Fábio Eiji Yamasaki
Marcelo Massayoshi Takizawa
Marcelo Toshio Uenoyama

Soluções de *Business Intelligence* baseadas em Software Livre

Projeto de Formatura apresentado à
Disciplina PCS 2502 – Projeto de Formatura II, da
Escola Politécnica da Universidade de São Paulo.

São Paulo
2005

Fábio Eiji Yamasaki
Marcelo Massayoshi Takizawa
Marcelo Toshio Uenoyama

Soluções de *Business Intelligence* baseadas em Software Livre

Projeto de Formatura apresentado à
Disciplina PCS 2502 – Projeto de Formatura II, da
Escola Politécnica da Universidade de São Paulo.

Área de Concentração:
Engenharia de Computação

Orientador:
Prof. Dr.
Pedro Luiz Pizzigatti Corrêa

São Paulo
2005

Autorização para reprodução

A todos que contribuíram direta ou indiretamente com o projeto.

Resumo

O presente trabalho tem como objetivo desenvolver uma solução *Business Intelligence* (BI) com base em aplicações *open source*. Esta arquitetura possibilita às empresas obter informações competitivas de mercado através do cruzamento das informações do dia-a-dia sobre os negócios, facilitando a obtenção de modelos de informação que auxiliam o processo de tomada de decisão. Além disso, a criação de relatórios é facilitada e a rapidez com que se obtém a informação do sistema é maior. As ferramentas de BI, através de suas ferramentas analíticas, auxiliam no gerenciamento do desempenho de indicadores estratégicos e táticos que resultam em uma melhora de produtividade pessoal no trabalho devido o maior nível de automação e informatização dos processos de negócios. Isso permite aos profissionais gastarem mais tempo planejando e analisando suas atividades ao invés de se preocupar em sua extração.

O fato da solução BI se utilizar de *software open source* apresenta vantagens de custo em relação às ferramentas de mercado, o que possibilita que pequenas e médias empresas tenham mais acesso a essa nova arquitetura, já que soluções de BI em geral são muito custosas.

Este projeto engloba o estudo de modelagem de dados para *data warehouses*, das ferramentas de OLAP para a análise das informações armazenadas, ETL para o povoamento do *data warehouse*, *Data Mining* para a elaboração de modelos e a organização de Meta Dados para suportar uma solução de BI.

Abstract

The current work has as an aim the development of a *Business Intelligence* (BI) solution based on open source applications. This architecture enables enterprises to get competitive market information by matching and organizing daily business information, that helps the attainment of information models that supports the decision process taking. Moreover, the reports creation is facilitated and the speed to get the information of the system is higher. The BI tools, through its analytical tools, assist in the management of the strategical performance and tactical score cards that result in a personal productivity improvement in the work due the biggest level of automation and computerization of the business-oriented processes. This allows the professionals to spend more time planning and analyzing their activities instead of being worried about data extraction.

The fact of a BI solution to use open source software has cost advantages in relation to the market tools, what makes possible that small and average companies to have more access to this new architecture, since BI solutions in general are very expensive.

This project covers the study of data modeling for data warehouses, OLAP tools for stored information analysis, ETL for data warehouse population, Data Mining for models elaboration and the Meta Data organization to support a BI solution.

SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

BI: *Business Intelligence*

OLAP: *Online Analytical Processing*

ROLAP: *Relational Online Analytical Processing*

MOLAP: *Multidimensional Online Analytical Processing*

ETL: *Extract, Transform and Load*

SGBD: *Sistema Gerenciador de Banco de Dados*

DASD: *Direct Access Storage Device*

DW : *Data Warehouse*

EIS: *Executive Information Systems*

DSS: *Decision Support System*

ERP: *Enterprise Resource Planning*

PENSA: *Programa de Estudos dos Negócios do Sistema Agroindustrial*

RI: *Referential Integrity*

DBA: *Data Base Administrator*

CASE: *Computer-Aided-Software Engineering*

LAN: *Local Area Network*

WAN: *Wide Area Network*

FIA: *Fundação Instituto de Administração*

CPM: *Corporate Performance Management*

API: *Application Programming Interface*

CSV: *Comma Separated Values*

1 INTRODUÇÃO

1.1 O que é Business Intelligence?

Dia a dia uma grande quantidade de dados é coletada pelas empresas. São informações sobre encomendas, inventários, contas a pagar, vendas em cada estabelecimento da rede, dados populacionais e, principalmente os consumidores de seus produtos. Mas as pesquisas feitas em 2000 (REINSCHMIDT, 2000) mostram que a maior parte dessas informações, em torno de 93%, não são úteis para o processo de tomada de decisão atualmente.

Consolidar, organizar e cruzar as informações para realizar uma gestão empresarial eficiente é uma necessidade para a liderança na vantagem competitiva e, aprendendo a descobrir e a alavancar essas vantagens são os objetivos de uma solução de *Business Intelligence*.

O conceito de *Business Intelligence* não é recente. Há milhares de anos atrás, povos do Oriente como os fenícios, persas, egípcios e outros, utilizavam esse princípio quando cruzavam informações obtidas junto à natureza em benefício próprio. Observar e analisar o comportamento das marés, os períodos de seca e de chuvas, a posição dos astros, entre outras, eram formas de obter informações que eram utilizadas para tomar as decisões que permitissem a melhoria de vida de suas respectivas comunidades (EGIDERIA, Web).

Hoje, a necessidade de cruzamento de informações é uma realidade tão verdadeira quanto fora no passado. Análises e projeções, de forma a agilizar processos às tomadas de decisões é o que *Business Intelligence*, termo criado por Howard Dresner, vice-presidente do grupo Gartner, visa a oferecer. Assim como ele, os norte-americanos ganharam fama pelo desenvolvimento das modernas ferramentas de *Business Intelligence* ou BI, como se costuma falar.

Business Intelligence é um conjunto de metodologias de gestão implementadas através de ferramentas de software, cuja função é proporcionar ganhos nos processos decisórios gerenciais e da alta administração nas organizações. Baseia-se na capacidade analítica das ferramentas que integram em um só lugar todas as informações necessárias ao processo decisório. BI transforma informação em conhecimento e, fornece a informação certa para o usuário certo na hora certa para suportar o processo de tomada de decisões (REINSCHMIDT, 2000).

1.2 Contexto Geral

Como já abordado anteriormente, o conceito BI é muito antigo, mas o desenvolvimento tecnológico ocorreu a partir da década de 70 e nos anos posteriores possibilitou a criação de ferramentas que vieram a facilitar todo o processo de captação, extração, armazenamento, filtragem, disponibilidade e personalização dos dados.

Entre 1992 e 1993 surgiu o *Data Warehouse* (ou DW) (DWBRASIL, Web) que é uma grande base de dados , ou seja, um repositório único de dados (os quais foram consolidados, limpos e uniformizados) considerado pelos especialistas no assunto como a peça essencial para a execução prática de um projeto de *Business Intelligence*. No entanto, quando se trata de BI, as opiniões nem sempre são unânimes. Na avaliação de alguns consultores é importante que a empresa que deseja implementar ferramentas de análise disponha de um repositório específico para reunir os dados já transformados em informações. Esse repositório não precisa ser, necessariamente, um *Data Warehouse*, mas algo menos complexo como, por exemplo, um *Data Mart* (banco de dados desenhado de forma personalizada para assuntos ou áreas específicas), ou um banco de dados relacional comum, mas separado do ambiente transacional (operacional) e dedicado a armazenar as informações usadas como base para a realização de diferentes análises e projeções.

Por volta do final de 1996, o conceito de BI começou a ser difundido como um processo de evolução do EIS – *Executive Information Systems* - um sistema criado no final da década 70, a partir dos trabalhos desenvolvidos pelos pesquisadores do MIT (*Massachusets Institute of Tecnology* - EUA). *Executive Information System* (EIS) é, na verdade, um software que objetiva fornecer informações empresariais a partir de uma base de dados. É uma ferramenta de consulta às bases de dados das funções empresariais para a apresentação de informações de forma simples e amigável, atendendo às necessidades, principalmente, dos executivos da alta administração. Permite os acompanhamentos diários de resultados, tabulando dados de todas as áreas funcionais da empresa para depois exibi-los de forma gráfica e simplificada, sendo de fácil compreensão para aqueles que não possuem profundos conhecimentos sobre tecnologia.

Em termos simples o EIS permite a esses profissionais o acesso amigável a uma série de informações por meio eletrônico, apresentadas de forma clara e visualmente atraente. A navegação é feita através do uso do mouse ou do sistema *touchscreen* (tela sensível ao toque) o que não requer habilidade, nem prática e nem necessidade de assistência. O principal objetivo do EIS é oferecer ao seu usuário, em curto espaço de tempo, uma visão gerencial da organização, mostrando como funcionam seus processos de trabalho e como ela se relaciona com o mundo externo dos negócios, clientes e fornecedores.

Com o passar dos anos o termo *Business Intelligence* ganhou maior abrangência, dentro de um processo natural de evolução, abarcando uma série de ferramentas, como o próprio EIS, e mais as soluções DSS (*Decision Support System* - sistema de suporte à decisão), Planilhas Eletrônicas, Geradores de Consultas e de Relatórios, *Data Marts*, *Data Mining*, Ferramentas OLAP (*Online Analytical Processing*), entre tantas outras (cada um destes módulos serão explicados no decorrer do trabalho), que têm como objetivo promover agilidade comercial, dinamizar a capacidade de tomar decisões e refinar estratégias de relacionamento com clientes, respondendo às necessidades do setor corporativo.

A história do *Business Intelligence* também está profundamente atrelada ao ERP (*Enterprise Resource Planning*) sigla que representa os sistemas integrados de gestão empresarial cuja função é facilitar o aspecto operacional das empresas. Esses sistemas registram, processam e documentam cada fato novo na engrenagem corporativa e distribuem a informação de maneira clara e segura, em tempo real.

Mas as empresas que implantaram esses sistemas logo perceberam que apenas armazenar grande quantidade de dados de nada valia se essas informações se encontravam repetidas, incompletas e espalhadas em vários sistemas dentro da corporação. Percebeu-se que era preciso dispor de ferramentas que permitissem reunir esses dados numa base única e trabalhá-los de forma a que possibilitassem realizar diferentes análises sob variados ângulos. Por essa razão, a maioria dos fornecedores de ERP passou a embutir em seus pacotes os módulos de BI, que cada vez mais estão se sofisticando.

Tradicionalmente, o *Business Intelligence* pertenceu ao domínio da área de TI e dos especialistas em pesquisa de mercado, responsáveis pela extração de dados, pela implantação de

processos e pela divulgação dos resultados aos executivos responsáveis pela tomada de decisões. Mas o crescimento da Internet mudou tudo. Se até então a aplicação deste conceito era a de levar informação a poucos empregados selecionados de uma empresa, para que fizessem uso em suas decisões, a Internet transformou esse cenário. Hoje, a rede permite disponibilizar soluções de BI para um número maior de pessoas.

A *Web* - e particularmente, o comércio eletrônico - também acelerou os negócios em todos os níveis. Some-se a isso o novo consumidor, que se apresenta de modo virtual, e para quem é preciso direcionar ações em razão de suas reais necessidades. Para saber quais são essas necessidades cada vez mais uma empresa precisa ter agilidade comercial, capacidade de tomar decisões e refinamento nas estratégias de clientes, tudo isso dentro do menor tempo possível.

Também nas empresas, atingir as metas passou a exigir um envolvimento corporativo maior e, ao mesmo tempo, a democratização da informação. Internamente o BI não mudou de mãos, mas ganhou mais mãos e, principalmente, mais cabeças pensantes e com acesso às informações. O *Business Intelligence* passou a ser encarado como uma aplicação estratégica integrada, estando disponível através de simples *desktops*, estações de trabalho e nos servidores da empresa.

Atualmente, corporações de pequeno, médio e grande porte necessitam do BI para auxiliá-las nas mais diferentes situações para a tomada de decisão, e ainda para otimizar o trabalho da organização, reduzir custos, eliminar a duplicação de tarefas, permitir previsões de crescimento da empresa como um todo e contribuir para a elaboração de estratégias. Não importa o porte da empresa, mas a necessidade do mercado. As maiorias dos analistas vêem a aplicabilidade eficiente de BI em todas as empresas, inclusive naquelas que apresentam faturamento reduzido, desde que analisado o fator custo x benefício. Para que um projeto de BI leve a empresa rumo ao melhor desempenho é preciso analisar muito bem alguns fatores: o quanto vai se gastar e o que se espera obter, ou seja, é preciso o alinhamento objetivo do projeto com os interesses e as estratégias da empresa.

Existem, ao redor do mundo, vários exemplos de implantação. No Brasil, soluções de *Business Intelligence* estão em bancos de varejo, em empresas de telecomunicações, seguradoras e em toda instituição que perceba a tendência da economia globalizada, em que a informação

precisa chegar de forma rápida, precisa e abundante porque a sobrevivência no mercado será medida pela capacidade de "gerar conhecimento". E somente quem fizer uma boa gestão do conhecimento irá fundamentar políticas e estratégias corporativas.

O retorno que se espera de um sistema de BI depende das prioridades de cada empresa. As ferramentas de BI continuam evoluindo porque o mercado possui enorme potencial de crescimento. A velocidade imposta pelos negócios na *Web* exige que se dê, a quem decide, disposição e autonomia para agir. (O grupo) Gartner, do mesmo Howard que deu nome ao BI, reconheceu que o início do século 21 trouxe uma mudança na visão da aplicabilidade dos softwares. O que se pode imaginar para o futuro é muito menos o que podemos chamar de ferramentas e muito mais o que o mercado competitivo necessita com urgência: soluções.

Muitas empresas já colhem os bons frutos possibilitados pelas soluções de BI. Há cerca de dois anos, a *General Motors* do Brasil (GM) padronizou sua infra-estrutura de análise de dados (MICROSTRATEGY, Web), seguindo diretrizes da corporação mundial, com plataforma de *Business Intelligence*. São atendidas pela solução as áreas de *Marketing* e Vendas, focadas no processo *Order to Delivery*, que reflete as informações desde um pedido até sua entrega ao consumidor; além das áreas de Manufatura , Finanças e Compras, responsáveis pela compra de materiais indiretos, previsão de vendas de veículos (*demand sensing*), análises de vendas *on-line* e análise da performance de processos internos ligados ao consumidor final. O uso da plataforma de BI permitiu, ainda, à GM, a troca de informações entre seus escritórios regionais em todo Brasil, além de ajudar a GM a entender melhor o perfil dos consumidores dos veículos da montadora .

Antes dessa opção, a GM já possuía vários processos e áreas dependentes de informações derivadas de diferentes negócios, para a tomada de decisão. Porém, na maioria das vezes, eram projetos elaborados manualmente, com diferentes sistemas e planilhas, que não interagiam entre si. "Havia a necessidade de se estabelecer uma estratégia de tecnologia para suportar as ações da empresa de maneira consistente e integrada. Foi, então, criada uma área específica denominada '*Executive Information Management*', com a missão de otimizar o potencial de uso da solução de BI", explica Hélio Avelino da Silva, gerente de tecnologia em planejamento estratégico e gerenciamento de informações da GM.

A implementação das soluções foi iniciada em um projeto para cerca de 20 pessoas. Hoje, conta com mais de 600 usuários, entre analistas, supervisores, coordenadores, gerentes e diretores. Trabalha com as soluções *MicroStrategy Intelligence Server*, OLAP *Server*, *Narrowcast*, *Web Analyst* e *Desktop Analyst*. A mais recente aquisição foi a plataforma de BI totalmente integrada e baseada na *Web*. A opção por uma solução completa de BI, segundo Silva, foi uma estratégia para que a empresa obtivesse informações competitivas de mercado. Graças a essa iniciativa, a GM do Brasil comemora o fato de ser mais rápido e mais fácil obter o cruzamento das informações de seu dia-a-dia sobre os negócios, para a obtenção de modelos de informação que auxiliam em muito o processo de tomada de decisão.

Outro benefício constatado é a facilidade na criação de relatórios. "Passamos a contar com mais rapidez na obtenção de qualquer informação do sistema e maior facilidade no cruzamento dos dados existentes, como por exemplo, filtros por região, tempo ou modelos dos veículos comercializados. Com ele, todo dia na parte da manhã, os executivos da empresa podem ler os relatórios eletrônicos para saber quanto foi vendido no dia anterior", acrescenta Silva. Para comportar todas essas informações, a GM possui vários bancos de dados de portes médio e grande que constituem *Data marts* especializados. "Temos como perfil, a adoção de tecnologias maduras, provindas de empresas que tenham infra-estrutura adequada e que nos ofereçam todo suporte e consultoria no país".

Outro exemplo é o da Vésper (VALIM, Web). Operadora local da Embratel, a Vésper implantou o sistema de gestão de processos de negócio da Fuego, que agilizou o atendimento de banda larga sem fio a usuários finais. Cerca de 70% dos usuários que adquirem o serviço Giro, segundo a operadora, são atendidos em 24 horas. O restante é atendido em até 48 horas. A ferramenta permite sincronismos operacionais, tornando a operadora capaz de gerenciar todas as atividades relacionadas ao negócio, inclusive os processos das empresas terceirizadas, com uma equipe bastante reduzida: quatro funcionários.

Nos primeiros três meses de operação do Giro, a solução já garantiu uma eficiência de 80% na entrega dos terminais. Em seis meses, o percentual chegou a 95%, proporcionando um grande diferencial competitivo. No início do projeto foram mapeadas 30 macro atividades relacionadas a operações. Com o sistema elas foram reduzidas para cinco, otimizando recursos e identificando onde poderiam ser feitas melhorias.

O mercado de BI está em alta, as vendas de licenças de ferramentas de BI no mundo cresceram 12% em 2004 (DAMALZO, 2005). Até 2009, o grupo Gartner estima que o crescimento anual médio será de 7,4% para o setor. Como justificativas de fatores que têm impulsionado este mercado estão a popularização e iniciativas de adoção de ferramentas voltadas à avaliação de dados de negócios, como CPM (*Corporate Performance Management*) e de comunicação corporativa em tempo real.

No Brasil, empresas como a MicroStrategy crescem mais, em torno de 25% em 2005 (ÂNGELO, F. K., 2005). Os bons resultados são atribuídos especialmente à crescente demanda do mercado de médias empresas por produtos de BI. "Além dos gastos do governo e do contínuo movimento de expansão das grandes empresas, esse segmento mostrou-se bastante interessado por BI este ano", afirma Flávio Bolieiro, *country manager* da empresa. "As médias empresas têm obtido resultados mais claros e efetivos com nossas ferramentas - redução de riscos de fraude e aumento nas vendas, por exemplo", explica. Segundo ele, os clientes de médio porte ainda representam apenas entre 10% e 15% da base da MicroStrategy.

Atualmente são as grandes corporações em sua maioria que usufruem dos benefícios da utilização de BI. Já as médias, atualmente em porcentagem menor, vêm aumentando a sua adesão. Sendo assim, os nichos de pequenas e médias empresas têm um potencial grande a ser explorado, que pode ser obtido por ferramentas *open source*.

1.3 Objetivo

O objetivo do projeto de formatura é desenvolver uma solução de *Business Intelligence* (BI) baseada em software livre Este último remete não a um produto ou sistema, mas sim a uma arquitetura, uma coleção de aplicações operacionais integradas bem como aplicações de suporte a decisões e banco de dados que provém à comunidade de negócios fácil acesso aos dados de negócio.

O projeto visa, por fim, prover uma solução de B.I. utilizando e integrando ferramentas de software livre com interface *Web* que consigam prover o usuário final uma análise de dados mais

sofisticada, fácil de usar, um recurso poderoso compartilhado, com boa relação custo x benefício e de escalabilidade às necessidades.

Para validar a solução proposta, utilizaremos um estudo de caso de um projeto da organização PENSA (Programa de Estudos dos Negócios do Sistema Agroindustrial – Faculdade de Administração e Economia da Universidade de São Paulo) sobre análises de produção, área plantada, importação, exportação e perdas da área de agronegócios.

1.4 Motivação

Atualmente existem várias soluções no mercado sobre *Business Intelligence* como por exemplo, as soluções distribuídas pela Cognos, Execplan, Ascential, Microstrategy, SAS Institute, IBM Brasil, Business Object, Hyperion, SSP, Extend Software, Microsoft, e Hummingbird, entre outras. Soma-se a esse grande universo, também dos módulos de BI oferecidos pelas empresas desenvolvedoras de sistemas de gestão empresarial (ERP – *Enterprise Resource Planning*), entre as quais se incluem a SAP, PeopleSoft , Datasul (NEXTG, Web), entre outras. No entanto, se tratam de soluções privadas e restritas que em geral custam muito caro e de acesso somente a grandes empresas.

Visto tal cenário, a motivação para o desenvolvimento de tal projeto provém do interesse de fornecer soluções de BI que utilizem ferramentas *open source* que sejam mais acessíveis para uso e aplicação, além da oportunidade de aquisição de *know how* deste conceito de aplicabilidade prática no mercado que promete ascensão.

1.5 Escopo da Solução

O projeto tem como escopo prover uma solução de *Business Intelligence* através de um Portal *Web* para o projeto de Agronegócios da organização PENSA. Neste portal, as ferramentas de BI estarão integradas de modo que facilite o seu uso pelo usuário.

Utilizando a metodologia, avaliando as limitações e potenciais das ferramentas de BI, o projeto estará abordando o desenvolvimento e configuração de uma arquitetura que engloba o uso de ETL, *Data Warehouse* e o OLAP. Ou seja, para o usuário de negócios, a solução irá prover uma coleta corretiva de dados (através do uso de ETL) para uma base de dados informacional (*Data Warehouse*), na qual terá dados históricos e limpos, e será possível fazer consultas, emitir relatórios e gráficos se utilizando do poder de análises multidimensionais (OLAP). Tendo como benefícios uma automação no processo de geração de análises, permitindo que se possa se esforçar mais no processo decisório e científico, no caso.

O desenvolvimento da base de dados de origem operacional era de responsabilidade externa. No entanto, por contratempos e problemas com comprometimento, não pôde ser desenvolvido. Passou então a fazer parte do escopo do projeto de BI o seu desenvolvimento e carga de dados para que uma solução fim a fim de BI pudesse ser implementada e verificada.

1.6 Organização

O presente documento está organizado em 7 capítulos e 5 anexos. Esta organização visa seguir o padrão definido pelo Serviço de Bibliotecas da Escola Politécnica da USP no documento "Diretrizes para apresentação de dissertações e teses" editado em 2004.

O primeiro capítulo, a Introdução, apresenta o projeto explicando o conceito de BI, os objetivos, o escopo e o fator motivador para sua realização. Nesse capítulo também é abordado o contexto no qual se insere *Business Intelligence*, suas origens, evoluções históricas, e como ele pode contribuir para o dia a dia das pessoas.

O segundo capítulo, Aspectos Tecnológicos e Conceituais, apresenta as informações técnicas importantes para o entendimento do projeto. Nele são abordados e explicados as tecnologias, os conceitos e as metodologias utilizadas para o desenvolvimento de uma solução de BI.

Já o terceiro capítulo, Projeto de Implementação, apresenta a metodologia utilizada em Projetos de *Business Intelligence* e os aspectos mais técnicos das ferramentas. Além disso, são apresentados todos os requisitos do projeto e suas características.

O quarto capítulo, Ferramentas de BI, apresenta as informações sobre as ferramentas de ETL, OLAP e de portais open source que foram pesquisadas ao longo do projeto. Alem disso, apresenta particularidades das ferramentas escolhidas pelo grupo para a solução do estudo de caso.

O quinto capítulo, Estudo de Caso, apresenta um caso em que podemos usar a solução de *Business Intelligence* e a forma como as ferramentas foram utilizadas em conjunto para solucionar o problema.

O sexto capítulo, Considerações finais, apresenta as considerações finais do projeto e possíveis temas para dar continuidade ao mesmo.

O sétimo capítulo, Referencias, apresenta a lista de obras que foram alvo de pesquisa pelo grupo. Após este capítulo, o documento trás a lista de anexos.

2 ASPECTOS TECNOLÓGICOS E CONCEITUAIS

2.1 *Visão geral da solução de Business Intelligence*

Analisando sob um ponto de vista macro, uma solução de *Business Intelligence* visa extrair informações do Banco de Dados Operacional de qualquer área (comercial, governamental, hospitalar, educacional e outros), tratar e armazenar esses dados em um Banco de Dados Informacional e através dele executar as consultas e pesquisas desejadas pelo usuário final para análise dos dados.

A Figura 1 apresenta um diagrama explicativo que descreve o fluxo do sistema.

Figura 1: Fluxo em uma solução de *Business Intelligence*

Analisando esse diagrama de uma maneira mais detalhada, desmembrando o chamado Banco de Dados Informacional, têm-se os seguintes componentes apresentados na Figura 2, que possuem funcionalidades específicas.

