

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
ESPECIALIZAÇÃO EM BANCO DE DADOS**

IGOR ALLEN BEZERRA DE MAGALHÃES RITZMANN

**O USO DE *DATA WAREHOUSE* NO AUXÍLIO DE GESTÃO DO
AGRONEGÓCIO**

MONOGRAFIA DE ESPECIALIZAÇÃO

**PATO BRANCO
2017**

IGOR ALLEN BEZERRA DE MAGALHÃES RITZMANN

**USO DE *DATA WAREHOUSE* NO AUXÍLIO DE GESTÃO DO
AGRONEGÓCIO**

Trabalho de Conclusão de Curso, apresentado ao II Curso de Especialização em Banco de Dados, da Universidade Tecnológica Federal do Paraná, campus Pato Branco, como requisito parcial para obtenção do título de Especialista.

Orientador: M. Sc. Wellton Costa de Oliveira

**PATO BRANCO
2017**



TERMO DE APROVAÇÃO

O USO DE DATA WAREHOUSE NO AUXÍLIO DE GESTÃO DO AGRONEGÓCIO.

por

IGOR ALLEN BEZERRA DE MAGALHÃES RITZMANN

Este Trabalho de Conclusão de Curso foi apresentado em 24 fevereiro de 2017 como requisito parcial para a obtenção do título de Especialista em Banco de Dados. O(a) candidato(a) foi arguido(a) pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Wellton Costa de Oliveira
Prof.(a) Orientador(a)

Viviane Dal Molin de Souza
Membro titular

Marcelo Teixeira
Membro titular

“O Termo de Aprovação assinado encontra-se na Coordenação do Curso”

AGRADECIMENTOS

Agradeço a Deus, até mesmo pelo ar que respiro, “Ele” o alfa e o ômega. Reconheço sua onipresença e plenitude.

A minha querida e amada esposa, presente em todos os momentos de minha vida, sonhando os mesmos sonhos, andando na mesma estrada.

A meus pais, exemplos de vida, fonte de inspiração, vocês meus queridos, fazem o mundo parecer melhor.

A todos os professores do Curso de Especialização em Banco de Dados da Universidade Tecnológica Federal do Paraná, visto que não faltaram esforços no compartilhar do melhor de seus conhecimentos e experiências.

Em especial ao Professor Orientador Mestre Wellton Costa pela gentileza e carinho em aceitar ser orientador do presente estudo.

Aos colegas, pelos momentos que juntos vivenciamos, alguns contatos permanecerão, outros quem sabe, se perderão pelo caminho, mas, foi muito bom este tempo.

Os que não conhecem a história estão
condenados a repeti-la.

George Santayana

RESUMO

RITZMANN, Igor Allen Bezerra de Magalhães. O Uso de Data Warehouse no Auxílio de Gestão do Agronegócio. 2016. 51 fl. Monografia (II Curso de Especialização em Banco de dados) - Universidade Tecnológica Federal do Paraná. Pato Branco, 2017.

As empresas de agronegócio, mesmo suportadas por ERP (Enterprise Resource Planning) que auxiliam o gerenciamento das atividades do negócio, têm alguns problemas na agilidade de informações para o auxílio aos gestores em processos decisórios, que se baseiam em suas experiências do dia-a-dia de sua área para tomar tais decisões. Decisões baseadas em informações incorretas ou incompletas, seguramente prejudicarão o resultado da estratégia da empresa. O uso de data warehouse, no entanto, auxilia a gestão do agronegócio, apresentando uma arquitetura de consolidação de dados transacionais, para suportar e dar acesso às informações em um único ambiente integrado. O Objetivo deste trabalho é o estudo e a criação de um Data Warehouse para auxílio de uma gestão de agronegócio e são demonstrados através de uma aplicação desenvolvida pela Microsoft® com arquitetura em cloud computing (computação em nuvem) para o processo de ETL (Extract, Transform, Load).

Palavras-chave: *Data Warehouse; Data Mart; Cloud Computing; Agronegócio.*

ABSTRACT

RITZMANN, Igor Allen Bezerra de Magalhães. Using Data Warehouse to help managing Agribusiness. 2017. 51 p. Monography (II Specialization Course in Database) - Federal University of Technology - Parana. Pato Branco, 2017.

Agribusiness firms, even backed by Enterprise Resource Planning (ERP) that helps manage business activities, have some agile information issues to aid decision makers who are based on their day-to-day experiences, Day of your area to make such decisions. Decisions based on incorrect or incomplete information will surely undermine the outcome of the company's strategy. The use of data warehousing, however, assists agribusiness management by presenting a transactional data consolidation architecture to support and give access to information in a single integrated environment. The objective of this work is the study and creation of a Data Warehouse to aid in agribusiness management and are demonstrated through an application developed by Microsoft® with architecture in cloud computing for the process of ETL (Extract, Transform, Load).

Keywords: Data Warehouse; Data Mart; Cloud Computing; Agribusiness.

LISTA DE SIGLAS

| | |
|-------|-----------------------------------------------------|
| ACID | Atomicidade, Consistência, Isolamento, Durabilidade |
| BI | <i>Bussiness Intelligence</i> |
| DF | <i>Dependência Funcional</i> |
| ETL | <i>Extract, Transform, Load</i> |
| ER | <i>Entidade Relacionamento</i> |
| ERP | <i>Enterprise Resource Planning</i> |
| FBNC | <i>Forma Normal Boyce Codd</i> |
| OLAP | <i>Processamento analítico online</i> |
| OLTP | <i>Online Transaction Processing</i> |
| PAAS | <i>Plataform As a Service</i> |
| SGBD | <i>Data Base Management System</i> |
| SQL | <i>Structured Query Language</i> |
| TI | <i>Tecnologia da Informação</i> |
| UTFPR | <i>Universidade Tecnológica Federal do Paraná</i> |

LISTA DE FIGURAS

| | | |
|-----------|--------------------------------------------------------------------|----|
| Figura 01 | - Um exemplo de esquema estrela..... | 14 |
| Figura 02 | - Fases do desenvolvimento de <i>data marts</i> independentes..... | 16 |
| Figura 03 | - Visualizações dos componentes arquiteturais básica..... | 18 |
| Figura 04 | - Processo de recebimento de grãos..... | 30 |
| Figura 05 | - Dados da entrada de pesagem..... | 31 |
| Figura 06 | - Tela do sistema ERP p ara entrada de romaneio..... | 31 |
| Figura 07 | - Arquitetura do <i>data warehouse</i> | 33 |
| Figura 08 | - Schema lógico..... | 35 |
| Figura 09 | - Sequência e <i>trigger</i> para o campo com auto incremento..... | 35 |
| Figura 10 | - Tabelas do <i>data warehouse</i> (a)..... | 36 |
| Figura 11 | - Tabelas do <i>data warehouse</i> (b)..... | 38 |
| Figura 12 | - Fábrica de dados..... | 38 |
| Figura 13 | - Dashboard gerencial..... | 39 |
| Figura 14 | - Dashboard publicado no Power BI..... | 40 |

LISTA DE QUADROS

| | | | |
|-----------|---|----------------------------------------------------------------------------------------------------------------------|----|
| Quadro 01 | - | Diferenças entre sistemas para dados operacionais (OLTP) e dados de suporte a decisões o <i>data warehouse</i> | 13 |
| Quadro 02 | - | Modelo de dados do <i>data warehouse</i> | 14 |
| Quadro 03 | - | Vantagens e desvantagens usando <i>data marts</i> independentes..... | 15 |
| Quadro 04 | - | Adaptação de componentes arquiteturais básicos..... | 17 |
| Quadro 05 | - | Vantagens e desvantagens da implementação <i>bottom up</i> | 20 |
| Quadro 06 | - | Fato recebimento de grãos..... | 34 |
| Quadro 07 | - | Códigos SQL..... | 36 |
| Quadro 08 | - | <i>Queries</i> | 37 |

