de Produto Pães, Setor



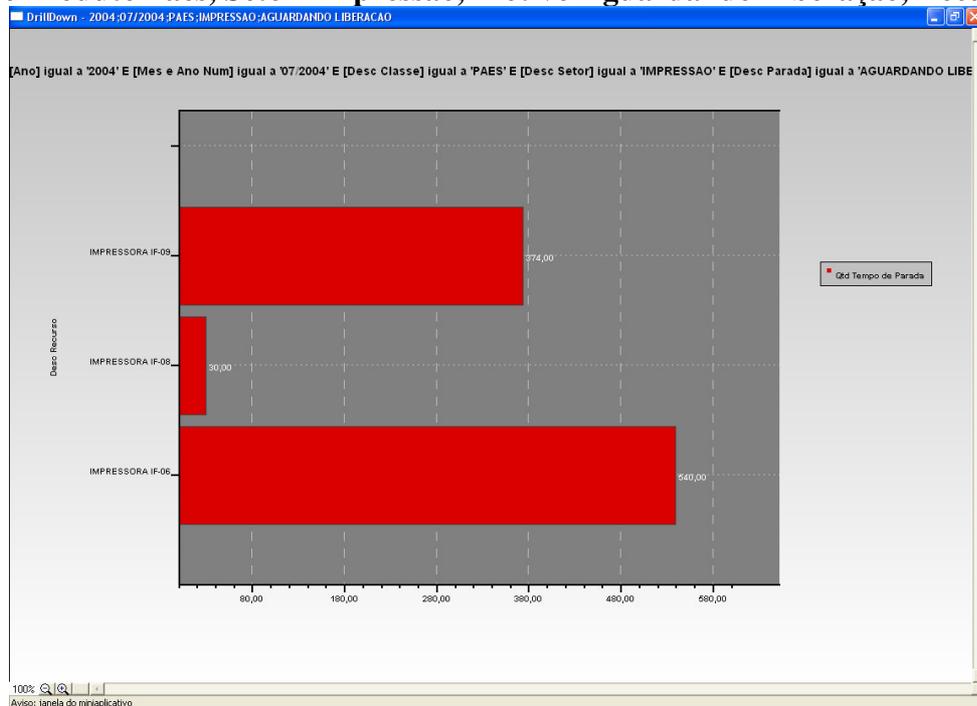
Fonte do Autor

Fig. 41: Gráfico Tempo de Produção Parada – Nível 3 Ano 2004, Período 07/2004, Classe de Produto Pães, Setor Impressão, Motivo



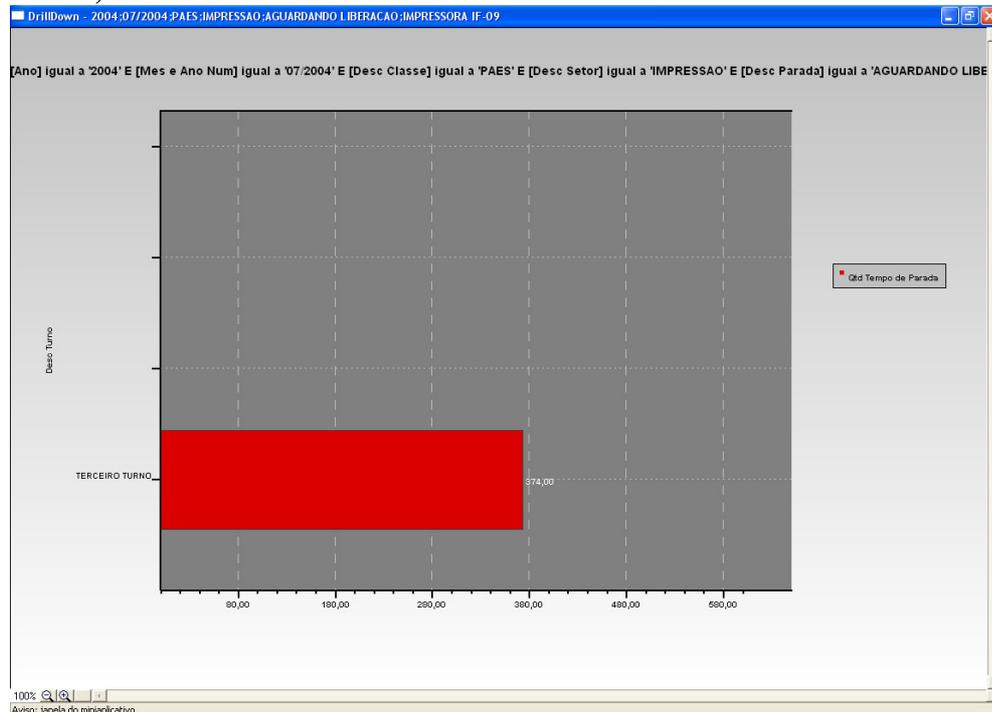
Fonte do Autor

Fig. 42: Gráfico Tempo de Produção Parada – Nível 4 Ano 2004, Período 07/2004, Classe de Produto Pães, Setor Impressão, Motivo Aguardando Liberação, Recurso



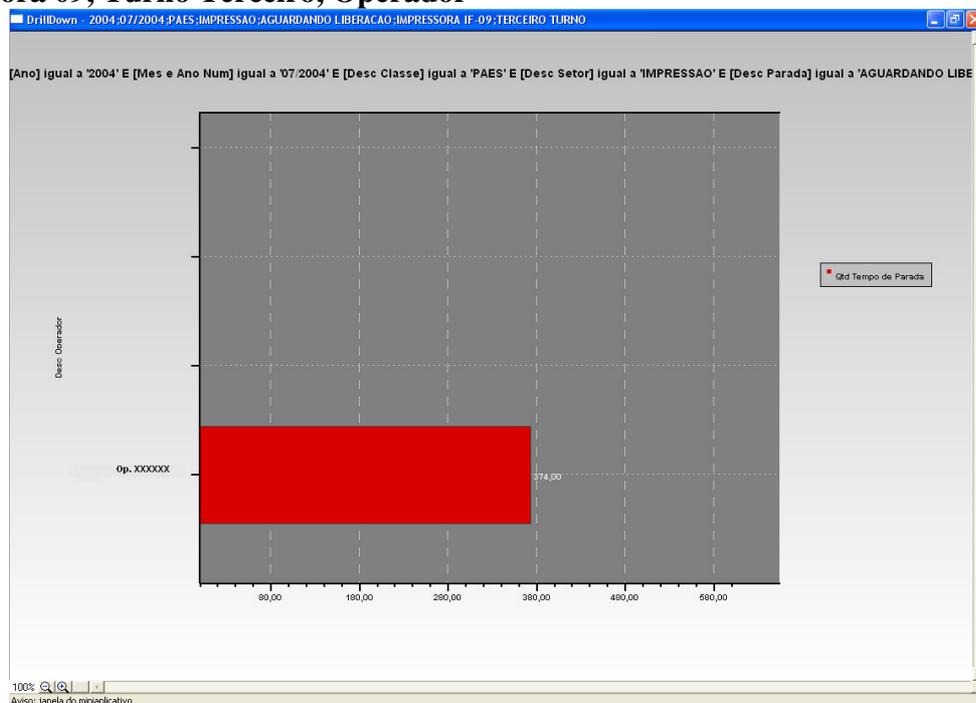
Fonte do Autor

Fig. 43: Gráfico Tempo de Produção Parada – Nível 5 Ano 2004, Período 07/2004, Classe de Produto Pães, Setor Impressão, Motivo Aguardando Liberação, Recurso Impressora 09, Turno



Fonte do Autor

Fig. 44: Gráfico Tempo de Produção Parada – Nível 6 Ano 2004, Período 07/2004, Classe de Produto Pães, Setor Impressão, Motivo Aguardando Liberação, Recurso Impressora 09, Turno Terceiro, Operador



Fonte do Autor

Nesta seqüência de análise, o decisor pode analisar com detalhes o tempo de produção parada da fábrica, identificando os motivos, recursos, produtos, turnos. Podendo navegar em todas as dimensões possíveis para esse fato.

No caso visto acima, pode-se analisar com mais detalhes o que tinha chamado a atenção no nível 1 da análise de produção (ver figura 35), podendo-se identificar uma anormalidade no turno 3, na classe de produtos Pães.

4.10 Implantação

As trilhas da tecnologia, dos dados e das aplicações analíticas convergem na implantação.

Para uma implantação bem sucedida deve-se avaliar criticamente a preparação do *Data Warehouse*, um pacote de treinamento e suporte para implantação.

Para o programa de treinamento deve-se considerar:

- O Entendimento do público-alvo; não o subjugando.
- Não treinar a comunidade de negócio antes da disponibilidade dos dados e das aplicações analíticas.
- Adiar o treinamento (e implantação) se o *Data Warehouse* não estiver pronto para ser lançado.
- Conseguir o comprometimento do patrocinador no sentido de uma política “sem treinamento, sem acesso”.

O suporte é muitas vezes organizado em uma estrutura de duas camadas: A primeira linha de experiência se encontra na área de negócio, ao passo que o suporte centralizado fornece uma linha secundária de defesa.

Como em qualquer teste de sistema, enfrentar problemas é perfeitamente normal, então se deve certificar de que há tempo suficiente na programação para inevitável retrabalho.

No projeto em questão, antes dos treinamentos e de disponibilizar o ambiente de consulta e análise, foram feitas, todas as devidas avaliações com os decisores, para só então se efetuar os respectivos treinamentos e posterior liberação do ambiente de consulta e análise aos usuários de *DW*.

4.11 Manutenção e Crescimento

Uma vez implantado o projeto, é preciso continuar a investir recursos nas seguintes áreas:

- Suporte;
- Treinamento;
- Suporte técnico;
- Suporte ao programa;

Se todos os envolvidos no projeto tiverem feito o trabalho de forma correta. Inevitavelmente, haverá demanda de crescimento para novos usuários, novos dados novas aplicações ou aprimoramentos maiores a produtos existentes.

No caso, do projeto em questão, a manutenção e crescimento estão sendo estudados e efetuados de acordo com a necessidade e o ritmo da empresa.

4.5 Conclusão

Neste capítulo pode-se compreender o ciclo de vida dimensional de negócio de Kimball (2002) e validar o mesmo para uma indústria de embalagens flexíveis.

Fica claro também que o ciclo de vida dimensional de negócio, como o próprio nome já o identifica, é um processo cíclico e não linear, ou seja, deve estar em constante revisão e manutenção.

No projeto em questão, teve-se uma preocupação muito grande em seguir os passos propostos por Kimball (2002), a fim de evitar retrabalho e ter uma avaliação correta quanto à corroboração do processo, em relação a uma indústria de embalagens flexíveis.

CONCLUSÃO

5.1 Conclusões

Nos primórdios dos sistemas de informação, houve uma preocupação muito grande com a automatização do operacional, não se preocupando com a integração dos setores. Num segundo momento, houve a necessidade de se integrar esses setores e com a popularização dos SGDBs (Sistemas Gerenciadores de Banco de Dados), começou-se também a criar uma base histórica.

Com o avanço dos computadores, a popularização dos bancos de dados, a queda considerável dos custos com tecnologia e o aumento da concorrência, devido à globalização, às empresas começaram a investir nos níveis superiores das camadas hierárquicas da pirâmide organizacional.

Neste novo cenário mundial, o *Data Warehouse* está se mostrando uma tecnologia de diferencial competitivo. Auxiliando os decisores a terem uma visão ampla e de forma rápida de toda a empresa. Segundo Haberkorn (2003), chega-se ao ápice da tecnologia da informação, mas com um, porém. A informática pode a partir de seus sistemas de informações, fornecer aos executivos todas as informações que eles necessitam, menos uma: a própria decisão.

Com o estudo de caso, conseguiu-se mostrar que é perfeitamente possível e viável a implantação de um *Data Warehouse*, segundo o ciclo de vida dimensional de negócio de Kimball, no setor produtivo de uma indústria de embalagens flexíveis.

Na análise de produção o decisor tem uma visão macro de todo o processo produtivo, comparando a quantidade produzida com a perda e o tempo de produção com o tempo de produção parada. Essa análise mostrou-se de extrema importância, pois sintetiza todo o processo produtivo, evidenciando a normalidade e mostrando as anomalias, como o tempo de parada e tempo de produção da classe Pães, demonstrada no capítulo 4.

Já as análises, Tempo de Produção Parada e Perda de Produção, complementam a análise de produção, onde o decisor pode verificar com mais detalhes qualquer anomalia que venha a se evidenciar na análise de produção.

Todas as análises mencionadas passaram pela aprovação do diretor de produção da empresa, ao qual legitimou a eficácia e eficiência das mesmas em responder as questões levantadas, na análise de requisitos.

No decorrer do trabalho notou-se que quanto mais centros de trabalhos (diferentes processos que o produto passa até chegar ao seu final) existem no setor de produção, mais complexos fica o controle e conseqüentemente mais complexa fica a análise da produção da empresa. No caso, de uma indústria de plásticos flexíveis, podem existir até sete processos de produção. O *DW* mostrou-se uma ferramenta essencial, complementando o sistema OLPT, trazendo agilidade e segurança nas decisões, tanto no nível gerencial, como no nível tático e estratégico.

No nível gerencial os fatos trabalhados podem como foi demonstrado no estudo feito no capítulo 4, estar auxiliando o gerente de produção a evidenciar anormalidades no processo em curto prazo, onde com a anormalidade identificada caberá ao gerente decidir a melhor atitude a tomar.

No nível tático e estratégico, além do decisor (que aqui podemos identificar como o diretor de produção) ter uma visão do que ocorre no setor produtivo (podem ter a mesma

visão do gerente) terá também uma visão macro de todos os processos da fábrica a curto e em longo prazo. Em curto prazo o decisor terá uma visão geral de todo o processo produtivo da fábrica e não de setores isolados, como no caso dos gerentes. E em longo prazo terá uma visão geral de mercado, podendo facilmente, através da base histórica, traçar um perfil dos diversos segmentos do mercado de embalagens flexíveis levando em conta a época do ano, o tipo de produto e a situação econômica.

Não existem limites no que o *Data Warehouse* pode fazer para auxiliar no processo decisório de uma indústria de embalagens flexíveis, o que existem são fatos mais ou menos complexos. Mas que devem ser implementados gradualmente com a necessidade dos decisores da empresa.

O que este estudo de caso levantou e implementou com sucesso foi um exemplo de que essa tecnologia esta preenchendo um espaço vago na pirâmide hierárquica da informação das empresas e que é comprovadamente um diferencial competitivo pela facilidade e simplicidade com que trata a informação.

5.2 Trabalhos Futuros

Tudo o que poderia ser feito e não coube neste trabalho sugeri-se como trabalho futuro. Mas, é importante salientar algumas sugestões levantadas pelo diretor de produção da empresa.

Uma delas, diz respeito à previsão de perdas e tempo de produção parada. Onde deverá ser alimentado o sistema com as devidas previsões, para serem vinculadas a análise de produção.

Outra sugestão é incorporar uma análise de produção faturada, juntamente com a meta orçamentária de faturamento periódico, a fim de comparar se o que está sendo produzido está

sendo faturado e se a produção está suprimindo a necessidade do departamento comercial da empresa.

Referências

BOGHI, Cláudio. **Sistemas de informação: um enfoque dinâmico**. São Paulo: Erica, 2002.

CARVALHO, André **O sistema de informação em saúde para municípios**. Acessado em 08/06/2003; disponível em: http://ids-saude.uol.com.br/SaudeCidadania/ed_06/02.html.

CHIAVENATO, Idalberto. **Administração: teoria, processo e prática**. 3. ed. São Paulo: Makron Books, 2000.