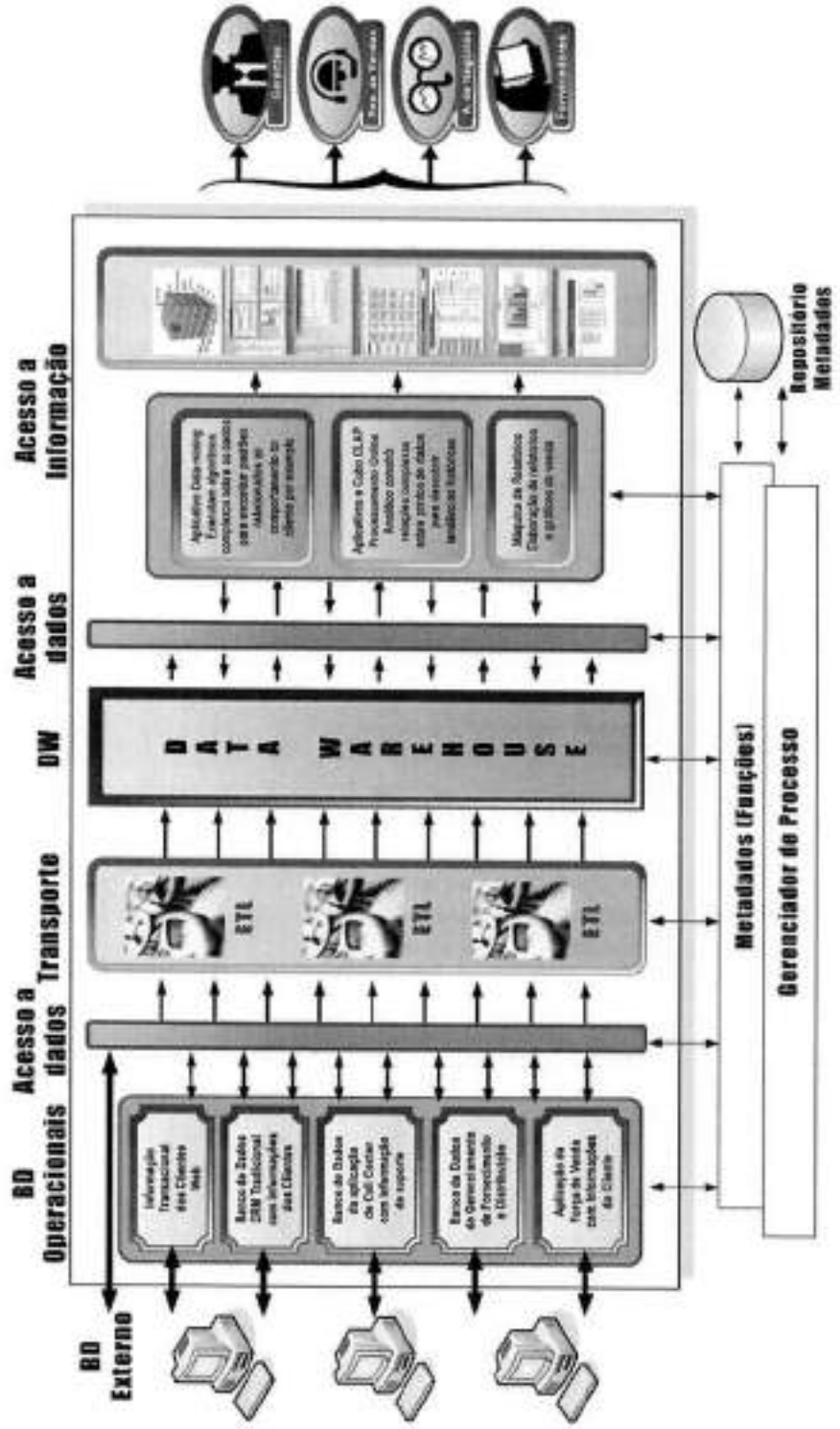


Figura 2: Fluxo detalhado da solução

A partir do Banco de Dados Operacional a ferramenta ETL (Extração, Transformação e Limpeza/Carga) como o próprio nome diz, extrai (*Extract*) as informações relevantes para a análise do negócio, no caso, definidos pelo usuário. Tal funcionalidade tem a capacidade de extrair dados de diversas origens para depois serem transformadas (*Transform*) pelo ETL para que o formato dos dados provenientes das diversas origens seja padronizado. Por fim, os dados passam pelo processo de limpeza (*Cleansing*) e carga (Load) na qual dados incompletos, com erros e inconsistentes são corrigidos de modo a poder funcionar segundo as regras de negócio e são carregados no *data warehouse*.

Depois de passar pela ferramenta de ETL, os dados podem tanto ser utilizados para a criação de meta dados que são utilizados para exibir relatórios e gráficos ao usuário, ou então para o OLAP para uma análise multidimensional mais complexa.

Meta Dados estruturam a informação em categorias, tópicos, grupos, hierarquias e assim por diante. É utilizado para prover informação dos dados dentro do banco de dados informacional como por exemplo serem "Orientados por assunto" baseado em abstrações de entidades do mundo real como por exemplo "projeto", "cliente", "organização" e outros. Também define o modo como o dado transformado é interpretado ("5/9/99" = 5 de Setembro de 1999 ou 9 de maio de 1999). Provê também informação a respeito dos dados relacionados dentro do banco de dados informacional e estima o tempo de resposta através de números de gravações processadas numa pesquisa.

OLAP consiste na construção de cubos lógicos de dados para análise multidimensional. Para isso os dados provenientes do ETL, deve antes passar por uma modificação lógica em sua estrutura pois provém de uma lógica de banco de dados relacional. Podem tanto mudar para uma estrutura conhecida como *star schema* (a mais popular) ou para a estrutura *snow-flake schema*. Ambas representam uma visão relacional de uma base de dados multidimensional.

Caso seja necessário um processo analítico com uma grande quantidade de dados ou se queira descobrir e explorar padrões nos dados e relacionamentos sistemáticos, pode-se utilizar ferramentas de *Data Mining*.

Por fim existe a interface com o usuário que deverá permitir uma visualização dessa análise através de relatórios e gráficos. Os dados desejados nesta análise são configurados pelo usuário também, e interferem logo na ferramenta ETL.

2.1.1 Arquitetura Genérica

A arquitetura apresentada também pode ser dividida em camadas, que permite padronizar e sistematizar os papéis de cada ferramenta dentro do contexto de uma arquitetura.

Esta arquitetura é composta pela camada dos dados operacionais e outras fontes de dados que são acessados pela camada de acesso aos dados. As camadas de gerenciamento de processos, transporte e DW formam o centro da arquitetura e são elas as responsáveis por manter e distribuir os dados. A camada de acesso à informação é formada por ferramentas que possibilitam os usuários extrair informações do DW. Todas as camadas desta arquitetura interagem com o dicionário de dados (meta dados) e com o gerenciador de processos.

- **Camadas de bancos de dados operacionais e fontes externas:** É composto pelos dados dos sistemas operacionais e informações provenientes de fontes externas que serão integradas para compor o DW.

- **Camada de acesso à informação:** Envolve o hardware e o software utilizado para obtenção de relatórios, planilhas, gráficos e consultas. É nesta camada que os usuários finais interagem com o ambiente, utilizando ferramentas de manipulação, análise e apresentação dos dados, incluindo-se as ferramentas de *data-mining* e visualização.

- **Camada de acesso aos dados:** Esta camada faz a ligação entre as ferramentas de acesso à informação e os bancos de dados operacionais. Esta camada se comunica com diferentes sistemas de bancos de dados, sistemas de arquivos e fontes sob diferentes protocolos de comunicação.

- **Camada de Meta Dados (Dicionário de dados):** Meta Dados são as informações que descrevem os dados utilizados pela empresa, isto envolve informações como descrições de registros, comandos de criação de tabelas, diagramas Entidade/Relacionamentos (E-R)

e dados de um dicionário de dados. É necessário que exista uma grande variedade de meta dados no ambiente de DW para que ele mantenha sua funcionalidade e os usuários não precisem se preocupar onde residem os dados ou a forma com que estão armazenados;

- **Camada de gerenciamento de processos:** É a camada responsável pelo gerenciamento dos processos que contribuem para manter o DW atualizado e consistente. Está envolvida com o controle das várias tarefas que devem ser realizadas para construir e manter as informações do dicionário de dados e do DW.

- **Camada de transporte:** Esta camada gerencia o transporte de informações pelo ambiente de rede. Inclui a coleta de mensagens e transações e se encarrega de entregá-las em locais e tempos determinados. Também é usada para isolar aplicações operacionais, do formato real dos dados nas duas extremidades.

- **Camada do *Data Warehouse*:** É o DW propriamente dito, corresponde aos dados utilizados para obter informações. O DW pode ser simplesmente uma visão lógica ou virtual dos dados, podendo não envolver o armazenamento dos mesmos ou armazenar dados operacionais e externos para facilitar seu acesso e manuseio.

2.2 Extração, Transformação e Carga

Todas as informações utilizadas pelo sistema de *Business Intelligence* são armazenadas em um banco informacional, em geral um *data warehouse* ou um *data mart*, por exemplo. O ETL é a ferramenta utilizada para extrair informações dos bancos de dados transacionais da empresa e efetuar o armazenamento destas no *Data Warehouse* (MOSS et. al.,2003).

Os dados de origem utilizados numa aplicação de BI são originados de diversas plataformas e são gerenciadas por uma variedade de sistemas e aplicações. O propósito do processo de ETL é reunir todos esses dados disponíveis em diversas plataformas heterogêneas, para um formato padrão no banco de dados destino (pode ser um *Data Warehouse* ou um *Data Mart*, por exemplo) que será utilizada na arquitetura de BI, como mostra a figura 3.

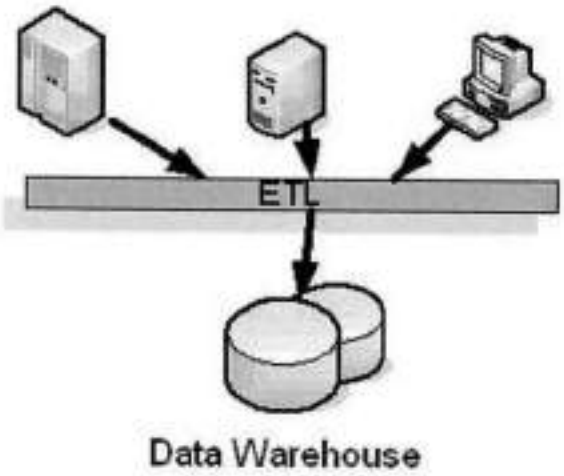


Figura 3: Diversidade nos dados de Origem

2.2.1.1 Preparação para o processo de ETL

O processo de ETL se inicia com a preparação para re-formatação, reconciliação e limpeza dos dados de origem.

- Re-formatação: Os dados de origem residem em diferentes arquivos e banco de dados de origem, cada um com o seu próprio formato. No processo de ETL, os dados devem ser unificados em um formato comum.
- Reconciliação: Muitas vezes os dados acabam se tornando redundantes, devido a imensa quantidade de dados nos banco de dados nas organizações. Isto acaba resultando em inconsistências de dados. Estes dados devem ser encontrados e reconciliados durante o processo de ETL.
- Limpeza: Os dados inválidos encontrados durante a análise de dados terão de ser apagados ou tratados durante este processo.

Antes de projetar o processo de ETL, é necessário revisar os seguintes tópicos:

- Formato dos registros do banco de dados atual assim como os dados históricos
- Especificações das regras de limpeza para os dados de origem.

As maiorias dos dados de origem carregados no processo de ETL são dados operacionais atuais, mas dependendo de alguns casos, os dados podem ser históricos.

Tabela 1 - Tipos de Carga ETL

Carga Inicial	Carga Histórica	Carga Incremental
Carregamento inicial no banco de dados destino com os dados operacionais atuais	Carregamento inicial no banco de dados destino com os dados históricos	Carga "incremental" do banco de dados, com os dados operacionais atuais.

Caso os requisitos da carga de dados incluam poucos anos de dados históricos, os 3 passos devem ser projetados e desenvolvidos como está listado na Tabela 01.

2.2.1.2 Carga Inicial

O processo de preparar a carga inicial é muito similar ao processo de conversão de sistemas, como no caso em que muitas corporações migram seus sistemas para um sistema ERP (*Enterprise Resource Planning*). Em geral, a primeira tarefa de um processo de conversão de sistema é mapear os campos dos dados do arquivo de origem ou banco de dados de origem para outros campos em arquivos ou banco de dados de destino mais apropriado, possuindo assim similaridades no nome, definição, tamanho, comprimento e funcionalidades.

A segunda tarefa é desenvolver a lógica para converter os dados de origem. Esta conversão deve também resolver problemas de registros duplicados, chaves primárias, *truncate* e tamanho dos campos dos dados.

2.2.1.3 Carga Histórica

O processo de carga histórica deve ser visto como uma extensão do processo de carga Inicial, pois este tipo de conversão é ligeiramente diferente devido aos dados históricos que são

estáticos. Em contraste com os dados operacionais dinâmicos, dados estáticos são utilizados com o propósito de armazenamento por dispositivos *offline*. A implicação disto é que os dados antigos expiram e alguns dados novos são adicionados ao passar dos anos, mas muitas vezes os formatos dos registros atuais não estão coerentes com os dados históricos. Conseqüentemente, as conversões utilizadas para os sistemas atuais não podem ser utilizadas em sistemas históricos sem que haja algumas mudanças.

2.2.1.4 Carga Incremental

Um outro processo que devem ser projetado é a carga incremental dos dados (mensal, semanalmente, ou diariamente), considerando que os processos para povoar as bases de dados destino do BI com dados iniciais e históricos foram planejados. As cargas incrementais podem ser realizadas de duas maneiras: extrair todos os registros ou as variações dos dados somente, como mostrados na Tabela 02. O projeto do processo de extração do ETL será diferenciado dependendo da opção escolhida.

Tabela 2 - Opções de carga Incremental

Extração de todos os registros	**Extração da variação apenas (delta)**
Extrai todos os registros dos dados de origem, independente se algum dado tenha tido seu valor alterado ou não independente de quando foi a última carga ETL.	Extrai os registros dos dados de origem que teve seus valores alterados desde a última carga do ETL.

Na maior parte dos casos, extrair todos os registros não é uma opção viável devida ao grande volume de dados envolvidos. Conseqüentemente, muitas organizações optam por uma extração do tipo delta (extrair apenas os registros que mudaram). Uma alternativa seria extrair uma cópia completa dos dados de origem para cada carga, e compará-la com a carga antiga para encontrar os registros que mudaram, criando assim um arquivo próprio com o delta.

2.2.2 Extração

De uma perspectiva operacional de sistemas, a maneira mais simples de criar cargas para extração de dados seria duplicar todo o conteúdo dos arquivos de origem e das bases de dados operacionais de origem e copiar os registros para o DW. Entretanto, os sistemas de ETL utilizariam enormes arquivos quando na verdade seria necessário somente um subconjunto do banco de dados transacionais.

Já na perspectiva do projeto do BI, a maneira a mais favorável de criação de cargas para extração deve classificar, filtrar, limpar e agregar todos os dados requeridos em uma etapa e fazê-los corretamente nos dados de origem. Entretanto, em algumas organizações isto impactaria os sistemas utilizados a tal ponto que as funções operacionais do negócio teriam que ser suspensas por diversas horas.

A solução é geralmente um acordo: os programas de extração de dados são projetados para o ETL ser o mais eficiente possível com relação ao seu processamento e sempre com um foco em obter os dados de origem requisitados o mais rápido possível. O objetivo é impactar os sistemas utilizados de forma mínima, de modo que as funções diárias de negócio não sejam afetadas. Este é um dito mais fácil do que feito, devido a uma série de razões.

Selecionar e unir (*merge*) os dados das fontes de origem pode se tornar um desafio por causa da redundância elevada nos sistemas transacionais. Os programas de extração devem saber quais são as redundâncias nos arquivos e nas bases de dados do sistema. Por exemplo, o mesmo elemento de dados da fonte de origem (por exemplo, nome do cliente) pode existir em dúzias de arquivos e bases de dados de origem. Estas ocorrências redundantes têm que ser classificadas e consolidadas, envolvendo um número de etapas de classificação e de fusão, guiadas por um número de tabelas que fazem referência às chaves e valores específicos dos dados.

Uma outra maneira de produzir pequenas cargas válidas de dados é extrair somente aqueles elementos que são necessários para a aplicação de BI e resolver somente aqueles problemas de qualidade dos dados que pertencem às regras de negócio definidas previamente, sem tentar classificar automaticamente e consolidar ocorrências redundantes dos dados.

Entretanto, em muitas organizações isso não funcionaria, pois o processo de limpeza retardaria o processo de extração, que por sua vez "amarraria" os sistemas por um tempo maior que o aceitável.

Em grandes organizações, o projeto de BI utiliza muito tempo processando os dados antes que eles tenham que ir para o *go live* no dia seguinte. Esta é a principal razão para popular o DW em três processos separados: a extração, transformação e carga (Figura 4).

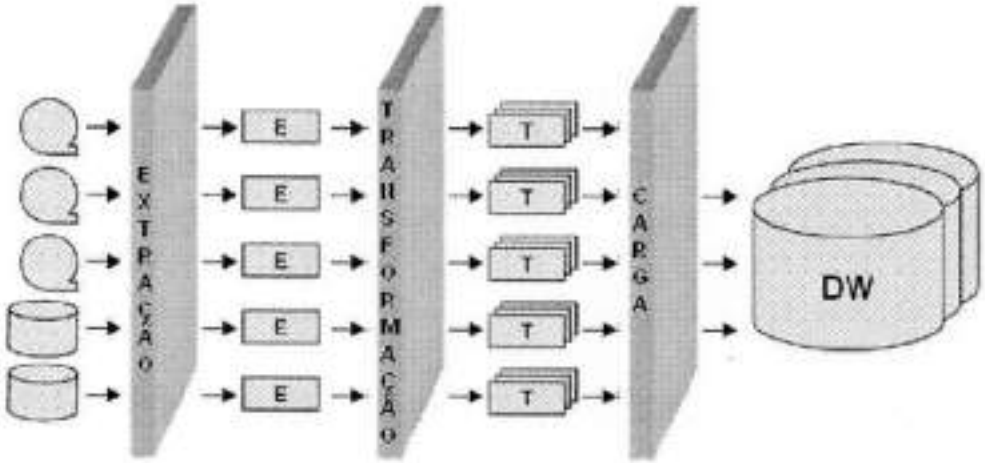


Figura 4: Processo ETL

2.2.3 Transformação

Usando a regra 80/20, 80 por cento do trabalho do ETL ocorrem na fase "T" (Transformação) onde ocorre a integração extensiva dos dados e as transformações necessárias, enquanto extração e carga representam apenas 20 por cento do processo de ETL.

2.2.3.1 Problemas com os dados de origem

O projeto de programas de transformação pode tornar-se muito complicado quando os dados são extraídos de um ambiente operacional heterogêneo. Alguns dos problemas típicos dos dados de origem são descritos abaixo.

- Inconsistência nas chaves primárias: As chaves primárias dos registros dos dados de origem não combinam com a nova chave primária nas tabelas do DW.

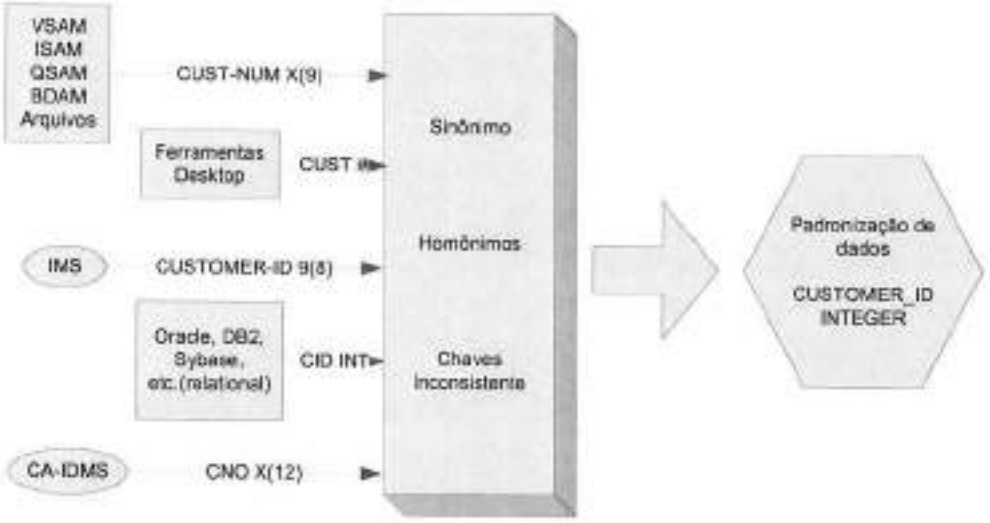


Figura 5: Resolução da inconsistência nas chaves primárias

- Valores inconsistentes dos dados: Muitas organizações acabam duplicando os seus dados e eles acabam possuindo valores completamente diferentes devido a problemas no processo de atualização.

- Formatos diferentes de dados: Os elementos de dados tais como datas e moedas correntes, podem ser armazenados em um formato completamente diferente no DW.

- Valores não precisos dos dados: A lógica de limpeza tem que ser definida para corrigir os valores inconsistentes dos dados

- Sinônimos e Homônimos: Os dados redundantes não são sempre fáceis de reconhecer porque o mesmo elemento de dados pode ter nomes diferentes

2.2.3.2 Transformação dos dados

Além da transformação dos dados de origem, por razões de incompatibilidade entre os diversos tipos de dados ou de inconsistência, uma grande parcela da lógica de transformação

envolverá cálculos prévios nos dados para o armazenamento multidimensional. Os dados nas bases de dados do DW podem ser dados completamente diferentes dos dados nos banco de dados operacionais. Alguns exemplos:

- Alguns dos dados serão renomeados seguindo os padrões definidos (os sinônimos e os homônimos não devem ser propagados no ambiente de *decision-support* do BI). Por exemplo, o elemento *Account Flag* pode agora ser chamado de *Product_Type_Code*.
- Alguns dados de diferentes sistemas serão combinados (*merge*) em uma única coluna em uma tabela do DW porque representam o mesmo elemento de dados lógico. Por exemplo, o *Cust-Nome* do arquivo CMAST, *Customer_Nm* da tabela de CRM_CUST, e *Cust_Acct_Nm* da tabela de CACCT podem agora ser combinados na coluna *Customer_Name* na tabela de BI_CUSTOMER.
- Os elementos de alguns dados serão divididos através de diferentes colunas na base de dados do DW porque estão sendo usados para finalidades diferentes pelos sistemas interligados. Por exemplo, os valores "A", "B", "C", "L", "M", "N", "X", "Y", e "Z" do Prod-Código do elemento de dados podem ser usados como segue pelo sistema: "A","B," e "C" descreve clientes; "L," "M," e "N" descreve fornecedores; e "X," "Y," e "Z" descreve confinamentos regionais. Em conseqüência, o Prod-Código pode agora ser dividido em três colunas:
    - Customer_Type_Code na tabela de BI_CUSTOMER
    - Supplier_Type_Code na tabela de BI_SUPPLIER
    - Regional_Constraint_Code na tabela de BI_ORG_UNIT
- Algumas informações dos dados serão traduzidas em mnemônicos. Por exemplo:
    - "A" pode ser traduzido como "corporação"
    - "B" pode ser traduzido como "parceiro"
    - "C" pode ser traduzido como "indivíduo"
    -

2.2.4 Carga

A etapa final no processo de ETL é carregar as bases de dados destino (*Data Warehouse*), e pode ser realizada de duas maneiras: (1) introduzindo os novos registros nas tabelas ou (2) usando a ferramenta de carga do SGBD parar executar uma carga volumosa. É muito mais eficiente utilizar a ferramenta de carga do SGBD, e a maioria de organizações escolhem esta abordagem. Uma vez terminada as etapas de extração e de transformação, não deveria ser demasiadamente complicado terminar o processo de carga. Entretanto, é ainda necessário fazer decisões do projeto sobre a integridade de dados e indexação.

2.3 Data Warehouse

2.3.1 Definições

Atualmente, os bancos de dados são de vital importância para as empresas, pois eles armazenam todos os dados relativos ao negócio. Estes dados permitem aos analistas de dados buscar informações que auxiliem na tomada de decisão através da identificação de tendências e, com isso, posicionar a empresa estrategicamente para ser mais competitiva diminuindo o índice de erros na tomada de decisão.

Mas existe uma dificuldade muito grande das empresas em extrair informações dos dados e analisá-las. Isso ocorre devido ao grande volume de dados, que na maioria das vezes, estão distribuídos em diferentes sistemas gerenciadores de banco de dados.

Numa tentativa de contornar estes problemas encontrados pelos sistemas de informação, foi introduzido o conceito de *data warehouse*. Existem várias definições para este conceito, mas, as mais aceitas são as de Willian H. Inmon e Ralph Kimball.

Willian H. Inmon define da seguinte forma: "Um *data warehouse* é um conjunto de dados baseado em assuntos, integrado, não volátil, variável em relação ao tempo e de apoio às decisões gerenciais." (INMON, 1997).

Em outras palavras, o *data warehouse* é orientado aos principais assuntos do negócio como clientes, vendas, produtos, etc. Ele é não volátil. Isso significa que os dados armazenados no *data warehouse* sofrem alteração com uma freqüência muito menor e pode ser considerado como não sendo de tempo-real, com atualizações periódicas, geralmente incrementais. Ele é integrado, pois os dados armazenados no *data warehouse* estão padronizados com relação aos termos e as estruturas técnicas que são utilizados nos sistemas de informações tradicionais e, por último, ele é variável no tempo, ou seja, o elemento tempo sempre está presente na estrutura de dados, o que não ocorre obrigatoriamente no banco de dados transacional (ELMASRI, 2003).

Ralph Kimball, por sua vez, define o *data warehouse* como sendo "uma cópia dos dados transacionais especificamente estruturado para buscas e análises." (KIMBALL, 2000). Esta definição é menos profunda que a anterior, mas, não deixa de ser correta.

2.3.2 Data Mart

Os primeiros projetos sobre *Data Warehouse* (DW) referiam-se a uma arquitetura centralizada. Embora fosse interessante fornecendo uniformidade, controle e maior segurança, a implementação desta abordagem não é uma tarefa fácil. Requer uma metodologia rigorosa e uma completa compreensão dos negócios da empresa. Esta abordagem pode ser longa e dispendiosa e por isto sua implementação exige um planejamento bem detalhado. Com o aparecimento de *data mart*(DOS SANTOS, 2001) ou *warehouse* departamental, a abordagem descentralizada passou a ser uma das opções de arquitetura *data warehouse*.