SUMÁRIO

| | | |
|----------|---------------------------------------------------------------------|----|
| 1 | INTRODUÇÃO | 9 |
| 1.1 | CONSIDERAÇÕES INICIAIS..... | 9 |
| 1.2 | OBJETIVOS..... | 9 |
| 1.2.1 | Objetivo Geral..... | 9 |
| 1.2.2 | Objetivos Específicos..... | 10 |
| 1.3 | JUSTIFICATIVA..... | 10 |
| 1.4 | ORGANIZAÇÃO DO TEXTO..... | 10 |
| 2 | REFERENCIAL TEÓRICO | 12 |
| 2.1 | DATA WAREHOUSE..... | 12 |
| 2.1.1 | Escopo do Data Warehouse..... | 12 |
| 2.1.2 | Modelo de Data Warehouse..... | 13 |
| 2.1.2.1 | Data Mart..... | 15 |
| 2.2 | ARQUITETURA DO DATA WAREHOUSE..... | 16 |
| 2.2.1 | Camada de Dados Operacionais e Dados Externos..... | 18 |
| 2.2.2 | Camada de Extração, Transformação e Carregamento de Dados (ETL). .. | 19 |
| 2.2.3 | Camada do Data Warehouse..... | 19 |
| 2.2.4 | Ferramenta de Consulta do Usuário Final (Interface) | 19 |
| 2.3 | IMPLEMENTAÇÃO..... | 20 |
| 2.3.1 | Bottom Up..... | 20 |
| 2.4 | GRANULARIDADE (NÍVEL DE AGREGAÇÃO) | 20 |
| 2.5 | PROJETO DA ESTRUTURA DO DATA WAREHOUSE..... | 21 |
| 2.5.1 | Projeto Lógico..... | 21 |
| 2.5.2 | Implicações do Projeto Físico..... | 22 |
| 2.5.2.1 | Particionamento..... | 22 |
| 2.5.2.2 | Índices (indexação) | 22 |
| 2.5.2.3 | Surrogate key (chave substituta) | 24 |
| 2.6 | COMPONENTES DE UM DATA WAREHOUSE..... | 24 |
| 2.6.1 | Integrado..... | 24 |
| 2.6.2 | Orientado por Assunto..... | 25 |
| 2.6.3 | Variável no Tempo..... | 25 |

| | | |
|----------|--------------------------------------------------|-----------|
| 2.6.4 | Não Volátil..... | 26 |
| 3 | MATERIAIS E MÉTODOS..... | 27 |
| 3.1 | MATERIAIS..... | 27 |
| 3.1.1 | Azure Data Factory..... | 27 |
| 3.1.2 | Data Management Gateway..... | 28 |
| 3.1.3 | SGBD Oracle 11g R2..... | 28 |
| 3.1.4 | SQL Developer..... | 28 |
| 3.1.5 | Power Architet..... | 29 |
| 3.1.6 | Power BI..... | 29 |
| 3.1.7 | Microsoft Visio..... | 29 |
| 3.2 | MODELAGEM E IMPLEMENTAÇÃO..... | 29 |
| 3.2.1 | Requisitos..... | 29 |
| 3.2.2 | Metodologia..... | 32 |
| 3.2.3 | Modelagem de Dados Dimensional..... | 32 |
| 3.2.4 | Criação do Ambiente e Arquitetura..... | 32 |
| 4 | RESULTADO E DISCUSSÃO..... | 33 |
| 4.1 | APRESENTAÇÃO DO DATA WAREHOUSE..... | 33 |
| 4.1.1 | Arquitetura do Data Warehouse..... | 33 |
| 4.2 | MODELAGEM DO DATA WAREHOUSE..... | 34 |
| 4.2.1 | Projeto Lógico..... | 34 |
| 4.2.2 | Projeto Físico..... | 35 |
| 4.3 | CRIAÇÃO DO AMBIENTE DE ETL E DATA WAREHOUSE..... | 36 |
| 4.3.1 | ETL..... | 37 |
| 4.3.2 | Extração, Transformação e Carga dos Dados..... | 37 |
| 4.4 | CRIAÇÃO DE DASHBOARD..... | 39 |
| 5 | CONSIDERAÇÕES FINAIS..... | 41 |
| | REFERÊNCIAS..... | 42 |

1 INTRODUÇÃO

1.1 CONSIDERAÇÕES INICIAIS

Apesar da importância do setor de agronegócios, há poucas soluções prontas no Brasil para gerenciar informações que auxiliam no processo de tomada de decisão do agronegócio (CORREA *et al.*, 2009).

Menos de 1% dos produtores e empresas ligadas ao agronegócio usam os dados em seu potencial máximo e, cerca de 10% o utilizam em alguma aplicação (FERREIRA, 2014). Segundo Rob e Coronel (2011, p. 535) “as organizações tendem a crescer e prosperar quando obtêm melhor compreensão do seu ambiente”.

O objetivo de sistemas de apoio a decisões é capturar (dados, informações, conhecimento) e responder a eventos de negócios que precisam tomar decisões rápidas e melhor (POPA, 2010).

Contudo, nas soluções de apoio a decisões, existentes nas empresas de agronegócio, nota-se certa demora no processamento das informações causando lentidão no sistema, concorrência entre ferramentas e *ERP* e a carga extra para o banco de dados de transação, principalmente quando envolve um grande período de análise por estas ferramentas.

Dessa forma, com o uso da tecnologia pode-se manter o controle, extraindo melhor estas informações entregando indicadores que antes seriam difícil de imaginar devido à complexidade envolvida, trazendo um ambiente propício para melhor tomada de decisão por parte dos gestores, no ramo do agronegócio.

No presente estudo, será construído um *data warehouse*, iniciando com um *data mart*, para auxiliar o departamento de grãos.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Desenvolver um modelo de *data warehouse* iniciando com um *data mart* orientado por assunto, no auxílio de gestão do agronegócio do setor de grãos, fornecendo um ambiente para a tomada de decisão.

1.2.2 Objetivos Específicos

- Identificar o processo de recebimento de grãos (gestão de pátio), sendo os aspectos a considerar a pesagem e classificação do produto, colaborando para a construção do modelo do *data mart*;
- Implantar arquitetura e ambiente *ETL* para o tratamento dos dados em nuvem;
- Contribuir para o melhoramento das empresas do agronegócio.

1.3 JUSTIFICATIVA

O desenvolvimento de um *data warehouse* justifica-se por resolver problemas de lentidão no processamento de informações gerenciais com base no passado, gerar relatórios que demandam um grande período de processamento, assim, não onerando o banco de dados de transação (*OLTP- Online Transaction Processing*).

O uso de serviços em nuvem sendo a plataforma como serviço (*PAAS*) disponibiliza poder de processamento pagando-se pelo uso, garantindo disponibilidade do serviço, elasticidade para aumentar ou diminuir os recursos de acordo com a necessidade de processamento e armazenamento do negócio, garantindo a continuidade do negócio e menor preocupação para a gestão de *TI (Tecnologia da Informação)*.

Com a aplicação das tecnologias de *data warehouse* a empresa em estudo tem o diferencial competitivo perante outras empresas que não tenham o mesmo ambiente para tomada de decisões.

1.4 ORGANIZAÇÃO DO TEXTO

O estudo foi organizado a partir de cinco capítulos principais, assim distribuídos: na introdução, apresentam-se o tema, objetivos e justificativa. No segundo capítulo, é contextualizada a revisão de literatura, onde são percorridos temas que darão suporte ao desenvolvimento final do estudo, enfatizando-se o *data warehouse*. No terceiro capítulo, estão detalhados os métodos empregados no desenvolvimento do presente estudo. No quarto capítulo, ocorrem os resultados e discussões, com apresentação do *data warehouse*, fundamentado em estudo de caso com dados (*dump* da base de dados operacional) de uma

empresa da área de agronegócio, destaca-se a modelagem física e lógica do *data mart* e a sua arquitetura. No quinto capítulo, ocorrem as considerações finais, seguidas das referências bibliográficas.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico utilizado para fundamentar o desenvolvimento do *data warehouse* proposto. Enfatiza-se a arquitetura e modelo de *data warehouse* e a preparação dos dados.

2.1 DATA WAREHOUSE

Como definição de *data warehouse* Rob e Coronel (2011) empregam o termo concebido por Bill Inmon, no qual *data warehouse* significa “um conjunto de dados integrado, orientado por assunto, variável no tempo e não volátil que fornece suporte à tomada de decisões”.

Segundo Elmasri e Navathe (2011, p. 422), vê-se que “o *data warehouse* tem como característica distintiva de servir principalmente para aplicações de apoio à decisão”.

De outra maneira, sobre *data warehouse* Rob e Coronel (2011 p. 541), assim se posicionam:

O data warehouse é o fundamento sobre o qual é construída a infraestrutura de BI, os dados são capturados a partir do sistema OLTP e posicionados no data warehouse em base quase de tempo real. O BI proporciona integração de dados de toda a empresa e recursos para responder a questões de negócios no tempo oportuno.