CIELO, Ivã. **Construindo um sistema de apoio à decisão - 02/01/2000**. Acessado em 08/06/2003; disponível em: <http://www.datawarehouse.inf.br/artigos/sad.asp>

----- **EIS - Executive Information System**. Acessado em 08/06/2003; disponível em: <http://www.datawarehouse.inf.br/artigos/eis.asp>.

----- **ETL - Extração, alterar e carga de dados**. Acessado em 08/06/2003; disponível em: <http://www.datawarehouse.inf.br/artigos/etl.asp>

----- **Um pouco sobre OLAP - 23/01/2000**. Acessado em 08/06/2003; disponível em: <http://www.datawarehouse.inf.br/artigos/olap.asp>

COLANGELO Filho, Lucio. **Implantação de sistemas ERP (Enterprise Resources Planing)**: um enfoque de longo prazo. São Paulo: Atlas, 2001.

DATAMARK Ltda. Acessado em 25/07/2004; disponível em : http://www.datamark.com.br/newdatamark/home_p.asp

Decision Warehouse, 2004a. *Turning Data Into Results - O que é Business Intelligence?* Acessado em 15/03/2004; disponível em: <http://www.decisionwarehouse.com.br/institucional/business.htm>

-----, 2004b - *Turning Data Into Results - BI no planejamento estratégico*. Acessado em 15/03/2004; disponível em: <http://www.decisionwarehouse.com.br/institucional/business.htm>

DWBRASIL,2003a. *Business Intelligence*. Acessado em 09/06/2003; disponível em: <http://www.dwbrasil.com.br/html/bi.html>

-----, 2003b. *Características do Data Warehouse*. Acessado em 09/06/2003; disponível em: http://www.dwbrasil.com.br/html/artdw_carac.html

_____, 2003c. *Data Warehouse:* Acessado em 09/06/2003; disponível em: <http://www.dwbrasil.com.br/html/dw.html>

-----, 2003d. Histórico do *Data Warehouse*. Acessado em 09/06/2003; disponível em: http://www.dwbrasil.com.br/html/artdw_hist.html

-----, 2003e Metadados. Acessado em 09/06/2003; disponível em: <http://www.dwbrasil.com.br/html/metadata.html>

-----, 2003f. OLAP. Acessado em 09/06/2003; disponível em: <http://www.dwbrasil.com.br/html/olap.html>

-----,2003g. Stage (ETL). Acessado em 09/06/2003; disponível em: <http://www.datawarehouse.inf.br/artigos/etl.asp>

-----,2004. Dividir para conquistar ou conquistar para dividir? Acessado em 03/05/2004; disponível em: http://www.dwbrasil.com.br/html/artdw_20030620.html

ERP.2003. Sistemas ERP: diferencial competitivo ou apenas mais um sistema de informação. Acessado em 09/06/2003; disponível em: http://www.uol.com.br/cultvox/novos_artigos/sistemaserp.htm

FEDICHINA,Márcio Antonio H.; GOZZI, Sérgio. A implementação de Sistemas Integrados ERP: Uma análise dos fatores críticos de sucesso. Acessado em 29/04/2004; disponível em: http://www.fgvsp.br/iberoamerican/Papers/0450_Artigo%20Congresso%20Ibero%20America no%202003%20-%20Marcio%20e%20Sergio.pdf

GONZALES, Pablo M.; ZEPKA, Marilene M. Portal de Embalagens. Rio Grande: FURG. Acessado em 25/07/2004; disponível em : <http://www.furg.br/furg/projet/embalagens/index.html>

GREENFIELD, Larry. *A Definition of Decision Support*. Acessado em 10/02/2004; disponível em : <http://www.dwinfocenter.org/decsup.html>

----- *What Decision Support Tools are Used For*. Acessado em 10/02/2004; disponível em : <http://www.dwinfocenter.org/decsup.html>

HABERKORN,Ernesto. **Teoria do ERP, *Enterprise Resource Planning***. São Paulo: Makron Books, 1999.

----- **Gestão empresarial com ERP** . São Paulo, 2003.

HERMINIA, Vânia Gestão de Sistema de Informação II - DATA WAREHOUSE; Acessada em 08/06/2003; disponível em: <http://www.ispu.ac.mz/formacao/trabalhos/dw2.doc>

HYPOLITO, Christiane Mendes, PAMPLONA, Edson de Oliveira. Sistemas de Gestão Integrada:Conceitos e Considerações em uma Implantação. Acessado em 22/06/2003; disponível em: <http://www.iem.efei.br/edson/download/Arterp.doc>

INMON, W. H.. *ERP and Data Warehouse: Reading the tea leaves* Acessado em 05/01/2004; disponível em: <http://www.inmoncif.com//library/articles/arterpfu.asp>

----- OLTP versus DSS. Acessado em 05/01/2004; disponível em:
<http://www.inmoncif.com/library/articles/dwvirt.asp>

----- *Virtual Data Warehouse: The Snake Oil Of The Nineties*. Acessado em
 05/01/2004; disponível em: <http://www.inmoncif.com/library/articles/oltpdss.asp>

-----*Definition of a Data Warehouse*. Acessado em 05/01/2004; disponível em:
<http://www.inmoncif.com/library/articles/dwdef.asp>

KEEN,P. G. W.; MORTON M. S. S. *Decision Support System: An Organizational
 Perspective*. Addison. USA.:Wesley Publishing Company, 1978.

KIMBALL, Ralph. *Fact Tables and Dimension Tables* - 01/01/2003. Acessado em
 18/11/2003; disponível em:
http://www.intelligententerprise.com/030101/602warehouse1_1.shtml/data_warehousing%7cware

----- *The Matrix*-07/12/1999. Acessado em 18/11/2003; disponível em:
http://www.intelligententerprise.com/db_area/archives/1999/990712/webhouse.jhtml?_requestid=14921

----- *The Soul of the Data Warehouse, Part One: Drilling Down* -20/03/2003.
 Acessado em 18/11/2003; disponível em:
http://www.intelligententerprise.com/030320/605warehouse1_1.jhtml?_requestid=14131

----- *The Soul of the Data Warehouse, Part Two: Drilling Across* -05/04/2003.
 Acessado em 18/11/2003; disponível em:
http://www.intelligententerprise.com/030405/606warehouse1_1.jhtml?_requestid=14298

----- *The Soul of the Data Warehouse, Part 3: Handling Time* -22/04/2003.
 Acessado em 18/11/2003; disponível em:
http://www.intelligententerprise.com/030422/607warehouse1_1.jhtml?_requestid=14600

----- traduzido e adaptado por Luiz Cláudio Paz; Como levar seu DW ao
 fracasso. Acessado em 08/06/2003; disponível em:
<http://www.datawarehouse.inf.br/artigos/fracasso.asp>

----- *Data Warehouse Toolkit*; Guia completo para modelagem dimensional;
 Tradução de Ana Beatriz Tavares,Daniela Lacerda. - Rio de Janeiro: Campus, 2002.

LAUDON, Kenneth C, Laudon, Jane P. Gerenciamento de sistemas de informação. 3. ed. Rio
 de Janeiro: LTC, 2001.

----- Sistemas de informação. 4. ed. Rio de Janeiro: LTC, 1999..

MACHADO, Felipe N. R. Projeto de *Data Warehouse*: Uma Visão Multidimensional - São
 Paulo, Érica, 2000.

NAZÁRIO, Paulo. A Importância de sistemas de informação para a Competitividade Logística; acessado em 09/06/2003; disponível em: <http://www.cel.coppead.ufrj.br/fs-busca.htm?fr-sist-info.htm>

OLIVEIRA, Djalma de P. R. Sistemas de informações gerenciais. 5.ed. São Paulo: Atlas, 1998.

----- Sistemas, organizações e métodos: Uma abordagem gerencial. 2. ed . – São Paulo, Atlas, 1988

OLIVEIRA, Jaur F. Sistemas de informação: um enfoque gerencial inserido no contexto empresarial e tecnológico. São Paulo: Érica, 2000.

PERENTE, Daniel. *Data Warehouse*: identificando os possíveis pontos que podem levá-lo ao fracasso. Acessado em 09/06/2003; disponível em : <http://www.datawarehouse.inf.br/artigos/identfracasso.asp>

PLASC, Plásticos Santa Catarina Ltda; disponível em : www.plasc.com.br

POWER,D. J. *A Brief History of Decision Support Systems*. Acessado em 05/04/2004; disponível em: <http://www.staff.livjm.ac.uk/bsnmyoll/dsshistory.html>

REZENDE, Denis Alcides. **Tecnologia da informação aplicada a sistemas de informação nas empresas**. 3.ed São Paulo: Atlas, 2003.

----- **Engenharia de software e sistemas de informação**. Rio de Janeiro: Brasport, 1999.

SANTOS, Cleison. Modelagem de dados com ErWin. **SQL Magazine**. Edição 1, Ano 1; Neofício Editora.

SELL, Denílson; **Uma arquitetura para distribuição de componentes tecnológicos de sistemas de informações baseados em Data Warehouse**. Florianópolis: UFSC, 2000.

STAR, Ralf M; REYNOLDS, George W. **Princípios de sistemas de informação: uma abordagem gerencial**. 4. ed. Rio de Janeiro: LTC, 2002.

----- **Princípios de sistemas de informação: uma abordagem gerencial**. 2. ed. Rio de Janeiro: LTC, 1998.

THOMSEN, Erik. **OLAP**: construindo sistemas de informações multidimensionais; tradução de Daniel Vieira. Rio de Janeiro: Campus, 2002.

TINFO NEWS.- Tecnologia & Tendências; ERP - Entendendo seu significado. Acessado em: 15/05/2004; disponível em: <http://www.tinfo.com.br/artigos.asp?newsid=8>

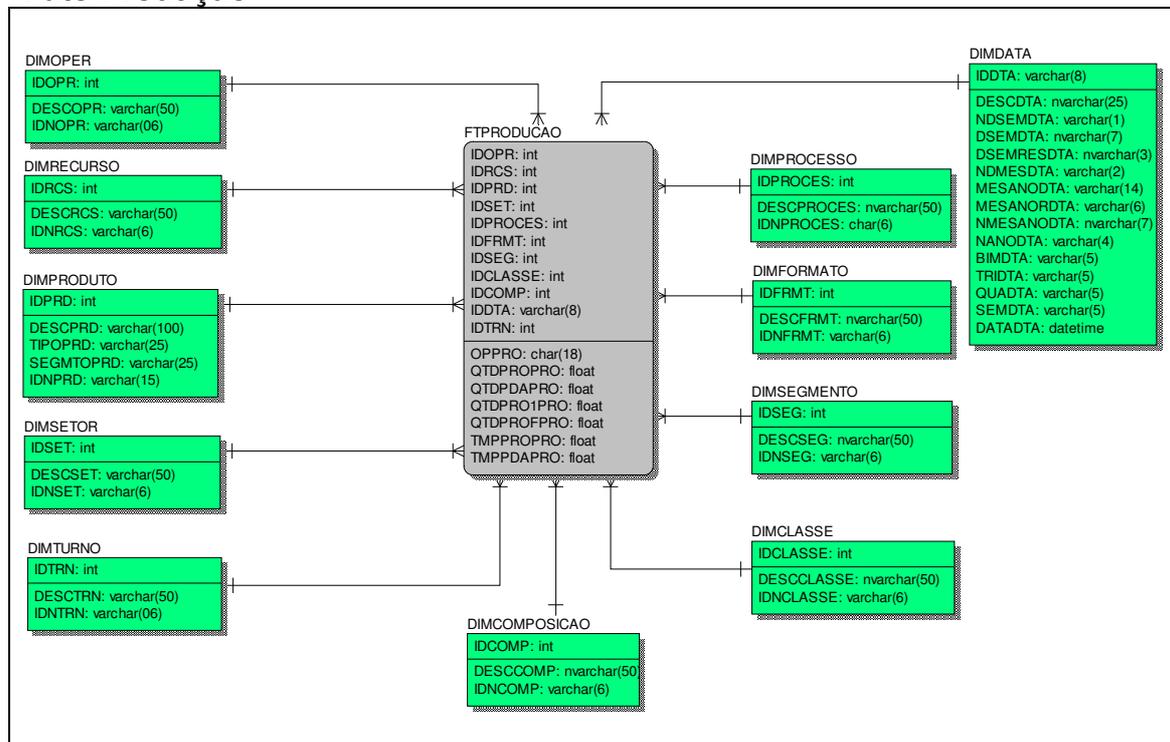
VIEIRA, Fernando. Alguns conceitos sobre DW. Acessado em 09/06/2003; disponível em: <http://www.datawarehouse.inf.br/artigos/conceitosdw.asp>

Apêndice A – Modelo Dimensional

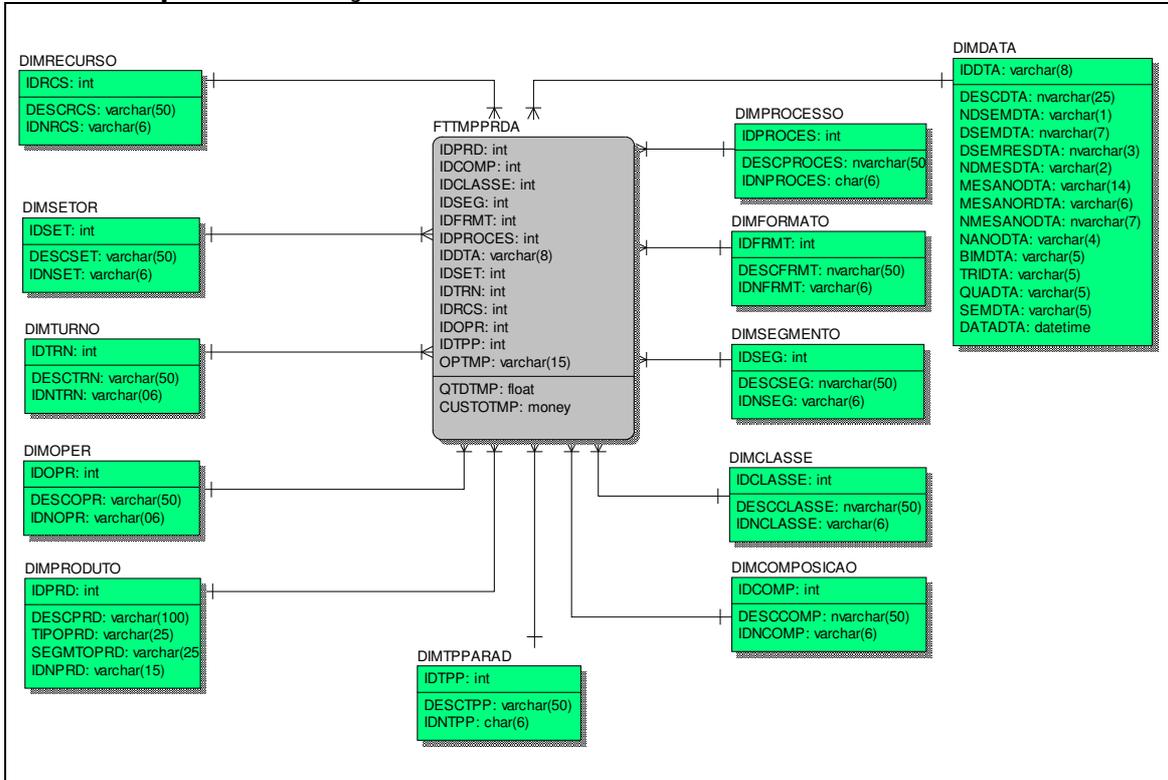
O modelo dimensional do projeto foi desenvolvido no *ER-WIN*. Uma Ferramenta *case* para modelagem de banco de dados, onde foram projetados o modelo lógico e o físico. Com o modelo físico pronto pode-se através da própria ferramenta se conectar com o banco de dados e gerar todo o modelo.