A tecnologia usada tanto no DW como no *Data Mart* é a mesma, as variações que ocorrem são mínimas, sendo em volume de dados e na complexidade de carga. A principal diferença é a de que os *Data Marts* são voltados somente para uma determinada área, já o DW é voltado para os assuntos da empresa toda.

Figura 6: Data Marts

2.3.3 Projeto Lógico do banco de dado

A filosofia de projeto entre o *data warehouse* e o banco de dados transacional são completamente diferentes.

O propósito do projeto do banco de dados transacional é o de prevenir o armazenamento do mesmo dado em vários lugares diferentes e, com isso, evitar anomalias decorrentes da atualização causada pela redundância dos dados. Isso é feito através do projeto de banco de dados normalizados, que garante que o dado seja criado, armazenado e modificado de uma forma não redundante e consistente.

A maioria dos sistemas transacionais é projetada seguindo uma filosofia conhecida como *data-in* (MOSS, 2003). O objetivo desta filosofia de projeto é a de efetuar a entrada de dados de forma eficiente, eliminando ou minimizando redundâncias dos dados. A redundância dos dados conduz a inconsistência dos mesmos e esta é, em muitos casos, a causa da baixa qualidade dos dados. Estes problemas são resolvidos através da normalização (ELMASRI, 2003).

Enquanto a normalização é uma ótima solução para os sistemas transacionais, os requisitos para sistemas de informação (sistemas de geração de relatórios) são bem diferentes dos requisitos de um sistema que segue a filosofia de *data in*. Os relatórios utilizam dados existentes e que não sofrerão alteração, ou seja, as anomalias decorrentes da atualização não ocorrerão. O

mesmo projeto que traz diversos benefícios aos sistemas transacionais, apresenta diversas dificuldades para os sistemas de informação.

Ao contrário da filosofia de projeto *data-in* dos sistemas transacionais, a filosofia de projeto *data-out* (MOSS, 2003), presentes em aplicações de B.I. possuem as seguintes considerações em seu projeto:

- São projetados para retornar dados para o usuário de forma simplificada e com alta performance.
- A redundância dos dados é aceita desde que ela traga uma simplificação na maneira de acessar os dados, mas esta redundância deve ser controlada.

A tabela abaixo apresenta uma comparação das principais características dos dois tipos de bancos de dados.

Tabela 3 - Banco de dados Transacional X *Data Warehouse*

Banco de Dados Transacionais	*Data Warehouse*
prioriza a eliminação da redundância, coordenando atualizações e repetindo as mesmas operações várias vezes	prioriza as buscas e geração de relatórios.
Um dos requisitos a serem atendidos é o tempo de resposta muito baixo (menor que um segundo)	Apesar do tempo de resposta ser importante, ele varia de alguns segundos até algumas horas, dependendo do sistema.
Altamente normalizado para facilitar a atualização e a manutenção da integridade *(referencial integrity)*	Desnormalizado para prover um rápido retorno de informação em sistemas que possuem um volume muito grande de dados.
Armazena poucos dados derivados. Os dados são derivados dinamicamente quando são necessários	Armazenam grande quantidade de dados derivados. Com isso obtem um ganho de tempo no caso de buscas na base de dados.
Não armazenam dados históricos. Estes dados são arquivados.	Armazenam grande quantidade de dados históricos.
Levemente resumidos	Muitos valores pré-calculados são armazenados no *data warehouse*

Na filosofia de projeto *data-out* as tabelas que compõem o *data warehouse* não são necessariamente normalizadas. A desnormalização das tabelas possibilita um ganho de

velocidade para retornar informações ao usuário, pois com isso, a busca é feita em um número menor de tabelas, reduzindo assim, o número de operações de *join* necessárias.

Outra característica, que podemos citar como sendo a principal diferença entre o *data warehouse* e o banco de dados transacional, é o seu modelo de dados multidimensional (LANE, 2002). Este modelo de dados multidimensional é ideal para as tecnologias de suporte a decisão, como o OLAP com suas operações especiais, que serão detalhados adiante.

No banco de dados transacional, os dados são armazenados em tabelas bidimensionais, como no exemplo abaixo, em que é mostrada uma tabela de vendas por localidades e por produtos em um período de tempo particular.

Tabela 4 - Modelo de dados bidimensional

	Localidade 1	Localidade 2	...	Localidade n
Produto 1	Venda 1,1	Venda 1,2		Venda 1,n
Produto 2	Venda 2,1	Venda 2,2		Venda 2,n
...				
Produto m	Venda m,1	Venda m,2		Venda m,n

Utilizando o modelo de dados multidimensional, podemos tirar vantagem através dos relacionamentos entre os dados, que são armazenados em tabelas multidimensionais representados por cubos. Ela permite a adição de um maior número de parâmetros para os dados armazenados. Isso pode ser demonstrado neste exemplo, através da inclusão do parâmetro tempo na tabela, representando a mesma através do cubo abaixo.

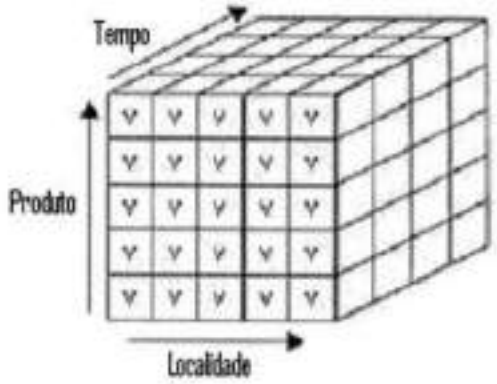


Figura 7: Representação multidimensional dos dados

Este modelo de dados multidimensional possui uma grande tabela central relacionada a outras tabelas secundarias menores. A tabela central é chamada de tabela fato e as demais tabelas secundárias são chamadas de dimensão. A tabela fato armazena valores numéricos chamados de *measures* que representam os resultados do negócio. Estes valores são mensuráveis e podem ser somadas. Esta tabela é a maior tabela no esquema devido aos dados históricos armazenados na mesma. Elas são longas e estreitas, pois ela possui um imenso número de linhas, mas com poucas colunas nestas tabelas. No exemplo acima, o valor armazenado no cubo é a quantidade de vendas realizadas. Estes valores armazenados na tabela fato são referentes a alguns parâmetros de negócios, que são armazenadas nas tabelas dimensão. Cada tabela dimensão armazena uma descrição de cada parâmetro do negócio e possui uma chave primária composta por apenas uma coluna. Na representação do cubo acima, as dimensões são representadas através das arestas dos cubos, no caso, a localidade, o produto e o tempo são dimensões. A tabela fato está atrelada as tabelas dimensão através de sua chave, que é composta pelas chaves estrangeiras das tabelas dimensão.

Os dados armazenados nas tabelas dimensão possuem uma hierarquia. Por exemplo, na dimensão de tempo são armazenados os meses. Estes meses são agrupados em trimestres, que por sua vez são agrupados em semestres, anos e década, conforme apresentado na figura abaixo.

Figura 8: Tabela dimensão de tempo

Esta forma hierárquica é estendida para todas as tabelas dimensão. Esta estrutura permite que o OLAP efetue as operações de *Drill down/up*, que nada mais é do que se deslocar através dos níveis dentro desta hierarquia.

Estas tabelas dimensão e fato são organizados mais popularmente de duas maneiras, seguindo as técnicas chamadas *star schema* e *snowflake schema* que serão descritas abaixo.

2.3.3.1 Star Schema

Na técnica chamada *star schema* (LANE, 2002), o dado é representado como um *array* de valores pré-calculados, que são os chamados fatos, a partir da qual é feita a análise. Estes fatos pré-calculados representam valores operacionais atômicos que foram sintetizados por certas dimensões, como fornecedores, produto e tempo. A dimensão em um *star schema* é similar a uma entidade no modelo lógico de dados tradicional. Ela é um objeto de negócio através da qual os dados são coletados para propósitos de negócios.

Através do *star schema* podemos efetuar as informações do negócio de maneira muito eficiente. Como o próprio nome diz, o *star schema* tem uma única tabela central, que é a tabela fato explicada acima, e nela são conectadas diversas outras tabelas, chamadas tabela dimensão. A Figura 9 apresenta um exemplo de *star schema*.

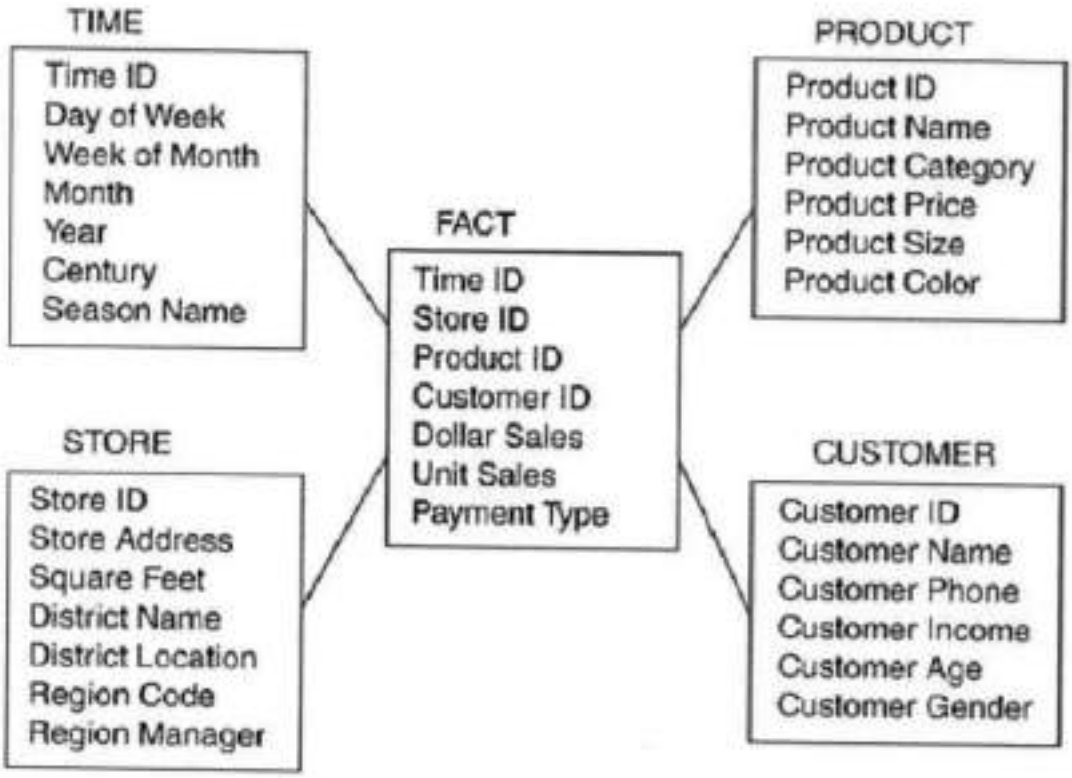


Figura 9: Star schema

O *star schema* possui dois, e apenas dois níveis: a tabela fato e uma série de tabelas dimensão de um nível só.

O *star schema* é a técnica de projeto para aplicações de *Business Intelligence* pelas seguintes razões:

- Apresenta a melhor performance em buscas de analises do negócio e relatórios que possuem anos de dados históricos
- Fornece máxima flexibilidade para análise de dados multidimensionais
- É compatível com a maior parte dos fabricantes de SGBDs relacionais com modificações para suas ferramentas de otimização.
- Sua simplicidade torna as analises complexas dos dados mais simples do que em modelos de dados normalizados.

2.3.3.2 Snowflake Schema

O *snowflake schema* (LANE, 2002) é uma variação do *star schema*, sendo a principal diferença entre as duas a presença de mais de dois níveis, como mostra a Figura 10:

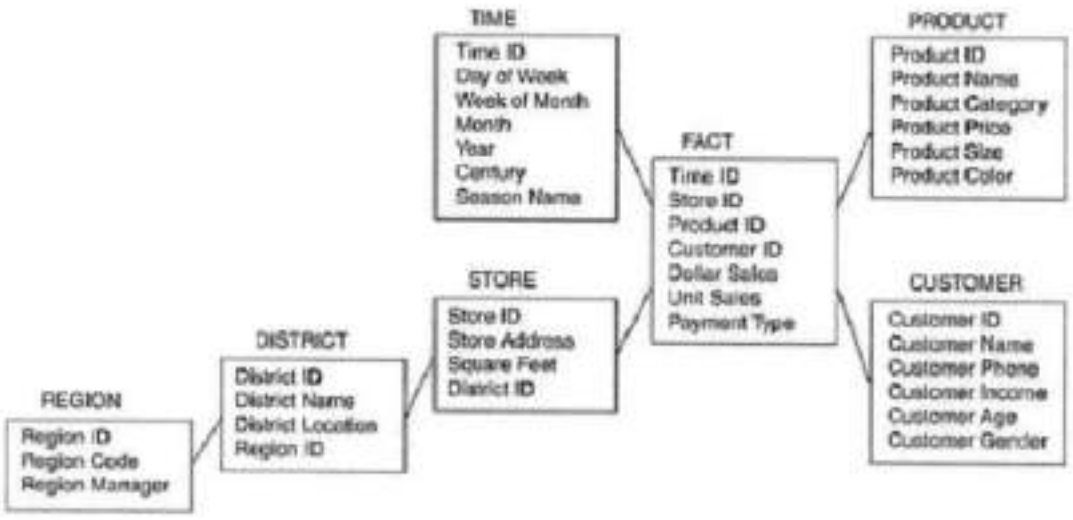


Figura 10: Snowflake schema.

No *snowflake schema*, os níveis de hierarquia nas tabelas dimensão são normalizadas e para isso, é feita a divisão das tabelas dimensão em várias outras, evitando redundância. A tabela abaixo apresenta as vantagens e desvantagens do *snowflake schema*.

Tabela 5 - Vantagens e desvantagens do Snowflake Schema

Vantagens	Desvantagens
O tamanho da tabela dimensão é reduzido e a redundância é evitada com a divisão da tabela dimensão	O crescimento do número de tabelas pode afetar na performance das buscas devido a quantidade adicional de *joins* a ser feito
Há um aumento na flexibilidade da aplicação	O esforço necessário para a manutenção do banco de dados aumenta, pois o número de tabelas a serem mantidas aumenta.

2.3.4 Projeto Físico do banco de dados

Os bancos de dados utilizados em BI costumam ser muito grandes, excedendo a casa dos *terabytes* em muitos casos. Tanto o projeto físico destes bancos de dados (LANE, 2002) como a manutenção do mesmo é um grande desafio, pois exige a utilização de recursos de alta performance do SGBD.

Os SGBDs em geral fornecem várias opções de implementação. A escolha dessas opções demanda uma experiência do DBA para decidir qual combinação das opções a seguir atingirá o nível de performance desejado.

As decisões a serem tomadas para a implementação incluem as seguintes decisões:

- Qual a quantidade de espaço livre a ser disponibilizado?
- Qual o tamanho do buffer a ser declarado?
- Qual deve ser o *blocksize*?
- Quando deve ser utilizada uma técnica de compactação?

Outros pontos que afetam o desempenho do banco de dados são apresentadas abaixo:

- **Alocação física dos dados** efetuada seguindo de uma série de medidas para atingir respostas rápidas. Estas medidas são o armazenamento de dados freqüentemente

utilizados em dispositivos mais rápidos, o processamento de diversas operações em paralelo, a alocação de diferentes níveis de agregação em plataformas diferentes (pode ser necessário até o armazenamento de dados em servidores distribuídos enquanto os dados detalhados são mantidos no *mainframe*) e o uso de diversos discos de menor capacidade ao invés de poucos discos de grande capacidade.

- **Particionamento** das tabelas através dos discos, permitindo a divisão dos dados de uma tabela lógica em vários *datasets*. Este ponto é particularmente importante para os bancos de dados muito grandes, em que as tabelas fato podem atingir alguns *gigabytes*.

- **Clustering:** técnica útil para melhorar a performance onde ocorrem acessos seqüenciais a uma grande quantidade de dados, o que proporciona um grande ganho no desempenho pois acessos aos dados de forma seqüencial é comum em aplicações de BI. Esta técnica é implementada através de *clustering index*, que determinam a ordem em que as linhas serão armazenadas nos *datasets*.

- **Indexação** de tabelas, que possui dois fatores a se considerar: A decisão de quais colunas indexar e a determinação da estratégia usada pelos índices (algoritmos). As colunas que possuem valores muito distribuídos e sofrem buscas freqüentes são normalmente indexadas. A escolha entre os algoritmos é baseada nas informações fornecidas por cada fabricante de SGBD.

- **Reorganizações** dos bancos de dados, tendo como atividades, a cópia do banco de dados para um outro dispositivo, reagrupar as linhas e carrega-las novamente no banco de dados.

- ***Backup*** **, Recuperação dos dados e** ***Disaster Recovery*** como prevenção a falhas, sendo estes procedimentos, muito importantes para o BI pois levaria muito tempo para se recriar o *data warehouse* através das fontes de dados originais.

- **Execução Paralela de Buscas** dividindo a mesma em várias partes e efetua-las concorrentemente. Quando várias partes de uma busca são executadas

paralelamente em diferentes processadores, o sistema apresenta um grande aumento no seu desempenho.

2.3.5 *Implementação do Data Warehouse*

A implementação de um *data warehouse* é uma tarefa árdua e o seu tempo de implementação varia muito com o nível de robustez que o projeto em questão exige. Podemos apresentar estes níveis de implementação agrupados em quatro grupos, como mostra a Figura 11. As principais diferenças entre estes grupos é a robustez do modelo, o tempo e custo da implementação destes modelos.

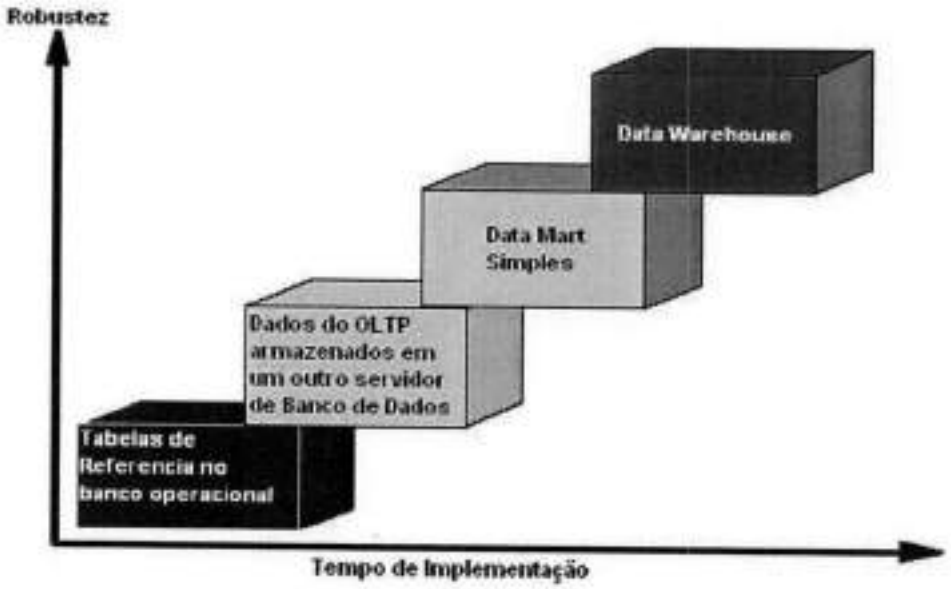


Figura 11: Implementações de BI

2.3.5.1 *Tabelas de Referência no banco operacional*

A implementação mais simples é a inserção de tabelas de referência no próprio banco de dados operacional que abrange apenas alguns requisitos da área de negócios da empresa. As vantagens apresentadas por este modelo são a minimização da necessidade de recursos de rede, a

implementação rápida e o ponto de acesso único dos dados. As desvantagens apresentadas pelo mesmo são a impossibilidade de isolar a carga de trabalho, o custo da atualização de processos, a galta de transformação dos dados, não ser apropriado para casos em quem nem todos os dados estão no OLTP em questão e o forte impacto causado no OLTP no caso de atualizações dos valores armazenados.

2.3.5.2 Dados do OLTP armazenados em outro servidor de Banco de Dados

Outra implementação um pouco mais robusta que a anterior é a de cópia periódica dos dados do OLTP para outro servidor de banco de dados. Nesta implementação não ocorre nenhuma mudança na estrutura do banco de dados, mas apenas a cópia dos dados do OLTP para um outro banco de dados. Este é o primeiro passo para desviar a carga de processamento do sistema OLTP para construir uma base de dados voltada para auxilio a tomada de decisão. Suas vantagens são o ganho de desempenho através do isolamento de cargas de trabalho e a implementação rápida do modelo. Suas desvantagens são a impossibilidade de usar meta dados, a não otimização do projeto de banco de dados e a limitação na flexibilidade.

Se esta implementação for efetuada em maquinas completamente isoladas, pode-se eliminar qualquer interdependência entre a busca de informação para BI e a carga de trabalho operacional.

O principal problema que persiste nesta implementação é a não otimização do banco de dados para busca de informação, pois o modelo lógico e físico do mesmo não sofreu qualquer alteração.

2.3.5.3 Data Mart

Outra implementação possível é a construção de *Data marts* simples. Normalmente este tipo de implementação é feito como uma prévia da implementação de um *data warehouse* e permite um crescimento ao longo do tempo. Este caso envolve um pouco mais de preparação, planejamento e investimentos. Suas vantagens são o ganho de desempenho devido ao isolamento das cargas de trabalho do *data mart* e do OLTP, a possibilidade de armazenar meta dados, a

existência de soluções específicas de mercado e o tempo de implementação mais curto que o de um *data warehouse*. A desvantagem deste modelo é o fato de expansões futuras poderem forçar o uso de novos programas para carga e limpeza dos dados de origem.

A implementação de um simples *data mart* pode ser feita de maneira muito rápida se o escopo da informação a ser armazenado está bem definido e limitado ao número adequado de *datasets*.

2.3.5.4 Data Warehouse

O último tipo de implementação citado acima é a implementação do *data warehouse* que é composto por três níveis:

- OLTP em bancos de dados operacionais
- Dados organizados seguindo as técnicas *star* e *snowflake schema* para otimizar a performance das buscas
- Diversos *data marts* agregados e pré-calculados para apresentar os dados ao usuário final

As características deste modelo são:

- *Data marts* departamentais mantêm as informações organizadas de forma específica para alguns requisitos.
- Dados históricos podem ser mantidos no *data warehouse*
- Os Meta Dados tem uma grande importância nesta arquitetura, pois facilita a navegação do usuário
- Limpeza e transformação dos dados implementados em um único ponto da arquitetura

- A Carga de trabalho gerada pelas análises não interferem no OLTP.

2.4 OLAP

2.4.1 Introdução

OLAP (PARRINI, 2002) é considerado uma categoria de software que cria novas informações para o negócio através de um conjunto de transformações e cálculos executados sobre os dados existentes no *data warehouse*, permitindo aos usuários obter respostas dentro dos dados com uma ampla variedade de possíveis visões. As ferramentas de OLAP permitem que o negócio de uma empresa possa ser visualizado e manipulado de forma multidimensional (utilizando a estrutura de dados multidimensional explicada anteriormente no *data warehouse*), permitindo a combinação dos dados na ordem em que se deseja em qualquer nível de detalhamento considerando diversos períodos de tempo. Os analistas podem projetar suas análises selecionando as dimensões através de simples *cliques* com o *mouse* e selecionando os elementos de dados desejados.

2.4.2 Vantagens da Ferramenta OLAP

A ferramenta de OLAP apresenta duas vantagens distintas:

- O foco do processamento analítico está sobre os dados, mais especificamente no aspecto multidimensional dos dados. Estes dados são separados em dimensões, que são naturalmente interrelacionadas, e geralmente possuem uma hierarquia.
- A facilidade de navegação através das dimensões efetuando operações como *drill down*, *drill up* e *drill across*, entre outros. Estas operações serão explicadas ao longo do texto.

A ferramenta de OLAP é um componente muito importante na arquitetura de BI pois, enquanto ferramentas convencionais de busca de informações em bancos de dados mostram apenas o que está armazenado no mesmo, o OLAP pode ser utilizado para responder o "porque" de certos eventos estarem corretos, comprovando ou negando hipóteses levantadas pelos analistas de negócio através de correlações entre alguns dados armazenados.

2.4.3 Características da Ferramenta OLAP

A ferramenta de OLAP se popularizou não somente por fazer com que os analistas de negócio se tornem auto-suficientes, mas também pelas ferramentas que fornecem maneiras inovadoras de analisar os dados.

- **Apresentação de uma visão multidimensional** dos dados, que são intuitivos para os analistas de negócio.
- **Fornecimento de somatórias e agrupamentos** dos dados em qualquer uma das intersecções entre as diversas dimensões presentes no banco de dados.
- **Buscas de informação no banco de dados de forma interativa e capacidade de análise dos dados**, que é uma funcionalidade muito apreciada pelos analistas de negócio, pois permite a eles efetuarem buscas no banco de dados e atuar nas buscas interativamente, mudando valores de alguns parâmetros e refazendo a busca para produzir novos resultados.
- **Fornece sustentação para operações de acesso a dados como *drill down, drill up, rotate, Slice and Dice* e *drill across*** (PARRINI, 2002) para análises multidimensionais.
    - ***Drill Up*:** somatória dos dados conforme diminui o nível de detalhamento dos dados apresentados.

- ***Drill Down*:** operação inversa ao *Drill Up* , em que é feita a somatória dos dados conforme aumenta o nivel de detalhamento dos dados apresentados, inserindo subdivisões do valor anterior.