Segundo estes autores, o uso de sistemas de banco de dados com a finalidade de oferecer apoio à decisão, são sistemas que auxiliam na análise de informações do negócio da empresa, onde a meta é auxiliar a administração definindo tendências, apontando os problemas e tomando decisões inteligentes, baseados nestas informações.

Pode-se afirmar que o *data warehouse* é um banco de dados otimizado para armazenar, extrair e processar os dados de diversas origens, dando suporte para gerar informações para tomada de decisão.

2.1.1 Escopo do *Data Warehouse*

Segundo Singh (2001) “o *data warehouse* pode funcionar como um repositório central para alocar grandes quantidades de informações”, pode-se criar *data marts* que

também contém dados sumarizados e tabelas (Dimensões e Fato) para o suporte as ferramentas de *processamento analítico online (OLAP)*.

Verifica-se no Quadro 01, as diferenças entre sistemas para dados operacionais (OLTP) e dados de suporte a decisões o *data warehouse*.

| CARACTERÍSTICA | DADOS OPERACIONAIS | DADOS DE SUPORTE A DECISÕES |
|--------------------------|---------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| Atualidade dos dados | Operações correntes Dados em tempo real | Dados históricos Retrato instantâneo dos dados da empresa Componente temporal (semana/mês/ano) |
| Granularidade | Dados detalhados do nível indivisível | Dados resumidos |
| Nível de resumo | Baixo; alguns resultados agregados | Alto; vários níveis de agregação |
| Modelo de dados | Altamente normalizado Principalmente SGBD relacional | Não normalizado Estrutura complexa Principalmente em SGBD multidimensional, embora com alguns aspectos relacionais |
| Tipo de transação | Principalmente atualizações | Principalmente consulta |
| Volumes de transação | Alto volume de atualizações | Cargas periódicas e cálculos resumidos |
| Velocidade de transação | As atualizações são fundamentais | As recuperações são fundamentais |
| Atividade de consulta | De baixa a média | Alta |
| Escopo da consulta | Pequena amplitude | Grande amplitude |
| Complexidade da consulta | De simples a média | Muito complexa |
| Volume de dados | Centenas de <i>megabytes</i> , até <i>gigabytes</i> | Centenas de <i>gigabytes</i> , até <i>terabytes</i> |

Quadro 01 - Diferenças entre sistemas para dados operacionais (OLTP) e dados de suporte a decisões o *data warehouse*

Fonte: Rob e Coronel (2011)

O quadro elencado acima explicita de maneira bastante clara a diferença entre sistemas para dados operacionais (OLTP) e dados de suporte a decisões o *data warehouse*, o que por sua vez dispensa complementos. Assim sendo, a seguir, discorre-se sobre no modelo de *data warehouse*.

2.1.2 Modelo de Data Warehouse

Segundo Cledes (2001), o modelo de *data warehouse* é uma abstração do mundo real que permite visualizar e definir a direção ou planejamento para implementar o *data warehouse*, no qual as opções dos modelos mais utilizados pode ser visto no Quadro 02:

| MODELO | CARACTERÍSTICAS |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1-E/R | Representação do modelo utilizando entidades e os relacionamentos entre estas entidades. Apresentação do modelo através do diagrama E/R. Principais componentes: entidades, relacionamentos e atributos |
| 2-Estrela | <p>É a estrutura básica do modelo dimensional. Basicamente é composto de uma grande tabela central (tabela fato) e um conjunto de pequenas tabelas (tabelas de dimensão) dispostas ao redor da tabela fato</p> <p>Dentro de cada categoria, existe uma única tabela fato histórica simples, contendo detalhes e dados resumidos, armazenados nos níveis de estrutura indicados em cada tabela dimensional</p> <p>A chave primária da tabela fato contém somente uma coluna de cada dimensão</p> <p>Cada chave é uma chave gerada pelo sistema</p> <p>Cada dimensão é representada por uma única tabela fato, usando também uma chave gerada pelo sistema</p> |
| 3-SnowFlake | O modelo dimensional típico inicia com uma tabela fato e um único nível de várias dimensões ao redor. O modelo <i>snowflake</i> é o resultado da decomposição de uma ou mais dessas dimensões. Algumas vezes, esta decomposição pode acontecer em uma dimensão que já foi decomposta, formando uma hierarquia |

Quadro 02 - Modelo de dados do *data warehouse*

Fonte: IBM (1998)

Na imagem abaixo (Figura 01), vemos as tabelas lógicas de dimensões e fatos para armazenar informações de vendas (tabela de fatos vendas). A combinação entre a tabela de fato e as tabelas de dimensão é chamado de esquema estrela, sendo um esquema comum, projetado para uso por aplicações *OLAP*. A maior parte dos dados fica na tabela fato, sem redundância, na Forma Normal de Boyce-Codd (FNBC), onde segundo Ramakrishnan e Gehrke (2008, p. 514) “a FNBC garante que nenhuma redundância pode ser detectada usando-se apenas as informações da DF” (dependência funcional), sendo a forma normal mais desejável partindo do ponto de vista de não existir redundância na tabela fato.

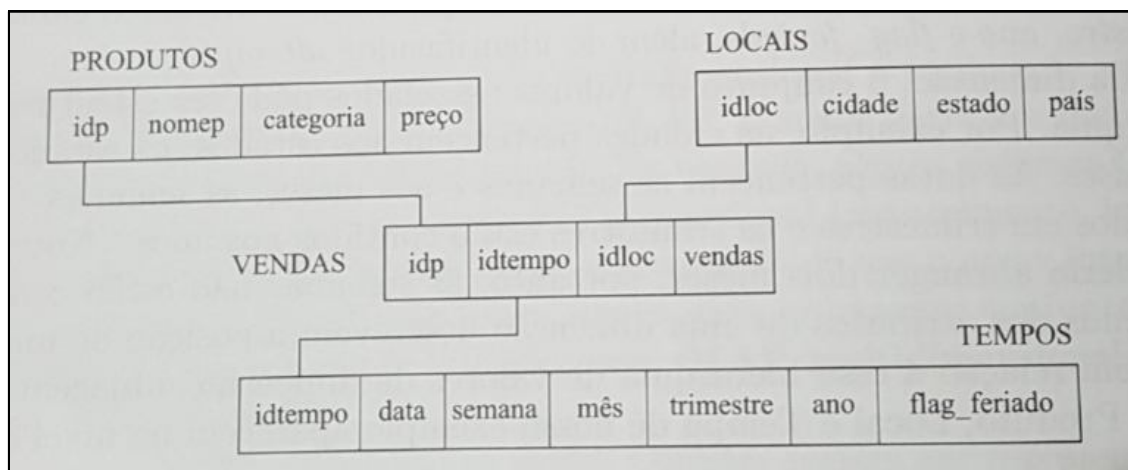


Figura 01 - Um exemplo de esquema estrela

Fonte: Ramakrishnan; Gehrke (2008)

Segundo Ramakrishnan e Gehrke (2008) os valores de dimensão são mantidos nas tabelas de dimensão, onde estas tabelas não são normalizadas por serem estáticas (não se têm atualizações) e o espaço economizado com a normalização das tabelas de dimensão são de menos, sendo mais interessante minimizar o tempo de computação, combinando fatos na tabela de fatos com informações de dimensões e evitar a divisão de uma tabela de dimensão em tabelas menores, pois pode levar a junções adicionais.

Interessante observar que em sistemas OLAP, o uso de tabelas materializadas de resumo geradas através de consultas usando agrupamento diminui o tempo de resposta. As consultas *ad hoc* feitas por usuários são respondidas usando as tabelas originais junto com os resumos previamente calculados, sendo o desafio saber quais tabelas de resumo devem ser materializadas para obter o melhor uso de memória disponível e consultas *ad hoc* com tempo de respostas interativo.

2.1.2.1 Data Mart

Segundo Rob e Coronel (2011), devido ao alto custo com tempo, dinheiro e esforço gerencial para se implantar um *data warehouse*, existe os *data marts* que são armazenamentos menores, um subconjunto de um *data warehouse* de um único assunto para fornecer suporte a decisão a um departamento ou grupo de pessoas. A única diferença entre o *data warehouse* e o *data mart* é o tamanho e o escopo do problema a ser resolvido.

Para Cledes (2001), o modelo de *data marts* independentes, é implementado de forma isolada, atendendo as necessidades específicas de um setor, grupo de trabalho, departamento ou um negócio da empresa, sem considerar todas as necessidades no primeiro momento.