Visto que o modelo físico é na verdade o modelo lógico escrito para o banco de dados ao qual serão criadas as tabelas, segue abaixo os modelos físicos dos fatos do projeto.

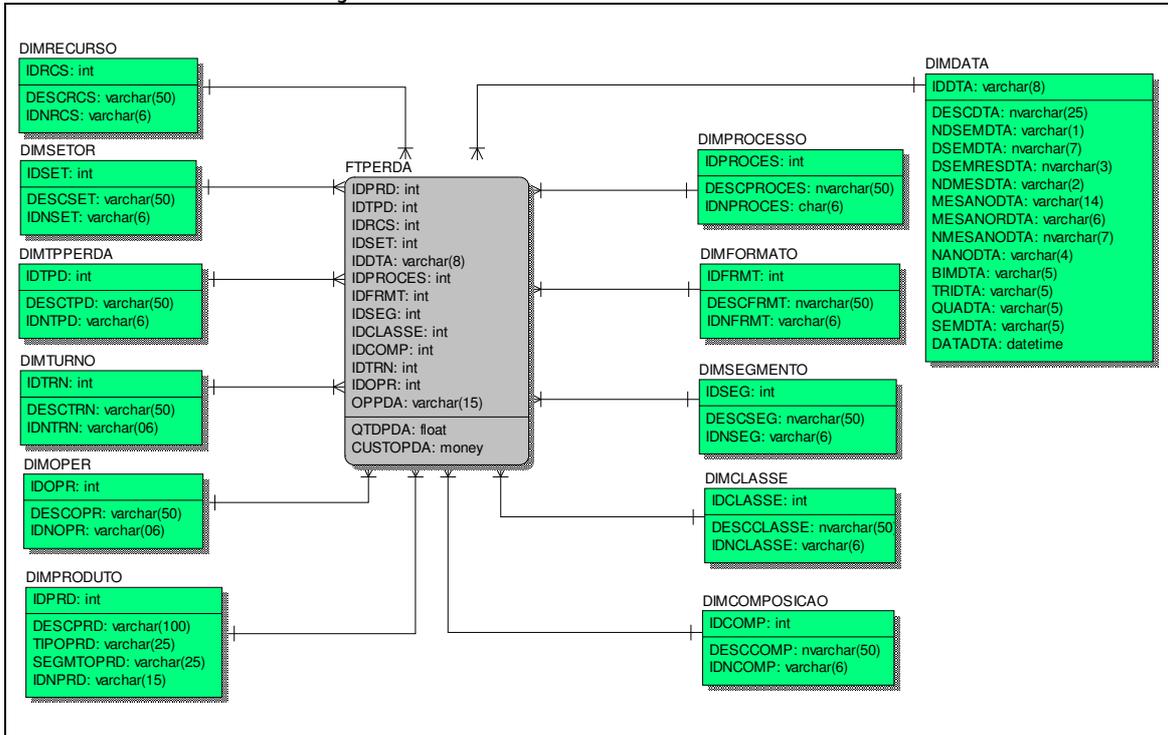
Fato Produção



Fato Tempo de Produção Parada



Fato Perda de Produção



Apêndice B – Dicionário de Dados

A ferramenta *case ER-WIN*, que foi utilizada para fazer a modelagem dimensional, também gera automaticamente o dicionário de dados, ao qual segue abaixo.

Nome da Tabela	Descrição da Tabela	Nome das Colunas	Descrição das Colunas	Tipo das Colunas
DIMCLASSE	Dimensão Classe	DESCCLASSE	descrição da Classe	nvarchar(50)
		IDCLASSE	Identificador da Classe	int
		IDNCLASSE	Chave nativa do OLPT	varchar(6)
DIMCOMPOSICAO	Dimensão	DESCCOMP	descrição da Composição	nvarchar(50)
		IDCOMP	Identificador da composição	int
		IDNCOMP	Chave nativa do OLPT	varchar(6)
DIMDATA	Dimensão Data	BIMDTA	Bimestre	varchar(5)
		DATADTA	Data	datetime
		DESCDTA	descrição completa (Sábado, 6 de setembro de 2003)	nvarchar(25)
		DSEMDTA	descrição do dia da semana(Domingo)	nvarchar(7)
		DSEMRESDTA	descrição resumida do dia da semana(Dom)	nvarchar(3)
		IDDTA	Identificador da dimensao data	varchar(8)
		MESANODTA	Mês e Ano (Janeiro/2004)	varchar(14)
		MESANORDTA	mês e Ano resumido (Jan/2004)	varchar(6)
		NANODTA	Ano (2004)	varchar(4)
		NDMESDTA	Dia do mês (01 , 02 .. 31)	varchar(2)
		NDSEMDTA	Numero do dia da semana(1 - Domingo)	varchar(1)
		NMESANODTA	mês e Ano em números (01/2004)	nvarchar(7)
		QUADTA	Quadrimestre	varchar(5)
		SEMDTA	Semestre	
TRIDTA	Trimestre			
DIMFORMATO	Dimensão Formato	DESCFRMT	descrição do Formato	nvarchar(50)
		IDFRMT	Identificador do Formato	int
		IDNFRMT	Chave nativa do OLPT	varchar(6)
DIMOPER	Dimensão Operador	DESCOPR	Descrição do Operador	varchar(50)
		IDNOPR	Chave nativa do OLPT	varchar(06)
		IDOPR	Identificador do Operador	int
DIMPROCESSO	Dimensão Processo	DESCPROCES	Descrição do Processo	nvarchar(50)
		IDNPROCES	Chave nativa do OLPT	char(6)
		IDPROCES	Identificador do Processo	int
DIMPRODUTO	Dimensão Produto	DESCPRD	descrição do produto	varchar(100)
		IDNPRD	Chave nativa do OLPT	varchar(15)
		IDPRD	Identificador do produto	int
		SEGMENTOPRD	Segmento do produto INDUSTRIAL CONSUMIDOR	varchar(25)

		TIPOPRD	Tipo de Produto"ACABADO""CONSUMO"etc...	
DIMRECURSO		DESCRCS	Decisão do recurso	varchar(50)
		IDNRCS	Identificador nativo do OLPT	varchar(6)
		IDRCS	Identificador da dimensao	int
DIMSEGMENTO	Dimensão Segmento	DESCSEG	descrição do Segmento	nvarchar(50)
		IDNSEG	Chave nativa do OLPT	varchar(6)
		IDSEG	Identificador do Segmento	int
DIMSETOR	Dimensão Setor	DESCSET	Descrição do setor	varchar(50)
		IDNSET	Chave nativa do OLPT	varchar(6)
		IDSET	Identificador do setor	int
DIMTPPARAD	Dimensão Descrição do Tempo de Produção Parada	DESCTPP	Descrição do tipo de parada	varchar(50)
		IDNTPP	Chave nativa do OLPT	char(6)
		IDTPP	Identificador do tipo de parada	int
DIMTPPERDA	Dimensão Descrição da Perda de Produção	DESCTPD	Descrição do tipo de perda	varchar(50)
		IDNTPD	Chave nativa do OLPT	varchar(6)
		IDTPD	Identificador do tipo de perda	int
DIMTURNO	Dimensão Turno	DESCTRN	Descrição do Turno	varchar(50)
		IDNTRN	Chave nativa do OLPT	varchar(06)
		IDTRN	Identificador do Turno	int
FTPERDA	Fato que detalha as perdas de produção.	CUSTOPDA	Custo da perda Custo de producao * quantidade da perda.	money
		IDCLASSE	Identificador da Classe	int
		IDCOMP	Identificador da composição	
		IDDTA	Identificador da dimensao data	varchar(8)
		IDFRMT	Identificador do Formato	int
		IDOPR	Identificador do Operador	
		IDPRD	Identificador do produto	
		IDPROCES	Identificador do Processo	
		IDRCS	Identificador da dimensao	
		IDSEG	Identificador do Segmento	
		IDSET	Identificador do setor	
		IDTPD	Identificador do tipo de perda	
		IDTRN	Identificador do Turno	
		OPPDA	Numero da ordem de producao no OLPT	varchar(15)
QTDPPDA	Quantidade da Perda	float		
FTPRODUCAO	Fato que mostra a produção por quantidade, tempo, perda e tempo de produção parada	IDCLASSE	Identificador da Classe	int
		IDCOMP	Identificador da composição	
		IDDTA	Identificador da dimensao data	varchar(8)
		IDFRMT	Identificador do Formato	int
		IDOPR	Identificador do Operador	
		IDPRD	Identificador do produto	
		IDPROCES	Identificador do Processo	
		IDRCS	Identificador da dimensao	
		IDSEG	Identificador do Segmento	
		IDSET	Identificador do setor	
IDTRN	Identificador do Turno			

		OPPRO	Numero da ordem de producao no OLPT	char(18)
		QTDPDAPRO	Quantidade de perda de producao	float
		QTDPRO1PRO	Quantidade de perda de producao no primeiro processo	
		QTDPROFPRO	Quantidade de perda de producao no ultimo processo	
		QTDPROPRO	Quantidade produzida em Kg	
		TMPPDAPRO	Tempo de producao parada em minutos	
		TMPPROPRO	Tempo de producao em minutos	
FTTMPPRDA	Fato que detalha o tempo de produção parada.	CUSTOTMP	Custo do tempo de producao parada	money
		IDCLASSE	Identificador da Classe	int
		IDCOMP	Identificador da composição	
		IDDTA	Identificador da dimensao data	varchar(8)
		IDFRMT	Identificador do Formato	int
		IDOPR	Identificador do Operador	
		IDPRD	Identificador do produto	
		IDPROCES	Identificador do Processo	
		IDRCS	Identificador da dimensao	
		IDSEG	Identificador do Segmento	
		IDSET	Identificador do setor	
		IDTPP	Identificador do tipo de parada	
		IDTRN	Identificador do Turno	
		OPTMP	Numero da ordem de producao no OLPT	varchar(15)
		QTDTMP	Tempo de producao parada em minutos	float

Apêndice C – ETL (*Extract, Transform and Load*)

Todo processo de ETL do projeto foi desenvolvido em *Transact SQL*, a linguagem de programação do banco de dados da *Microsoft*, o *SQL Server*. Os *scripts* foram escritos em forma de *store procedures* a fim de obterem uma melhor performance e de deixar a rotina de automação mais simples, ao qual roda através do *scheduler* do próprio banco.

A automação dos *scripts* foi dividida em duas partes: *scripts* de *ETL* de dimensões e *scripts* de *ETL* de fatos. Abaixo segue os respectivos *scripts*.

ETL das Dimensões

```

/*****
* Dimensao DIMOPER
* Operador
*****/

if exists (select * from sysobjects where id = object_id(N'[dbo].[USP_DIMOPER]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
    DROP PROC [dbo].[USP_DIMOPER]
GO

CREATE PROC USP_DIMOPER
AS

SELECT X5_TABELA, X5_CHAVE,X5_DESCRI
INTO DSA.dbo.TEMPSX5
FROM Dadosadv.dbo.SX5010
WHERE X5_TABELA = '97'
and NOT EXISTS ( SELECT DIMOPER2.IDNOPR
                 FROM DSA.dbo.DIMOPER DIMOPER2
                 WHERE DIMOPER2.DESCOPR = X5_DESCRI
                 AND DIMOPER2.IDNOPR = X5_CHAVE )

And D_E_L_E_T_ <> '*'

DECLARE @X5_CHAVE VARCHAR(06)
DECLARE @X5_DESCRI VARCHAR(50)
DECLARE TNames_CURSOR CURSOR
FOR SELECT X5_CHAVE,X5_DESCRI FROM DSA.dbo.TEMPSX5

OPEN TNames_CURSOR
FETCH NEXT FROM TNames_CURSOR INTO @X5_CHAVE,@X5_DESCRI
WHILE (@@FETCH_STATUS = 0 ) BEGIN

IF NOT EXISTS ( SELECT DIMOPER2.IDNOPR
                FROM DSA.dbo.DIMOPER DIMOPER2
                WHERE DIMOPER2.DESCOPR = @X5_DESCRI
                AND DIMOPER2.IDNOPR = @X5_CHAVE )

BEGIN
INSERT INTO DSA.dbo.DIMOPER (DESCOPR, IDNOPR)
                values (@X5_DESCRI,@X5_CHAVE )
END

FETCH NEXT FROM TNames_CURSOR INTO @X5_CHAVE,@X5_DESCRI

```

```

END

CLOSE TNames_CURSOR
DEALLOCATE TNames_CURSOR

IF NOT EXISTS (SELECT * FROM DSA.dbo.DIMOPER WHERE IDNOPR = 'ZZZ')
BEGIN
    INSERT INTO DSA.dbo.DIMOPER (DESCOPR, IDNOPR)
        values('DIVERSOS','ZZZ')
END

DROP TABLE DSA.dbo.TEMPSX5

GO

/*****
*   Dimensao DIMTURNO
*   Turno
*****/

if exists (select * from sysobjects where id = object_id(N'[dbo].[USP_DIMTURNO]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
    DROP PROC [dbo].[USP_DIMTURNO]
GO

CREATE PROC USP_DIMTURNO
AS

SELECT X5_TABELA,X5_CHAVE,X5_DESCRI
INTO DSA.dbo.TEMPSX5
FROM Dadosadv.dbo.SX5010
WHERE X5_TABELA = '98'
and NOT EXISTS ( SELECT DIMTURNO2.IDNTRN
                  FROM DSA.dbo.DIMTURNO DIMTURNO2
                  WHERE DIMTURNO2.DESCTRN = X5_DESCRI
                  AND DIMTURNO2.IDNTRN = X5_CHAVE )

And D_E_L_E_T_ <> '*'

DECLARE @X5_CHAVE VARCHAR(06)
DECLARE @X5_DESCRI VARCHAR(50)
DECLARE TNames_CURSOR CURSOR
FOR SELECT X5_CHAVE,X5_DESCRI FROM DSA.dbo.TEMPSX5

OPEN TNames_CURSOR
FETCH NEXT FROM TNames_CURSOR INTO @X5_CHAVE,@X5_DESCRI
WHILE (@@FETCH_STATUS = 0 ) BEGIN

IF NOT EXISTS ( SELECT DIMTURNO2.IDNTRN
                FROM DSA.dbo.DIMTURNO DIMTURNO2
                WHERE DIMTURNO2.DESCTRN = @X5_DESCRI
                AND DIMTURNO2.IDNTRN = @X5_CHAVE )

BEGIN

    INSERT INTO DSA.dbo.DIMTURNO (DESCTRN, IDNTRN)
        values (@X5_DESCRI,@X5_CHAVE )

END

    FETCH NEXT FROM TNames_CURSOR INTO @X5_CHAVE,@X5_DESCRI
END

CLOSE TNames_CURSOR
DEALLOCATE TNames_CURSOR

IF NOT EXISTS (SELECT * FROM DSA.dbo.DIMTURNO WHERE IDNTRN = 'Z')
BEGIN
    INSERT INTO DSA.dbo.DIMTURNO (DESCTRN, IDNTRN)
        values('DIVERSOS','Z')
END

DROP TABLE DSA.dbo.TEMPSX5

GO