- ***Rotate*:** Consiste na inserção, exclusão ou alteração das dimensões entre as dimensões visuais (por exemplo, incluir uma dimensão nas linhas da tabela ou passar a dimensão da linha para a coluna).

- ***Slice and Dice*:** operações para realizar navegação por meio dos dados na visualização de um cubo. A operação *Slice* fatia o cubo mas mantêm a mesma perspectiva de visualização dos dados. Ela funciona como um filtro restringindo uma dimensão à somente alguns de seus valores. A operação *Dice* efetua a mudança de perspectiva da visão multidimensional. É como se o cubo fosse rotacionado.

- ***Drill Across*:** processo de ligação entre duas ou mais tabelas fatos que tenham a mesma granularidade. Em alguns softwares OLAP este processo é transparente. Nem todas as ferramentas OLAP suportam este tipo de funcionalidade.

- **Disponibilização dos dados na forma gráfica** que oferece uma rápida visualização dos dados.

2.4.4 Arquitetura do OLAP

Conceitualmente, a ferramenta OLAP é dividida em três partes. Uma delas é o núcleo do OLAP propriamente dito. As outras duas estão ligadas à apresentação dos resultados e banco de dados.

A parte do OLAP relacionada com a apresentação dos dados fornece o elo entre os analistas de negócio e os dados armazenados. A maioria das empresas ainda pode ser definida como sendo ricas em dados, mas pobres em informações, pois a informação é o dado que pode ser analisado, sintetizado e usado em um contexto de negócios válido.

As pessoas que farão uso desta informação são analistas, gerentes e executivos da área de negócio, ou seja, são pessoas sem conhecimentos tecnológicos profundos. Por isso, estes dados têm que ser apresentados em um formato que possibilitem aos profissionais da área de negócio a desenvolver o planejamento estratégico das empresas. Portanto, esta *interface* de apresentação deve ser de fácil uso para os mesmos, e não para os profissionais de Tecnologia da informação. A interface gráfica deve ser intuitiva e apresentar termos familiares à área de negócio. Ela não deve apresentar a estrutura de dados nem mostrar os processos que são executados pelo OLAP para gerar os resultados apresentados para o usuário final. Ela deve ser flexível, pois cada analista possui suas necessidades e preferências individuais e estas variam conforme sua habilidade com o computador e seu nível de conhecimento da ferramenta.

O núcleo do OLAP deve prover uma ampla variedade de serviços, desde buscas simples com poucas dimensões até buscas poderosas envolvendo um número muito grande de variáveis. Alem disso, o OLAP deve ser capaz de integrar todo os processamentos analíticos, que envolve capacidade de gerar relatórios e análises multidimensionais. Isto faz do OLAP uma ferramenta útil para transformar dados em informação.

Para prover esta variedade de serviços, o OLAP executa as operações de *drill down*, *roll up*, *rotate*, *slice and dice* e, em alguns casos, *drill across*.

No que diz respeito ao banco de dados, o OLAP possui duas variações:

- ROLAP: ferramenta que acessa qualquer um dos SGBDs relacionais, desde que o modelo de dados utilizado no banco seja multidimensional, como por exemplo *star schemas*, *snowflake schemas* e *hybrid schemas*.
- MOLAP: ferramentas projetadas exclusivamente para acessar um SGBD multidimensional proprietário, que possui uma estruturação dos dados especial.

2.5 Meta Dados

De uma maneira simples, podemos definir meta dados como sendo "dados sobre dados". Ou seja, são dados que fazem referência a outros dados de uma forma abstrata. De acordo com Inmon, os meta dados mantêm informações sobre "o que está e onde" no ambiente de DW.

Nos meta dados estão boa parte das regras de negócios que provêem inteligência ao sistema além de desempenharem um papel importante na administração de dados. É a partir deles que as informações serão consultadas, processadas e atualizadas.

Todas as fases de um projeto de BI geram meta dados. Os aspectos sobre os quais os meta dados mantêm informações são:

- O modelo de dados
- A estrutura de Dados segundo a visão técnica
- A estrutura de Dados segundo a visão de negócios
- O histórico das extrações de dados
- A fonte de dados que alimenta o DW
- A transformação sofrida pelos dados no momento de sua migração para o DW
- Informações referentes a relatórios gerenciais
- Informações referentes às camadas semânticas
- Informações referentes aos processos de Carga.

2.5.1 Classificação de meta dados

De uma forma simples, podemos dividir os meta dados em 3 camadas diferentes:

- Meta Dados operacionais: definem a estrutura dos dados mantidos pelos bancos operacionais, usados pelas aplicações de produção da empresa.

- Meta Dados centrais do DW: são orientados por assunto e definem como os dados transformados devem ser interpretados, incluem definições de agregação e campos calculados, assim como visões sobre cruzamentos de assuntos.

- Meta Dados do nível do usuário: organizam os meta dados do DW para conceitos que sejam familiares e adequados aos usuários finais

2.6 Data Mining

Data mining (ou mineração de dados) é o processo de extrair informação válida, previamente desconhecida e de máxima abrangência a partir de grandes bases de dados, usando-as para efetuar decisões cruciais. *Data mining* vai muito além da simples consulta a um banco de dados, no sentido de que permite aos usuários explorar e inferir informação útil a partir dos dados, descobrindo relacionamentos escondidos no banco de dados. Pode ser considerada uma forma de descobrimento de conhecimento em bancos de dados, área de pesquisa de bastante evidência no momento, envolvendo Inteligência Artificial e Banco de Dados.

Data mining pode ser utilizada com os seguintes objetivos:

- Explicativo: explicar algum evento ou medida observada, tal como porque a venda de sorvetes caiu no Rio de Janeiro;

- Confirmação: confirmar uma hipótese. Uma companhia de seguros, por exemplo, pode querer examinar os registros de seus clientes para determinar se famílias de duas rendas têm mais probabilidade de adquirir um plano de saúde do que famílias de uma renda;

- Exploratório: analisar os dados buscando relacionamentos novos e não previstos. Uma companhia de cartão de crédito pode analisar seus registros históricos para determinar que fatores estão associados a pessoas que representam risco para créditos.

Especialmente devido ao alto custo envolvido, estas ferramentas vinham sendo usadas, até o momento, quase que unicamente por grandes corporações e instituições governamentais. Com o grande aumento do volume de dados nas empresas e com o crescimento do uso de tecnologia de banco de dados, especialmente de *Data Warehouse*, as técnicas de *Data Mining* assumiram papel importante no suporte aos processos de tomada de decisão e devem, aos poucos, ganhar mercado dentre empresas de menor porte.

2.6.1 Visão geral das tecnologias de implementação

O *Data Mining* é um campo que compreende atualmente muitas ramificações importantes. Cada tipo de tecnologia tem suas próprias vantagens e desvantagens, do mesmo modo que nenhuma ferramenta consegue atender todas as necessidades em todas as aplicações.

- Redes Neurais
- Indução de Regras
- Árvores de decisão
- Análise Estatística de Séries Temporais

2.6.2 Técnicas de Data Mining

As diversas técnicas de *Data Mining* dão suporte a um conjunto de operações que diferem entre si pelo tipo de problema que são capazes de resolver. São elas: associações, padrões seqüenciais, séries temporais similares, classificação e regressão, e *clusterização*.

2.6.2.1 Associação

Associações são relacionamentos significativos entre itens de dados armazenados. O objetivo da operação é encontrar tendências, a partir de grande número de transações, que possam

ser usadas para entender e explorar padrões de comportamento dos dados. Um exemplo seria o de varrer registros de terminais de ponto de venda e descobrir que tipos de itens são vendidos juntos, de forma a redefinir a disposição dos artigos na loja e sua promoção em campanhas publicitárias, permitindo explorar com maior eficácia essas associações.

2.6.2.2 Padrões seqüenciais e séries temporais similares

Enquanto a associação encontra eventos que ocorrem juntos a partir de coleções lógicas, a operação de padrões seqüenciais encontra eventos relacionados que ocorrem ao longo de um período de tempo. Um exemplo deste tipo de operação seria a identificação de padrões de sintomas e doenças em pesquisas médicas.

Séries temporais similares podem ser usadas para identificar séries similares coletadas ao longo de um período de tempo. Como exemplo, pode-se considerar a identificação de empresas com padrão de crescimento similares, ações ou fundos de investimento com movimentos de preços parecidos.

2.6.2.3 Classificação e regressão

Classificação e regressão usam dados existentes para criar modelos de comportamento de variáveis. A operação de classificação cria automaticamente um modelo a partir de um conjunto inicial de registros. Esse conjunto serve de exemplo e é chamado de conjunto de treinamento. Os registros do conjunto de treinamento devem pertencer a um pequeno grupo de classes predefinidas. O modelo é composto de padrões, essencialmente generalizações em relação aos registros, os quais são usados para diferenciar as classes. Uma vez obtido o modelo, este é usado para classificar automaticamente os demais registros.

2.7 Integração de ferramentas de BI

Com o intuito de integrar as ferramentas utilizadas na arquitetura de BI como o ETL e o OLAP, um portal *web* será utilizado que irá facilitar o acesso e a disponibilização dos serviços.

O portal será responsável também pelo controle de acesso das ferramentas. Apenas os usuários com o perfil previamente pré-estabelecidos poderão utilizar os serviços de *Business Intelligence*, uma vez que os dados providos pelo BI são estratégicos e por questões técnicas de desempenho, já que consome muito processamento, não é desejável o seu uso desenfreado por vários usuários.

2.7.1 *Portais* Web

A internet possibilita aos seus usuários um acesso rápido e fácil às informações, busca de produtos e serviços. Portais *Web* são uma classe de aplicações que permitem o acesso a essas informações, de diferentes provedores de informação através de uma única interface com o usuário, integrando em um só local os serviços disponíveis em diversos provedores disponíveis. O uso de portais *web* ajuda o usuário a encontrar a informação, produto ou serviço desejado dentro de um grande número de provedores sem ter que navegar por eles individualmente.

A relação entre portais e provedores pode ser classificada em dois modos:

- Modo Centralizado: o conteúdo de interesse dos provedores é enviado para o portal e estes conteúdos de diferentes provedores são mantidos pelo portal. Quando o portal recebe uma requisição, esta é atendida localmente. Existe um administrador que gerencia as atualizações de conteúdo, que são de responsabilidade de cada provedor.

- Modo Distribuído: o conteúdo de interesse de cada provedor é mantido por eles mesmos. Quando o portal recebe uma requisição, ele divide a mesma em sub-requisições menores e enviam cada uma aos seus respectivos provedores. Cada um dos provedores processa as requisições e retorna o resultado para o portal, na forma de um arquivo XML, que é formatado em HTML e é enviado para o usuário.

Portais possuem uma grande quantidade de funcionalidades como, por exemplo: gerenciamento de conteúdo *online*, ferramentas para acesso a informações em diferentes aplicações e plataformas, aplicações que permitem a personalização de conteúdo de páginas na internet, controle de acesso em diferentes hierarquias, provedor e gerenciador de serviços.

Aproveitando a capacidade do Portal de prover diversos serviços, aplicativos de ETL e OLAP serão inseridos de modo a facilitar o seu uso através de uma única interface.

3 PROJETO DA SOLUÇÃO DE BI

3.1 Fases de Projeto de BI

Como a grande maioria dos projetos de engenharia, é necessária uma aplicação para tal. Visando isso, foi proposto o desenvolvimento de uma aplicação de *Business Intelligence* para o Grupo PENSA, uma organização que integra os Departamentos de Economia e Administração da FEA-USP, São Paulo e Ribeirão Preto. Um dos principais objetivos da entidade é estudar a dinâmica do Sistema Agroindustrial, além de identificar e analisar as principais tendências dos negócios industriais visando, sobretudo a inserção competitiva do Brasil no Agribusiness internacional.

Assim como todo tipo de projeto de engenharia existem seis estágios como a concepção e a implementação, como ilustrado na Figura 12.

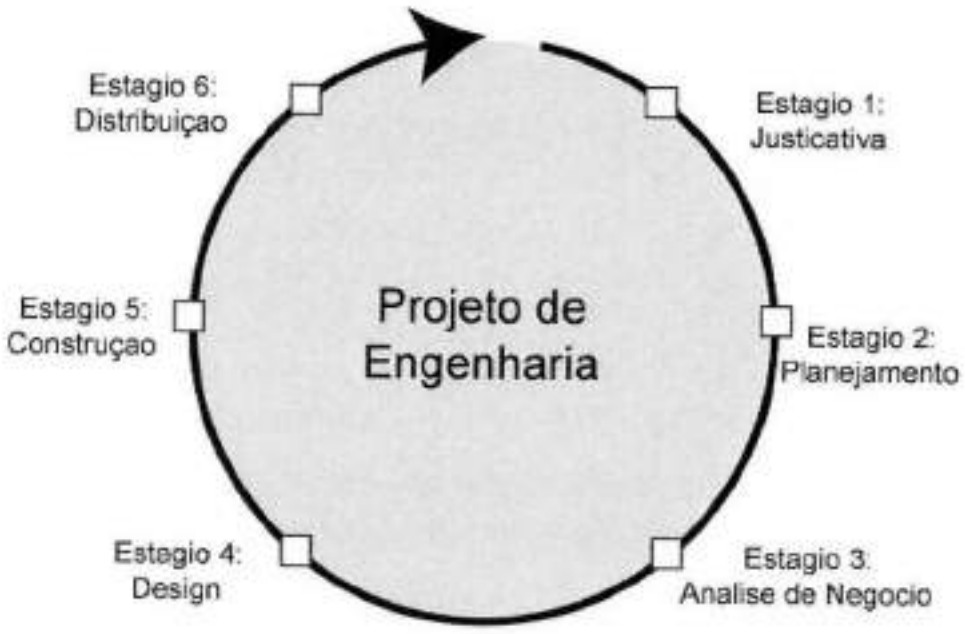


Figura 12: Estágios de Projeto de Engenharia

Como a seta na figura indica, processos de engenharia são interativos. Depois de distribuído, um produto é continuamente aprimorado baseado no *feedback* da comunidade que

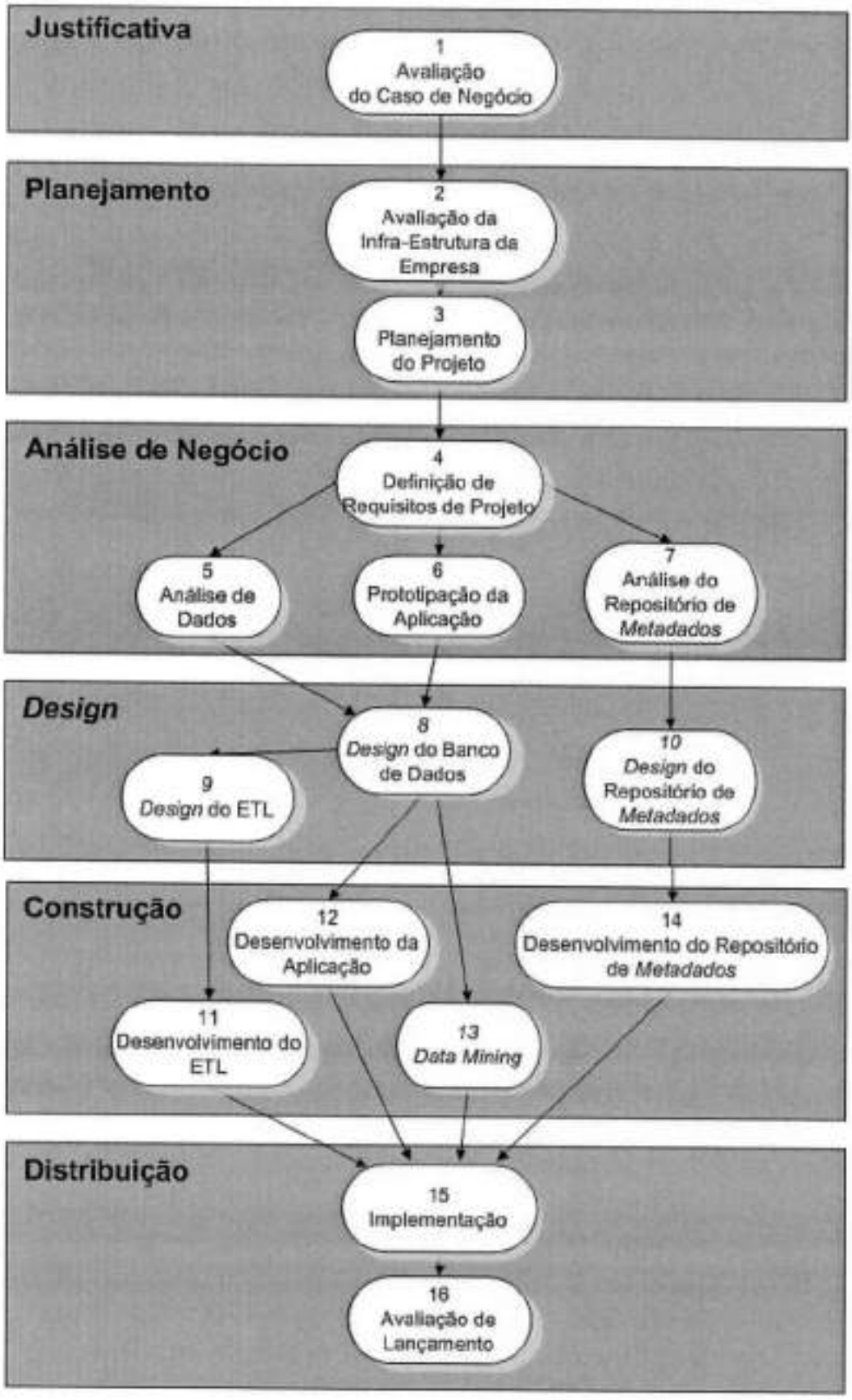
utiliza o produto. No corrente trabalho, ter-se-á a oportunidade de apenas passar uma vez por este ciclo.

Tabela 6 - Estágios de projeto de Engenharia.

Estágio 1.	Justificativa: Avaliar a necessidade de negócio que dá origem ao novo projeto de engenharia.	Estágio 4.	Desenho: Conceber um produto que resolve o problema de negócio ou possibilita uma oportunidade de negócio.
Estágio 2.	Planejamento: Desenvolver planos estratégicos e táticos, que guiam como o projeto de engenharia será realizado e completado.	Estágio 5.	Construção: Construir o produto, que deve prover um retorno sobre investimento dentro de um pré-determinado espaço de tempo.
Estágio 3.	Análise de Negócios: Fazer análise detalhada do problema de negócio ou oportunidade de negócio para ganhar um entendimento sólido dos requisitos de negócio para uma potencial solução (produto).	Estágio 6.	Distribuição: Implementar ou vender o produto acabado, então medir sua efetividade para determinar se a solução atende, excede ou falha em relação ao retorno sobre investimento esperado.

3.1.1 Estágios de Engenharia e passos de Desenvolvimento

Projetos de BI estão organizados de acordo com os mesmo seis estágios comuns a qualquer projeto de engenharia. Para cada estágio de engenharia, certos passos são seguidos para guiar o projeto para a sua completude. A metodologia adotada descreve 16 passos de desenvolvimento, como citados de modo resumido logo abaixo a figura a seguir.


Justificativa
1
Avaliação
do Caso de Negócio
Planejamento
2
Avaliação da
Infra-Estrutura da
Empresa
3
Planejamento
do Projeto
Análise de Negócio
4
Definição de
Requisitos de Projeto
5
Análise de
Dados
6
Prototipação da
Aplicação
7
Análise do
Repositório de
Metadados
Design
8
Design do Banco
de Dados
9
Design do ETL
10
Design do
Repositório de
Metadados
Construção
12
Desenvolvimento da
Aplicação
14
Desenvolvimento do Repositório
de Metadados
11
Desenvolvimento do
ETL
13
Data Mining
Distribuição
15
Implementação
16
Avaliação de
Lançamento

3.1.1.1 Estágio de Justificativa

Passo 1: Avaliação do Estudo de Caso

O problema ou oportunidade de negócio é definido e uma solução BI é proposta. Cada lançamento de aplicação BI deve ser justificada por custo e deve claramente definir os benefícios de tanto resolver um problema de negócio ou tomar vantagem de uma oportunidade de negócio.

3.1.1.2 Estágio de Planejamento

Passo 2: Avaliação da Infra-estrutura da Empresa

Já que aplicações BI são iniciativas *cross-organizational*, a infra-estrutura de uma empresa deve ser criada para suportá-la. Alguns componentes de infra-estrutura devem já estar posicionados antes que o primeiro projeto de BI é lançado. Outros componentes de infra-estrutura devem ser desenvolvidos no decorrer do tempo como parte dos projetos BI. A infra-estrutura de uma empresa tem dois componentes:

1. Infra-estrutura técnica, que inclui *hardware*, *software*, *middleware*, sistemas de gerenciamento de banco de dados, sistemas operacionais, componentes de rede, repositórios de meta dados, utilidades e assim por diante.
2. Infra-estrutura não-técnica, que inclui padrões de meta dados, padrões de nomes de dados, modelo lógico de dados da empresa, metodologias, guias bases, procedimento de testes, processos de mudança de controle, procedimentos para gerenciamento de problemas e resolução de conflitos e assim por diante.

Passo 3: Planejamento do Projeto

Projetos de suporte de decisão de BI são extremamente dinâmicos. Mudança de escopo, equipe de funcionários, orçamento, tecnologia, representantes de negócios e patrocinadores podem impactar severamente no sucesso do projeto. Conseqüentemente um planejamento de projeto deve ser detalhado, e o progresso atual deve ser analisado e reportado.

3.1.1.3 Estágio de Análise de Negócios

Passo 4: Definição dos requisitos do Projeto

Gerenciar o escopo de um projeto é uma das tarefas mais difíceis em projetos de suporte de decisão BI. A vontade de querer fazer tudo instantaneamente é difícil de conter, mas conter esta vontade é um dos mais importantes aspectos de negociar requisitos para cada *deliverable* (entregáveis). Times de projeto devem esperar que estes requisitos mudem ao longo do ciclo de desenvolvimento assim que o pessoal de negócio aprende mais sobre as possibilidade e limitações da tecnologia de BI durante o projeto.

Passo 5: Análise dos Dados

O maior desafio para todos os projetos de suporte de decisão BI é a qualidade dos dados de origem. Maus hábitos desenvolvidos através das décadas são difíceis de quebrar, e os danos resultantes de maus hábitos são muito caros, consomem tempo, e são tediosos para encontrar e corrigir. Para completar, análise de dados no passado está confinada apenas a visão de uma linha de negócios e nunca foi consolidada ou reconciliada com outras visões da organização.

Passo 6: Prototipação da Aplicação

Análise de *deliverables* funcionais, que costumavam ser chamadas de análises de sistema, é melhor feito através da prototipação para que possa ser combinada com o desenho da aplicação. Novas ferramentas e linguagens de programação permitem aos desenvolvedores a rapidamente provar que um conceito ou uma idéia são válidas. Prototipação também permite que o pessoal de

negócios possam ver o potencial e os limites da tecnologia, o que dá a oportunidade de ajustar seus requisitos de projeto e suas expectativas.

Passo 7: Análise do Repositório de Meta Dados

Ter mais ferramentas significa ter mais meta dados técnicos juntamente com os meta dados de negócios, que são geralmente capturados em uma ferramenta de modelagem CASE (*Computer-Aided-Software Engineering*). Meta dados técnicos precisam ser mapeados para os meta dados de negócios e, todos os meta dados devem ser armazenados em um repositório de meta dados. Repositórios de Meta dados pôem ser licenciados (comprados) ou construídos. Em qualquer caso, os requisitos para aquele tipo de meta dado para capturar e armazenar devem ser documentados em um meta modelo lógico que deve ser comparado com o meta modelo do fornecedor, se existir. Para acrescentar, os requisitos para entregar meta dados para a comunidade de negócios tem de ser analisadas (como por exemplo, função de ajuda *online*).

3.1.1.4 Estágio de Desenho

Passo 8: Desenho da Base de Dados

Um ou mais banco de dados de BI de destino irão armazenar dados de negócio de uma forma detalhada ou agregada, dependendo dos requisitos de reportagem da comunidade de negócios. Nem todos os requisitos de reportagem são estratégicos, e nem todos são multidimensionais. O design de *schemas* de banco de dados deve atender os requisitos de acesso de informação com a comunidade de negócios.

Passo 9: Desenho de ETL (*Extract/Transform/Load*)

O processo de ETL é o mais complicado do projeto de decisão de suporte de BI inteiro. É também o menos glamoroso. As janelas de processamento de ETL (janelas *batch*) são tipicamente pequenas, e ainda a baixa qualidade dos dados de origem geralmente querer muito

tempo para executar a transformação e limpeza dos programas. Finalizar o processo de ETL com as janelas *batch* disponíveis é um desafio para a maioria das organizações.

Passo 10: Desenho do Repositório de Meta Dados

Caso o repositório já existe, o processo de desenho deve apenas atualizar as características que foram documentadas em um modelo lógico e não foi fornecido pelo produto. Já no caso em que o repositório de meta dados esta sendo criado, a decisão que deve ser feita é a escolha do tipo do desenho do repositório: entidade-relacionamento ou orientado a objetos. Em qualquer caso, o desenho deve atender os requerimentos no modelo lógico dos dados.

3.1.1.5 Estágio de Construção

Passo 11: Desenvolvimento *Extract/Transform/Load*

Várias ferramentas estão disponíveis para o processo ETL, algumas sofisticadas e outras simples. Dependendo dos requisitos para a limpeza dos dados e da transformação desenvolvida durante o o passo de Análise de Dados (passo 5), e o desenho de ETL (passo 9), uma ferramenta ETL pode ou não ser a melhor solução. Em qualquer caso, pré-processar o dado e escrever extensões para suplementar as capacidades da ferramenta de ETL são freqüentemente requeridas.