As vantagens e desvantagens do uso de *data marts* independentes estão elencadas no Quadro 03:

| VANTAGENS | DESVANTAGENS |
|--------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Apresentação de resultados efetuada de forma rápida e barata | Baixa integração entre os <i>data marts</i> |
| Poucos recursos necessários para iniciar o projeto | Acesso aos dados de um <i>data mart</i> serão acessíveis somente aos seus proprietários |
| Extração dos dados não necessita de suporte da equipe de TI | Identificação das necessidades de informação, levantamento das fontes de dados e implementação dos processos de extração replicados em cada <i>data mart</i> |

| | |
|----------------------------------------------------------------|--|
| Gerenciamento facilitado devido a área de abrangência reduzida | |
|----------------------------------------------------------------|--|

Quadro 03 - Vantagens e desvantagens usando *data marts* independentes
 Fonte: Clemes (2001)

Na Figura 02 apresentam-se as fases de desenvolvimento para os *data marts* independentes.

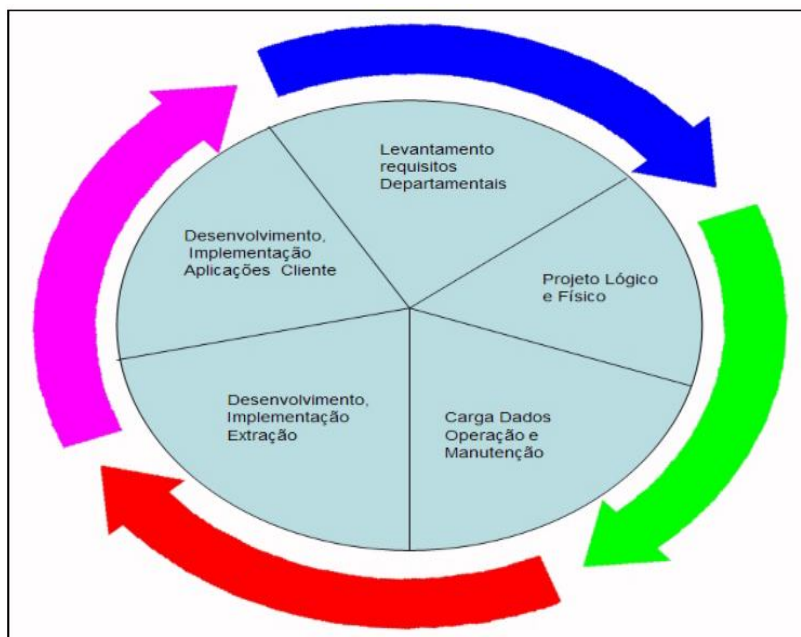


Figura 02 - Fases do desenvolvimento de *data marts* independentes
 Fonte: Clemes (2001)

Uma vez apresentadas as fases de desenvolvimento para os *data marts* independentes, se faz necessário apresentar a arquitetura do *data warehouse*

2.2 ARQUITETURA DO DATA WAREHOUSE

A arquitetura visa cobrir uma faixa de tecnologia e aplicações para o gerenciamento de todo o ciclo de vida dos dados, da aquisição ao armazenamento, transformando, integrando, analisando, monitorando, apresentando e arquivando, indo da simples coleta e extração de dados a complexas aplicações de análises e apresentação das informações. (ROB; CORONEL, 2011)

Verificam-se no Quadro 04, os componentes arquiteturais básicos de BI (*Business Intelligence*):

| COMPONENTE | DESCRIÇÃO |
|---------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Ferramenta de extração, transformação e carregamento de dados (ETL) | Esse componente é encarregado de coletar, filtrar, integrar e agregar dados operacionais a serem salvos em um armazenamento de dados otimizado para o suporte a decisões. Como o nome diz, esse componente extrai os dados, filtra os dados extraídos para selecionar os registros relevantes e empacota os dados no formato certo para ser adicionado ao componente de armazenamento de dados. |
| Armazenamento de dados | O armazenamento de dados é otimizado para o suporte a decisões e costuma ser representado por um <i>data warehouse</i> ou <i>data mart</i> . Ele contém dados de negócios extraídos de bancos operacionais e de fontes externas. Esses dados são armazenados em estruturas otimizadas para a velocidade de análise e consulta. As fontes de dados externos fornecem dados que não podem ser encontrados no interior da empresa, mas que são relevantes para os negócios como preço de ações, indicadores de mercado, informações de marketing (como as demográficas) e dados de competidores. |
| Ferramenta de consulta e análise de dados | Esse componente executa as tarefas de recuperação, análise e mineração, utilizando os dados no armazenamento e os modelos de análise de dados de negócios. Tal componente é utilizado pelo analista de dados para criar as consultas que acessam o banco. Dependendo da implementação, a ferramenta de consulta acessa tanto o banco de dados operacional como, o que é mais comum, o armazenamento de dados. Essa ferramenta orienta o usuário sobre quais dados selecionar e como construir um modelo de dados confiável. Tal componente costuma aparecer na forma de uma ferramenta OLAP. |
| Ferramenta de apresentação e visualização de dados | Esse componente é encarregado de apresentar os dados ao usuário final de vários modos. É utilizado pelo analista de dados para organizar e apresentar os dados. Essa ferramenta ajuda o usuário final a selecionar o formato de apresentação mais adequado, como relatório resumido, mapa, gráfico de pizza ou barra ou gráfico mistos. A ferramenta de consulta e a ferramenta de apresentação são a extremidade final do ambiente de BI. |

Quadro 04 – Adaptação Componentes arquiteturais básicos

Fonte: Rob; Coronel (2011)

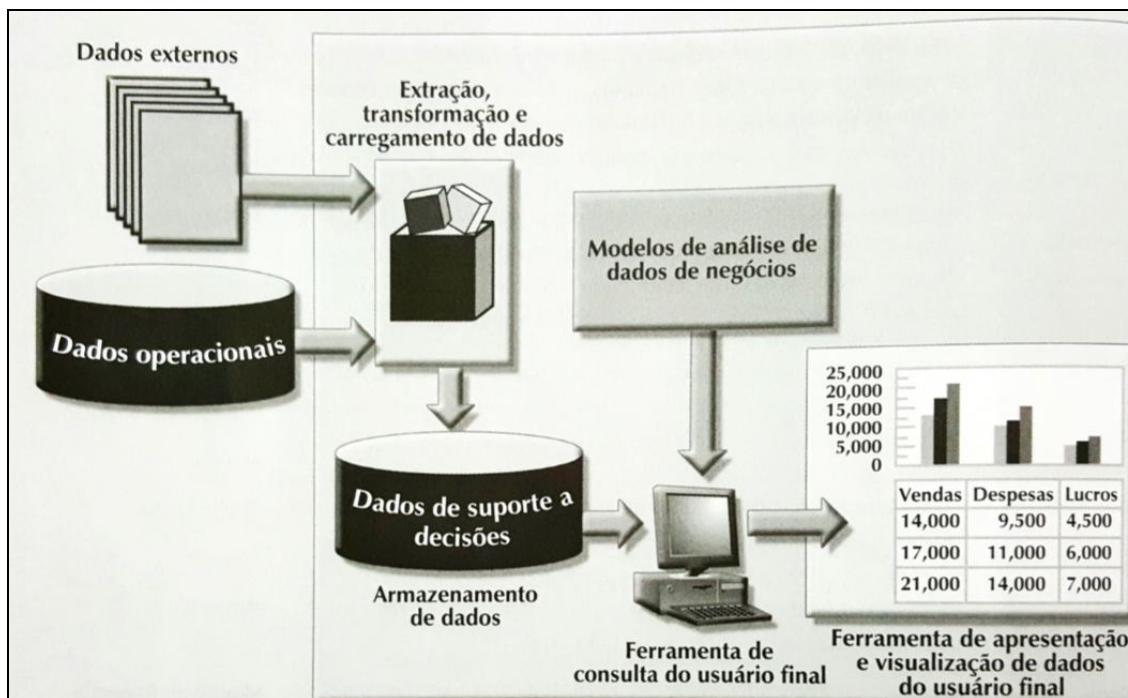


Figura 03 – Visualizações dos componentes arquiteturais básica
 Fonte: Rob; Coronel (2011)

Descrita a Arquitetura do *Data Warehouse* (adaptação componentes arquiteturais básicos e visualizações dos componentes arquiteturais básicos), se faz imprescindível apresentar a camada de dados operacionais e dados externos para compor o *data warehouse*.