```

```

/*****
* Dimensao DIMTPPERDA *
* Tipo de Perda *
*****/

if exists (select * from sysobjects where id = object_id(N'[dbo].[USP_DIMTPPERDA]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
    DROP PROC [dbo].[USP_DIMTPPERDA]
GO

CREATE PROC USP_DIMTPPERDA
AS

SELECT X5_TABELA,X5_CHAVE,X5_DESCRI
INTO DSA.dbo.TEMPSX5
FROM Dadosadv.dbo.SX5010
WHERE X5_TABELA = '43'
and NOT EXISTS ( SELECT DIMTPPERDA2.IDNTPD
                  FROM DSA.dbo.DIMTPPERDA DIMTPPERDA2
                  WHERE DIMTPPERDA2.DESCTPD= X5_DESCRI
                  AND DIMTPPERDA2.IDNTPD= X5_CHAVE )

And D_E_L_E_T_ <> '*'

DECLARE @X5_CHAVE VARCHAR(06)
DECLARE @X5_DESCRI VARCHAR(50)
DECLARE TNames_CURSOR CURSOR
FOR SELECT X5_CHAVE,X5_DESCRI FROM DSA.dbo.TEMPSX5

OPEN TNames_CURSOR
FETCH NEXT FROM TNames_CURSOR INTO @X5_CHAVE,@X5_DESCRI
WHILE (@@FETCH_STATUS = 0 ) BEGIN

    INSERT INTO DSA.dbo.DIMTPPERDA (DESCTPD, IDNTPD)
        values (@X5_DESCRI,@X5_CHAVE )

    FETCH NEXT FROM TNames_CURSOR INTO @X5_CHAVE,@X5_DESCRI
END

CLOSE TNames_CURSOR
DEALLOCATE TNames_CURSOR

IF NOT EXISTS (SELECT * FROM DSA.dbo.DIMTPPERDA WHERE IDNTPD= 'ZZ')
    BEGIN
        INSERT INTO DSA.dbo.DIMTPPERDA (DESCTPD, IDNTPD)
            values ('DIVERSOS','ZZ')
    END

DROP TABLE DSA.dbo.TEMPSX5

GO

/*****
* Dimensao DIMTPPARAD *
* Tipo de Parada *
*****/

if exists (select * from sysobjects where id = object_id(N'[dbo].[USP_DIMTPPARAD]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
    DROP PROC [dbo].[USP_DIMTPPARAD]
GO

CREATE PROC USP_DIMTPPARAD
AS

SELECT X5_TABELA,X5_CHAVE,X5_DESCRI
INTO DSA.dbo.TEMPSX5
FROM Dadosadv.dbo.SX5010
WHERE X5_TABELA = '44'
and NOT EXISTS ( SELECT DIMTPPARAD2.IDNTPP
                  FROM DSA.dbo.DIMTPPARAD DIMTPPARAD2
                  WHERE DIMTPPARAD2.DESCTPP = X5_DESCRI
                  AND DIMTPPARAD2.IDNTPP = X5_CHAVE )

And D_E_L_E_T_ <> '*'

```

```

DECLARE @X5_CHAVE VARCHAR(06)
DECLARE @X5_DESCRI VARCHAR(50)
DECLARE T_NAMES_CURSOR CURSOR
FOR SELECT X5_CHAVE,X5_DESCRI FROM DSA.dbo.TEMPSX5

OPEN T_NAMES_CURSOR
FETCH NEXT FROM T_NAMES_CURSOR INTO @X5_CHAVE,@X5_DESCRI
WHILE (@@FETCH_STATUS = 0 ) BEGIN

    INSERT INTO DSA.dbo.DIMTPPARAD (DESCTPP, IDNTPP)
        values (@X5_DESCRI,@X5_CHAVE )

    FETCH NEXT FROM T_NAMES_CURSOR INTO @X5_CHAVE,@X5_DESCRI
END

CLOSE T_NAMES_CURSOR
DEALLOCATE T_NAMES_CURSOR

IF NOT EXISTS (SELECT * FROM DSA.dbo.DIMTPPARAD WHERE IDNTPP = 'ZZZ')
BEGIN
    INSERT INTO DSA.dbo.DIMTPPARAD (DESCTPP, IDNTPP)
        values ('DIVERSOS','ZZZ')
END

DROP TABLE DSA.dbo.TEMPSX5

GO

/*****
*   Dimensao DIMSETOR
*           Setor
*****/

if exists (select * from sysobjects where id = object_id(N'[dbo].[USP_DIMSETOR]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
    DROP PROC [dbo].[USP_DIMSETOR]
GO

CREATE PROC USP_DIMSETOR
AS

SELECT * INTO DSA.dbo.TEMPSHB
FROM Dadosadv.dbo.SHB010
WHERE NOT EXISTS ( SELECT DIMSETOR2.IDNSET
                    FROM DSA.dbo.DIMSETOR DIMSETOR2
                    WHERE DIMSETOR2.DESCSET = HB_NOME
                    AND DIMSETOR2.IDNSET = HB_COD )
And D_E_L_E_T_ <> '*'

DECLARE @HB_COD VARCHAR(03)
DECLARE @HB_NOME VARCHAR(50)
DECLARE T_NAMES_CURSOR CURSOR
FOR SELECT HB_COD,HB_NOME FROM DSA.dbo.TEMPSHB

OPEN T_NAMES_CURSOR
FETCH NEXT FROM T_NAMES_CURSOR INTO @HB_COD,@HB_NOME
WHILE (@@FETCH_STATUS = 0 ) BEGIN

    INSERT INTO DSA.dbo.DIMSETOR (DESCSET, IDNSET)
        values (@HB_NOME,@HB_COD)

    FETCH NEXT FROM T_NAMES_CURSOR INTO @HB_COD,@HB_NOME
END

CLOSE T_NAMES_CURSOR
DEALLOCATE T_NAMES_CURSOR

IF NOT EXISTS (SELECT * FROM DSA.dbo.DIMSETOR WHERE IDNSET = 'ZZZZZ')
BEGIN
    INSERT INTO DSA.dbo.DIMSETOR (DESCSET, IDNSET)
        values ('DIVERSOS','ZZZZZ')

```

```

END

DROP TABLE DSA.dbo.TEMPSHB

GO

/*****
*   Dimensao DIMRECURSO
*           Recurso
*****/

if exists (select * from sysobjects where id = object_id(N'[dbo].[USP_DIMRECURSO]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
    DROP PROC [dbo].[USP_DIMRECURSO]
GO

CREATE PROC USP_DIMRECURSO
AS

SELECT H1_CODIGO,H1_DESCRI
INTO DSA.dbo.TEMPSH1
FROM Dadosadv.dbo.SH1010
WHERE NOT EXISTS ( SELECT DIMRECURSO2.IDNRCS
                    FROM DSA.dbo.DIMRECURSO DIMRECURSO2
                    WHERE DIMRECURSO2.DESCRCS = H1_DESCRI
                    AND DIMRECURSO2.IDNRCS = H1_CODIGO )

And D_E_L_E_T_ <> '*'

DECLARE @H1_CODIGO VARCHAR(06)
DECLARE @H1_DESCRI VARCHAR(50)

DECLARE TNAMES_CURSOR CURSOR
FOR SELECT H1_CODIGO,H1_DESCRI FROM DSA.dbo.TEMPSH1
OPEN TNAMES_CURSOR
FETCH NEXT FROM TNAMES_CURSOR INTO @H1_CODIGO,@H1_DESCRI
WHILE (@@FETCH_STATUS = 0 ) BEGIN

    INSERT INTO DSA.dbo.DIMRECURSO (DESCRCS, IDNRCS)
        values (@H1_DESCRI,@H1_CODIGO)

    FETCH NEXT FROM TNAMES_CURSOR INTO @H1_CODIGO,@H1_DESCRI
END

CLOSE TNAMES_CURSOR
DEALLOCATE TNAMES_CURSOR

IF NOT EXISTS (SELECT * FROM DSA.dbo.DIMRECURSO WHERE IDNRCS = 'ZZZZZ')
    BEGIN
        INSERT INTO DSA.dbo.DIMRECURSO (DESCRCS, IDNRCS)
            values ('DIVERSOS','ZZZZZ')
    END

DROP TABLE DSA.dbo.TEMPSH1

GO

/*****
*   Dimensao DIMPROCESSO
*           Processo
*****/

if exists (select * from sysobjects where id = object_id(N'[dbo].[USP_DIMPROCESSO]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
    DROP PROC [dbo].[USP_DIMPROCESSO]
GO

CREATE PROC USP_DIMPROCESSO
AS

SELECT X5_TABELA,X5_CHAVE,X5_DESCRI
INTO DSA.dbo.TEMPSX5
FROM Dadosadv.dbo.SX5010
WHERE X5_TABELA = '90'

```

```

and NOT EXISTS ( SELECT DIMPROCESSO2.IDNPROCES
                  FROM DSA.dbo.DIMPROCESSO DIMPROCESSO2
                  WHERE DIMPROCESSO2.DESCPROCES = X5_DESCRI
                  AND DIMPROCESSO2.IDNPROCES = X5_CHAVE )
And D_E_L_E_T_ <> '*'

DECLARE @X5_CHAVE VARCHAR(06)
DECLARE @X5_DESCRI VARCHAR(50)
DECLARE T_NAMES_CURSOR CURSOR
FOR SELECT X5_CHAVE,X5_DESCRI FROM DSA.dbo.TEMPSX5

OPEN T_NAMES_CURSOR
FETCH NEXT FROM T_NAMES_CURSOR INTO @X5_CHAVE,@X5_DESCRI
WHILE (@@FETCH_STATUS = 0 ) BEGIN

IF NOT EXISTS ( SELECT DIMPROCESSO2.IDNPROCES
                  FROM DSA.dbo.DIMPROCESSO DIMPROCESSO2
                  WHERE DIMPROCESSO2.DESCPROCES = @X5_DESCRI
                  AND DIMPROCESSO2.IDNPROCES = @X5_CHAVE )

BEGIN
INSERT INTO DSA.dbo.DIMPROCESSO (DESCPROCES, IDNPROCES)
values (@X5_DESCRI,@X5_CHAVE )

END

FETCH NEXT FROM T_NAMES_CURSOR INTO @X5_CHAVE,@X5_DESCRI
END

CLOSE T_NAMES_CURSOR
DEALLOCATE T_NAMES_CURSOR

IF NOT EXISTS (SELECT * FROM DSA.dbo.DIMPROCESSO WHERE IDNPROCES = 'ZZZZZ')
BEGIN
INSERT INTO DSA.dbo.DIMPROCESSO (DESCPROCES, IDNPROCES)
values ('DIVERSOS','ZZZZZ')

END

DROP TABLE DSA.dbo.TEMPSX5

GO

/*****
* Dimensao DIMFORMATO *
* Formato *
*****/

if exists (select * from sysobjects where id = object_id(N'[dbo].[USP_DIMFORMATO]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
DROP PROC [dbo].[USP_DIMFORMATO]
GO

CREATE PROC USP_DIMFORMATO
AS

SELECT X5_TABELA,X5_CHAVE,X5_DESCRI
INTO DSA.dbo.TEMPSX5
FROM Dadosadv.dbo.SX5010
WHERE X5_TABELA = '89'
and NOT EXISTS ( SELECT DIMFORMATO2.IDNFRMT
                  FROM DSA.dbo.DIMFORMATO DIMFORMATO2
                  WHERE DIMFORMATO2.DESCFRMT = X5_DESCRI
                  AND DIMFORMATO2.IDNFRMT = X5_CHAVE )
And D_E_L_E_T_ <> '*'

DECLARE @X5_CHAVE VARCHAR(06)
DECLARE @X5_DESCRI VARCHAR(50)
DECLARE T_NAMES_CURSOR CURSOR
FOR SELECT X5_CHAVE,X5_DESCRI FROM DSA.dbo.TEMPSX5

OPEN T_NAMES_CURSOR
FETCH NEXT FROM T_NAMES_CURSOR INTO @X5_CHAVE,@X5_DESCRI
WHILE (@@FETCH_STATUS = 0 ) BEGIN

```

```

IF NOT EXISTS ( SELECT DIMFORMATO2.IDNFRMT
                FROM DSA.dbo.DIMFORMATO DIMFORMATO2
                WHERE DIMFORMATO2.DESCFRMT = @X5_DESCRI
                AND DIMFORMATO2.IDNFRMT = @X5_CHAVE )

BEGIN
INSERT INTO DSA.dbo.DIMFORMATO (DESCFRMT, IDNFRMT)
        values (@X5_DESCRI,@X5_CHAVE )

END

        FETCH NEXT FROM TNames_CURSOR INTO @X5_CHAVE,@X5_DESCRI
END

CLOSE TNames_CURSOR
DEALLOCATE TNames_CURSOR

IF NOT EXISTS (SELECT * FROM DSA.dbo.DIMFORMATO WHERE IDNFRMT = 'ZZZZZ')
BEGIN
INSERT INTO DSA.dbo.DIMFORMATO (DESCFRMT, IDNFRMT)
        values ('DIVERSOS','ZZZZZ')

END

DROP TABLE DSA.dbo.TEMPSX5

GO

/*****
*   Dimensao DIMSEGMENTO                               *
*           Segmento                                   *
*****/

if exists (select * from sysobjects where id = object_id(N'[dbo].[USP_DIMSEGMENTO]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
        DROP PROC [dbo].[USP_DIMSEGMENTO]
GO

CREATE PROC USP_DIMSEGMENTO
AS

SELECT X5_TABELA,X5_CHAVE,X5_DESCRI
INTO DSA.dbo.TEMPSX5
FROM Dadosadv.dbo.SX5010
WHERE X5_TABELA = '88'
and NOT EXISTS ( SELECT DIMSEGMENTO2.IDNSEG
                FROM DSA.dbo.DIMSEGMENTO DIMSEGMENTO2
                WHERE DIMSEGMENTO2.DESCSEG = X5_DESCRI
                AND DIMSEGMENTO2.IDNSEG = X5_CHAVE )

And D_E_L_E_T_ <> '*'

DECLARE @X5_CHAVE VARCHAR(06)
DECLARE @X5_DESCRI VARCHAR(50)
DECLARE TNames_CURSOR CURSOR
FOR SELECT X5_CHAVE,X5_DESCRI FROM DSA.dbo.TEMPSX5

OPEN TNames_CURSOR
FETCH NEXT FROM TNames_CURSOR INTO @X5_CHAVE,@X5_DESCRI
WHILE (@@FETCH_STATUS = 0 ) BEGIN

IF NOT EXISTS ( SELECT DIMSEGMENTO2.IDNSEG
                FROM DSA.dbo.DIMSEGMENTO DIMSEGMENTO2
                WHERE DIMSEGMENTO2.DESCSEG = @X5_DESCRI
                AND DIMSEGMENTO2.IDNSEG = @X5_CHAVE )

BEGIN
INSERT INTO DSA.dbo.DIMSEGMENTO (DESCSEG, IDNSEG)
        values (@X5_DESCRI,@X5_CHAVE )

END

        FETCH NEXT FROM TNames_CURSOR INTO @X5_CHAVE,@X5_DESCRI