Passo 12: Desenvolvimento da Aplicação – OLAP

Assim que o esforço da prototipação estabeleça a maioria (se não todos) dos requisitos funcionais, o desenvolvimento de aplicação de análise pode se iniciar. Desenvolver uma aplicação pode ser uma simples finalização de um protótipo operacional, ou então um desenvolvimento que requer mais esforço usando diferentes, ou mais ferramentas de análise ou acesso mais robustos. Em qualquer caso, as atividades de desenvolvimento da aplicação *front-end* geralmente são feitas em paralelo com as atividades de *back-end* como o desenvolvimento ETL e o desenvolvimento do repositório de meta dados.

Passo 13: *Data Mining*

Muitas organizações não utilizam seus ambientes de BI na sua totalidade. Aplicações de BI geralmente são limitadas a reportagens pré-escritas, algumas das quais não são mesmo novos tipos de reportagens, mas sim substitutos dos antigos. O verdadeiro retorno vem da informação escondida nos dados da organização, que podem ser descobertos somente através das ferramentas de mineração.

Passo 14: Desenvolvimento de repositório de Meta dados

Se a decisão escolhida for desenvolver repositório de meta dados do que licenciar um, um time separado é geralmente encarregado com o processo de desenvolvimento. O que torna assim um subprojeto menor dentro do projeto total de BI.

3.1.1.6 Estágio de Distribuição

Passo 15: Implentação

Uma vez que a equipe testou completamente todos os componentes da aplicação do BI, a equipe faz um *roll out* das bases de dados e as aplicações. O treinamento é programado para a equipe de funcionários do negócio e outros *stakeholders* que estarão usando a aplicação de BI e o repositório de meta dados. As funções de suporte iniciam-se mantendo os trabalhos das bases de dados do BI, programando e executando os *batch jobs* do ETL, monitorando o desempenho e ajustando a bases de dados.

Passo 16: Avaliação do lançamento

Após o final da implementação da aplicação, é muito importante beneficiar-se das lições aprendidas de projetos anteriores. Quaisquer *deadlines* ultrapassados, custos adicionais, conflitos e resolução de conflitos devem ser examinados, e ajustamento de processos devem ser feitos antes que o próximo lançamento se inicie. Quaisquer ferramentas, técnicas, guias bases e processos que não forem úteis devem ser reavaliados, ajustados ou possivelmente até mesmo descartados.

Os passos de desenvolvimento não necessariamente precisam ser executados em seqüência, a maioria dos projetos em times tendem a executá-los de modo paralelo. No entanto existem dependências entre um estágio e outro, o que cria um certo gargalo no desenvolvimento do projeto.

4 FERRAMENTAS DE BI

Neste capítulo serão apresentadas as ferramentas de BI pesquisadas para o projeto. Tratam-se de ferramentas de ETL (*Extract,Transform, Load*), OLAP (*Online Analytical Processing*) e de Portais *Web*, todas *open source*.

Em um projeto de BI, essas ferramentas são pesquisadas em suas respectivas etapas da metodologia, lembrando que uma solução de BI deve ser sempre orientada ao negócio e não pela tecnologia. A apresentação da pesquisa aqui neste capítulo é útil na medida em que serve como material de referência de ferramentas de BI *open source*. Ferramentas de *Data Mining* não foram pesquisadas, pois não se adequam ainda ao negócio, já que não há dados em quantidade suficiente que justifique o seu emprego.

4.1 Ferramentas de ETL Open Source

Octopus

- *Enhydra Octopus* é uma avançada ferramenta de Extração, Transformação e Carga de dados.
- Pode ser conectar em qualquer *JDBC data sources*
- Transformações são definidas em XML.
- Suporta *Ant* e *Junit*
- Inclue *drivers* para CSV e XML.
- Status de desenvolvimento: Estável, Maduro.
- Ambiente: Console (*Text Based*)
- Licença: GNU General *Public License* (GPL), GNU *Lesser General Public License* (LGPL)
- Sistema Operacional: Windows NT/2000, *Linux*
- Linguagem de Programação: Java, PL/SQL

- Site: http://octopus.objectweb.org/

Clover

- *CloverETL* é um ETL baseado em Java e pode ser utilizado para transformação estruturada de dados.
- Pode ser utilizado de forma independente ou embarcado na aplicação
- Licença: LGPL.
- Tipos de dados suportados: *String*, Numérico, Data e Byte.
- Converte dados de/para diversos tipos de caracteres (ASCII, UTF-8, ISO-8859-1,ISO-8859-2)
- Compatível com diverso banco de dados através de *drivers* JDBC.
- Define valores padrões para campos.
- Site: http://cloveretl.berlios.de/

BEE

- O BEE pode ser definido como um conjunto de ferramentas que suporta a implementação de um projeto de *Business Intelligence* incluindo uma ferramenta de ETL (Extração, Transformação e Carregamento dos dados) e um simples OLAP.
- São programas desenvolvidos na linguagem C e *Perl*
- Banco de dados: *Perl* DBI/DBD, *MySQL, Oracle, Other network-based* DBMS, *PostgreSQL* (pgsql)
- Licença : GNU General Public License (GPL)

- Sistema Operacional : All POSIX (Linux/BSD/UNIX-like OSes), Linux
- Translations :Theco, English
- Interface: X Window System (X11), Web-based
- Site: http://sourceforge.net/projects/bee/

Outras ferramentas levantadas mas que não foram consideradas neste estudo:

JETSTREAM

- http://sourceforge.net/projects/jetstream

Pequel Data Transformer

- http://sourceforge.net/projects/pequel

Cplusql ETL tool

- http://sourceforge.net/projects/cplusql

OpenDigger

- http://sourceforge.net/projects/opendigger

Simple ETL

- http://sourceforge.net/projects/simpleetl

OpenETL

- http://sourceforge.net/projects/openetl

Open Source ETL project

- http://sourceforge.net/projects/opnsrcetl

Migration Machine ETL

- http://sourceforge.net/projects/freemetal

4.1.1 Funcionalidades do ETL

De acordo com o estudo das principais ferramentas de ETL *Open source*, escolheu-se a ferramenta *Octopus* já que ela se mostrou a mais estável e madura com relação às outras. Além disso, o *Octopus* apresenta diversas características que são aderentes à necessidade de implementação do projeto.

O *Octopus* realiza transformações durante a carga de dados de um banco de dados ou arquivo de origem para um banco de destino. Além disso, o *Octopus* cria chaves artificiais, executa comandos SQL antes, durante e depois do processo de ETL, cria tabelas, índices, chaves primárias, chaves estrangeiras e carga de dados.

Outras características: Carga de dados através de JDBC *data source* (banco de dados) para JDBC *data target* (banco de dados) além de desempenhar transformações definidas em XML.

A aplicação pode utilizar diversos tipos de banco de dados: MSSQL, MySql, Access, Excel, Csv, PostgreSQL, Qed, InstantDB, XML, BorlandJDataStore, Oracle, HSQL, McKoi, DB2, Sybase e Paradox database.

A aplicação possui outras características como: criação de banco de dados, criação de tabelas, inserção de dados em um banco de dados vazio, inserção de dados em um banco não vazio, atualização de colunas, atualização de colunas com o horário do sistema, atualiza colunas com o Id do usuário e executa comandos SQL.

- **Arquitetura do Octopus**

Basicamente o *Octopus* é dividido em 2 partes: *OctopusGenerator* e *OctopusLoader*.

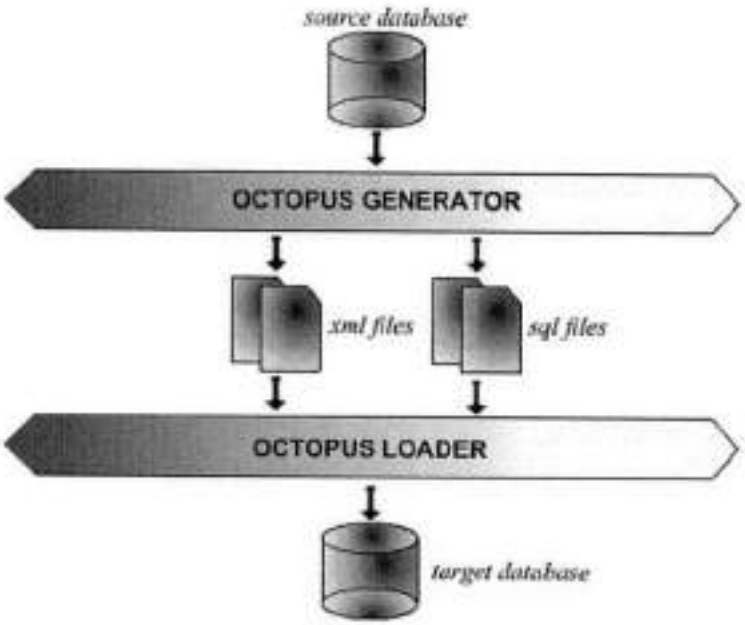


Figura 13: Arquitetura do Octopus

O principal objetivo do *Octopus Generator* é criar arquivos SQL e XML através dos dados de origem.

Os arquivos SQL incluem criação de banco de dados, tabelas, chaves primárias e chaves estrangeiras. O usuário pode escolher qual arquivo SQL será criado.

Os arquivos XML descrevem as relações entre os dados e as tabelas (loadJob.xml and importDefinition.xml). As regras de transformações são escritas no XML também. O número de transformações em um arquivo XML loadJob não é limitado.

Já o *OctopusLoader* é uma ferramenta *Java-based* de Extração, Transformação e Carga que transfere dados de um banco de dados JDBC para um outro banco de dados JDBC. Ele pode se conectar com qualquer banco de dados JDBC e executar transformações definidas no XML. Ou seja, o *Octopus Loader* sempre necessitará dos arquivos XML e SQL criados pelo Octopus Generator. O *OctopusGenerator* e OctopusLoader dividem a mesma GUI (*graphic user interface*).

Além disso, o *Octopus* pode também ser usado para '*backup*' e '*restore*' de banco de dados.

4.1.2 Configuração do ETL

Através do *Octopus loader* é possível importar e transformar os dados. As regras de importação e transformação podem ser definidas nos arquivos XML.

Para importar e mapear os dados, devem ser configurado o arquivo ImportDefinition.xml e definir alguns dos principais campos:

Tabela 7 - Importação de dados

Atributos	Descrição
Name	Nome da definição de importação
TableName	Nome da tabela de origem no banco de dados de origem
SelectStatement	Comando SQL que carrega os dados da tabela de origem
CommitCount	Número de registros que sofrerão confirmação de gravação.
LogMode	Define o tipo de log.
objectIDIncrement	Número de Identificação
ObjectIDTableName	Define o nome da tabela
ObjectIDColumnName	Define o nome da coluna

No caso de transformações deve ser configurado o seguinte arquivo: ImportTransformation.xml e definir as regras através de tags do tipo <transformations>

Tabela 8 - Transformação de dados

Atributo	Descrição
Name	Nome da transformação
transformatorClassName	Nome da classe que transforma os dados (eg. org.webdocwf.util.loader.TransformMyData)
transformatorConfig	Parâmetro que pode ser enviado para a classe TransformMyData

Além disso, é necessária a criação de uma classe Java que contém toda a lógica da transformação. Exemplo:

```
package org.webdocwf.util.loader;

import java.util.List;
public interface Transformer {
/**
 * Configura transformação
 */
public void configure(String s);
public void release();
/**
 * Este método retorna a lista com os valores transformados para a coluna de origem
 */
public List transformValue(List valueToTransform);
}
```

4.2 Ferramentas OLAP Open Source

Nesta fase foi realizada inicialmente uma pesquisa sobre as ferramentas de BI *open source* disponíveis no mercado. Uma dificuldade que se teve nesta fase foi a de que muitas alternativas de OLAP estão em fases iniciais de desenvolvimento e não possuem uma versão estável para testes. As alternativas pesquisadas estão listadas abaixo.

JPivot

- O JPivot é uma ferramenta em JSP (Java Servlet Pages) que renderiza tabelas OLAP e gráficos. Os usuários podem efetuar operações típicas do OLAP como *drill down*, *slice* e *dice*. Para estas operações, ele utiliza a ferramenta Mondrian.
- http://jpivot.sourceforge.net/

Mondrian

- O Mondrian é uma ferramenta OLAP desenvolvida em Java. Ela executa buscas no banco de dados escritas na linguagem MDX e busca informações de uma base de dados relacional e apresenta os resultados em um formato multidimensional.
- http://mondrian.sourceforge.net/

BEE

- O BEE pode ser definido como um conjunto de ferramentas que suporta a implementação de um projeto de *Business Intelligence* incluindo uma ferramenta de ETL (Extração, Transformação e Carregamento dos dados) e um simples OLAP.
- São programas desenvolvidos na linguagem C e Perl
- Banco de dados: Perl DBI/DBD, MySQL, Oracle, Other network-based DBMS, PostgreSQL (pgsql)
- Licença : GNU General Public License (GPL)
- Sistema Operacional : All POSIX (Linux/BSD/UNIX-like OSes), Linux
- Translations :Theco, English
- Interface: X Window System (X11), Web-based

Outras ferramentas pesquisadas

OpenRolap

- http://sourceforge.net/projects/openrolap

OLAP4J

- http://sourceforge.net/projects/olap4j

GnuOLAP

- http://sourceforge.net/projects/gnuolap

Cubeb

- http://sourceforge.net/projects/cubeb

Generic Open Database Interface

- http://sourceforge.net/projects/godi

Macondo OLAP

- http://sourceforge.net/projects/macondoolap

Open OLAP

- http://sourceforge.net/projects/openolap

Apos este estudo, a opção feita pelo grupo foi a ferramenta Mondrian, pois é um projeto estável, desenvolvido em Java, o que permite a utilização do mesmo em outros sistemas operacionais e possui uma equipe de desenvolvedores muito ativa, que disponibiliza atualizações com freqüência.

4.2.1 Funcionalidades

O Mondrian é uma ferramenta poderosa que implementa grande parte das operações apresentadas acima. Além disso, ela disponibiliza funcionalidades importantes para o analista de negócios como a geração de diversos tipos de gráfico e a exportação dos dados para uma planilha Excel.

Dentre as operações citadas no capítulo referente ao OLAP, o mondrian executa o *drill up/down, rotate e slice*. A operação de *drill across* não é permitida, pois o Mondrian não reconhece mais do que uma tabela fato. Isso não gera impacto no nosso projeto, pois esta operação não é necessária devido a presença de apenas uma tabela fato no modelo lógico do banco de dados.

A ferramenta Mondrian é composta por quatro camadas: a camada de apresentação, a camada dimensional, a camada *star* e a camada de armazenamento.

A primeira camada é a camada de apresentação determina as informações que o usuário final tem acesso em seu monitor e como ele pode interagir para efetuar novas buscas de informação. Existem várias formas de apresentação dos dados multidimensionais, entre eles, tabelas, gráficos do tipo pizza, gráfico de barras e ferramentas avançadas de visualização.

A segunda camada é a camada dimensional. Esta camada dimensional efetua uma análise gramatical (*parse*), valida e executa as *queries* MDX. O modelo dimensional é descrito por um meta dado e é através deste que se faz o mapeamento em um modelo relacional.

A terceira camada é a camada *star*, e é responsável por manter um *cache* de agregação. Uma agregação é um conjunto de *measure values* armazenadas na memória, qualificadas através de um conjunto de valores das dimensões envolvidas. A camada dimensional envia requisições de conjuntos de registros, que inicialmente são verificadas no *cache* de agregação. Caso estes registros não sejam encontradas no *cache* ou não possam ser derivadas através de *drill up/down* de alguma agregação em *cache* o gerenciador de agregações invia a requisição para a camada de armazenamento.

A quarta camada é a camada de armazenamento, que é representada por um SGBD relacional. Esta camada é responsável por fornecer registros agregados e membros das tabelas de dimensão.

Estes componentes podem estar situados no mesmo computador ou podem estar distribuídos. As camadas dois e três, que compõem o Mondrian Server, devem estar no mesmo computador. A camada de armazenamento pode estar em outro computador, que será acessado através de um JDBC remoto. Em um sistema multi-usuário, a camada de apresentação pode estar nos computadores de cada usuário final.

4.2.2 Configuração do OLAP

Para que o Mondrian seja executado corretamente, primeiramente foi definida a forma de acesso ao banco de dados. Configurou-se uma aplicação chamada MySQL Connector ODBC para que o Mondrian possa ter acesso as informações armazenados no banco de dados MySQL.

Após isso, o próximo passo foi o reconhecimento dos dados armazenados no *data warehouse*. O Mondrian faz este reconhecimento através de um mapeamento feito em um arquivo XML. Este arquivo possui um esquema que define um banco de dados multidimensional. Nele está contido um modelo lógico (que no nosso caso é um *star schema*), que é formado por cubos, hierarquias, níveis e seus membros.

4.3 Ferramentas de Portal Open Source

Uportal

- **Descrição:** É um portal gratuito, desenvolvido por instituições de ensino (principalmente universidades) de forma colaborativa, através dos esforços dos membros das instituições.Utiliza Java, XML, JSP e J2EE.
- **Site:** http://mis105.mis.udel.edu/ja-sig/uportal/index.html
- **Versão:** 2.3.3
- **Língua:** Inglês
- **Requisitos:** Java - JDK, Jakarta Ant, Servlet container (Apache TomCat, por exemplo), Handling SQL database queries via JDBC
- **Licença:** Grátis (GPL)

Exo platform

- **Descrição:** O eXo platform é um poderoso portal *Open Source* construído através de diversos módulos É baseada no Java Server Faces, Pico Container, JbossMX e AspectJ.
- **Site:** http://exo.sourceforge.net/
- **Versão:** 1.0
- **Língua:** Inglês, Francês, Português.

- **Requisitos:** Java
- **Licença:** Grátis (GPL)

Liferay

- **Descrição:** Liferay é um portal que foi projetado para suportar *portlets* que aderem a norma JSR168.
- **Site:** http://www.liferay.com/home/index.jsp
- **Versão:** 1.5
- **Licença:** Grátis (GPL)

GridSphere

- **Descrição:** O GridSphere portal *framework* fornece *portlets open-source* baseados em portais Web. GridSphere permite que desenvolvedores possam utilizar e integrar *portlets* de maneira simples e administradas através de *portlet* container.
- **Site:** http://www.gridsphere.org/
- **Versão:** 2.0
- **Língua:** Inglês.
- **Requisitos:** Java, Jakarta TomCat, Jakart Ant.
- **Licença:** Grátis (GPL)

JetSpeed

- **Descrição:** JetSpeed é um *framework* portal *web* que permite que seus usuários possam personalizar suas páginas de forma bem simples. Os usuários podem construir suas páginas selecionando os *portlets* disponíveis.
- **Site:** http://portals.apache.org/jetspeed-1/
- **Versão:** 1.5
- **Língua:** Inglês
- **Requisitos:** Java

4.3.1.1 Funcionalidades

O Jetspeed é um *framework* para construção de portais baseados em *Portlets*. Na implementação do projeto, foi adotada a versão 1.5. Uma das principais vantagens do *Jetspeed* é a sua robustez além da facilidade para utilizar o grande número de *portlets* prontos disponíveis, como por exemplo notícias, pesquisas e fóruns de discussão.

O JetSpeed costuma ser utilizado como solução para Portais, *Intranets*, *Extranets* e Sistemas de Gerenciamento de Conteúdo.

5 ESTUDO DE CASO

5.1 Avaliação do Estudo de Caso

5.1.1 Justificativa de negócio

Criar um ambiente de BI geralmente custa milhões de dólares e, uma organização considerando tais iniciativas necessita de uma estratégia de BI e uma justificativa de negócio para mostrar o balanço entre custos envolvidos e os benefícios ganhos. Uma decisão de iniciativa de BI provê inúmeros benefícios, não somente benefícios tangíveis como aumentar o volume de vendas, mas também benefícios intangíveis como aumentar a reputação de uma organização. Muitos destes benefícios, especialmente os intangíveis, são difíceis de quantificar em termos de valor monetário. Apesar dos benefícios genéricos de iniciativas de BI estar documentadas largamente, eles não podem justificar a iniciativa de um projeto a menos que se possam associar tais benefícios com os problemas de negócios específicos da organização e das metas estratégicas de negócio.

A Justificativa de uma iniciativa de BI deve sempre ser direcionada pela área de negócio e não pela tecnologia. Aliás, não seria inteligente configurar um ambiente de suporte de decisão de BI caro somente para experimentar uma nova tecnologia. Conseqüentemente, cada aplicação BI proposta deva reduzir mensuravelmente o chamado *business pain* (MOSS, 2003) (problemas que afetam o lucro ou a eficiência de uma organização).

No caso de estudo corrente, o provimento de uma solução de BI é válida já que se trata de prover uma solução para o grupo PENSA, que visa fazer estudos na área de agronegócios sobre tendências dos negócios industriais. Como benefícios para a organização, tem-se a automação no processo de obtenção e geração de relatórios e gráficos. Que poderão tirar proveito das análises multidimensionais e históricas, que são características dos benefícios provenientes de ferramentas como OLAP juntamente com o *Data warehouse*.

Com um sistema de *Business Intelligence*, o grupo PENSA terá um grande auxílio na obtenção de seus estudos, fornecendo assim subsídios à tomada de decisão e ao planejamento estratégico de organizações privadas e públicas de forma mais simples e ágil.

Pela justificativa de custo, é visivelmente vantajoso uma vez que se trata de um projeto acadêmico sem fins lucrativos e também pela postura de se adotar ferramentas *Open Source*, de uso gratuito.

5.2 Avaliação da Infra-estrutura da Empresa

A infraestrutura de uma empresa consiste em dois grandes componentes:

- Infraestrutura técnica, como hardware, *middleware*, o sistemas de gerenciamento de banco de dados (SGDB)
- Infraestrutura não técnica, como padrões, meta dados, regras de negócios e políticas

5.2.1 Avaliação da Infra-estrutura técnica da Empresa

Hardware

Um novo hardware não será necessário, podendo ser utilizado as máquinas atuais do grupo PENSA, não necessitando de máquinas muito potentes. Ainda mais que a carga de dados é feita teoricamente mensalmente e, por isso o tempo de processamento não é crítico. Ao contrário do que acontece na maioria das empresas. Isso de deve aos dados já virem sumarizados.

Rede

Não se aplica ao caso de estudo, já que se trata de um piloto que não entrará em produção a princípio e também porque o seu uso é muito limitado a poucas pessoas.

Middleware

Não há *middleware* sendo utilizado anteriormente, sendo portanto especificado pelo projeto de BI.

A arquitetura operacional de origem viria de um banco de dados relacional operacional na qual seriam cadastrados os dados através de um portal. Como era uma dependência externa que teve contratempos, já que ela não foi mais desenvolvida por falta de comprometimento externo,

foi incorporado no escopo do projeto o seu desenvolvimento e a sua carga de dados para uso no projeto de BI.

Sistema de Gerenciamento de Banco de Dados

Por ser um trabalho acadêmico voltado para um grupo acadêmico e ser um piloto, optou-se pelo uso de SGDB gratuitos, no caso o MySQL. Que é compatível os sistemas operacionais Linux e Windows. Juntamente com o MySQL, ferramentas gratuitas como o *MySQL Control Center* (*software desktop*) e o *phpMyAdmin* (*web*) podem ser utilizadas para o gerenciamento.

Ferramentas e Padrões

Atualmente a ferramenta utilizada para o estudo é o Excel da Microsoft. Seria desejável que a solução de BI fosse compatível com esta ferramenta.

5.2.2 Avaliação da Infra-estrutura não técnica

Modelo Lógico de Dados

Existia um modelo de base de dados para o desenvolvimento de um portal, que era objetivo inicial do grupo PENSA desenvolvido pela FIA. No entanto este teve seu projeto descontinuado devido a questão de patrocínio, e por isso o projeto do Portal foi descontinuado. Os modelos de dados que existem são as planilhas em Excel que contém os dados de produção e área colhida, por exemplo. Tais planilhas têm modelos, cada uma e, a partir desses arquivos em Excel será feito a extração e seu povoamento nas tabelas de banco de dados operacional, para simular o cenário inicial da proposição de solução BI.

Meta Dados

Não existe repositório de meta dados.

Padrões, Linhas Bases e Procedimentos

Por se tratar de uma organização acadêmica, muitos dos itens não são aplicáveis, o mais relevante é a linha mestre de qualidade de dados para mensuração de dados sujos e triagem de limpeza de dados que não existe, e pretende-se ser solucionada com o projeto de BI.

5.3 Planejamento do Projeto

O planejamento de projeto incluir criar um *project charter*, que define o projeto em termos de:

- Objetivos e metas
- Escopo (*deliverable* experado do projeto)
- Riscos
- Recursos
- Prazos

Como o planejamento do projeto em questão foi realizado e entregue em outra etapa, ele não será descrito no corrente documento.

5.4 Definição dos requisitos de projeto

5.4.1 Requisitos Funcionais

Os principais requisitos levantados para o projeto de *Business Intelligence* são:

- ***Web*** – Aplicação numa plataforma *web-based* onde será possível utilizar o sistema através de qualquer navegador compatível.
- **Planilha** – Funcionalidade que permita a exportação de tabelas em formato Excel e CSV.

- **Gráficos** – Exibição de gráficos para facilitar a análise das informações.
- **Filtros**- Filtros que permitam o usuário a selecionar a faixa de dados que devem ser apresentados na tabela.
- **Cruzamento de Informações** – O sistema deve permitir a interação do usuário para a montagem da tabela, através da seleção das informações a serem cruzadas.