2.2.1 Camada de Dados Operacionais e Dados Externos

Nesta camada constam os dados operacionais e as fontes de dados externas, os quais serão tratados e integrados para compor o *data warehouse*.

O banco de dados de suporte a decisões é criado, em grande parte, pela extração de dados do banco operacional (banco transacional) e pela importação de dados adicionais chamados de fonte externa, onde o SGBD deve dar suporte a ferramentas avançadas de extração e filtragem de dados, suporte a várias fontes de dados e tipos (banco de dados de arquivos, hierárquico, em rede, relacional), fornecedores diferentes, resolução de conflito de formato de dados (dados filtrado e purificado) e também minimizar o impacto do banco de dados operacional, onde os recursos de extração de dados devem permitir a extração de dados em batch e programada (ROB; CORONEL, 2011).

2.2.2 Camada de Extração, Transformação e Carregamento de Dados (ETL)

A ferramenta de extração, transformação e carregamento de dados (ETL), é encarregada de coletar, filtrar, integrar e agregar dados operacionais a serem salvos em um armazenamento de dados otimizado para o suporte a decisões. Esse componente extrai os dados, filtra os dados extraídos, para selecionar os registros relevantes, e empacota os dados no formato certo para ser adicionado ao componente de armazenamento de dados (ROB; CORONEL, 2011).

2.2.3 Camada do *Data Warehouse*

Camada do *Data Warehouse* é a camada onde os dados são armazenados e manipulados, oferecendo acesso para as ferramentas de consultas utilizarem os dados armazenados.

De acordo com Rob e Coronel (2011) o esquema deve proporcionar representação de dados complexos, a recuperação das consultas de leitura deve ser otimizada com suporte a recursos como índices de bitmap e particionamento de dados, que aumentam a velocidade das buscas e suporte a estruturas não normalizadas e complexas.

2.2.4 Ferramenta de Consulta do Usuário Final (Interface)

Nesta camada os usuários finais utilizam ferramentas de consultas de terceiros, como planilha eletrônica ou geradores de relatórios para obterem as informações do *data warehouse*. As consultas devem ser otimizadas para o rápido processamento devido às informações geradas pela consulta serem fundamentais para os tomadores de decisões, a interface analítica do usuário final é um dos componentes mais importantes, quando adequadamente implantado permite ao usuário navegar pelos dados, simplificando e acelerando o processo de tomada de decisões.

2.3 IMPLEMENTAÇÃO

A escolha de uma forma de implementar o *data warehouse* parte da ideia de se ter um curto tempo de implementação, baixo custo de *hardware* e manutenção, estas escolhas nos trazem a forma de implementação *bottom up* por estar baseada em *data marts* incrementais e aos poucos ir avançando até cobrir toda a empresa e consolidar o *data warehouse* centralizado no ambiente da empresa. Cledes (2001).

2.3.1 Bottom Up

Segundo Cledes (2001), “a opção por esta implementação permite resultados mais rápidos, pois está baseada na construção de *data marts*, que tem uma menor complexidade de projeto do que um *data warehouse* global”. No Quadro, 05 observam-se as vantagens e desvantagens do uso.

| VANTAGENS | DESVANTAGENS |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Apresenta resultados imediatos | Pode ocorrer redundância de dados e inconsistência entre os <i>data marts</i> |
| Menor complexidade na elaboração do projeto | A integração dos <i>data marts</i> em um <i>data warehouse</i> global pode ser difícil até uma determinada etapa do projeto |
| Necessidades iniciais de hardware e outros recursos têm custo relativamente baixo | Pode ser necessário retrabalho quando novas áreas são atendidas, ou quando mudanças naquelas que já estão implementadas são necessárias |

Quadro 05 – Vantagens e desvantagens da implementação *bottom up*

Fonte: Cledes (2001)

2.4 GRANULARIDADE (NÍVEL DE AGREGAÇÃO)

Os dados de suporte a decisão devem ser apresentados em diferentes níveis de agregação, altamente resumidos a quase indivisíveis, podendo ser decompostos (*drill down*) em componentes menos divisíveis (dados refinados em menor nível de agregação) e ao agrupar os dados (*roll up*), agrega-se em um nível mais alto. (ROB; CORONEL, 2011)

De acordo com Singh (2001, p. 147) sobre agregação, “agregação é o processo pelo qual os dados de nível baixo são previamente sumarizados e incluídos em tabelas que armazenam informações “agregadas” ou sumarizadas”.

Sendo assim, estas tabelas agregadas eliminam a necessidade de repetir cálculos consumidores de recursos, não sendo necessário a sua execução toda vez que for solicitada.

2.5 PROJETO DA ESTRUTURA DO *DATA WAREHOUSE*

Segundo Date (2003), o projeto de banco de dados sempre deve ser feito em pelo menos duas fases, a fase lógica e depois a fase física:

- Projeto lógico: sua função é a correção relacional onde as tabelas têm que representar relações de forma apropriada, garantindo que as operações de relações funcionem como se deseja para produzir resultados esperados. Especifica os tipos, as colunas e suas dependências definidas e com essas informações prosseguir com a normalização, com a definição das restrições de integridade.
- Projeto físico: será derivado do projeto lógico, com foco em eficiência de armazenamento e desempenho. A organização física dos dados exige ter transformação que preserve a informação e expressa na álgebra relacional entre o esquema lógico e físico.

2.5.1 Projeto Lógico

Algumas decorrências do projeto lógico segundo Date (2003):

- Combinações de colunas e menor número de dependências: As consultas de apoio a decisão tratam combinação de colunas como uma unidade onde não se tem acesso as colunas individuais, exemplo: endereço, uma combinação de colunas seria uma coluna composta e no projeto lógico as colunas compostas se comportam como se não fossem compostas, tendo o efeito final, o número total de dependências reduzidas com o projeto lógico simples, com menos colunas e menos tabelas;

- Restrições de integridade em geral: Os bancos de dados de apoio a decisão são somente leitura e sua integridade é verificada quando o banco de dados é carregado ou renovado. As restrições de integridade são pouco usadas no esquema lógico, devido ao custo adicional em operações de carga e renovação, porém, se ganha na declaração de restrições, pois fornecem um meio para informar aos usuários o que os dados significam;
- Chaves temporais: Os bancos de dados operacionais envolvem dados atuais e os bancos de dados de apoio à decisão envolvem dados históricos, tendendo a incluir *timestamp*, tradução livre carimbo de tempo, incluindo no banco de dados colunas do tipo *timestamp*.

2.5.2 Implicações do Projeto Físico

2.5.2.1 Particionamento

O Particionamento (fragmentação) divide uma tabela em conjunto de partições/fragmentos disjuntos para fins de armazenamento físico, melhorando o gerenciamento de acesso da tabela. (DATE, 2003)

Cada tabela recebe seus recursos de hardware diminuindo a disputa por recursos entre partições. As tabelas são particionadas horizontalmente (poucos produtos o fazem) por meio da função de particionamento que usa os valores das colunas selecionadas (chave de partição) e retorna um número ou endereço de partição. Os tipos de função de particionamento são o particionamento de intervalo (*range partitioning*), de *hashing* e de rodízio (*round-robin*) dentre outros (DATE, 2003).

2.5.2.2 Índices (indexação)

O uso correto de índice reduz a E/S (entrada e saída), onde o primeiro tipo disponível foi o *b-tree* (árvore b) e com o passar dos anos surgiram os tipos *bitmap*, *hashing*, multi-tabelas, *booleanos*, funcionais, dentre outros. (THOMAZ, 2010)

Segundo Elmasri e Navathe (2011, p. 422), índices são estruturas de acesso auxiliares que “são utilizadas para agilizar a recuperação de registros em resposta a certas

condições de pesquisa”, permitindo o acesso eficiente aos registros, com base nos campos de indexação, sendo os mais predominantes baseados em arquivo ordenado (índices de único nível) e estruturas em árvore (índices multinível) *b-tree*.