```

```

END

CLOSE TNames_CURSOR
DEALLOCATE TNames_CURSOR

IF NOT EXISTS (SELECT * FROM DSA.dbo.DIMSEGMENTO WHERE IDNSEG = 'ZZZZZZ')
BEGIN
    INSERT INTO DSA.dbo.DIMSEGMENTO (DESCSEG, IDNSEG)
        values ('DIVERSOS', 'ZZZZZZ')
END

DROP TABLE DSA.dbo.TEMPSX5

GO

/*****
*   Dimensao DIMCLASSE
*           CLASSE
*****/

if exists (select * from sysobjects where id = object_id(N'[dbo].[USP_DIMCLASSE]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
    DROP PROC [dbo].[USP_DIMCLASSE]
GO

CREATE PROC USP_DIMCLASSE
AS

SELECT X5_TABELA, X5_CHAVE, X5_DESCRI
INTO DSA.dbo.TEMPSX5
FROM Dadosadv.dbo.SX5010
WHERE X5_TABELA = '87'
and NOT EXISTS ( SELECT DIMCLASSE2.IDNCLASSE
                FROM DSA.dbo.DIMCLASSE DIMCLASSE2
                WHERE DIMCLASSE2.DESCCLASSE = X5_DESCRI
                AND DIMCLASSE2.IDNCLASSE = X5_CHAVE )

And D_E_L_E_T_ <> '*'

DECLARE @X5_CHAVE VARCHAR(06)
DECLARE @X5_DESCRI VARCHAR(50)
DECLARE TNames_CURSOR CURSOR
FOR SELECT X5_CHAVE, X5_DESCRI FROM DSA.dbo.TEMPSX5

OPEN TNames_CURSOR
FETCH NEXT FROM TNames_CURSOR INTO @X5_CHAVE, @X5_DESCRI
WHILE (@@FETCH_STATUS = 0 ) BEGIN

IF NOT EXISTS ( SELECT DIMCLASSE2.IDNCLASSE
                FROM DSA.dbo.DIMCLASSE DIMCLASSE2
                WHERE DIMCLASSE2.DESCCLASSE = @X5_DESCRI
                AND DIMCLASSE2.IDNCLASSE = @X5_CHAVE )

BEGIN
    INSERT INTO DSA.dbo.DIMCLASSE (DESCCLASSE, IDNCLASSE)
        values (@X5_DESCRI, @X5_CHAVE )
END

    FETCH NEXT FROM TNames_CURSOR INTO @X5_CHAVE, @X5_DESCRI
END

CLOSE TNames_CURSOR
DEALLOCATE TNames_CURSOR

IF NOT EXISTS (SELECT * FROM DSA.dbo.DIMCLASSE WHERE IDNCLASSE = 'ZZZZZZ')
BEGIN
    INSERT INTO DSA.dbo.DIMCLASSE (DESCCLASSE, IDNCLASSE)
        values ('DIVERSOS', 'ZZZZZZ')
END

```

```

DROP TABLE DSA.dbo.TEMPSX5

GO

/*****
* Dimensao DIMCOMPOSICAO
* COMPOSICAO
*****/

if exists (select * from sysobjects where id = object_id(N'[dbo].[USP_DIMCOMPOSICAO]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
    DROP PROC [dbo].[USP_DIMCOMPOSICAO]
GO

CREATE PROC USP_DIMCOMPOSICAO
AS

SELECT X5_TABELA, X5_CHAVE, X5_DESCRI
INTO DSA.dbo.TEMPSX5
FROM Dadosadv.dbo.SX5010
WHERE X5_TABELA = '86'
and NOT EXISTS ( SELECT DIMCOMPOSICAO2.IDNCOMP
                  FROM DSA.dbo.DIMCOMPOSICAO DIMCOMPOSICAO2
                  WHERE DIMCOMPOSICAO2.DESCCOMP = X5_DESCRI
                  AND DIMCOMPOSICAO2.IDNCOMP = X5_CHAVE )

And D_E_L_E_T_ <> '*'

DECLARE @X5_CHAVE VARCHAR(06)
DECLARE @X5_DESCRI VARCHAR(50)
DECLARE T_NAMES_CURSOR CURSOR
FOR SELECT X5_CHAVE, X5_DESCRI FROM DSA.dbo.TEMPSX5

OPEN T_NAMES_CURSOR
FETCH NEXT FROM T_NAMES_CURSOR INTO @X5_CHAVE, @X5_DESCRI
WHILE (@@FETCH_STATUS = 0 ) BEGIN

IF NOT EXISTS ( SELECT DIMCOMPOSICAO2.IDNCOMP
                FROM DSA.dbo.DIMCOMPOSICAO DIMCOMPOSICAO2
                WHERE DIMCOMPOSICAO2.DESCCOMP = @X5_DESCRI
                AND DIMCOMPOSICAO2.IDNCOMP = @X5_CHAVE )

    BEGIN
        INSERT INTO DSA.dbo.DIMCOMPOSICAO (DESCCOMP, IDNCOMP)
            values (@X5_DESCRI, @X5_CHAVE )
    END

    FETCH NEXT FROM T_NAMES_CURSOR INTO @X5_CHAVE, @X5_DESCRI
END

CLOSE T_NAMES_CURSOR
DEALLOCATE T_NAMES_CURSOR

IF NOT EXISTS (SELECT * FROM DSA.dbo.DIMCOMPOSICAO WHERE IDNCOMP = 'ZZZZZ')
    BEGIN
        INSERT INTO DSA.dbo.DIMCOMPOSICAO (DESCCOMP, IDNCOMP)
            values ('DIVERSOS', 'ZZZZZ')
    END

DROP TABLE DSA.dbo.TEMPSX5

GO

/*****
* Dimensao DIMPRODUTO
* PRODUTO
*****/

if exists (select * from sysobjects where id = object_id(N'[dbo].[USP_DIMPRODUTO]') and

```

```

OBJECTPROPERTY(id, N'IsProcedure') = 1)
DROP PROC [dbo].[USP_DIMPRODUTO]
GO

CREATE PROC USP_DIMPRODUTO
AS

SELECT
B1_COD,B1_DESC,B1_BIPROC,B1_BIFORM,B1_BISEG,B1_BICLASS,B1_BICOMP, 'B1_TIOPRD'=X5_DESCRI,B1_FAMPB,B1_SEGMTO,
'B1_TIPOEMB'= CASE
                WHEN B1_TIPOEMB = 'I' THEN 'INDUSTRIAL'
                WHEN B1_TIPOEMB = 'C' THEN 'CONSUMIDOR'
                ELSE ' '
            END
INTO DSA.dbo.TEMPSB1A
FROM Dadosadv.dbo.SB1010 SB1A
JOIN Dadosadv.dbo.SX5010 SX5A ON X5_TABELA = '02' AND B1_TIPO = X5_CHAVE
WHERE SB1A.D_E_L_E_T_ <> '*' AND SX5A.D_E_L_E_T_ <> '*'

UPDATE DSA.dbo.TEMPSB1A
SET     B1_FAMPB      = 'ZZZ',
        B1_SEGMTO    = 'ZZZ',
        B1_BIPROC    = 'ZZZZZZ',
        B1_BIFORM    = 'ZZZZZZ',
        B1_BISEG     = 'ZZZZZZ',
        B1_BICLASS   = 'ZZZZZZ',
        B1_BICOMP    = 'ZZZZZZ'
WHERE  B1_FAMPB      = ' '
      OR B1_SEGMTO   = ' '
      OR B1_BIPROC   = ' '
      OR B1_BIFORM   = ' '
      OR B1_BISEG    = ' '
      OR B1_BICLASS  = ' '
      OR B1_BICOMP   = ' '

SELECT B1_COD,B1_DESC,B1_TIOPRD,B1_TIPOEMB
INTO DSA.dbo.TEMPSB1B
FROM DSA.dbo.TEMPSB1A
WHERE NOT EXISTS ( SELECT DIMPRODUTO2.IDNPRD
                  FROM DSA.dbo.DIMPRODUTO DIMPRODUTO2
                  WHERE DIMPRODUTO2.DESCPRD = B1_DESC
                    AND DIMPRODUTO2.IDNPRD = B1_COD )

DECLARE @B1_TIOPRD VARCHAR(50)
DECLARE @B1_TIPOEMB VARCHAR(50)
DECLARE @B1_DESC VARCHAR(250)
DECLARE @B1_COD VARCHAR(15)

DECLARE TNames_CURSOR CURSOR
FOR SELECT B1_DESC,B1_TIOPRD,B1_TIPOEMB,B1_COD FROM DSA.dbo.TEMPSB1B

OPEN TNames_CURSOR
FETCH NEXT FROM TNames_CURSOR INTO @B1_DESC,@B1_TIOPRD,@B1_TIPOEMB,@B1_COD
WHILE (@@FETCH_STATUS = 0 ) BEGIN

IF NOT EXISTS ( SELECT DIMPRODUTO2.IDNPRD
              FROM DSA.dbo.DIMPRODUTO DIMPRODUTO2
              WHERE DIMPRODUTO2.DESCPRD = @B1_DESC
                AND DIMPRODUTO2.IDNPRD = @B1_COD )

BEGIN
INSERT INTO DSA.dbo.DIMPRODUTO (DESCPRD, TIOPRD, SEGMTOPRD, IDNPRD)
VALUES (@B1_DESC, @B1_TIOPRD, @B1_TIPOEMB, @B1_COD )
END
FETCH NEXT FROM TNames_CURSOR INTO @B1_DESC, @B1_TIOPRD, @B1_TIPOEMB, @B1_COD
END

CLOSE TNames_CURSOR
DEALLOCATE TNames_CURSOR

DROP TABLE DSA.dbo.TEMPSB1A

```

```
DROP TABLE DSA.dbo.TEMPSB1B
```

```
GO
```

ETL dos Fatos

```

/*****
* Fato FTPRODUCAO
* Producao Realizada
*****/

if exists (select * from sysobjects where id = object_id(N'[dbo].[USP_FTPRODUCAO]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
DROP PROC [dbo].[USP_FTPRODUCAO]
GO

CREATE PROC USP_FTPRODUCAO
AS

/*****
* Cria o arquivo temporario de Producao
*****/
SELECT
'BIPROC'=ISNULL(B1_BIPROC, 'ZZZZZ'),
'BIFORM'=ISNULL(B1_BIFORM, 'ZZZZZ'),
'BISEG'=ISNULL(B1_BISEG, 'ZZZZZ'),
'BICLASS'=ISNULL(B1_BICLASS, 'ZZZZZ'),
'BICOMP'=ISNULL(B1_BICOMP, 'ZZZZZ'),
'PRODUTO'= B1_COD,
'CTRAB'=isnull(H1_CTRAB, 'ZZZZZ'),
'TURNO' =isnull(H6_TURNO, 'ZZZZZ'),
'OPERADO'=isnull(H6_OPERADO, 'ZZZZZ'),
'RECURSO'=isnull(H6_RECURSO, 'ZZZZZ'),
'DATA' = H6_DATAFIN,
'OP'=H6_OP,
'QTDPRO'=(CASE WHEN H1_CTRAB IN ('3','5') AND B1_UM <> 'KG' THEN
(((B1_LARG+B1_TRANSPA)/100)*(B1_COMP/100)*B1_ESPE)*B1_DENSID)*H6_QTDPROD
WHEN H1_CTRAB IN ('4') AND B1_UM <> 'KG' AND B1_BICLASS = '000002' THEN
(((B1_LARG+B1_TRANSPA)/100)*(B1_COMP/100)*B1_ESPE)*B1_DENSID)*H6_QTDPROD
ELSE
H6_QTDPROD
END),
'TMPPRO'=(CAST(SUBSTRING(H6_TEMPO,1,2) AS INT)*60)+CAST(SUBSTRING(H6_TEMPO,4,2) AS INT),
'TMPPDA'=0,
'QTDPPDA'=0,
'QTDPRO1'=Dadosadv.dbo.udsSetPrdIF('I',H1_CTRAB,B1_COD,H6_RECURSO,(CASE WHEN H1_CTRAB IN
('3','5') AND B1_UM <> 'KG' THEN
(((B1_LARG+B1_TRANSPA)/100)*(B1_COMP/100)*B1_ESPE)*B1_DENSID)* H6_QTDPROD
WHEN H1_CTRAB IN ('4') AND B1_UM <> 'KG' AND B1_BICLASS = '000002' THEN
(((B1_LARG+B1_TRANSPA)/100)*(B1_COMP/100)*B1_ESPE)*B1_DENSID)* H6_QTDPROD
ELSE
H6_QTDPROD
END)),
'QTDPROF'=Dadosadv.dbo.udsSetPrdIF('F',H1_CTRAB,B1_COD,H6_RECURSO,(CASE WHEN H1_CTRAB IN
('3','5') AND B1_UM <> 'KG' THEN
(((B1_LARG+B1_TRANSPA)/100)*(B1_COMP/100)*B1_ESPE)*B1_DENSID)* H6_QTDPROD
WHEN H1_CTRAB IN ('4') AND B1_UM <> 'KG' AND B1_BICLASS = '000002' THEN
(((B1_LARG+B1_TRANSPA)/100)*(B1_COMP/100)*B1_ESPE)*B1_DENSID)* H6_QTDPROD
ELSE
H6_QTDPROD
END))

```

```

INTO DSA.dbo.tmpFatoProd_H6
FROM Dadosadv.dbo.SH6010 SH6
left JOIN Dadosadv.dbo.SB1010 SB1 ON SB1.B1_COD = SH6.H6_PRODUTO
left JOIN Dadosadv.dbo.SH1010 SH1 ON SH1.H1_CODIGO = SH6.H6_RECURSO
WHERE SH6.D_E_L_E_T_ = ' '
AND SB1.D_E_L_E_T_ = ' '

/*****
* Cria o arquivo temporario de Perda *
*****/
SELECT
'BIPROC'=ISNULL(B1_BIPROC, 'ZZZZZ'),
'BIFORM'=ISNULL(B1_BIFORM, 'ZZZZZ'),
'BISEG'=ISNULL(B1_BISEG, 'ZZZZZ'),
'BICLASS'=ISNULL(B1_BICLASS, 'ZZZZZ'),
'BICOMP'=ISNULL(B1_BICOMP, 'ZZZZZ'),
'PRODUTO'= B1_COD,
'CTRAB'=isnull(H1_CTRAB, 'ZZZZZ'),
'TURNO' =isnull(Z3_TURNO, 'ZZZZZ'),
'OPERADO'=isnull(Z3_OPERADO, 'ZZZZZ'),
'RECURSO'=isnull(Z3_RECURSO, 'ZZZZZ'),
'DATA' = Z3_DATA,
'OP'=Z3_OP,
'QTDPRO'=0,
'TMPPRO'=0,
'TMPPDA'=0,
'QTDPCA'=Z3_QUANTID,
'QTDPRO1'=0,
'QTDPROF'=0
INTO DSA.dbo.tmpFatoProd_Z3
FROM Dadosadv.dbo.SZ3010 SZ3
left JOIN Dadosadv.dbo.SB1010 SB1 ON Z3_PRODUTO = B1_COD
left JOIN Dadosadv.dbo.SH1010 SH1 ON Z3_RECURSO = H1_CODIGO
where SZ3.D_E_L_E_T_ = ' '
AND SB1.D_E_L_E_T_ = ' '
AND SH1.D_E_L_E_T_ = ' '

/*****
* Cria o arquivo temporario de tempo de parada *
*****/
SELECT
'BIPROC'=ISNULL(B1_BIPROC, 'ZZZZZ'),
'BIFORM'=ISNULL(B1_BIFORM, 'ZZZZZ'),
'BISEG'=ISNULL(B1_BISEG, 'ZZZZZ'),
'BICLASS'=ISNULL(B1_BICLASS, 'ZZZZZ'),
'BICOMP'=ISNULL(B1_BICOMP, 'ZZZZZ'),
'PRODUTO'= B1_COD,
'CTRAB'=isnull(H1_CTRAB, 'ZZZZZ'),
'TURNO' =isnull(Z4_TURNO, 'ZZZZZ'),
'OPERADO'=isnull(Z4_OPERADO, 'ZZZZZ'),
'RECURSO'=isnull(Z4_RECURSO, 'ZZZZZ'),
'DATA' = Z4_DATA,
'OP'=Z4_OP,
'QTDPRO'=0,
'TMPPRO'=0,
'TMPPDA'= (FLOOR(Z4_HORAS) *60)+FLOOR((Z4_HORAS-FLOOR(Z4_HORAS)) *100),
'QTDPCA'=0,
'QTDPRO1'=0,
'QTDPROF'=0
INTO DSA.dbo.tmpFatoProd_Z4
FROM Dadosadv.dbo.SZ4010 SZ4
left JOIN Dadosadv.dbo.SB1010 SB1 ON ' ' = B1_FILIAL AND Z4_PRODUTO = B1_COD
left JOIN Dadosadv.dbo.SH1010 SH1 ON '01' = H1_FILIAL AND Z4_RECURSO = H1_CODIGO
where SZ4.D_E_L_E_T_ = ' '
AND SB1.D_E_L_E_T_ = ' '
AND SH1.D_E_L_E_T_ = ' '

/*****
* Junta os temporarios todos ao temporario de Producao *
*****/
DECLARE @BIPROC VARCHAR(6)
DECLARE @BIFORM VARCHAR(6)
DECLARE @BISEG VARCHAR(6)