5.4.2 Requisitos Não Funcionais

- **UI** – Interface do usuário amigável que facilite a utilização do sistema.
- **Segurança de acesso (*security*)** – O acesso será limitado através de autenticação no portal por senha simples
- **Manutenibilidade** – Não há requisito
- **Disponibilidade** – Não há requisito
- **Portabilidade** – Não há requisito
- **Segurança (*safety*)** - Não aplicável
- **Desempenho** – Não há requisito. (Geralmente, em empresas, este é um requisito crítico já que o processamento de dados pode levar em torno de 5 horas para ser realizado dependendo dos casos, devendo ser executados em processos batch, geralmente a noite, quando não há atividade na empresa para não influenciar nos processos operacionais. Como a entrada de novos dados é mensal e de baixo volume, o quesito desempenho não será levado em consideração.)

5.5 Análise de Dados

Para dar início ao projeto do *data warehouse*, de acordo com as fases consideradas) foi feito um levantamento das informações que poderiam ser armazenadas no mesmo. As informações são armazenadas em tabelas Excel e seu conteúdo é referente a valores de produção, importação, exportação, área plantada, área colhida e perdas do agronegócio.

A análise de dados externos ao sistema é feito pelos participantes do projeto PENSA, quando os mesmos efetuam a coleta de dados de sua origem para planilhas Excel. Após esta etapa, algumas diferenças ainda persistem, como a quantificações dos produtos exportados. Em alguns casos, estes valores são quantificados em toneladas e em outros casos, são quantificados em sacas. Estas diferenças serão solucionadas na implementação do ETL.

Outro ponto importante nesta etapa de analise dos dados é a qualidade dos mesmos. Este ponto é verificado no mesmo momento que a análise dos dados externos ao sistema, pelos participantes do projeto PENSA. Portanto, a qualidade dessas planilhas é de responsabilidade da área de negócios.

Como o *data warehouse* ainda não possui dados armazenadas e todas as informações serão inseridas seguindo esta política, não teremos problemas quanto aos dados históricos armazenados de forma errônea.

O segundo passo efetuado foi a divisão destas informações nas tabelas fato e dimensão. As informações presentes nos arquivos de origem foram agrupadas em cinco tabelas dimensão e uma tabela fato. As tabelas dimensão são: *GeographicDimension, InformationSourceDimension, ProductDimension, TimeDimension* e *TransactionTypeDimension*.

5.6 Projeto da Base de Dados

O projeto lógico do banco de dados foi baseado na técnica *star schema* devido às vantagens apresentadas com relação ao *snowflake schema*, como a menor quantidade de tabelas no banco de dados para gerenciar e isso traz como conseqüência um ganho na performance, pois a quantidade de *joins* necessários para as buscas de informação também diminui.

De acordo com o que foi citado no capítulo de Análise de dados, as tabelas dimensão criadas são: *GeographicDimension*, *InformationSourceDimension*, *ProductDimension*, *TimeDimension* e *TransactionTypeDimension*.

Estas tabelas dimensão e fato estão interligados entre si, formando o *star schema* apresentado na figura abaixo.

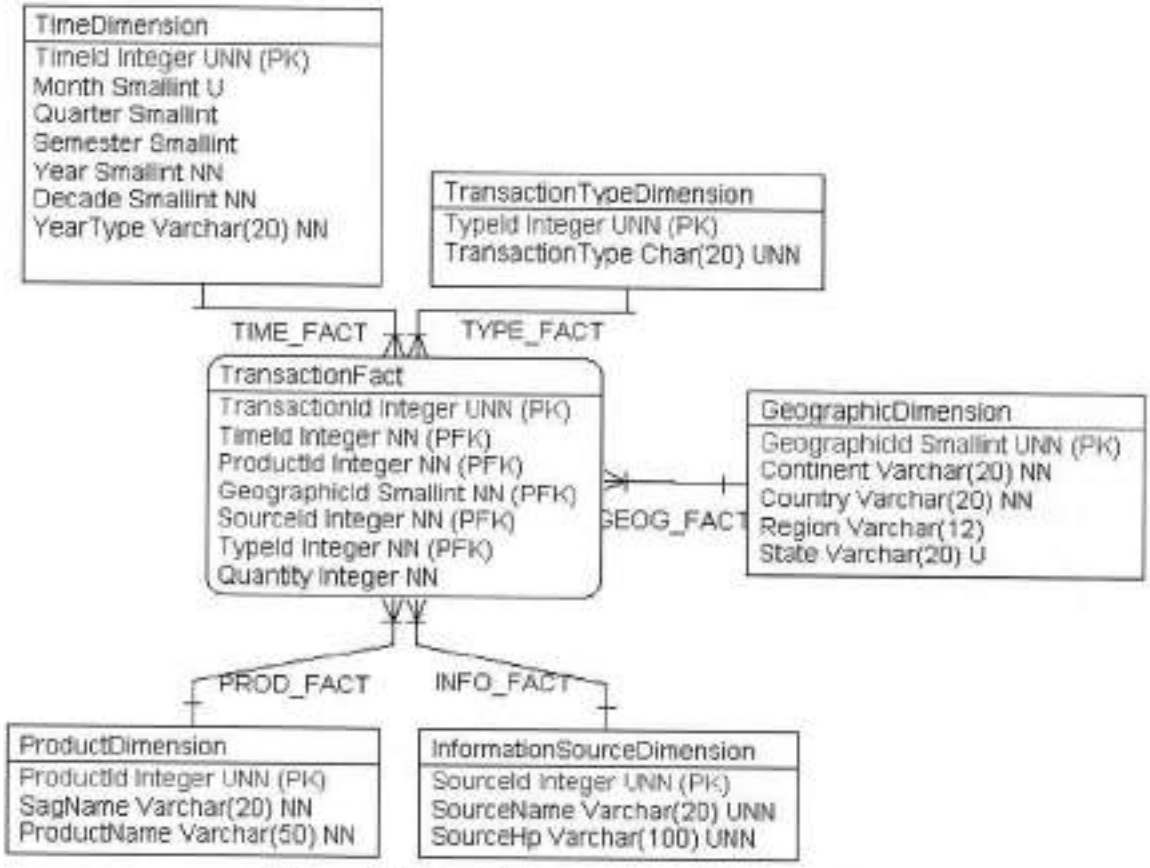


Figura 14: Star Schema para os dados do PENSA

A dimensão do lugar geográfico (*GeographicDimension*), armazena informações a respeito da localidade a que o dado se refere. Para armazenar esta informação, esta tabela foi projetada de modo a formar uma hierarquia entre suas colunas. As informações armazenadas nesta tabela são: o continente (coluna *Continent*), o país (coluna *Country*), a região (coluna *Region*) e o estado (coluna *State*). Estas duas últimas colunas são utilizadas somente no caso do Brasil, sendo desconsiderados nos outros países devido a ausência desses dados na tabela de origem, o que não impossibilita que o sistema possa vir a guardar estes dados futuramente. Alem destas colunas, a tabela possui uma coluna que serve como identificador da mesma (*GeographicId*). A tabela resultante é apresentada na figura abaixo.

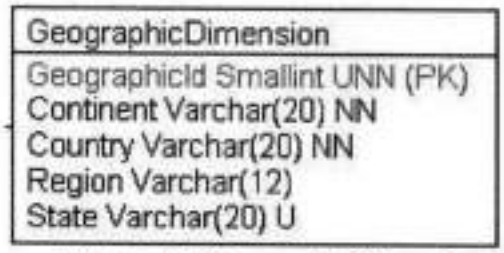


Figura 15: GeographicDimension

A dimensão da fonte de dados (*InformationSourceDimension*), armazena a origem das informações armazenadas. As informações armazenadas nesta tabela são: o nome da fonte de informação (coluna *SourceName*) e o endereço do site na internet (coluna *SourceHP*). Diferentemente da dimensão anterior, as colunas não possuem relações hierárquicas. A coluna que possui a função de identificador é a *SourceId*. A tabela resultante é apresentada na figura abaixo.

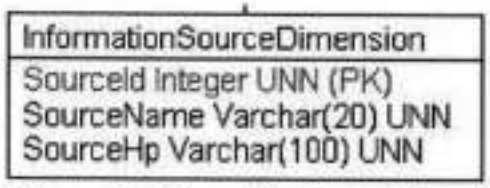


Figura 16: InformationSourceDimension

A dimensão do produto (*ProductDimension*), armazena informações a respeito do produto a que o dado se refere. Da mesma forma que a tabela *GeographicDimension*, esta tabela também possui uma hierarquia entre suas colunas. As informações armazenadas nesta tabela são: um subconjunto definido pela área de negócio (PENSA) que foi chamado como SAG (Sistema Agrário) (coluna SagName) e o nome do produto (coluna *ProductName*). A coluna que exerce a função de identificador é a coluna *ProductId*. A tabela resultante é apresentada na figura abaixo.

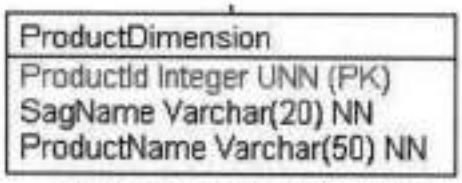


Figura 17: ProductDimension

A dimensão do tempo (*TimeDimension*), armazena informações do tempo a que o dado se refere. Esta tabela também possui hierarquia entre suas colunas. As informações armazenadas nesta tabela são: a década (coluna *Decade*), o ano (coluna *Year*), o semestre (coluna *Semester*), o trimestre (coluna *Quarter*) e o mês (coluna *Month*), alem do tipo do ano considerado (coluna *YearType*), que não se encaixa na hierarquia formada pelas outras colunas. Esta última coluna é necessária pois os dados estão em função de dois tipos de ano: o ano civil e o ano safra. A coluna que exerce a função de identificador é a coluna *TimeId*. A tabela resultante é apresentada na figura abaixo.

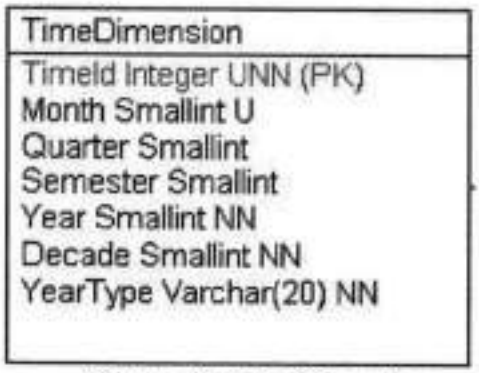


Figura 18: TimeDimension

A dimensão do tipo de transação (*TransactionTypeDimension*), armazena informações a respeito do tipo de transação que o dado se refere. Esta tabela possui apenas duas colunas. Uma das colunas armazena o tipo de transação. As opções para esta coluna são: Produção, Importação, Exportação, Área Plantada, Área Colhida e Perdas. A outra coluna exerce a função de identificador (*TypeId*). Esta tabela proporciona certa flexibilidade com relação aos dados a serem armazenados na tabela fato. Caso a área de negócio deseje armazenar outro tipo de dado, como por exemplo, o crescimento da área plantada com relação ao ano anterior, não será necessário uma mudança no modelo lógico do *data warehouse*, mas apenas a inclusão de um registro nesta tabela. A tabela resultante é apresentada na figura abaixo.

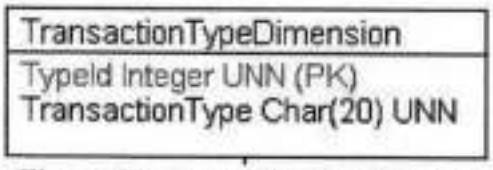


Figura 19: TransactionTypeDimension

A tabela fato (*TransactionFact*) armazena um dado numérico (coluna *Quantity*) referente a transação indicada na tabela *TransactionTypeDimension*, no periodo de tempo indicado na tabela *TimeDimension*, do produto indicado na tabela *ProductDimension*, da localidade indicada na tabela *GeographicDimension* e que foi extraido da fonte indicada na tabela *InformationSourceDimension*. Com isso, ela relaciona o valor a ser armazenado a um determinado valor de todas as tabelas dimensão através das colunas *TimeId*, *ProductId*, *GeographicId*, *SourceId* e *TypeId*. Alem destas colunas, a tabela possui uma coluna que serve como identificador da mesma (*TransactionId*). A tabela resultante é apresentada na figura abaixo.

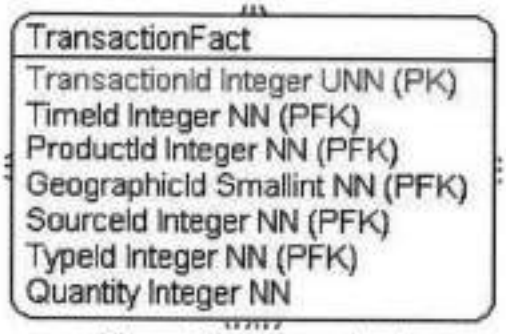


Figura 20: TransactionFact

Com relação aos pontos levantados para o projeto fisico do banco de dados, a alocação fisica dos dados, o particionamento, o *clustering* e o backup e recuperação dos dados não foram levados em conta para o projeto da base de dados do estudo de caso em questão devido ao volume de dados pequeno que este envolve.

A indexação foi considerada no projeto do *data warehouse*. As colunas selecionadas para indexação no modelo foram:

Tabela *TimeDimension*

- Coluna *Year*

Tabela *GeographicDimension*

- Coluna *Country*

Tabela *ProductDimension*

- Coluna *ProductName*

Estas colunas foram selecionadas pois estas colunas sofrem buscas frequentemente e os seus dados são bem distribuídos.

5.7 Projeto do ETL

5.7.1.1 Introdução

A metodologia utilizada na implementação do ETL consistiu basicamente num estudo inicial sobre as principais tecnologias *open sources* disponíveis no mercado, criação do mapeamento entre o banco de dados de origem e o banco de dados destino além da definição e implementação das regras de transformações.

Foram consideradas 2 bases de dados distintas que serão centralizadas no DW:

- Base de dados em *Mysql* que possui os dados de café (SAGCAFE)
- Arquivo Excel que possui os dados de cana-de-açúcar (CANA)

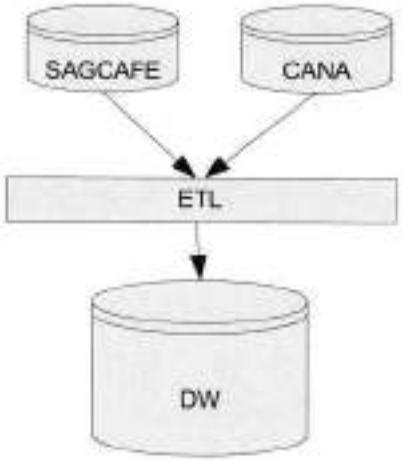


Figura 21: ETL – Estudo de caso

Maiores informações referente ao banco de dados operacional pode ser obtidas no Anexo D.

5.7.1.2 Mapeamento

Verificar anexo C.

5.7.1.3 Fluxo dos dados

- SAGCAFE

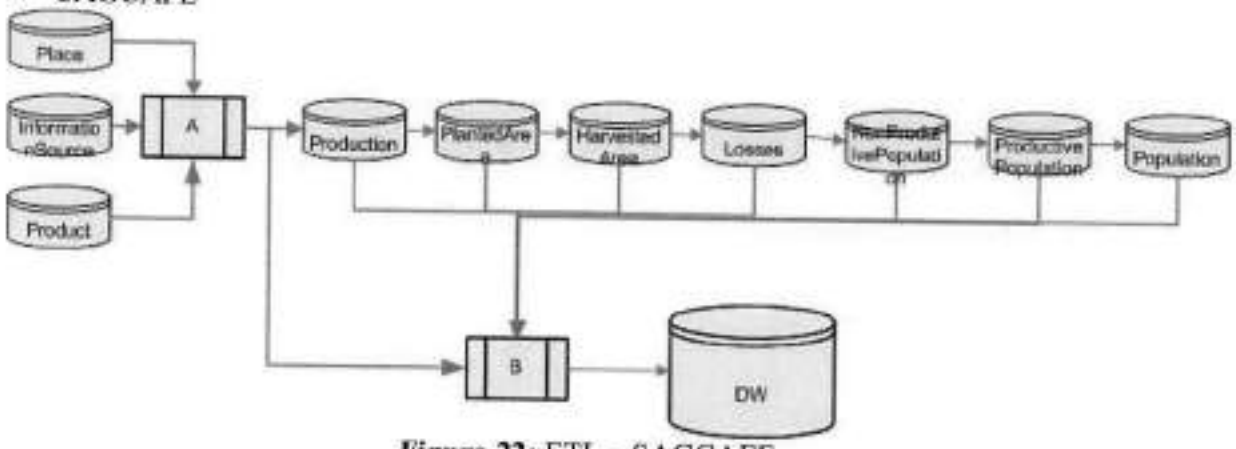


Figura 22: ETL – SAGCAFE

Processos utilizados para povoar o *Data Warehouse* através da base SAGCAFE:

Processo A:

Extração de dados das seguintes tabelas:

- *PlaceTable*
- *ContinentTable*
- *ProductTable*
- *SAGTable*
- *InformationSourceTable*

Processo B:

Extração de dados das seguintes tabelas:

- *ProductionTable*

- *PlantedAreaTable*
- *HarvestedAreaTable*
- *LossesTable*
- *NonProductivePopulationTable*
- *ProductivePopulationTable*
- *PopulationTable*

Além disso é realizada as seguintes transformações:

- Transformação da quantidade para a mesma unidade a partir das Colunas *Unit* e *Quantity*
- Os *campos Month,Quarter,Semester,Year,Decade* serão obtidos a partir da coluna *Time*
- *TransactionType* será obtido através do Nome da Tabela (*Production,PlantedArea, Harvested Area, LossesTables, NonProdutivePopulation, ProductivePopulation, Population*)

- **CANA**

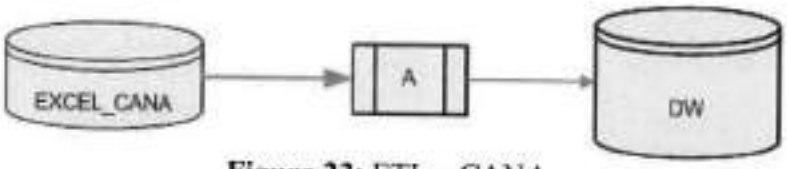


Figura 23: ETL – CANA

Processo A:

Povoamento do DW através da extração das informações contidas nas colunas do arquivo Excel.

Transformações:

- Todos os dados de Quantidade são transformados na mesma unidade.
- Década é obtida a partir do campo *Year*
- *Region* é obtido a partir do *States*

5.8 Desenho do repositório de META DADOS

Abaixo segue a apresentação de quatro modelos de repositório de meta dados.

5.8.1 Repositório de Meta Dados Centralizado

Solução mais popular e fácil de implementar porque existe apenas um banco de dados, tanto relacional quanto orientado a objetos e apenas uma aplicação para manter. Como no caso de estudo há várias aplicações a serem integradas, tal solução não será aplicada.

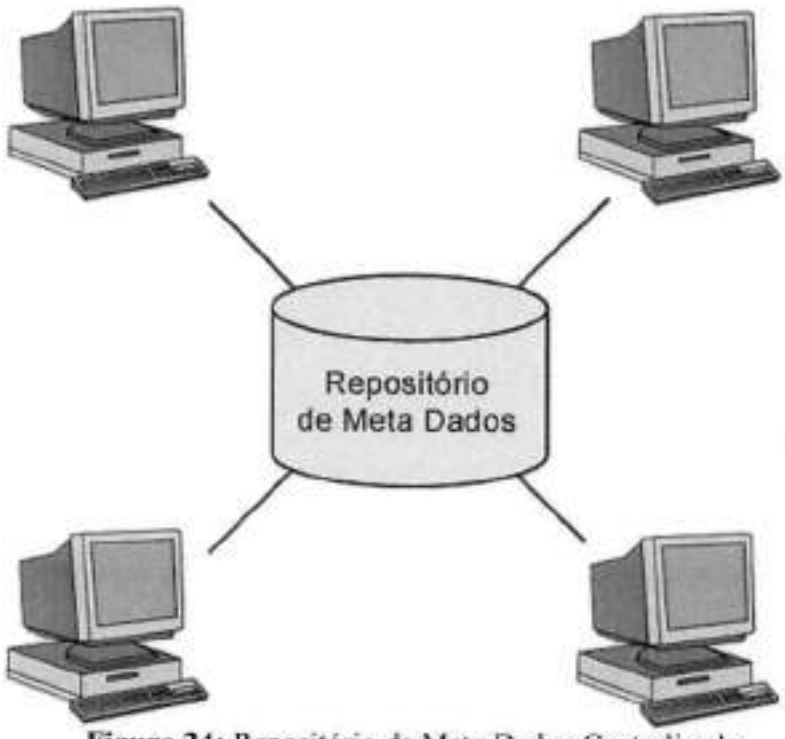


Figura 24: Repositório de Meta Dados Centralizado

Repositório construído de modo customizado

Tabela 9 - Vantagens e desvantagens de um Repositório de Meta Dados Centralizado Customizado

Vantagens	Desvantagens
Um desenho personalizado do banco de dados incorpora todos os requisitos de meta dados.	O tempo integral da equipe responsável é necessário para mandar a base de dados do repositório e seus relatórios.
O acesso *front end* e interfaces para ferramentas (ETL, OLAP, e assim por diante) são desenhados personalizadamente para atender todos os requisitos.	O acesso *front end* e ferramentas de interface devem ser programados e passarem por manutenção, o que consome tempo.
Relatórios assim como funcionalidades de ajuda são desenhadas exatamente como desejado.	O repositório de meta dados teria de ser aprimorado periodicamente (algumas vezes redesenhado) porque ele não pode ser construído com todas as funcionalidades desde o início.
Técnicos tem controle total do desenho e funcionalidade do repositório de meta dados.	Conteúdo pode ficar fora de sincronia com os dicionários de dados das ferramentas e do SGBD.

Repositório licenciado

Tabela 10 - Vantagens e desvantagens de um Repositório de Meta Dados Centralizado Licenciado

Vantagens	Desvantagens
Economia de tempo por não necessitar desenvolver um repositório de base de dados, interfaces, *front end* e relatórios.	A versão "*plain vanilla*" do produto licenciado provavelmente não irá satisfazer todos os requisitos de meta dados. Ainda assim, um administrador em tempo integral será

	necessário para fazer manutenção e aprimorar o produto licenciado.
A maioria dos produtos de repositórios de meta dados licenciados vem com interfaces e, a maioria vem com uma série de APIs *(Application Programming Interface)*.	Existirá uma linha de aprendizado para ficar familiar com a arquitetura do produto, interfaces e APIs.
Se o produto do repositório de meta dados é certificado para as ferramentas onde os meta dados residem, ele irá prover ferramentas de interface.	Quanto mais sofisticado o repositório de meta dados fora, mais caro ele será e mais expertise os técnicos precisarão ter para mantê-lo.

5.8.2 Repositório de Meta Dados Descentralizado

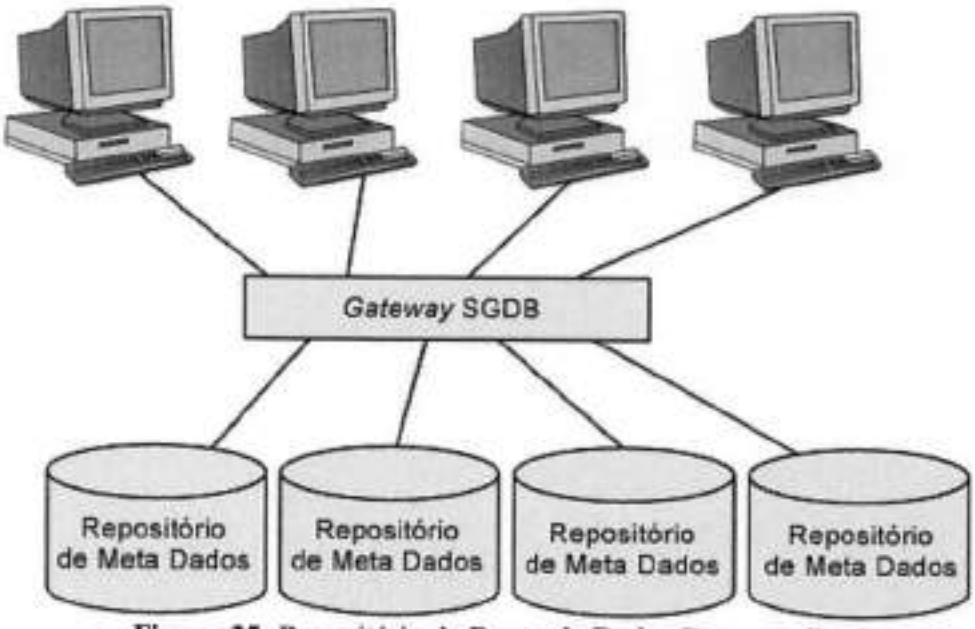


Figura 25: Repositório de Banco de Dados Descentralizado

Tabela 11 - Vantagens e desvantagens de um Repositório de Meta Dados Descentralizado

Vantagens	Desvantagens

Vários donos podem manter e gerenciar seu conjunto de meta dados separadamente.	Controlar redundância através de múltiplos repositórios de meta dados e manter a consistência é difícil.
Fáceis de usar porque cada banco de dados contém somente aqueles componentes de meta dados que são de interesse para um grupo específico do pessoal de negócio.	Leva mais tempo para fazer manutenção e gerenciar múltiplos bancos de dados em múltiplas plataformas. Pode haver também problemas de sincronização com novos lançamentos de SGBD.
Cada repositório de meta dados pode ter seu próprio meta modelo, que é seu próprio desenho customizado.	Comunicação através dos vários repositórios de meta dados terão de aumentar. Ainda mais, irá ser requerido um meta-meta modelo de manutenção, que é uma integração da arquitetura geral de múltiplos meta modelos.
Relatórios podem ser customizados para cada repositório de meta dados individualmente.	Relatar meta dados através de vários bancos d dados pode ser difícil. Por exemplo, dados de negócio não é automaticamente vinculado ao meta dado técnico se eles residem em banco de dados diferentes.
	A arquitetura desta solução é mais complicada e a curva de aprendizado para usar múltiplos bancos de dados com desenhos potencialmente diferentes deve ser alto.