- Índice *B-tree*: proporciona acesso eficiente para consultas de intervalos com atualização relativamente eficiente.
- Índice *bitmap*: Este tipo de índice é eficiente para consultas envolvendo conjuntos de valores (perde eficiência quando os conjuntos ficam grandes demais), aceita várias operações relacionais (junção, união, restrição de igualdade...), executadas inteiramente dentro dos índices por meio de operações booleanas simples (*AND*, *OR*, *NOT*) sobre os vetores de bits. O acesso aos dados reais é feito quando o conjunto do resultado final for retornado, tendo atualização de índice de bitmap relativamente ineficiente.
- Índice *hashing*: Esse tipo de índice conhecido como endereçamento hashing, são eficientes para acesso a linhas específicas (não intervalo), seu custo computacional é proporcional ao número de linhas com a regra de não estender para acomodar valores de chaves adicionais. Usado para implementar junções de modo eficiente.
- Índice multitabela: Esse tipo de índice conhecido como índice de junção contém em uma entrada de índice de multitabela ponteiros para linhas de várias tabelas ao invés de ponteiros para linhas em uma única tabela podendo melhorar o desempenho de junções e verificação de integridade de multitabelas de banco de dados.
- Índice booleano: Esse tipo de índice conhecido como índice de expressões, indica quais linhas de determinada tabela uma expressão booleana envolvendo colunas da tabela é avaliada como *true* (verdadeiro), sendo índices valiosos quando a expressão booleana relevante é um componente comum das condições de restrição.
- Índices funcionais: Esse tipo de índice executa a indexação das linhas de uma tabela, com base no resultado da chamada de alguma função especificada sobre os valores.

2.5.2.3 *Surrogate key* (chave substituta)

Lima (2011, p. 5) descreve sobre chave substituta “levando esse conceito para uma modelagem dimensional, aprendemos que chaves compostas em dimensões causam uma certa complexidade na estruturação de um SQL, além de afetar o tempo de execução desse SQL”. Assim sendo, quando uma tabela origem de uma dimensão possui uma chave composta para a identificação única da linha na tabela, “usamos uma *SURROGATE KEY* única para representar essa *primary key* composta. Mesmo que não tenhamos uma chave composta no sistema de origem, criamos uma *Surrogate Key* na dimensão (com a finalidade de *Primary Key*), para permitir um controle histórico dos dados”.

A seguir apresentam-se os componentes de um data Warehouse.

2.6 COMPONENTES DE UM *DATA WAREHOUSE*

A seguir apresentam-se algumas definições e componentes de um *data warehouse* de acordo com Rob e Coronel (2011).

2.6.1 Integrado

Segundo Rob e Coronel (2011, p 548), “o *data warehouse* é um banco de dados consolidado e centralizado que integra dados provenientes de toda a empresa, de várias fontes e formatos”, e para complementar essa consolidação, “Entende-se por integração que todas as entidades comerciais, elementos e características de dados e métricas de negócios estão descritas do mesmo modo em toda a empresa”. Para não haver confusões, os dados no *data warehouse* precisam se adequar a um formato comum aceito em toda a empresa (exemplo: andamento do pedido definido com o texto “aberto” em um departamento, “recebido” em outro e “cancelado” em outro), a integração consome tempo, porém, aprimora a tomada de decisões e ajuda os gestores a compreender melhor as operações da empresa, vislumbrar oportunidades no negócio.

2.6.2 Orientado por Assunto

Os dados no *data warehouse* são dispostos e otimizados fornecendo respostas a perguntas derivadas de diversos setores da empresa, resumidos por tema (vendas, marketing, finanças, distribuição, transporte etc.), cada tema é relacionado a assuntos de interesse específicos (produto, clientes, departamentos, regiões, promoções etc.), sendo essa forma de organização de dados diferente das disposições mais orientada a processos ou funcional comum das transações, por exemplo, projetar estrutura de dados normalizado dando suporte ao processo de negócio armazenando dados na tabela fatura, o *data warehouse* tem orientação por assunto, onde a tabela fatura focou nos dados e não no processo que modifica os dados onde ao invés de o *data warehouse* armazenar uma fatura, armazena seu componente de venda por produto e venda por cliente devido a atividade de suporte a decisões exige a recuperação de resumos de vendas por produto ou cliente. (ROB; CORONEL, 2011).

2.6.3 Variável no Tempo

Diferente dos dados operacionais que focam nas transações correntes, no *data warehouse* o fluxo é representado no tempo, conter dados de projeções geradas por estatísticas ou modelos. Uma vez que os dados são carregados periodicamente no *data warehouse*, todas as agregações dependentes do tempo são recalculadas, um exemplo, os dados de vendas da semana passada carregados no *data warehouse* influencia o carregamento dos agregados da semana, mês e ano e outros períodos para os produtos, clientes, lojas e outras variáveis. Sendo os dados como um retrato instantâneo do histórico da empresa conforme medido por suas variáveis, o componente tempo é peça fundamental, onde o *data warehouse* tem uma *ID* de tempo utilizado para gerar resumos e agregações por semana, mês, trimestre, ano... e quando um dado é inserido a *ID* atribuída não pode ser alterada. (ROB; CORONEL, 2011).

2.6.4 Não Volátil

Quando inserido um dado no *data warehouse* o mesmo nunca será removido, representando o histórico da empresa, com os dados operacionais inseridos continuamente, crescendo em volume de dados com vários gigabytes, se diferenciando dos bancos de dados operacionais (transações). (ROB; CORONEL, 2011).

3 MATERIAIS E MÉTODOS

Este capítulo apresenta os materiais e o método utilizado para realização deste trabalho. Os materiais, referem-se às tecnologias como linguagens e ferramentas para a modelagem e a implementação do sistema. O método contém as etapas com os principais procedimentos utilizados para o desenvolvimento do *data warehouse*, abrangendo o levantamento dos requisitos aos testes.

O método utilizado no presente estudo foi fundamentado em estudo de caso com dados (*dump* da base de dados operacional) de uma empresa da área de agronegócio.

3.1 MATERIAIS

As ferramentas e as tecnologias utilizadas para compor a modelagem, a arquitetura e a implementação do *data warehouse* são:

- a) *Azure Data factory* para executar o *ETL*;
- b) *Data Management Gateway* para enviar os dados do servidor transacional para o serviço fábrica de dados-*data fabric* processar;
- c) *SGBD Oracle 11g R2* para armazenar o banco de dados do *data warehouse*;
- d) *SQL Developer* para administrar consultas *SQL*;
- e) *Power Architet* para criar o modelo lógico *E/R* e dimensional com tabela fato e dimensões;
- f) *Power BI* ferramenta para demonstrar consulta ao *data warehouse*;
- g) *Microsoft Visio* para desenhar a arquitetura do ambiente *on premisse*.

3.1.1 *Azure Data Factory*

Segundo o site Channel 9, o Azure Data factory é um serviço baseado em nuvem totalmente gerenciado para compor o armazenamento e o processamento de dados de forma simplificada, com os *pipelines* (dutos) agregando os dados, transformando e carregando no destino. Abaixo alguns componentes da fábrica de dados.

- a) Fábrica de dados: onde os dados serão trabalhados *ETL* em nuvem

- b) Conjunto de dados: *input* e *output dataset*.
 - *Inputdataset*: cria-se os dados que serão obtidos no banco de dados de transação
 - *Outputdataset*: cria-se os dados que serão obtidos no banco de dados do *data warehouse*.
- c) *Pipelines*: cria o túnel para o tráfego dos dados entre a origem (banco de dados transacional) e destino (*data warehouse*). Também aqui se define a frequência e intervalo do envio dos dados, as políticas de cópia para o *data warehouse*.
- d) *Gateway* de dados: O *gateway* de dados é onde se parametriza as propriedades para se conectar nos agentes instalados nos ambientes *onpremise* que faz a ligação com os *pipelines* para poder entregar os dados na fábrica de dados.

O resultado será a orquestração e transformação dos dados com redução de custos operacionais e alta disponibilidade (tolerância a falhas).

3.1.2 Data Management Gateway

O *Data Management Gateway* irá transferir os dados do SGBD transacional através de dutos (*pipeline*), e enviado para o destino. (RAD, 2015)

3.1.3 SGBD Oracle 11g R2

Oracle 11g R2 (64 Bit): Responsável por armazenar as tabelas fato e dimensões no *data warehouse*.

3.1.4 SQL Developer

Utilizado para criar as consultas, agregações, tabelas fato, tabelas dimensões, consultas e estrutura do *data warehouse*.

3.1.5 Power Architect

Utilizado para criar o modelo dimensional estrela e o modelo lógico das tabelas vinculadas do banco de dados de transação.

3.1.6 Power BI

Power BI ferramenta para criar os relatórios e dashboard e publicar em frontend, navegador de internet, para consulta pelo usuário final das informações e indicadores.

3.1.7 Microsoft Visio

Aplicativo usado para desenhar a infraestrutura do *data warehouse*.