```

```

DECLARE @BICLASS          VARCHAR(6)
DECLARE @BICOMP           VARCHAR(6)
DECLARE @PRODUTO         VARCHAR(15)
DECLARE @CTRAB           VARCHAR(1)
DECLARE @TURNO           VARCHAR(1)
DECLARE @OPERADO         VARCHAR(3)
DECLARE @RECURSO         VARCHAR(5)
DECLARE @DATA            VARCHAR(8)
DECLARE @OP              VARCHAR(11)
DECLARE @QTDPRO          float
DECLARE @TMPPRO          float
DECLARE @TMPPDA          float
DECLARE @QTDPPDA         float
DECLARE @QTDPRO1         float
DECLARE @QTDPROF         float

--* Perda
--***** /
DECLARE TNames_CURSOR CURSOR
FOR SELECT
BIPROC, BIFORM, BISEG, BICLASS, BICOMP, PRODUTO, CTRAB, TURNO, OPERADO, RECURSO, DATA, OP, QTDPRO, TMPPRO,
TMPPDA, QTDPPDA, QTDPRO1, QTDPROF
FROM DSA.dbo.tmpFatoProd_Z3
OPEN TNames_CURSOR
FETCH NEXT FROM TNames_CURSOR INTO
@BIPROC, @BIFORM, @BISEG, @BICLASS, @BICOMP, @PRODUTO, @CTRAB, @TURNO, @OPERADO, @RECURSO, @DATA, @OP,
@QTDPRO, @TMPPRO, @TMPPDA, @QTDPPDA, @QTDPRO1, @QTDPROF

WHILE (@@FETCH_STATUS = 0 ) BEGIN
BEGIN
INSERT INTO
DSA.dbo.tmpFatoProd_H6 (BIPROC, BIFORM, BISEG, BICLASS, BICOMP, PRODUTO, CTRAB, TURNO, OPERADO, RECURSO,
DATA, OP, QTDPRO, TMPPRO, TMPPDA, QTDPPDA, QTDPRO1, QTDPROF)

values (@BIPROC, @BIFORM, @BISEG, @BICLASS, @BICOMP, @PRODUTO, @CTRAB, @TURNO, @OPERADO, @RECURSO, @DATA,
@OP, @QTDPRO, @TMPPRO, @TMPPDA, @QTDPPDA, @QTDPRO1, @QTDPROF)
END
FETCH NEXT FROM TNames_CURSOR INTO
@BIPROC, @BIFORM, @BISEG, @BICLASS, @BICOMP, @PRODUTO, @CTRAB, @TURNO, @OPERADO, @RECURSO, @DATA, @OP,
@QTDPRO, @TMPPRO, @TMPPDA, @QTDPPDA, @QTDPRO1, @QTDPROF
END

CLOSE TNames_CURSOR
DEALLOCATE TNames_CURSOR

--* Tempo de parada
--***** /
DECLARE TNames_CURSOR CURSOR
FOR SELECT
BIPROC, BIFORM, BISEG, BICLASS, BICOMP, PRODUTO, CTRAB, TURNO, OPERADO, RECURSO, DATA, OP, QTDPRO, TMPPRO,
TMPPDA, QTDPPDA, QTDPRO1, QTDPROF
FROM DSA.dbo.tmpFatoProd_Z4

OPEN TNames_CURSOR
FETCH NEXT FROM TNames_CURSOR INTO
@BIPROC, @BIFORM, @BISEG, @BICLASS, @BICOMP, @PRODUTO, @CTRAB, @TURNO, @OPERADO, @RECURSO, @DATA, @OP,
@QTDPRO, @TMPPRO, @TMPPDA, @QTDPPDA, @QTDPRO1, @QTDPROF
WHILE (@@FETCH_STATUS = 0 ) BEGIN
BEGIN
INSERT INTO
DSA.dbo.tmpFatoProd_H6 (BIPROC, BIFORM, BISEG, BICLASS, BICOMP, PRODUTO, CTRAB, TURNO, OPERADO, RECURSO,
DATA, OP, QTDPRO, TMPPRO, TMPPDA, QTDPPDA, QTDPRO1, QTDPROF)

values (@BIPROC, @BIFORM, @BISEG, @BICLASS, @BICOMP, @PRODUTO, @CTRAB, @TURNO, @OPERADO, @RECURSO,
@DATA, @OP, @QTDPRO, @TMPPRO, @TMPPDA, @QTDPPDA, @QTDPRO1, @QTDPROF)
END
FETCH NEXT FROM TNames_CURSOR INTO
@BIPROC, @BIFORM, @BISEG, @BICLASS, @BICOMP, @PRODUTO, @CTRAB, @TURNO, @OPERADO, @RECURSO, @DATA, @OP,
@QTDPRO, @TMPPRO, @TMPPDA, @QTDPPDA, @QTDPRO1, @QTDPROF
END

CLOSE TNames_CURSOR
DEALLOCATE TNames_CURSOR

```

```

/*****
* Cria um arquivo temporario sumariizando o de Producao *
*****/
SELECT
BIPROC,
BIFORM,
BISEG,
BICLASS,
BICOMP,
PRODUTO,
CTRAB,
TURNO,
OPERADO,
RECURSO,
DATA,
OP,
'QTDPRO'=SUM(QTDPRO),
'TMPPRO'=SUM(TMPPRO),
'TMPPDA'=SUM(TMPPDA),
'QTDPA'=SUM(QTDPA),
'QTDPRO1'=SUM(QTDPRO1),
'QTDPROF'=SUM(QTDPROF)
INTO DSA.dbo.tmp1FatoProd
FROM DSA.dbo.tmpFatoProd_H6
GROUP BY
BIPROC,
BIFORM,
BISEG,
BICLASS,
BICOMP,
PRODUTO,
CTRAB,
TURNO,
OPERADO,
RECURSO,
DATA,
OP

/*****
* corrige os dados em branco para valores genericos *
*****/
update DSA.dbo.tmp1FatoProd set BIPROC = 'ZZZZZ'
where LEN(LTRIM(RTRIM(BIPROC))) <= 0
OR NOT EXISTS ( SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '90' AND
X5_CHAVE = BIPROC )
update DSA.dbo.tmp1FatoProd set BIFORM = 'ZZZZZ'
where LEN(LTRIM(RTRIM(BIFORM))) <= 0
OR NOT EXISTS ( SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '89' AND
X5_CHAVE = BIFORM )
update DSA.dbo.tmp1FatoProd set BISEG = 'ZZZZZ'
where LEN(LTRIM(RTRIM(BISEG))) <= 0
OR NOT EXISTS ( SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '88' AND
X5_CHAVE = BISEG )
update DSA.dbo.tmp1FatoProd set BICLASS = 'ZZZZZ'
where LEN(LTRIM(RTRIM(BICLASS))) <= 0
OR NOT EXISTS ( SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '87' AND
X5_CHAVE = BICLASS )
update DSA.dbo.tmp1FatoProd set BICOMP = 'ZZZZZ'
where LEN(LTRIM(RTRIM(BICOMP))) <= 0
OR NOT EXISTS ( SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '86' AND
X5_CHAVE = BICOMP )
update DSA.dbo.tmp1FatoProd set PRODUTO = 'ZZZZZZZZZZZZZZ'
where LEN(LTRIM(RTRIM(PRODUTO))) <= 0
OR ( NOT EXISTS (SELECT B1_COD FROM Dadosadv.dbo.SB1010 WHERE B1_COD = PRODUTO and
D_E_L_E_T_ <> '*') AND PRODUTO <> 'ZZZZZZZZZZZZZZ')
update DSA.dbo.tmp1FatoProd set RECURSO = 'ZZZZZ'
where LEN(LTRIM(RTRIM(RECURSO))) <= 0
OR ( NOT EXISTS (SELECT H1_CODIGO FROM Dadosadv.dbo.SH1010 WHERE H1_CODIGO = RECURSO
and D_E_L_E_T_ <> '*') AND RECURSO <> 'ZZZZZ')
update DSA.dbo.tmp1FatoProd set CTRAB = 'ZZZZZ'
where LEN(LTRIM(RTRIM(CTRAB))) <= 0

```

```

OR ( NOT EXISTS (SELECT HB_COD FROM Dadosadv.dbo.SHB010 WHERE HB_COD = CTRAB and
D_E_L_E_T_ <> '*') AND CTRAB <> 'ZZZZZ')
update DSA.dbo.tmp1FatoProd set TURNO = 'Z'
where LEN(LTRIM(RTRIM(TURNO))) <= 0
OR ( NOT EXISTS (SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '98' AND
X5_CHAVE = TURNO and D_E_L_E_T_ <> '*') AND TURNO <> 'Z')
update DSA.dbo.tmp1FatoProd set OPERADO = 'ZZZ'
where LEN(LTRIM(RTRIM(OPERADO))) <= 0
OR ( NOT EXISTS (SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '97' AND
X5_CHAVE = OPERADO and D_E_L_E_T_ <> '*') AND OPERADO <> 'ZZZ')

/*****
* Cria indice da tabela temporaria *
*****/

CREATE INDEX tmp1FatoProd1 ON DSA.dbo.tmp1FatoProd (PRODUTO)
CREATE INDEX tmp1FatoProd2 ON DSA.dbo.tmp1FatoProd (RECURSO)
CREATE INDEX tmp1FatoProd3 ON DSA.dbo.tmp1FatoProd (CTRAB)
CREATE INDEX tmp1FatoProd4 ON DSA.dbo.tmp1FatoProd (DATA)
CREATE INDEX tmp1FatoProd5 ON DSA.dbo.tmp1FatoProd (TURNO)
CREATE INDEX tmp1FatoProd6 ON DSA.dbo.tmp1FatoProd (OPERADO)
CREATE INDEX tmp1FatoProd7 ON DSA.dbo.tmp1FatoProd (BIPROC)
CREATE INDEX tmp1FatoProd8 ON DSA.dbo.tmp1FatoProd (BIFORM)
CREATE INDEX tmp1FatoProd9 ON DSA.dbo.tmp1FatoProd (BISEG)
CREATE INDEX tmp1FatoProdA ON DSA.dbo.tmp1FatoProd (BICLASS)
CREATE INDEX tmp1FatoProdB ON DSA.dbo.tmp1FatoProd (BICOMP)

/*****
* Cria o arquivo temporario ja vinculado às dimensões *
*****/

Select
DIMPROCESSO.IDPROCES,
DIMFORMATO.IDFRMT,
DIMSEGMENTO.IDSEG,
DIMCLASSE.IDCLASSE,
DIMCOMPOSICAO.IDCOMP,
DIMPRODUTO.IDPRD,
DIMRECURSO.IDRCS,
DIMSETOR.IDSET,
DIMDATA.IDDTA,
DIMENTURNO.IDTRN,
DIMOPER.IDOPR,
'OPPRO'=OP,
'QTDPROPRO'=SUM(QTDPRO),
'TMPPROPRO'=SUM(TMPPRO),
'TMPPDAPRO'=SUM(TMPPDA),
'QTDPDAPRO'=SUM(QTDPDA),
'QTDPRO1PRO'=SUM(QTDPRO1),
'QTDPROFPRO'=SUM(QTDPROF)
INTO DSA.dbo.tmp2FatoProd
from DSA.dbo.tmp1FatoProd
left join DSA.dbo.DIMPROCESSO DIMPROCESSO ON BIPROC = IDNPROCES
left join DSA.dbo.DIMFORMATO DIMFORMATO ON BIFORM = IDNFRMT
left join DSA.dbo.DIMSEGMENTO DIMSEGMENTO ON BISEG = IDNSEG
left join DSA.dbo.DIMCLASSE DIMCLASSE ON BICLASS = IDNCLASSE
left join DSA.dbo.DIMCOMPOSICAO DIMCOMPOSICAO ON BICOMP = IDNCOMP
left join DSA.dbo.DIMPRODUTO DIMPRODUTO ON PRODUTO = IDNPRD
left join DSA.dbo.DIMRECURSO DIMRECURSO ON RECURSO = IDNRCS
left join DSA.dbo.DIMSETOR DIMSETOR ON CTRAB = IDNSET
left join DSA.dbo.DIMDATA DIMDATA ON DATA = IDDTA
left join DSA.dbo.DIMENTURNO DIMTURNO ON TURNO = IDNTRN
left join DSA.dbo.DIMOPER DIMOPER ON OPERADO = IDNOPR
group by
DIMPROCESSO.IDPROCES,
DIMFORMATO.IDFRMT,
DIMSEGMENTO.IDSEG,
DIMCLASSE.IDCLASSE,
DIMCOMPOSICAO.IDCOMP,
DIMPRODUTO.IDPRD,
DIMRECURSO.IDRCS,
DIMSETOR.IDSET,
DIMDATA.IDDTA,