5.8.3 Solução de Meta Dados com uso de XML Distribuído

Ao invés de armazenar meta dados através de múltiplos bancos de dados, em uma solução com XML, os meta dados permanecem nos seus locais originais de origem, ou seja, em várias ferramentas de dicionário de dados.

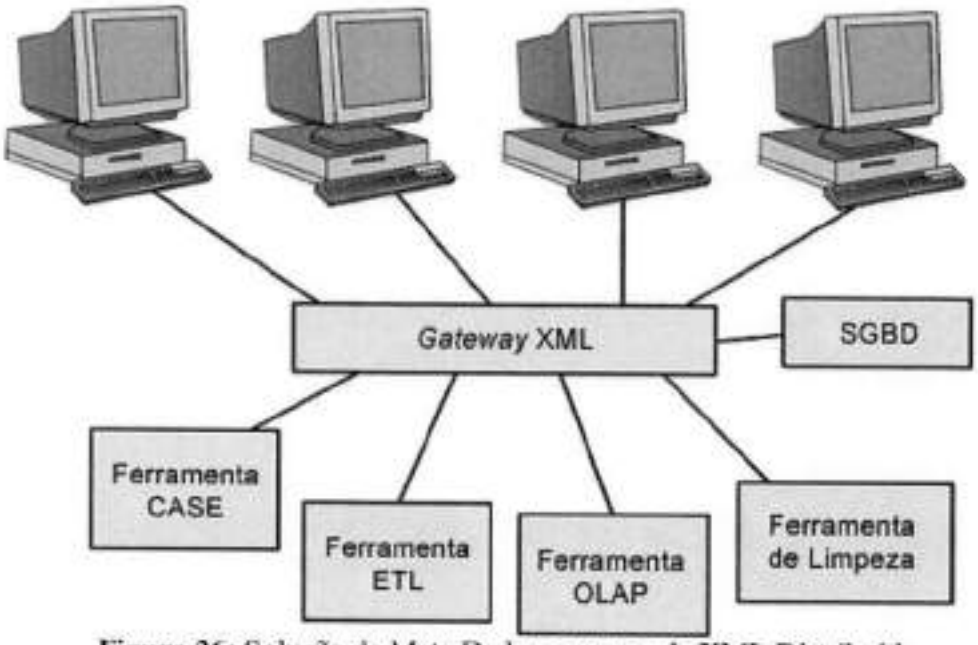


Figura 26: Solução de Meta Dados com uso de XML Distribuído

Um *gateway* atua como um diretório para vários lugares onde os componentes de meta dados são armazenados (por exemplo, o SGBD cataloga as tabelas ou o dicionário da ferramenta ETL). Vários fornecedores estão explorando esta solução porque reduz a necessidade de manutenção dupla dos meta dados. Dupla manutenção se refere a manter os meta dados nas fontes originais (SGBD e dicionários de ferramentas) assim como manter em separados banco de dados de meta dados.

Tabela 12 - Vantagens e desvantagens de uma Solução de Meta Dados com uso de XML Distribuído.

Vantagens	Desvantagens
XML *tags* habilitam o acesso de meta dados através de qualquer tipo de armazenamento de dados através de categorização padronizada.	O *tagging* inicial de todo meta dado com *tags* XML é um processo manual e delongado. Ainda mais, *tagging* XML não pode ser utilizado para todos os meta dados.

Meta dado nunca precisa ser duplicado ou movido da sua fonte original (exceto por motivo de relatório).	*Tags* XML adicionam requisitos de armazenamento para dicionário de banco de dados que armazenam meta dados (SGBC e dicionário de dados de ferramentas).
Um *gateway* faz a localização dos meta dados transparente para a pessoa que acessa.	Um meta-meta modelo tem de ser criado como um mapa de todos os tipos de armazenamento de meta dados, cada um dos quais é desenhado de acordo com seu único meta modelo.
Web search engines devem ser capazes de localizar os meta dados em qualquer lugar.	SGBD e ferramentas de fornecedores devem seguir padrões de mercado para *tags* XM para permitir acesso aos meta dados através de todos os produtos. Múltiplos padrões precisam ser suportados.
Meta dados e dados de negócios podem ser acopl ados e transmitidos simultaneamente.	Nem todos os SGBD e ferramentas utilizam XML. Esta é uma tecnologia nova e não provada.

No caso de estudo, está sendo utilizado este último modelo 'Solução de Meta Dados com uso de XML Distribuído', isso já em decorrência das ferramentas *open source* utilizadas (tanto OLAP quanto ETL) que já apresentam o uso por XML.

5.9 Desenvolvimento do ETL

Os detalhes técnicos da implementação podem ser obtidos no Anexo C.

5.10 Desenvolvimento da Aplicação – OLAP

O processo de desenvolvimento do OLAP foi dividido em três fases. Na primeira fase foi feito um estudo de alternativas de software e a instalação de algumas delas para verificar suas funcionalidades. Na segunda fase, a parte da implementação, o software escolhido foi configurado para o estudo de caso do projeto PENSA e na terceira fase foram efetuados alguns testes para verificar o correto funcionamento da ferramenta.

5.11 Desenvolvimento do repositório de META DADOS

Assim como visto anteriormente, será utilizada a solução de meta dados por XML distribuído, em decorrência que as ferramentas utilizadas já seguem esse modelo. Logo não necessita desenvolvimento.

5.11.1 Implementação da Aplicação – OLAP

A primeira parte do XML (Anexo E) representa uma dimensão, definindo a tabela que possui as informações a respeito dessa dimensão, e os atributos relacionados a mesma. Na segunda parte do XML podemos verificar os componentes cubo e medidas (*measures*). O cubo reúne informações das tabelas de dimensão e armazena a medida de interesse do usuário.

5.12 Data Mining

Para se obter um resultado preciso e consistente através de *Data Mining*, é necessário uma grande quantidade de dados. Como a quantidade de dados disponíveis para análise é pequena, esta ferramenta não será implementada.

5.13 Desenvolvimento do Portal

Para integrar as ferramentas utilizadas no sistema de BI, será utilizado um Portal *Web* que irá centralizar as ferramentas. Foi feito um estudo inicial para escolher a ferramenta mais adequada e posteriormente a sua configuração.

No projeto em si, foi criado 2 *portlets* baseados em JSP:

- *Portlet* do OLAP que invoca a aplicação OLAP (*Mondrian*)

- *Portlet* do ETL que inicia o processo de carga (ETL). Esse recurso está disponível somente para o administrador do sistema. Todo o controle e administração de usuários , feito pelo *JetSpeed*.

5.14 Resultados Finais

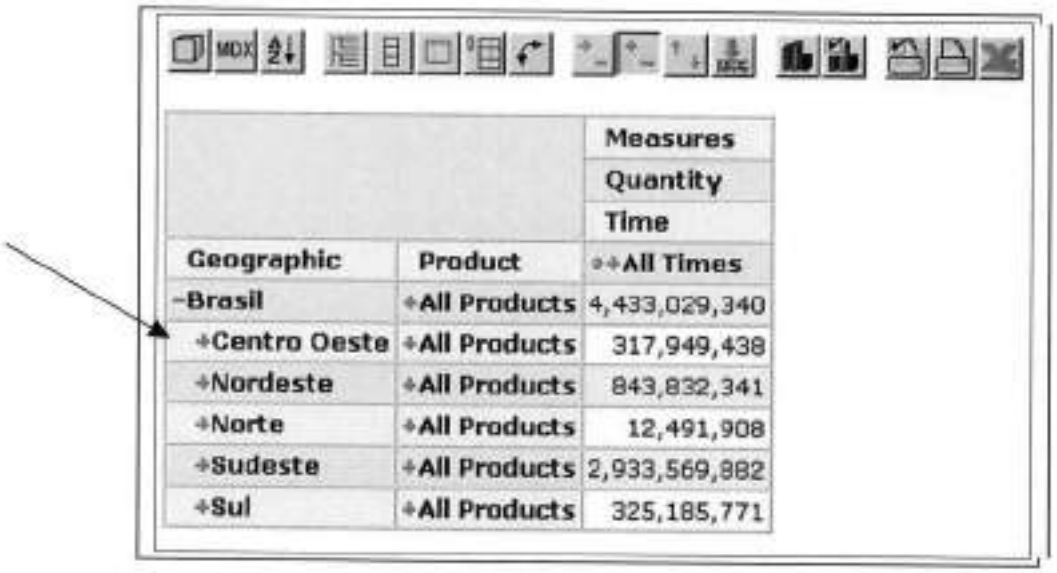
Atualmente, o ETL não está funcionando corretamente devido a uma série de problemas relacionados com a tecnologia utilizada. Além disso, ocorreram alguns problemas internos de comunicação, que acabaram gerando um atraso muito grande na construção do ETL. Toda a base de dados operacional utilizada pelo PENSA seria entregue pronta para nós e isto não ocorreu. Foi necessário um esforço extra para construção de 2 bases "extras" além de todo um processo inicial de pré-carga e tratamento dos dados iniciais.

O projeto lógico do *data warehouse* foi concluído por completo. No caso do projeto físico do *data warehouse*, somente o item sobre indexação foi levado em conta devido ao volume de dados envolvido no estudo de caso. Atualmente o *data warehouse* possui alguns registros para teste que foram povoados através de scripts pois, como foi dito no parágrafo anterior, o ETL ainda não está funcionando corretamente.

No caso do OLAP, ele se encontra funcionando corretamente, reconhecendo todas as dimensões definidas no *data warehouse* e efetuando corretamente todas as operações que ele é capaz de executar (*drill up/down*, *rotate* e *slice*) de maneira satisfatória. Como foi dita anteriormente, a deficiência da ferramenta OLAP para execução do *drill across* não foi notada, pois o modelo de dados utilizado pelo grupo possui apenas uma tabela fato.

Com relação ao Portal, ele se encontra em estágio de desenvolvimento e será concluído na semana final.

A seguir iremos demonstrar através de alguns *screenshots* as possíveis operações efetuadas pelo OLAP escolhido. A Figura 27 apresenta uma tela do OLAP em que são apresentadas informações sobre produção armazenadas no *data warehouse* divididas por regiões brasileiras.

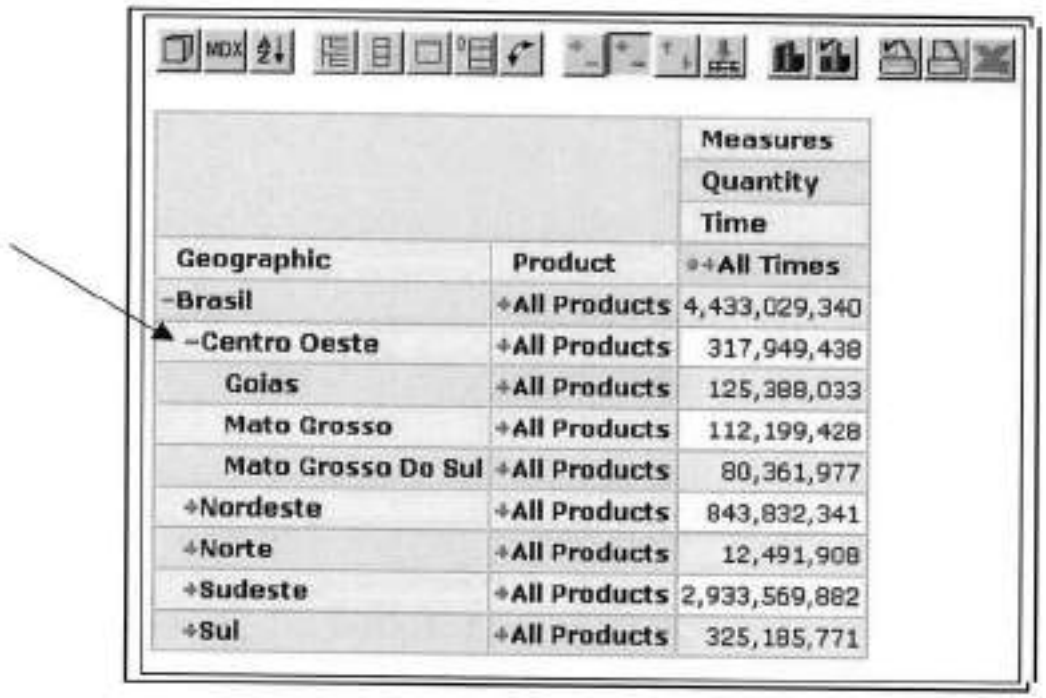


		Measures
		Quantity
		Time
Geographic	Product	++All Times
-Brasil	+All Products	4,433,029,340
+Centro Oeste	+All Products	317,949,438
+Nordeste	+All Products	843,832,341
+Norte	+All Products	12,491,908
+Sudeste	+All Products	2,933,569,882
+Sul	+All Products	325,185,771

Figura 27: Drill Down passo 1

No caso, para obtermos informações sobre a produção por estados da região Centro-Oeste, executa-se a operação de *drill down* clicando no sinal de + conforme indicado pela seta na Figura 27, obtendo um detalhamento maior das informações apresentadas para o usuário, como apresentado na Figura 28.

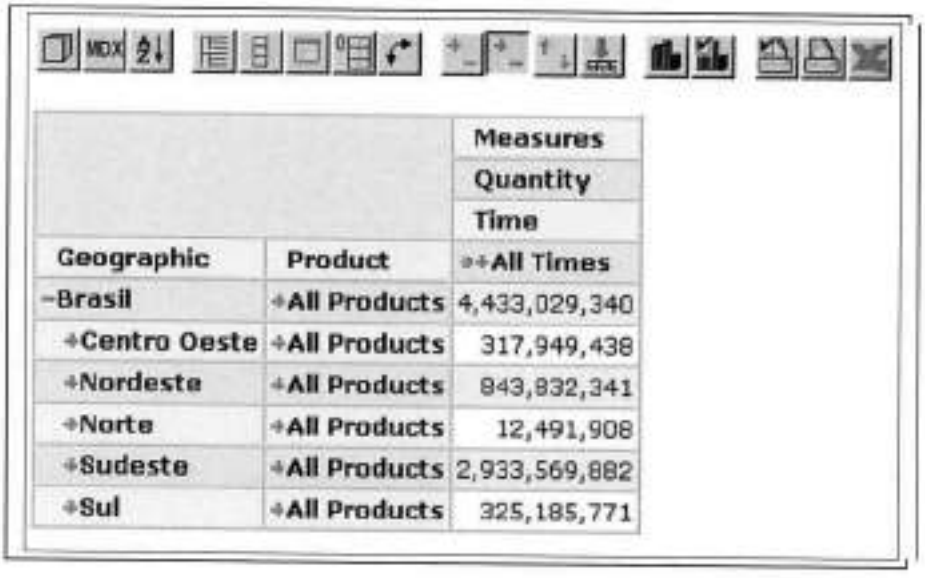
A operação de *drill up* é a operação inversa à operação de *drill down*, em que temos inicialmente informações mais detalhadas, como a apresentada na Figura 28 e queremos apresentar as informações como mostradas na Figura 29. Para isso, clicamos no sinal – indicado pela seta na figura 28.



		Measures
		Quantity
		Time
Geographic	Product	All Times
-Brasil	All Products	4,433,029,340
-Centro Oeste	All Products	317,949,438
Goias	All Products	125,388,033
Mato Grosso	All Products	112,199,428
Mato Grosso Do Sul	All Products	80,361,977
+Nordeste	All Products	843,832,341
+Norte	All Products	12,491,908
+Sudeste	All Products	2,933,569,882
+Sul	All Products	325,185,771

Figura 28: Drill Down passo 2

Podemos também efetuar a operação de *rotate*. Um dos casos da operação *rotate*, em que uma dimensão deixa de ser apresentada como sendo uma linha da tabela e passa a ser apresentada nas colunas da tabela, é exibida abaixo. Para abrir o menu que permite a execução da operação de *rotate*, clicar com o *mouse* no ícone indicado na Figura 29. O menu é apresentado na Figura 30.



		Measures
		Quantity
		Time
Geographic	Product	All Times
-Brasil	All Products	4,433,029,340
+Centro Oeste	All Products	317,949,438
+Nordeste	All Products	843,832,341
+Norte	All Products	12,491,908
+Sudeste	All Products	2,933,569,882
+Sul	All Products	325,185,771

Figura 29: Drill Up

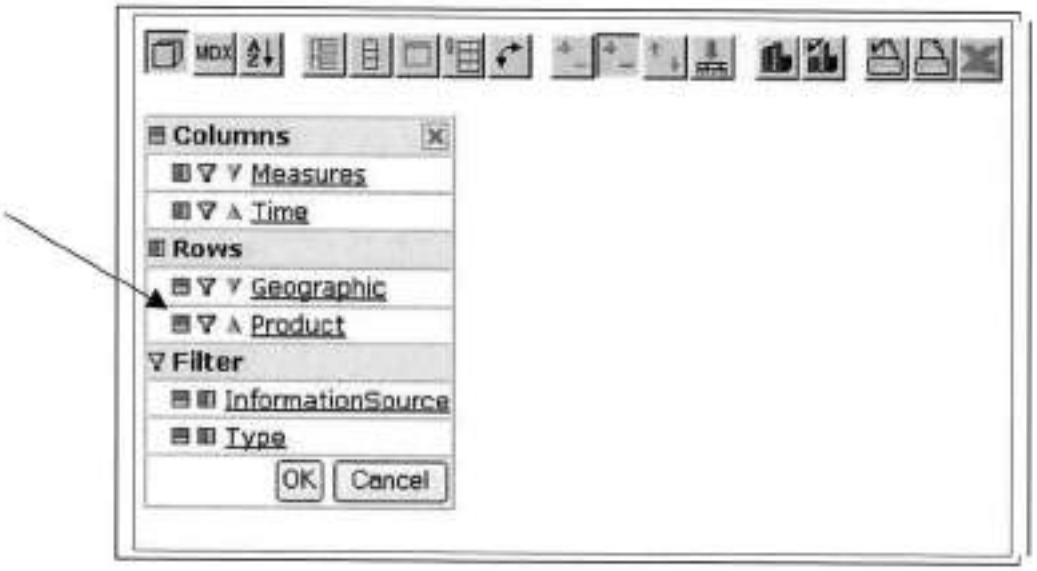
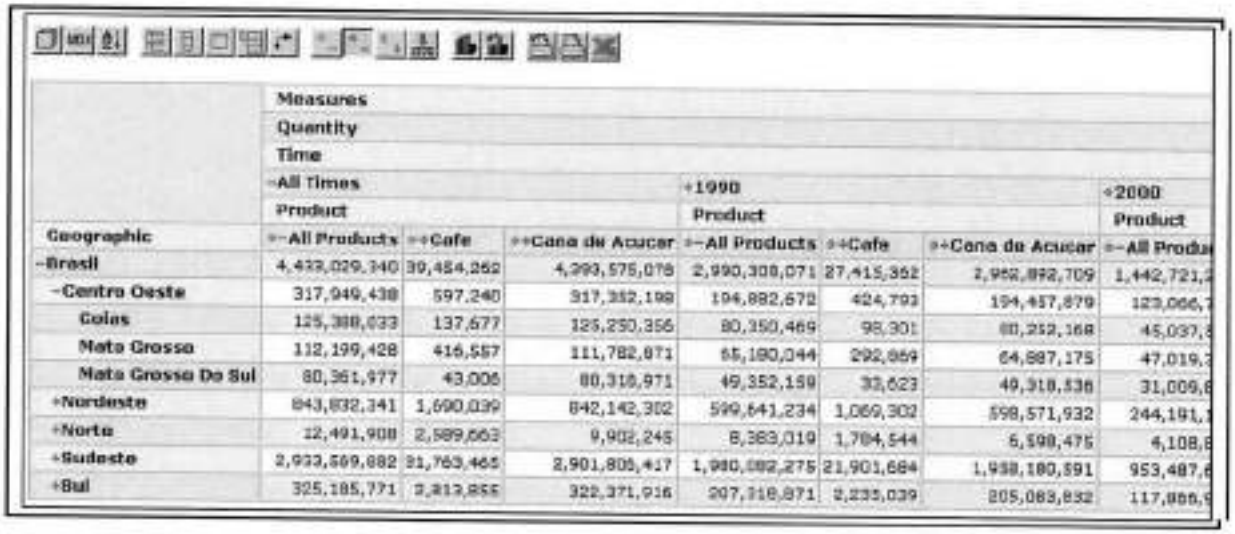


Figura 30: Rotate Passo 1

Para retirarmos a dimensão Produto da linha e coloca-lo na coluna da tabela, clicar no ícone conforme mostrado na Figura 30. Após esta operação, a dimensão produto passa a ser apresentado como sendo uma coluna, conforme a Figura 31. Aceitando esta alteração e efetuando algumas operações de *drill down*, obtemos uma tabela como a apresentada na Figura 32.

Figura 31: Rotate Passo 2



	Measures						
	Quantity						
	Time						
	-All Times			+1990			+2000
	Product			Product			Product
Geographic	--All Products	++Cafe	++Cana de Acucar	--All Products	++Cafe	++Cana de Acucar	--All Produ
-Brasil	4,433,029,340	39,454,262	4,293,575,078	2,990,308,071	27,415,352	2,952,892,709	1,442,721,2
-Centro Oeste	317,949,438	597,240	317,352,198	194,882,672	424,793	194,457,879	123,066,7
Goias	125,388,033	137,677	125,250,356	80,350,469	98,301	80,252,168	45,037,3
Mato Grosso	112,199,428	416,557	111,782,871	65,180,044	292,869	64,887,175	47,019,2
Mato Grosso Do Sul	80,361,977	43,006	80,318,971	49,352,159	33,623	49,318,536	31,009,8
+Nordeste	843,832,341	1,690,039	842,142,302	599,641,234	1,069,302	598,571,932	244,191,1
+Norte	12,491,908	2,589,663	9,902,245	8,383,019	1,784,544	6,598,475	4,108,8
+Sudeste	2,933,589,682	31,763,465	2,901,805,417	1,980,092,275	21,901,684	1,938,180,591	953,487,6
+Sul	325,185,771	2,813,855	322,371,916	207,318,871	2,235,039	205,083,832	117,866,9

Figura 32: Rotate com granularidade maior

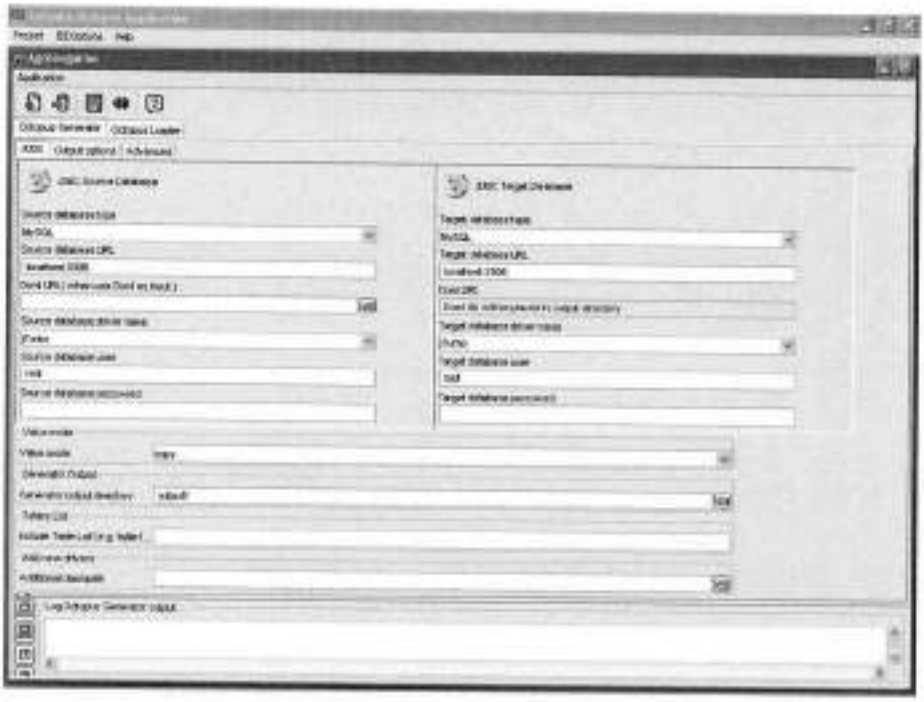


Figura 33: Tela do Octopus Generator(ETL)

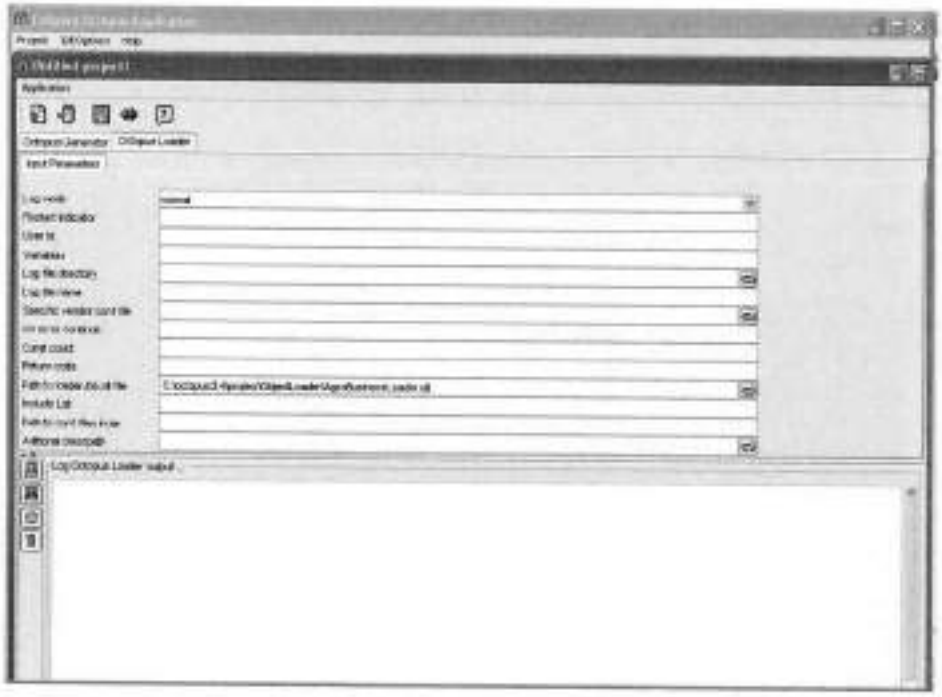


Figura 34: Tela do Octopus Loader (ETL)

6 CONSIDERAÇÕES FINAIS

6.1 Conclusões

O objetivo deste trabalho foi o estudo, pesquisa e aplicação de uma solução de *Business Intelligence* utilizando ferramentas *open source*. *Business Intelligence*, como o nome supõe, agrega inteligência ao negócio de maneira a automatizar o processo de coleta de dados, extração, armazenamento, filtragem, fazer seu tratamento e disponibilizar ao usuário final de uma maneira personalizada que facilite e agilize a tomada de decisão. Isso permite que se gaste menos tempo com a tarefa operacional de coleta de dados e possa assim oferecer ao usuário tomador de decisão mais tempo para se focar em questões estratégicas.