3.2 MODELAGEM E IMPLEMENTAÇÃO

O método utilizado para a realização deste trabalho expõe de maneira sequencial as atividades para obter o resultado. O escopo será em criar um *data mart* para uso no departamento comercial de grãos e conforme a necessidade ir expandindo o escopo do *data warehouse* para agregar mais valor. A seguir, apresentam-se as fases definidas para este trabalho.

3.2.1 Requisitos

As definições dos requisitos será disponibilizar informações que suporte atender indicadores na parte de entrada de grãos, validando o *data warehouse*. Uma das atividades que tem como entrada de informações da originação de grãos proveniente de produtos agrícolas como soja, trigo, milho, feijão, é o lançamento dos romaneios através de ERP.

Assim gera-se o seguinte indicador através do *data warehouse*:

- Recebimento: recebimento do produto nas filiais.

- Resultados: Quantidade de grãos de entrada no armazém, podendo selecionar por filial, por cliente, por tipo de grãos acumulados no ano.

Abaixo a Figura 04, mostra todo o processo de recebimento de grãos na gestão de pátio utilizado pelo ERP Maxicon de Toledo - PR.

1. Entrada na balança
2. *Imput* dos dados:
3. Dados de pesagem
4. Classificação
5. Geração do romaneio de entrada/recebimento do grão

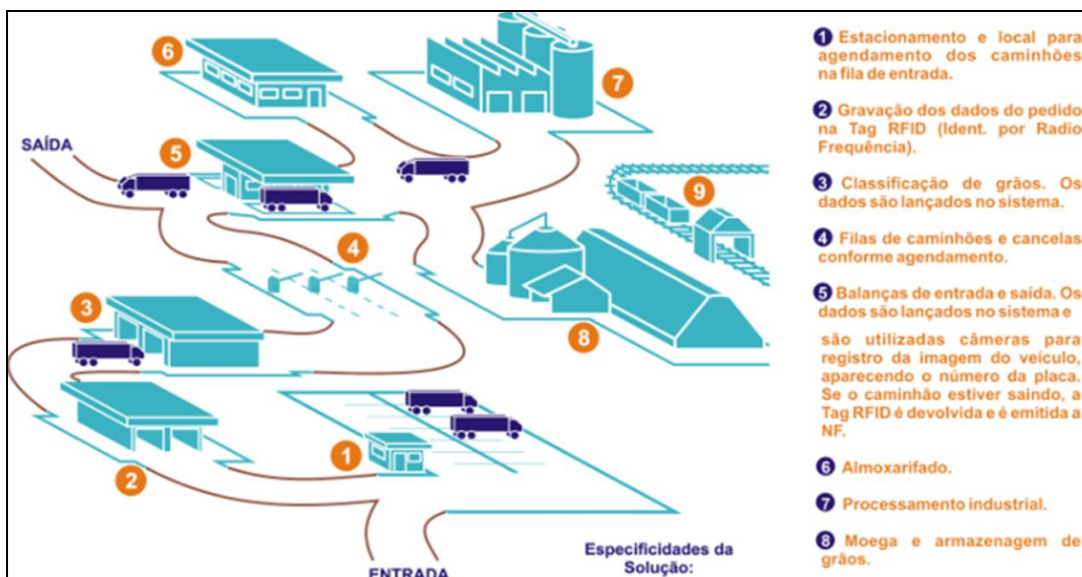


Figura 04 – Processo de recebimento de grãos
Fonte: Site Maxicon Sistemas

Abaixo, na Figura 05, apresenta-se exemplo do processo de lançamento de entrada de romaneio do sistema Maxicon e Viasoft:

- Configuração: traz a configuração do lançamento da entrada do romaneio, identificando qual o tipo de produto que está entrando (soja, trigo, entre outros)
- Pessoa: A pessoa/ produtor que deu entrada dos grãos
- Romaneio: o número do romaneio
- Item: qual é o produto
- Safra: qual a safra
- Local de estoque: onde o produto se encontra
- Peso total: Peso total com a tara

- Tara: o peso do caminhão
- Peso líquido: peso tirando a tara
- Descontos: informações de descontos como impureza e umidade
- Classificação: bônus calculado pelo peso bruto

Figura 05 – Dados da entrada de pesagem

Fonte: Maxicon Sistemas

| Cód | Desconto | Tab | % Tot. Cont. | Quantidade | Item |
|-----|------------|-------|--------------|------------|------|
| 1 | UMIDADE | 13,00 | | | |
| 2 | IMPUREZA | 0,50 | 0,50 | 88 | |
| 5 | DESCARTE I | | | | |
| 4 | DANIFICADO | | | | |
| 3 | AVARIADOS | | | | |
| 6 | RETENÇAO | | | | |
| 15 | QUEBRA TEI | | | | |

| Cód | Classificação | % | Quantidade | PH |
|-----|---------------|--------|------------|----|
| | PADRAO | 100,00 | | |

Figura 06 - Tela do sistema ERP para entrada de romaneio

Fonte: Viasoft (2017)

3.2.2 Metodologia

O *data warehouse* se dá em ambiente *onpremise* de forma centralizada no ambiente da empresa. O modelo de dados dimensional utiliza-se o *schema* estrela devida ter mais benefícios em geração de relatórios se comparado com a modelo entidade/relacionamento (*E/R*). Utilizado técnica *top/down* no desenvolvimento do trabalho, os níveis de granularidade (agregação), desenvolvido pensando no menor tempo de processamento. O processamento ETL para a carga, extração, transformação feito utilizou-se o *Azure data fabric*, onde na origem e destino usa-se SGBD relacional Oracle 11gR2.

Com as necessidades levantadas construí-se o *data mart* e dimensões de acordo com o método Segundo Kimball *et al.* (1998) onde:

1. Escolher o *data mart*;
2. Escolher a granularidade (detalhamento);
3. Escolher as dimensões;
4. Escolher a tabela fato.

3.2.3 Modelagem de Dados Dimensional

A modelagem de dados dimensional em estrela aplicada para agrupar os dados, criar as tabelas de dimensões e fato para atendendo os requisitos.

3.2.4 Criação do Ambiente e Arquitetura

O Ambiente e arquitetura se dá de forma centralizado em ambiente *onpremise*, com recursos de rede local, ou seja, dentro da empresa implantado o *data warehouse*, sendo que somente o ETL dá-se processado como serviço em nuvem.

4 RESULTADO E DISCUSSÃO

Este capítulo apresenta o *data warehouse* desenvolvido como resultado deste trabalho. Inicialmente é apresentada a descrição e em seguida é apresentada a sua modelagem.

4.1 APRESENTAÇÃO DO DATA WAREHOUSE

Abaixo, apresenta-se o desenvolvimento do projeto lógico e físico do *data warehouse*.

4.1.1 Arquitetura do Data Warehouse

Ambiente desenvolvido para que se tenha um sistema de *data warehouse* utilizando banco de dados que as empresas de agronegócio da região utilizam (*Oracle*) e que o ambiente em nuvem Azure possa disponibilizar para trabalhar a fase de carga dos dados no *data warehouse* (*ETL*).

Vemos abaixo a arquitetura do *data warehouse* implementado de forma centralizada para compartilhar uma arquitetura em comum, com dimensões e fatos de acordo a necessidade.

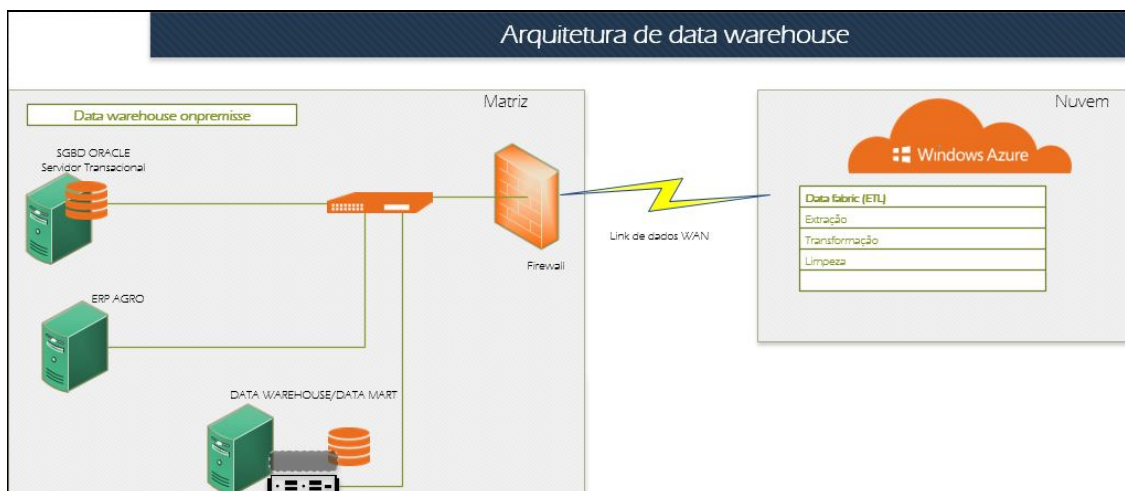


Figura 07 - Arquitetura do *data warehouse*

Fonte: do autor

4.2 MODELAGEM DO DATA WAREHOUSE

O Desenho do data warehouse inicialmente foi definido um modelo em estrela para implementar o *data mart*, a escolha do *data mart* para atender o assunto recebimento de grãos do setor de grão e a limitação para somente um assunto, deve-se a necessidade de proporcionar o resultado do trabalho, em curto espaço de tempo.