```

```

DIMTURNO.IDTRN,
DIMOPER.IDOPR,
OP

/*****
* ATUALIZA FATO *
*****/
DECLARE @IDPROCES int
DECLARE @IDFRMT int
DECLARE @IDSEG int
DECLARE @IDCLASSE int
DECLARE @IDCOMP int
DECLARE @IDPRD int
DECLARE @IDRCS int
DECLARE @IDSET int
DECLARE @IDDTA varchar(8)
DECLARE @IDTRN int
DECLARE @IDOPR int
DECLARE @OPPRO varchar(11)
DECLARE @QTDPROPRO float
DECLARE @TMPPROPRO float
DECLARE @TMPPDAPRO float
DECLARE @QTDPDAPRO float
DECLARE @QTDPRO1PRO float
DECLARE @QTDPROFPRO float

DECLARE TNames_CURSOR CURSOR
FOR SELECT
IDPROCES, IDFRMT, IDSEG, IDCLASSE, IDCOMP, IDPRD, IDRCS, IDSET, IDDTA, IDTRN, IDOPR, OPPRO, QTDPROPRO,
TMPPROPRO, TMPPDAPRO, QTDPDAPRO, QTDPRO1PRO, QTDPROFPRO
FROM DSA.dbo.tmp2FatoProd
OPEN TNames_CURSOR
FETCH NEXT FROM TNames_CURSOR INTO
@IDPROCES, @IDFRMT, @IDSEG, @IDCLASSE, @IDCOMP, @IDPRD, @IDRCS, @IDSET, @IDDTA, @IDTRN, @IDOPR, @OPPRO,
@QTDPROPRO, @TMPPROPRO, @TMPPDAPRO, @QTDPDAPRO, @QTDPRO1PRO, @QTDPROFPRO
WHILE (@@FETCH_STATUS = 0 ) BEGIN

    IF EXISTS ( SELECT * FROM DSA.dbo.FTPRODUCAO
WHERE IDPROCES = @IDPROCES AND IDFRMT = @IDFRMT AND IDSEG = @IDSEG
AND IDCLASSE = @IDCLASSE AND IDCOMP = @IDCOMP AND IDPRD = @IDPRD
AND IDRCS = @IDRCS AND IDSET = @IDSET AND IDDTA = @IDDTA
AND IDTRN = @IDTRN AND IDOPR = @IDOPR AND OPPRO = @OPPRO)
BEGIN
UPDATE DSA.dbo.FTPRODUCAO
SET QTDPROPRO = @QTDPROPRO,
TMPPROPRO = @TMPPROPRO,
TMPPDAPRO = @TMPPDAPRO,
QTDPDAPRO = @QTDPDAPRO,
QTDPRO1PRO= @QTDPRO1PRO,
QTDPROFPRO= @QTDPROFPRO

WHERE IDPROCES = @IDPROCES AND IDFRMT = @IDFRMT AND IDSEG = @IDSEG
AND IDCLASSE = @IDCLASSE AND IDCOMP = @IDCOMP AND IDPRD = @IDPRD
AND IDRCS = @IDRCS AND IDSET = @IDSET AND IDDTA = @IDDTA
AND IDTRN = @IDTRN AND IDOPR = @IDOPR AND OPPRO = @OPPRO
END
ELSE
BEGIN

INSERT INTO DSA.dbo.FTPRODUCAO (IDPROCES, IDFRMT, IDSEG, IDCLASSE, IDCOMP,
IDPRD, IDRCS, IDSET, IDDTA, IDTRN, IDOPR, OPPRO, QTDPROPRO,
TMPPROPRO, TMPPDAPRO, QTDPDAPRO, QTDPRO1PRO, QTDPROFPRO)

values (@IDPROCES, @IDFRMT, @IDSEG, @IDCLASSE, @IDCOMP, @IDPRD, @IDRCS, @IDSET,
@IDDTA, @IDTRN, @IDOPR, @OPPRO, @QTDPROPRO, @TMPPROPRO, @TMPPDAPRO, @QTDPDAPRO,
@QTDPRO1PRO, @QTDPROFPRO)

END

FETCH NEXT FROM TNames_CURSOR INTO
@IDPROCES, @IDFRMT, @IDSEG, @IDCLASSE, @IDCOMP, @IDPRD, @IDRCS, @IDSET, @IDDTA, @IDTRN, @IDOPR, @OPPRO,
@QTDPROPRO, @TMPPROPRO, @TMPPDAPRO, @QTDPDAPRO, @QTDPRO1PRO, @QTDPROFPRO
END

```

```

CLOSE TNAMES_CURSOR
DEALLOCATE TNAMES_CURSOR

DROP TABLE DSA.dbo.tmpFatoProd_H6
DROP TABLE DSA.dbo.tmpFatoProd_Z3
DROP TABLE DSA.dbo.tmpFatoProd_Z4
DROP TABLE DSA.dbo.tmp1FatoProd
DROP TABLE DSA.dbo.tmp2FatoProd

go

/*****
*   Fato FTTMPPRDA
*   Tempo de Parada de Producao
*****/

if exists (select * from sysobjects where id = object_id(N'[dbo].[USP_FTTMPPRDA]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
    DROP PROC [dbo].[USP_FTTMPPRDA]
GO

CREATE PROC USP_FTTMPPRDA
AS

SELECT
'B1_BIPROC'=ISNULL(B1_BIPROC,'ZZZZZ'),
'B1_BIFORM'=ISNULL(B1_BIFORM,'ZZZZZ'),
'B1_BISEG'=ISNULL(B1_BISEG , 'ZZZZZ'),
'B1_BICLASS'=ISNULL(B1_BICLASS,'ZZZZZ'),
'B1_BICOMP'=ISNULL(B1_BICOMP,'ZZZZZ'),
'B1_PRODUTO'= B1_COD,
'Z4_MOTIVO'=isnull(Z4_MOTIVO,'ZZZZZ') ,
'Z4_RECURSO'=isnull(Z4_RECURSO,'ZZZZZ'),
'H1_CTRAB'=isnull(H1_CTRAB,'ZZZZZ') ,
'Z4_DATA' = Z4_DATA,
'Z4_TURNO' =isnull(Z4_TURNO,'ZZZZZ'),
'Z4_OPERADO'=isnull(Z4_OPERADO,'ZZZZZ'),
Z4_OP,
'Z4_HORAS' = (FLOOR(Z4_HORAS)*60)+FLOOR((Z4_HORAS-FLOOR(Z4_HORAS))*100)
INTO DSA.dbo.tmp1FatoTmpParada
FROM Dadosadv.dbo.SZ4010 SZ4
left JOIN Dadosadv.dbo.SB1010 SB1 ON ' ' = B1_FILIAL AND Z4_PRODUTO = B1_COD
left JOIN Dadosadv.dbo.SH1010 SH1 ON '01' = H1_FILIAL AND Z4_RECURSO = H1_CODIGO
where SZ4.D_E_L_E_T_ = ' '
AND SB1.D_E_L_E_T_ = ' '
AND SH1.D_E_L_E_T_ = ' '

update DSA.dbo.tmp1FatoTmpParada set B1_BIPROC = 'ZZZZZ'
where LEN(LTRIM(RTRIM(B1_BIPROC))) <= 0
OR NOT EXISTS ( SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '90' AND
X5_CHAVE = B1_BIPROC )
update DSA.dbo.tmp1FatoTmpParada set B1_BIFORM = 'ZZZZZ'
where LEN(LTRIM(RTRIM(B1_BIFORM))) <= 0
OR NOT EXISTS ( SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '89' AND
X5_CHAVE = B1_BIFORM )
update DSA.dbo.tmp1FatoTmpParada set B1_BISEG = 'ZZZZZ'
where LEN(LTRIM(RTRIM(B1_BISEG))) <= 0
OR NOT EXISTS ( SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '88' AND
X5_CHAVE = B1_BISEG )
update DSA.dbo.tmp1FatoTmpParada set B1_BICLASS = 'ZZZZZ'
where LEN(LTRIM(RTRIM(B1_BICLASS))) <= 0
OR NOT EXISTS ( SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '87' AND
X5_CHAVE = B1_BICLASS )
update DSA.dbo.tmp1FatoTmpParada set B1_BICOMP = 'ZZZZZ'
where LEN(LTRIM(RTRIM(B1_BICOMP))) <= 0
OR NOT EXISTS ( SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '86' AND
X5_CHAVE = B1_BICOMP )
update DSA.dbo.tmp1FatoTmpParada set B1_PRODUTO = 'ZZZZZZZZZZZZZZ'
where LEN(LTRIM(RTRIM(B1_PRODUTO))) <= 0
OR ( NOT EXISTS (SELECT B1_COD FROM Dadosadv.dbo.SB1010 WHERE B1_COD = B1_PRODUTO and

```

```

D_E_L_E_T_ <> '*' ) AND B1_PRODUTO <> 'ZZZZZZZZZZZZZZ')
update DSA.dbo.tmp1FatoTmpParada set Z4_MOTIVO = 'ZZZ'
  where LEN(LTRIM(RTRIM(Z4_MOTIVO))) <= 0
  OR ( NOT EXISTS (SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '44' AND
X5_CHAVE = Z4_MOTIVO and D_E_L_E_T_ <> '*' ) AND Z4_MOTIVO <> 'ZZZ')
update DSA.dbo.tmp1FatoTmpParada set Z4_RECURSO = 'ZZZZZZ'
  where LEN(LTRIM(RTRIM(Z4_RECURSO))) <= 0
  OR ( NOT EXISTS (SELECT H1_CODIGO FROM Dadosadv.dbo.SH1010 WHERE H1_CODIGO = Z4_RECURSO
and D_E_L_E_T_ <> '*' ) AND Z4_RECURSO <> 'ZZZZZZ')
update DSA.dbo.tmp1FatoTmpParada set H1_CTRAB = 'ZZZZZZ'
  where LEN(LTRIM(RTRIM(H1_CTRAB))) <= 0
  OR ( NOT EXISTS (SELECT HB_COD FROM Dadosadv.dbo.SHB010 WHERE HB_COD = H1_CTRAB and
D_E_L_E_T_ <> '*' ) AND H1_CTRAB <> 'ZZZZZZ')
update DSA.dbo.tmp1FatoTmpParada set Z4_TURNO = 'Z'
  where LEN(LTRIM(RTRIM(Z4_TURNO))) <= 0
  OR ( NOT EXISTS (SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '98' AND
X5_CHAVE = Z4_TURNO and D_E_L_E_T_ <> '*' ) AND Z4_TURNO <> 'Z')
update DSA.dbo.tmp1FatoTmpParada set Z4_OPERADO = 'ZZZ'
  where LEN(LTRIM(RTRIM(Z4_OPERADO))) <= 0
  OR ( NOT EXISTS (SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '97' AND
X5_CHAVE = Z4_OPERADO and D_E_L_E_T_ <> '*' ) AND Z4_OPERADO <> 'ZZZ')

CREATE INDEX tmp1FatoTmpParada1 ON DSA.dbo.tmp1FatoTmpParada (B1_PRODUTO)
CREATE INDEX tmp1FatoTmpParada2 ON DSA.dbo.tmp1FatoTmpParada (Z4_MOTIVO)
CREATE INDEX tmp1FatoTmpParada3 ON DSA.dbo.tmp1FatoTmpParada (Z4_RECURSO)
CREATE INDEX tmp1FatoTmpParada4 ON DSA.dbo.tmp1FatoTmpParada (H1_CTRAB)
CREATE INDEX tmp1FatoTmpParada5 ON DSA.dbo.tmp1FatoTmpParada (Z4_DATA)
CREATE INDEX tmp1FatoTmpParada6 ON DSA.dbo.tmp1FatoTmpParada (Z4_TURNO)
CREATE INDEX tmp1FatoTmpParada7 ON DSA.dbo.tmp1FatoTmpParada (Z4_OPERADO)
CREATE INDEX tmp1FatoTmpParada8 ON DSA.dbo.tmp1FatoTmpParada (B1_BIPROC)
CREATE INDEX tmp1FatoTmpParada9 ON DSA.dbo.tmp1FatoTmpParada (B1_BIFORM)
CREATE INDEX tmp1FatoTmpParadaA ON DSA.dbo.tmp1FatoTmpParada (B1_BISEG)
CREATE INDEX tmp1FatoTmpParadaB ON DSA.dbo.tmp1FatoTmpParada (B1_BICLASS)
CREATE INDEX tmp1FatoTmpParadaC ON DSA.dbo.tmp1FatoTmpParada (B1_BICOMP)

Select
DIMPROCESSO.IDPROCES,
DIMFORMATO.IDFRMT,
DIMSEGMENTO.IDSEG,
DIMCLASSE.IDCLASSE,
DIMCOMPOSICAO.IDCOMP,
DIMPRODUTO.IDPRD,
DIMTPPARAD.IDTPP,
DIMRECURSO.IDRCS,
DIMSETOR.IDSET,
DIMDATA.IDDTA,
DIMTURNO.IDTRN,
DIMOPER.IDOPR,
'OPTMP'=Z4_OP,
'QTDTMP'=SUM(Z4_HORAS)
INTO DSA.dbo.tmp2FatoTmpParada
from DSA.dbo.tmp1FatoTmpParada

left join DSA.dbo.DIMPROCESSO DIMPROCESSO ON B1_BIPROC = IDNPROCES
left join DSA.dbo.DIMFORMATO DIMFORMATO ON B1_BIFORM = IDNFRMT
left join DSA.dbo.DIMSEGMENTO DIMSEGMENTO ON B1_BISEG = IDNSEG
left join DSA.dbo.DIMCLASSE DIMCLASSE ON B1_BICLASS = IDNCLASSE
left join DSA.dbo.DIMCOMPOSICAO DIMCOMPOSICAO ON B1_BICOMP = IDNCOMP
left join DSA.dbo.DIMPRODUTO DIMPRODUTO ON B1_PRODUTO = IDNPRD
left join DSA.dbo.DIMTPPARAD DIMTPPARAD ON Z4_MOTIVO = IDNTPP
left join DSA.dbo.DIMRECURSO DIMRECURSO ON Z4_RECURSO = IDNRCS
left join DSA.dbo.DIMSETOR DIMSETOR ON H1_CTRAB = IDNSET
left join DSA.dbo.DIMDATA DIMDATA ON Z4_DATA = IDDTA
left join DSA.dbo.DIMTURNO DIMTURNO ON Z4_TURNO = IDNTRN
left join DSA.dbo.DIMOPER DIMOPER ON Z4_OPERADO = IDNOPR
group by
DIMPROCESSO.IDPROCES,
DIMFORMATO.IDFRMT,
DIMSEGMENTO.IDSEG,
DIMCLASSE.IDCLASSE,
DIMCOMPOSICAO.IDCOMP,
DIMPRODUTO.IDPRD,
DIMTPPARAD.IDTPP,

```

```

DIMRECURSO.IDRCS,
DIMSETOR.IDSET,
DIMDATA.IDDTA,
DIMITURNO.IDTRN,
DIMOPER.IDOPR,
Z4_OP

DECLARE @IDRCS      int
DECLARE @IDPRD      int
DECLARE @IDTPP      int
DECLARE @IDCOMP     int
DECLARE @IDCLASSE  int
DECLARE @IDSEG      int
DECLARE @IDFRMT     int
DECLARE @IDPROCES  int
DECLARE @IDDTA      varchar(8)
DECLARE @IDSET      int
DECLARE @IDTRN      int
DECLARE @IDOPR      int
DECLARE @OPTMP      char(15)
DECLARE @CUSTOTMP   money
DECLARE @QDTTMP     float

DECLARE TNames_CURSOR CURSOR
FOR SELECT
IDRCS, IDPRD, IDTPP, IDCOMP, IDCLASSE, IDSEG, IDFRMT, IDPROCES, IDDTA, IDSET, IDTRN, IDOPR, OPTMP, QDTTMP
FROM DSA.dbo.tmp2FatoTmpParada
OPEN TNames_CURSOR
FETCH NEXT FROM TNames_CURSOR INTO
@IDRCS, @IDPRD, @IDTPP, @IDCOMP, @IDCLASSE, @IDSEG, @IDFRMT, @IDPROCES, @IDDTA, @IDSET, @IDTRN, @IDOPR,
@OPTMP, @QDTTMP
WHILE (@@FETCH_STATUS = 0 ) BEGIN

    IF EXISTS ( SELECT * FROM DSA.dbo.FTTMPPRDA
                WHERE IDRCS = @IDRCS AND IDPRD = @IDPRD AND IDTPP = @IDTPP
                AND IDCOMP = @IDCOMP AND IDCLASSE = @IDCLASSE AND IDSEG = @IDSEG
                AND IDFRMT = @IDFRMT AND IDPROCES = @IDPROCES AND IDDTA = @IDDTA
                AND IDSET = @IDSET AND IDTRN = @IDTRN AND IDOPR = @IDOPR
                AND OPTMP = @OPTMP )
        BEGIN
            UPDATE DSA.dbo.FTTMPPRDA
            SET QDTTMP = @QDTTMP
            WHERE IDRCS = @IDRCS AND IDPRD = @IDPRD AND IDTPP = @IDTPP
            AND IDCOMP = @IDCOMP AND IDCLASSE = @IDCLASSE AND IDSEG = @IDSEG
            AND IDFRMT = @IDFRMT AND IDPROCES = @IDPROCES AND IDDTA = @IDDTA
            AND IDSET = @IDSET AND IDTRN = @IDTRN AND IDOPR = @IDOPR
            AND OPTMP = @OPTMP
        END
    ELSE
        BEGIN
            INSERT INTO DSA.dbo.FTTMPPRDA (IDRCS, IDPRD, IDTPP, IDCOMP, IDCLASSE,
            IDSEG, IDFRMT, IDPROCES, IDDTA, IDSET, IDTRN, IDOPR, OPTMP, QDTTMP)
            values (@IDRCS, @IDPRD, @IDTPP, @IDCOMP, @IDCLASSE, @IDSEG, @IDFRMT, @IDPROCES,
            @IDDTA, @IDSET, @IDTRN, @IDOPR, @OPTMP, @QDTTMP)
        END

    FETCH NEXT FROM TNames_CURSOR INTO
    @IDRCS, @IDPRD, @IDTPP, @IDCOMP, @IDCLASSE, @IDSEG, @IDFRMT, @IDPROCES,
    @IDDTA, @IDSET, @IDTRN, @IDOPR, @OPTMP, @QDTTMP
END

CLOSE TNames_CURSOR
DEALLOCATE TNames_CURSOR

drop table DSA.dbo.tmp1FatoTmpParada
drop table DSA.dbo.tmp2FatoTmpParada

GO

/*****