Business Intelligence não é *software*, mas sim um conjunto de técnicas e métodos de gestões implementadas através de ferramentas de *software*. Neste trabalho foi apresentada e utilizada um método para projetos de *Business Intelligence* que utiliza etapas muito similares a qualquer outro projeto de engenharia, no entanto, com algumas peculiaridades e visões focadas para projetos de BI.

Como se pode observar no mercado, a grande fatia dos que usufruem os benefícios de BI são as grandes empresas, sendo que as médias e pequenas empresas representam uma parcela em minoria. Mesmo com o crescimento do interesse pelas médias empresas, o acesso a soluções de BI é difícil devido a questões de custos. Visando amenizar tal problema, o corrente trabalho visa fornecer um levantamento de ferramentas de BI *open source* que são válidas em uma solução de BI completa.

Neste trabalho foram pesquisadas ferramentas de ETL (*Extract Transform Load*), que se encarregam da extração (*Extract*) de dados dos bancos de dados operacionais da empresa e de sistemas legados. Essas mesmas ferramentas se encarregam de filtrar, corrigir e limpar erros dos dados (*Transform*) para que possam ser armazenados (*Load*) no banco de dados de BI (*Data Warehouse*).

O *Data Warehouse* tem como função armazenar os dados históricos relevantes da empresa, que serve de fonte para análise de BI, ainda mais que contém dados limpos que podem ser adquiridos através da ferramenta ETL. Dentro deste *Data Warehouse*, os dados podem estar

agrupados em diversos modelos, sendo que em BI, os mais conhecidos são os modelos *Star* e *Snowflake schema* para banco de dados relacionais. Tais modelos servem para permitir que possa ser feita uma análise multidimensional dos dados. Tarefa executada pelas ferramentas de OLAP (*Online Analytical Processing*) na qual análises *ad hoc* podem ser feitas em tempo real.

Das ferramentas OLAP, existem as modalidades mais famosas que são os MOLAP (*Multidimensional Online Analytical Processing*) e o ROLAP (*Relational Online Analytical Processing*). A primeira se caracteriza por acessar bancos de dados multidimensionais, em geral proprietários, que possui uma estrutura especial, e que confere melhor performance para análises multidimensionais. Já o ROLAP, acessa bancos de dados relacionais que estejam modelados segundos as estruturas multidimensionais *star* e *snow flake schema*.

Data Mining é um processo que pode ser incorporado em projeto de BI, no entanto não foi feito uma pesquisa de ferramentas a seu respeito, já que não tem uso justificado no estudo de caso corrente. Sendo assim, foi apresentado seu conceito e técnicas de obtenção de modelos.

Para a aplicação da solução utilizou-se um caso de estudo de um projeto de Agronegócios do grupo PENSA (Programa de Estudos dos Negócios do Sistema Agroindustrial) na qual foram configuradas e utilizadas as ferramentas acima. Como produto obteve-se um sistema piloto que provê relatórios e gráficos de análises multidimensionais para que auxiliem o grupo em sua pesquisa acadêmica.

6.2 Sugestões para trabalhos futuros

Como visto no Estudo de Caso, uma ferramenta que pode ser muito explorada seria a de *Data Mining*, que faria sentido o seu uso com um volume maior de dados. Logo, um povoamento com um volume maior de dados no *Data Warehouse* poderia ser feito, assim como a inclusão de dados mais detalhados, como por exemplo dados de clima, para descobrir a influência do clima na produção, dados econômicos para descobrir a relação com exportação e importação por exemplo. Ou então, até mesmo aumentar a granularidade de estados para cidades, para se obter

uma análise mais minuciosa do mercado de agronegócio, já que no caso de estudo os dados que alimentavam o *Data Warehouse* já vinham sumarizados por estados.

Outro trabalho que poderia ser feito é um aprimoramento do gerenciamento do repositório de meta dados, em face do crescimento da complexidade do projeto e inclusão de novas funcionalidades.

Colocar em produção seria um trabalho futuro já que implicaria em criar toda uma infra-estrutura técnica e não técnica e de treinamento de pessoas para fazerem manutenção no sistema.

7 REFERÊNCIAS

ÂNGELO, F. K. - *MicroStrategy* cresce 25% no Brasil em 2005 – COMPUTER WORLD – Disponível em < http://computerworld.uol.com.br/AdPortalv5/adCmsDocumentShow.aspx?GUID=A32F9C02-992B-46C2-90E5-077DABE1B2BC&ChannelID=22 > Acesso em 02 de novembro

DAMALZO, L. - Gartner: BI deve crescer 7,4% ao ano até 2009 – COMPUTER WORLD – Disponível em < http://computerworld.uol.com.br/AdPortalv5/adCmsDocumentShow.aspx?GUID=9E454F10-CBB0-43EE-B547-CEC9B407A224&ChannelID=22 > Acesso em 02 de novembro

DOS SANTOS, J. G. Ofertas de Produtos com Configurações Customizadas. 2001,106 p. Dissertação (Mestrado) – Universidade Federal de Santa Catarina, Florianópolis, 2001.

DWBRASIL, TIME - Histórico do *Data Warehouse* – Disponível em < http://www.dwbrasil.com.br/html/artdw_hist.html > Acesso 29 de novembro

EGIDERIA, YVES-MICHEL MARTI - *Europe's Information and Economic Inteligence culture* – Disponível em: < http://www.egideria.com/bieurope.html > Acesso 29 de novembro.

ELMASRI, R.; NAVATHE, S. B. *Fundamentals of Database Systems*. Boston, Editora Addison Wesley, 2003, 1030 p.

FIDALGO,F. Aplicações de Informática no Mundo Empresarial. Disponível em:<http://www.est.ipcb.pt/cursos/informatica/inf_SEMINARIO/grupo3_0203/Aplicacoes_Informaticas.pdf>. Acesso em: 06 de maio.

GANT,J. , GANT, D.B. - *Web portal functionality and State government E-service* – IEEE *Proceedings of the 35th Hawaii International Conference on System Sciences* - 2002

INMON, W. H. Como construir o *data warehouse*. Rio de Janeiro, Editora Campus, 1997, 387 p.

KIMBALL, RALPH. *The Database Market Splits.* Revista *DBMS Magazine.* Set 1995. Disponível em http://www.dbmsmag.com/9509d05.html. Acesso em: 05 Nov 2005

KIMBALL, R.; ROSS, M. *Data Warehouse Toolkit*. Rio de Janeiro, Makron Books, 2000, 388 p.

LANE, P., A. *Oracle9i Data Warehousing Guide Release* 2(9.2). Oracle Corporation, 2002, (Certification Guide, Part Nº A96520-01).

MAHDAVI M., SHEPHERD, J. - *Enabling Dynamic Content Caching in Web Portals* – IEEE - *Proceedings of the 14th International Workshop on Research Issues on Data Engineering: Web Services for E-Commerce and E-Government Applications* (RIDE'04) 0-7695-2095-2/04 2004

MICROSTRATEGY – *General Motors - Microstrategy* – Clientes – *Cases* – Disponível em < http://www.microstrategy.com.br/Customers/Cases/GM.asp > Acesso 29 de novembro

MOSS, LARISSA T., ATRE, SHAKU. *Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications* Editora Addison Wesley. Edição de 2003.

NEXTG – *Next Generation Center – Business Intelligence* – Módulo 3 – Disponível em < http://www.nextg.com.br > Acesso em 29 de novembro

PARRINI, E. Gestão do Conhecimento no Suporte à Decisão em Ambiente OLAP. 2002,157 p. Dissertação (Mestrado) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2002.

REINSCHMIDT, J.; FRANCOISE, *A. Business Intelligence Certification Guide*. San Jose, IBM Corporation, 2000, (Certification Guide, SG24-5747-00).

TELECO. Página com informações sobre tecnologias e conceitos de redes de telefonia e de computadores. Disponível em www.teleco.com.br.

VALIM, C. E. - Um novo campo estratégico – ITWeb - Disponível em < http://www.itweb.com.br/solutions/gestao_empresarial/gerenciamento_de_processos/artigo.asp?id=51191 > Acesso em 29 de novembro

Anexo A – Grupo Pensa

Apresentação
O PENSA é uma organização que integra os Departamentos de Economia e Administração da FEA-USP, São Paulo e Ribeirão Preto. Foi instalado pelo reitor da Universidade de São Paulo em 17 de junho de 1990. Sua finalidade é promover estudos sobre o Agribusiness Brasileiro.

Missão
Criar um espaço interativo entre alunos, professores e lideranças do Agribusiness nacional, por meio da pesquisa, ensino e extensão.

Objetivos
Estudar a dinâmica do Sistema Agroindustrial, fornecendo subsídios à tomada de decisão e ao planejamento estratégico de organizações privadas e públicas.

Identificar e analisar as principais tendências dos negócios agroindustriais visando sobretudo a inserção competitiva do Brasil no Agribusiness internacional.

Formar e capacitar recursos humanos para a gestão do Sistema Agroindustrial Brasileiro

Metodologia
A metodologia de trabalho do PENSA fundamenta-se na análise sistêmica dos negócios agroindustriais, focalizando especialmente as interfaces e redes criadas, entre os diversos setores (insumos, agropecuária, indústria, distribuição). Esta abordagem reconhece a dinâmica própria de cada um dos setores e as limitações impostas pelas suas inter-relações tecnológicas e econômicas.

A metodologia é complementada ainda por dois princípios: a análise das questões que circunscrevem o processo decisório das organizações e preocupação em aproximar a Universidade do meio empresarial.

Relacionamento com as Fundações da USP:

O PENSA é um programa multi-institucional que desenvolve suas atividades por meio das Fundações que integram a Universidade de São Paulo. O Programa encontra-se sediado na cidade de São Paulo na Fundação Instituto de Administração (FIA), onde desenvolve a maioria dos projetos. As atividades do PENSA também são desenvolvidas pela Fundação Instituto de Pesquisas Econômicas (FIPE) e pela Fundação para a Pesquisa e Desenvolvimento da Administração, Economia e Contabilidade (FUNDACE).

Anexo B – Origem das informações -Agronegócios

ABIC - Associação Brasileira da Indústria de Café
http://www.abic.com.br/

Estoques
Produção
Produtividade
Preço no Varejo
Preço pago ao produtor
Exportações
Consumo
Ranking das Maiores Empresas

OIC - International Coffee Organization
http://www.ico.org/

Abecitrus - Associação Brasileira dos Exportadores de Cítricos
http://www.abecitrus.com.br/menu_br.html
Exportações
Produção
Preços

Embrapa Gado de Leite
http://www.cnpgl.embrapa.br/
Produção Mundial
Produção
Produtividade
Preços
Demanda
Ranking Maiores Produtores
Ranking Maiores Laticínios
Industrialização
Produção Queijos
Produção Leite em pó
Importações

ABIOVE - Associação Brasileira das Indústrias de Óleos Vegetais
http://www.abiove.com.br/

Farelo/ Óleo/ Grão
Produção agrícola
Produtividade
Preços
Estoques
Esmagamento
Importações

Exportações
Consumo
Capacidade Instalada
Preços

Alice - SECEX - Ministério do Desenvolvimento
http://aliceweb.desenvolvimento.gov.br/

IBGE - Censo Agropecuário 2004
http://www.ibge.gov.br/

Embrapa Milho
http://www.cnpms.embrapa.br/

Embrapa Trigo
http://www.cnpt.embrapa.br/
Trigo e Farinha de trigo
Produção Nacional e Mundial
Produtividade
Exportações
Importações
Consumo per capita trigo, pão, massas e biscoitos
Capacidade de Moagem e Números de Moinhos
Empregos diretos

Abitrigo – Associação Brasileira da Indústria do Trigo
http://www.abitrigo.com.br/

Conab - Companhia Nacional de Abastecimento
http://www.conab.gov.br/

Unica – União da Agroindústria Canavieira de São Paulo
http://www.portalunica.com.br/

Produção Brasil, Centro-sul e Nordeste
Preços
Exportações
Carro a álcool
Ranking

Embrapa Algodão
http://algodao.cnpa.embrapa.br/

FAO – STAT – Food and Agriculture Organization of the United Nations
http://faostat.fao.org/faostat/collections?subset=agriculture

USDA - United States Department of Agriculture

http://www.usda.gov/wps/portal/!ut/p/_s.7_0_A/7_0_1OB?navid=DATA_STATISTICS&parentnav=MARKETING_TRADE&navtype=RT

Embrapa Arroz e Feijão
http://www.cnpaf.embrapa.br/

SBS – Sociedade Brasileira de Silvicultura
http://www.sbs.org.br/
Área reflorestada
Volume consumido
Quantidade de mudas
Mão de obra empregada
Faturamento do setor
Impostos e taxas pagas
Produção de papel
Produção de celulose
Carvão vegetal

ABIEC – Associação Brasileira das Indústrias Exportadoras de Carne
http://www.abiec.org.br/abiec/
Exportação
Importação
Abate
Consumo per capita
Rebanho

Embrapa Gado de Corte
http://www.cnpgc.embrapa.br/eventos/eventos.html

Embrapa Soja
http://www.cnpso.embrapa.br/index.php?op_page=120&cod_pai=87

ABEF - Associação Brasileira dos Exportadores de Frango
http://www.abef.com.br/
Produção Interno e Mundial
Consumo Interno e Mundial
Exportações

UBA – União Brasileira de Avicultura
http://www.uba.org.br/

Associação Nacional para Difusão de Adubos
http://www.anda.org.br/portug/

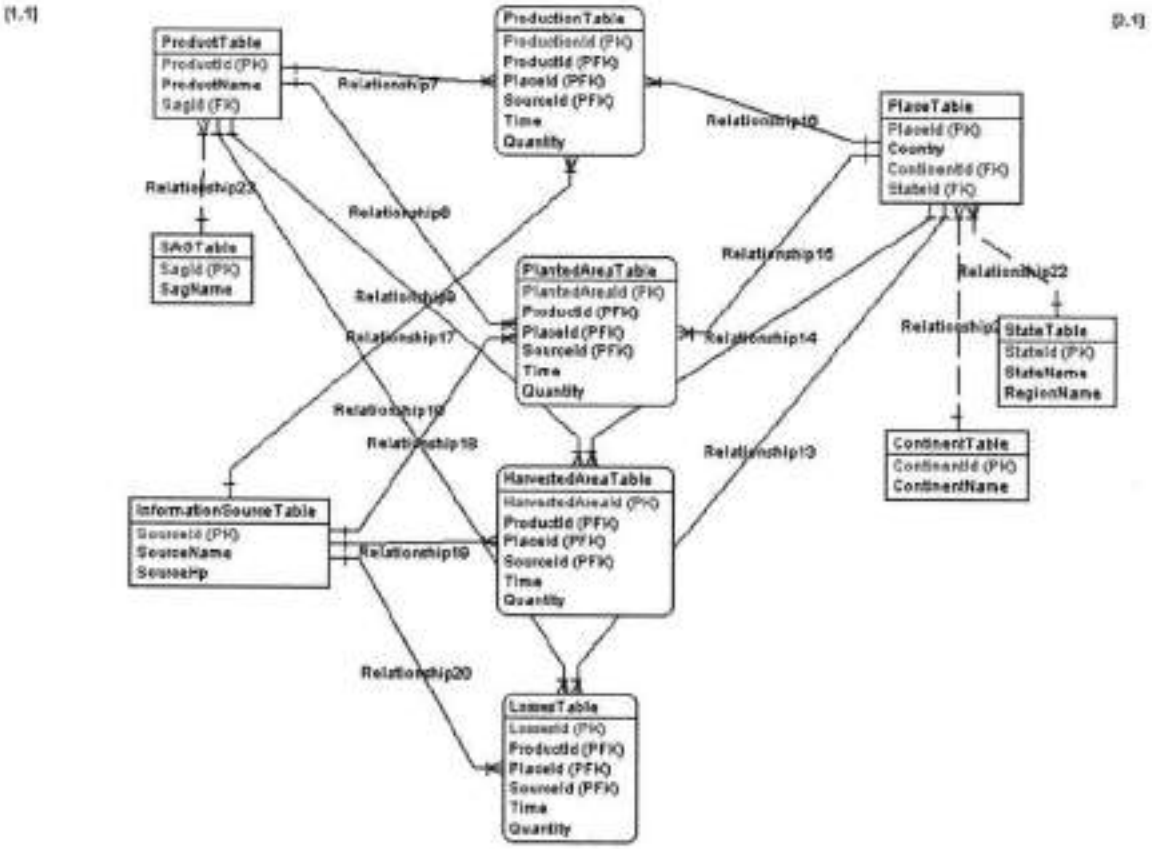
International Fertilizer Industry Association
http://www.fertilizer.org/ifa/statistics.asp

Anexo C:-Mapeamento origem-destino do ETL

Tabela	Coluna	Tabela	Coluna	Arquivo/BD	Transformações
TimeDimension	Month	Transformacao	Time	SAGCAFE	Month obtido a partir do Time
TimeDimension	Quarter	Transformacao	Time	SAGCAFE	Quarter obtido a partir do Time
TimeDimension	Semester	Transformacao	Time	SAGCAFE	Semester obtido a partir do Time
TimeDimension	Year	Transformacao	Time	SAGCAFE	Year obtido a partir do Time
TimeDimension	Decade	Transformacao	Time	SAGCAFE	Decada obtido a partir do Time
TimeDimension	YearType	Transformacao	YearType	SAGCAFE	
GeographicDimension	GeographicId			SAGCAFE	
GeographicDimension	Continent	ContinentTable	ContinentName	SAGCAFE	
GeographicDimension	Country	PlaceTable	Country	SAGCAFE	
GeographicDimension	Region	-	-	SAGCAFE	
GeographicDimension	State	-	-	SAGCAFE	
InformationSourceDimension	SourceId			SAGCAFE	
InformationSourceDimension	SourceName	InformationSourceTable	SourceName	SAGCAFE	
InformationSourceDimension	SourceHp	InformationSourceTable	SourceHp	SAGCAFE	
ProductDimension	ProductId			SAGCAFE	
ProductDimension	SagName	SAGTable	SagName	SAGCAFE	
ProductDimension	ProductName	ProductTable	ProductName	SAGCAFE	
TransactionTypeDimension	TypeId			SAGCAFE	
TransactionTypeDimension	Transaction	Transformacao	Transformacao	SAGCAFE	Nome da tabela origem
TransactionFact	TransactionId			SAGCAFE	
TransactionFact	TimeId			SAGCAFE	
TransactionFact	ProductId			SAGCAFE	
TransactionFact	GeographicId			SAGCAFE	
TransactionFact	SourceId			SAGCAFE	
TransactionFact	TypeId			SAGCAFE	
TransactionFact	Quantity	Transformacao	Quantity	SAGCAFE	Transformação para unidade padrão
TimeDimension		-		EXCEL_CANA	

TimeDimension	Month	-	-	EXCEL_CANA	
TimeDimension	Quarter	-	-	EXCEL_CANA	
TimeDimension	Semester	-	-	EXCEL_CANA	
TimeDimension	Year	-	COLUNA B	EXCEL_CANA	
TimeDimension	Decade	-	Transf.	EXCEL_CANA	Obtido a partir do Year
TimeDimension	YearType	-	"Safra"	EXCEL_CANA	
GeographicDimension	GeographicId	-		EXCEL_CANA	
GeographicDimension	Continent	-	"America do Sul"	EXCEL_CANA	
GeographicDimension	Country	-	"BRASIL"	EXCEL_CANA	
GeographicDimension	Region	-	Transf.	EXCEL_CANA	Obtido a partir do State
GeographicDimension	State	-	COLUNA A	EXCEL_CANA	
InformationSourceDimension	SourceId	-		EXCEL_CANA	
InformationSourceDimension	SourceName	-	COLUNA D	EXCEL_CANA	
InformationSourceDimension	SourceHp	-	COLUNA E	EXCEL_CANA	
ProductDimension	ProductId	-		EXCEL_CANA	
ProductDimension	SagName	-	"Cana"	EXCEL_CANA	
ProductDimension	ProductName	-	"Cana"	EXCEL_CANA	
TransactionTypeDimension	TypeId	-		EXCEL_CANA	
TransactionTypeDimension	Transaction	-	COLUNA G	EXCEL_CANA	
TransactionFact	TransactionId	-		EXCEL_CANA	
TransactionFact	TimeId	-		EXCEL_CANA	
TransactionFact	ProductId	-		EXCEL_CANA	
TransactionFact	GeographicId	-		EXCEL_CANA	
TransactionFact	SourceId	-		EXCEL_CANA	
TransactionFact	TypeId	-		EXCEL_CANA	
TransactionFact	Quantity	-	COLUNA C	EXCEL_CANA	Transformar tudo pra mesma unidade

Anexo D: Banco de Dados SAGCAFE



Anexo E: OLAP

Arquivo de configuração do OLAP utilizando a ferramenta Mondrian.

```
<?xml version="1.0"?>
<Schema name="FoodMart">

<!-- Shared dimensions -->

  <Dimension name="Time">
    <Hierarchy hasAll="true" primaryKey="TimeId">
      <Table name="TimeDimension"/>
      <Level name="Decade" column="Decade" uniqueMembers="false"/>
      <Level name="Year" column="Year" uniqueMembers="false"/>
      <Level name="Semester" column="Semester" uniqueMembers="false"/>
      <Level name="Quarter" column="Quarter" uniqueMembers="false"/>
      <Level name="Month" column="Month" uniqueMembers="false"/>
      <Property name="YearType" column="YearType"/>
<!--      <Level name="YearType" column="YearType" uniqueMembers="false"/> -->
    </Hierarchy>
  </Dimension>

  <Dimension name="Product">
    <Hierarchy hasAll="true" primaryKey="ProductId">
      <Table name="ProductDimension"/>
      <Level name="SagName" column="SagName" uniqueMembers="false"/>
      <Level name="ProductName" column="ProductName" uniqueMembers="false"/>
    </Hierarchy>
  </Dimension>

  <Dimension name="Geographic">
    <Hierarchy hasAll="true" primaryKey="GeographicId">
      <Table name="GeographicDimension"/>
      <Level name="Continent" column="Continent" uniqueMembers="false"/>
      <Level name="Country" column="Country" uniqueMembers="false"/>
      <Level name="Region" column="Region" uniqueMembers="false"/>
      <Level name="State" column="State" uniqueMembers="false"/>
    </Hierarchy>
  </Dimension>

  <Dimension name="InformationSource">
    <Hierarchy hasAll="true" primaryKey="SourceId">
      <Table name="InformationSourceDimension"/>
      <Level name="SourceName" column="SourceName" uniqueMembers="false"/>
      <Level name="SourceHp" column="SourceHp" uniqueMembers="false"/>
    </Hierarchy>
  </Dimension>

  <Dimension name="Type">
    <Hierarchy hasAll="true" primaryKey="TypeId">
      <Table name="TransactionTypeDimension"/>
      <Level name="TransactionType" column="TransactionType"
uniqueMembers="false"/>
    </Hierarchy>
  </Dimension>
```

```
<Cube name="Transaction">
  <Table name="TransactionFact"/>
  <DimensionUsage name="Time" source="Time" foreignKey="TimeId"/>
  <DimensionUsage name="Product" source="Product" foreignKey="ProductId"/>
  <DimensionUsage name="Geographic" source="Geographic"
foreignKey="GeographicId"/>
  <DimensionUsage name="InformationSource" source="InformationSource"
foreignKey="SourceId"/>
  <DimensionUsage name="Type" source="Type" foreignKey="TypeId"/>
  <Measure name="Quantidade" column="Quantity" aggregator="sum"
formatString="Standard"/>
</Cube>

</Schema>
```