```

```

* Fato FTPERDA *
* Perda de Producao *
*****/

if exists (select * from sysobjects where id = object_id(N'[dbo].[USP_FTPERDA]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1)
    DROP PROC [dbo].[USP_FTPERDA]
GO

CREATE PROC USP_FTPERDA
AS

SELECT
'B1_BIPROC'=ISNULL(B1_BIPROC,'ZZZZZ'),
'B1_BIFORM'=ISNULL(B1_BIFORM,'ZZZZZ'),
'B1_BISEG'=ISNULL(B1_BISEG , 'ZZZZZ'),
'B1_BICLASS'=ISNULL(B1_BICLASS,'ZZZZZ'),
'B1_BICOMP'=ISNULL(B1_BICOMP,'ZZZZZ'),
'B1_PRODUTO'= B1_COD,
'Z3_MOTIVO'=isnull(Z3_MOTIVO,'ZZZZZ'),
'Z3_RECURSO'=isnull(Z3_RECURSO,'ZZZZZ'),
'H1_CTRAB'=isnull(H1_CTRAB,'ZZZZZ'),
'Z3_DATA' = Z3_DATA,
'Z3_TURNO' =isnull(Z3_TURNO,'ZZZZZ'),
'Z3_OPERADO'=isnull(Z3_OPERADO,'ZZZZZ'),
Z3_OP,
Z3_QUANTID
INTO DSA.dbo.tmp1FatoPerda
FROM Dadosadv.dbo.SZ3010 SZ3
left JOIN Dadosadv.dbo.SB1010 SB1 ON ' ' = B1_FILIAL AND Z3_PRODUTO = B1_COD
left JOIN Dadosadv.dbo.SH1010 SH1 ON '01' = H1_FILIAL AND Z3_RECURSO = H1_CODIGO
where SZ3.D_E_L_E_T_ = ' '
AND SB1.D_E_L_E_T_ = ' '
AND SH1.D_E_L_E_T_ = ' '

update DSA.dbo.tmp1FatoPerda set B1_BIPROC = 'ZZZZZ'
    where LEN(LTRIM(RTRIM(B1_BIPROC))) <= 0
    OR NOT EXISTS ( SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '90' AND
X5_CHAVE = B1_BIPROC )
update DSA.dbo.tmp1FatoPerda set B1_BIFORM = 'ZZZZZ'
    where LEN(LTRIM(RTRIM(B1_BIFORM))) <= 0
    OR NOT EXISTS ( SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '89' AND
X5_CHAVE = B1_BIFORM )
update DSA.dbo.tmp1FatoPerda set B1_BISEG = 'ZZZZZ'
    where LEN(LTRIM(RTRIM(B1_BISEG))) <= 0
    OR NOT EXISTS ( SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '88' AND
X5_CHAVE = B1_BISEG )
update DSA.dbo.tmp1FatoPerda set B1_BICLASS = 'ZZZZZ'
    where LEN(LTRIM(RTRIM(B1_BICLASS))) <= 0
    OR NOT EXISTS ( SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '87' AND
X5_CHAVE = B1_BICLASS )
update DSA.dbo.tmp1FatoPerda set B1_BICOMP = 'ZZZZZ'
    where LEN(LTRIM(RTRIM(B1_BICOMP))) <= 0
    OR NOT EXISTS ( SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '86' AND
X5_CHAVE = B1_BICOMP )
update DSA.dbo.tmp1FatoPerda set B1_PRODUTO = 'ZZZZZZZZZZZZZZ'
    where LEN(LTRIM(RTRIM(B1_PRODUTO))) <= 0
    OR ( NOT EXISTS (SELECT B1_COD FROM Dadosadv.dbo.SB1010 WHERE B1_COD = B1_PRODUTO and
D_E_L_E_T_ <> '*') AND B1_PRODUTO <> 'ZZZZZZZZZZZZZZ')
update DSA.dbo.tmp1FatoPerda set Z3_MOTIVO = 'ZZ'
    where LEN(LTRIM(RTRIM(Z3_MOTIVO))) <= 0
    OR ( NOT EXISTS (SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '43' AND
X5_CHAVE = Z3_MOTIVO and D_E_L_E_T_ <> '*') AND Z3_MOTIVO <> 'ZZ')
update DSA.dbo.tmp1FatoPerda set Z3_RECURSO = 'ZZZZZ'
    where LEN(LTRIM(RTRIM(Z3_RECURSO))) <= 0
    OR ( NOT EXISTS (SELECT H1_CODIGO FROM Dadosadv.dbo.SH1010 WHERE H1_CODIGO = Z3_RECURSO
and D_E_L_E_T_ <> '*') AND Z3_RECURSO <> 'ZZZZZ')
update DSA.dbo.tmp1FatoPerda set H1_CTRAB = 'ZZZZZ'
    where LEN(LTRIM(RTRIM(H1_CTRAB))) <= 0
    OR ( NOT EXISTS (SELECT HB_COD FROM Dadosadv.dbo.SHB010 WHERE HB_COD = H1_CTRAB and
D_E_L_E_T_ <> '*') AND H1_CTRAB <> 'ZZZZZ')
update DSA.dbo.tmp1FatoPerda set Z3_TURNO = 'Z'

```

```

        where LEN(LTRIM(RTRIM(Z3_TURNO))) <= 0
        OR ( NOT EXISTS (SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '98' AND
X5_CHAVE = Z3_TURNO and D_E_L_E_T_ <> '*' )AND Z3_TURNO <> 'Z')
update DSA.dbo.tmp1FatoPerda set Z3_OPERADO = 'ZZZ'
        where LEN(LTRIM(RTRIM(Z3_OPERADO))) <= 0
        OR ( NOT EXISTS (SELECT X5_TABELA FROM Dadosadv.dbo.SX5010 WHERE X5_TABELA = '97' AND
X5_CHAVE = Z3_OPERADO and D_E_L_E_T_ <> '*' )AND Z3_OPERADO <> 'ZZZ')

CREATE INDEX tmp1FatoPerda1 ON DSA.dbo.tmp1FatoPerda (B1_PRODUTO)
CREATE INDEX tmp1FatoPerda2 ON DSA.dbo.tmp1FatoPerda (Z3_MOTIVO)
CREATE INDEX tmp1FatoPerda3 ON DSA.dbo.tmp1FatoPerda (Z3_RECURSO)
CREATE INDEX tmp1FatoPerda4 ON DSA.dbo.tmp1FatoPerda (H1_CTRAB)
CREATE INDEX tmp1FatoPerda5 ON DSA.dbo.tmp1FatoPerda (Z3_DATA)
CREATE INDEX tmp1FatoPerda6 ON DSA.dbo.tmp1FatoPerda (Z3_TURNO)
CREATE INDEX tmp1FatoPerda7 ON DSA.dbo.tmp1FatoPerda (Z3_OPERADO)
CREATE INDEX tmp1FatoPerda8 ON DSA.dbo.tmp1FatoPerda (B1_BIPROC)
CREATE INDEX tmp1FatoPerda9 ON DSA.dbo.tmp1FatoPerda (B1_BIFORM)
CREATE INDEX tmp1FatoPerdaA ON DSA.dbo.tmp1FatoPerda (B1_BISEG)
CREATE INDEX tmp1FatoPerdaB ON DSA.dbo.tmp1FatoPerda (B1_BICLASS)
CREATE INDEX tmp1FatoPerdaC ON DSA.dbo.tmp1FatoPerda (B1_BICOMP)

Select
DIMPROCESSO.IDPROCES,
DIMFORMATO.IDFRMT,
DIMSEGMENTO.IDSEG,
DIMCLASSE.IDCLASSE,
DIMCOMPOSICAO.IDCOMP,
DIMPRODUTO.IDPRD,
DIMITPPERDA.IDTPD,
DIMRECURSO.IDRCS,
DIMSETOR.IDSET,
DIMDATA.IDDTA,
DIMTURNO.IDTRN,
DIMOPER.IDOPR,
'OPPPDA'=Z3_OP,
'QTDPPDA'=SUM(Z3_QUANTID)
INTO DSA.dbo.tmp2FatoPerda
from DSA.dbo.tmp1FatoPerda

left join DSA.dbo.DIMPROCESSO DIMPROCESSO ON B1_BIPROC = IDNPROCES
left join DSA.dbo.DIMFORMATO DIMFORMATO ON B1_BIFORM = IDNFRMT
left join DSA.dbo.DIMSEGMENTO DIMSEGMENTO ON B1_BISEG = IDNSEG
left join DSA.dbo.DIMCLASSE DIMCLASSE ON B1_BICLASS = IDNCLASSE
left join DSA.dbo.DIMCOMPOSICAO DIMCOMPOSICAO ON B1_BICOMP = IDNCOMP
left join DSA.dbo.DIMPRODUTO DIMPRODUTO ON B1_PRODUTO = IDNPRD
left join DSA.dbo.DIMITPPERDA DIMITPPERDA ON Z3_MOTIVO = IDNTPD
left join DSA.dbo.DIMRECURSO DIMRECURSO ON Z3_RECURSO = IDNRCS
left join DSA.dbo.DIMSETOR DIMSETOR ON H1_CTRAB = IDNSET
left join DSA.dbo.DIMDATA DIMDATA ON Z3_DATA = IDDTA
left join DSA.dbo.DIMTURNO DIMTURNO ON Z3_TURNO = IDNTRN
left join DSA.dbo.DIMOPER DIMOPER ON Z3_OPERADO = IDNOPR
group by
DIMPROCESSO.IDPROCES,
DIMFORMATO.IDFRMT,
DIMSEGMENTO.IDSEG,
DIMCLASSE.IDCLASSE,
DIMCOMPOSICAO.IDCOMP,
DIMPRODUTO.IDPRD,
DIMITPPERDA.IDTPD,
DIMRECURSO.IDRCS,
DIMSETOR.IDSET,
DIMDATA.IDDTA,
DIMTURNO.IDTRN,
DIMOPER.IDOPR,
Z3_OP

DECLARE @IDPROCES int
DECLARE @IDFRMT int
DECLARE @IDSEG int
DECLARE @IDCLASSE int
DECLARE @IDCOMP int
DECLARE @IDPRD int
DECLARE @IDTPD int

```

```

DECLARE @IDRCS                int
DECLARE @IDSET                int
DECLARE @IDDTA                varchar(8)
DECLARE @IDTRN                int
DECLARE @IDOPR                int
DECLARE @OPPDA                varchar(15)
DECLARE @QTDPA                float

DECLARE TNAMES_CURSOR CURSOR
FOR SELECT
IDPROCES, IDFRMT, IDSEG, IDCLASSE, IDCOMP, IDPRD, IDTPD, IDRCS, IDSET, IDDTA, IDTRN, IDOPR, OPPDA, QTDPA
FROM DSA.dbo.tmp2FatoPerda
OPEN TNAMES_CURSOR
FETCH NEXT FROM TNAMES_CURSOR INTO
@IDPROCES, @IDFRMT, @IDSEG, @IDCLASSE, @IDCOMP, @IDPRD, @IDTPD, @IDRCS, @IDSET, @IDDTA, @IDTRN, @IDOPR,
@OPPDA, @QTDPA
WHILE (@@FETCH_STATUS = 0 ) BEGIN

    IF EXISTS ( SELECT * FROM DSA.dbo.FTPERDA
                WHERE IDPROCES = @IDPROCES AND IDFRMT = @IDFRMT AND IDSEG = @IDSEG
                  AND IDCLASSE = @IDCLASSE AND IDCOMP = @IDCOMP AND IDPRD = @IDPRD
                  AND IDTPD = @IDTPD AND IDRCS = @IDRCS AND IDSET = @IDSET
                  AND IDDTA = @IDDTA AND IDTRN = @IDTRN AND IDOPR = @IDOPR
                  AND OPPDA = @OPPDA )
        BEGIN
            UPDATE DSA.dbo.FTPERDA
            SET QTDPA = @QTDPA
            WHERE IDPROCES = @IDPROCES AND IDFRMT = @IDFRMT AND IDSEG = @IDSEG
            AND IDCLASSE = @IDCLASSE AND IDCOMP = @IDCOMP AND IDPRD = @IDPRD
            AND IDTPD = @IDTPD AND IDRCS = @IDRCS AND IDSET = @IDSET
            AND IDDTA = @IDDTA AND IDTRN = @IDTRN AND IDOPR = @IDOPR
            AND OPPDA = @OPPDA
        END
    ELSE
        BEGIN
            INSERT INTO DSA.dbo.FTPERDA (IDPROCES, IDFRMT, IDSEG, IDCLASSE, IDCOMP,
                                         IDPRD, IDTPD, IDRCS, IDSET, IDDTA, IDTRN, IDOPR, OPPDA, QTDPA)
            values (@IDPROCES, @IDFRMT, @IDSEG, @IDCLASSE, @IDCOMP, @IDPRD,
                  @IDTPD, @IDRCS, @IDSET, @IDDTA, @IDTRN, @IDOPR, @OPPDA, @QTDPA)
        END
    END

    FETCH NEXT FROM TNAMES_CURSOR INTO
    @IDPROCES, @IDFRMT, @IDSEG, @IDCLASSE, @IDCOMP, @IDPRD, @IDTPD, @IDRCS,
    @IDSET, @IDDTA, @IDTRN, @IDOPR, @OPPDA, @QTDPA
END

CLOSE TNAMES_CURSOR
DEALLOCATE TNAMES_CURSOR

drop table DSA.dbo.tmp1FatoPerda
drop table DSA.dbo.tmp2FatoPerda

GO

```

Apêndice D – Lista de Abreviaturas

BD	Banco de dados
BI	Business Intelligence
DW	Data Warehouse
DM	Data Mart
DSS	Decision Support System
EIS	Executive Information System
ERP	Enterprise Resource Planning
ESS	Executive Support Systems
ETL	Extract, Transform and Load (Extração, Transformação e Carga)
MD	Mineração de Dados (Data Mining)
OLAP	On-Line Analytical Processing (Processamento analítico on-line)
OLTP	On-Line Transaction Processing (Processamento transacional on-line)
SAD	Sistemas de Apoio à Decisão
SAG	Sistemas de Apoio Gerencial
SIE	Sistemas de Informações Executivas
SIG	Sistemas de Informação Gerenciais
SPT	Sistemas de Processamento Transacional
SI	Sistemas de Informação
SIE	Sistemas de Informação Executiva
SIG	Sistemas de Informação Gerencial
TI	Tecnologia da Informação
UML	Unified Modeling Language (Linguagem de Modelagem Unificada)
XML	Extensible Markup Language - Linguagem de Marcação Expansível.
Web	World Wide Web