Tem-se um assunto recebimento de grãos, onde existe uma única tabela fato histórica (FATO_ROMA_REC), contendo os recebimentos e dados resumidos.

A chave primária (PK) da tabela FATO_ROMA_REC é uma chave substituta para garantir a unicidade das informações, a tabela fato contém colunas de cada dimensão, cada chave é uma chave gerada pelo SGBD nas dimensões, cada dimensão é representada por uma única tabela fato.

| ASSUNTO (FATO) | DIMENSÃO |
|----------------|----------------------------------------------------------------------------|
| FATO_ROMA_REC | DIME_ESTAB, DIME_CLIENTE, DIME_ITEMAGRO, DIME_TEMPO, DIME_ROMA |

Quadro 06 – Fato recebimento de grãos

Fonte: do autor (2017)

Cada chave primária das dimensões representa uma chave primária da tabela fato, basicamente é composto de uma grande tabela central (FATO_ROMA_REC) e um conjunto de tabelas de dimensão dispostas ao redor da tabela fato.

4.2.1 Projeto Lógico

Na Figura 08 esta elencada o diagrama em estrela visualizando as tabelas envolvidas no schema lógico:

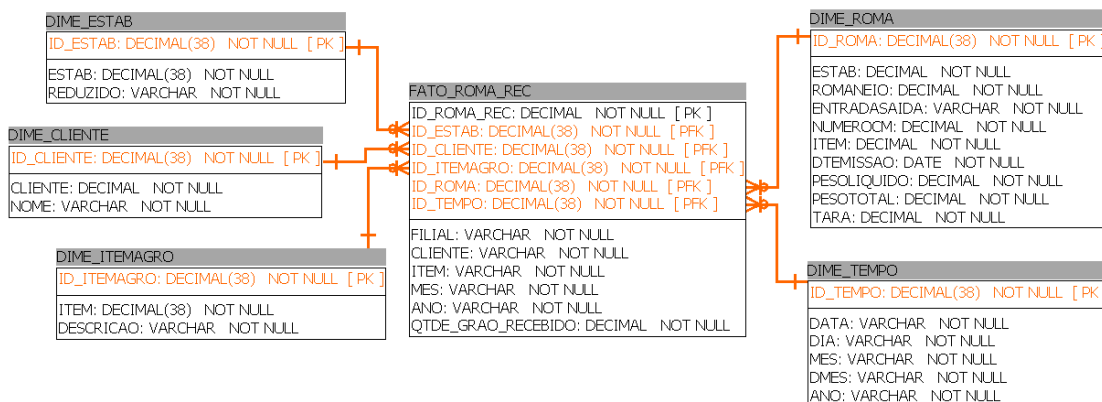


Figura 08 - Schema lógico

Fonte: do autor

4.2.2 Projeto Físico

Devido o produto Oracle (SGBD) não ter um tipo de dados de auto incremento, simula-se com sequência e *trigger* para o campo com auto incremento, no caso as chaves substitutas das tabelas.

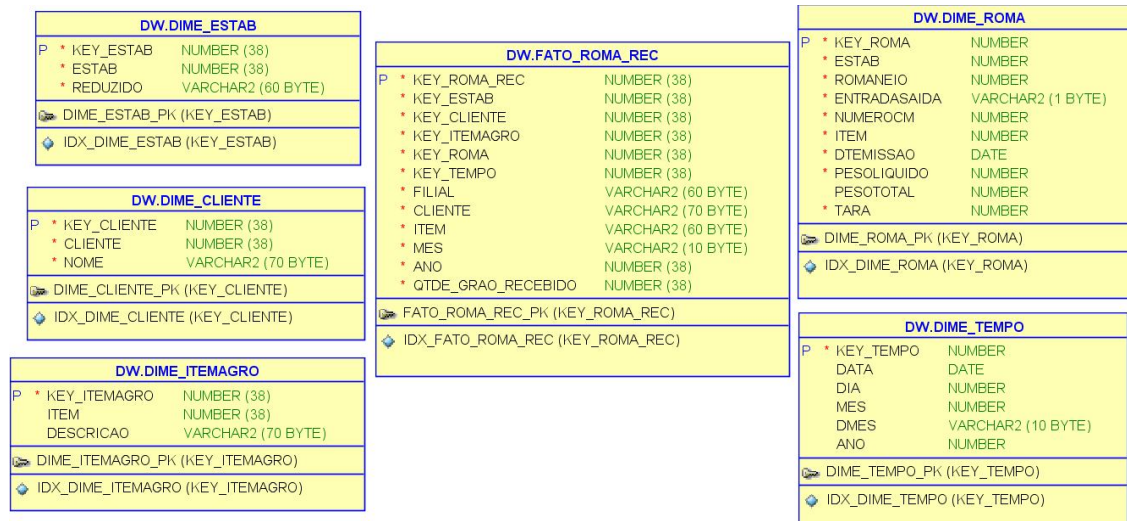


Figura 09 - Sequência e *trigger* para o campo com auto incremento

Fonte: do autor

Através do SQL Developer criou-se o schema DW, *datafiles*, tabelas fato e dimensões, índices *b-tree* e sequência (garantindo a unicidade).

Os códigos SQL utilizados se encontram no Anexo 1. No Quadro 07 abaixo vê-se os vínculos das tabelas origem e destino, e na Figura 10 os objetos (tabelas) criadas no *data warehouse*:

| | |
|------------|---------------|
| "FILIAL" | DIME_ESTAB |
| "CONTAMOV" | DIME_CLIENTE |
| "ITEMAGRO" | DIME_ITEMAGRO |
| "ROMA" | DIME_ROMA |
| | DIME_TEMPO |

Quadro 07 - códigos SQL

Fonte: Do autor

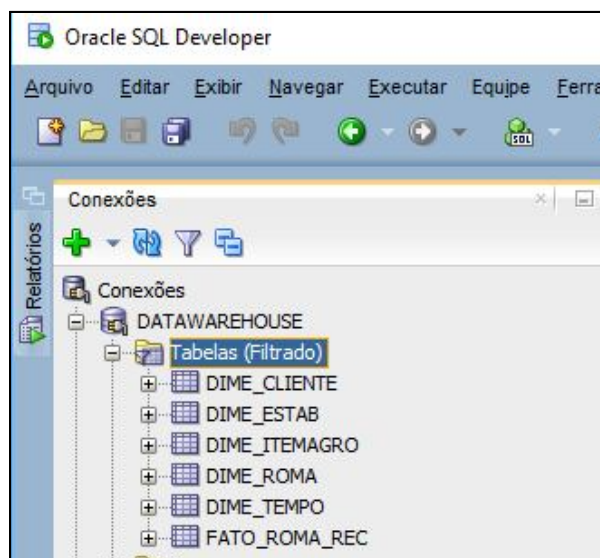


Figura 10 (a) - Tabelas do *data warehouse*

Fonte: Do autor

4.3 CRIAÇÃO DO AMBIENTE DE ETL E *DATA WAREHOUSE*

Nesta etapa se faz a preparação dos dados para que sejam alimentadas no *data warehouse*.

Os indicadores e respostas que serão implementados para a validação do ambiente de *data warehouse* são:

- Total de grãos recebido por cliente;
- Total de grãos recebido por filial;
- Total de grãos recebido por item.

4.3.1 ETL

Abaixo no Quadro 08 consta o resumo das *queries* executadas no *data factory* para extrair, limpar/transformar e carregar os dados para as tabelas de dimensões do *data warehouse*. Maior detalhe poderá ser verificado no Anexo 2:

| SQL QUERY |
|---------------------------------------------------|
| 1-Cria a estrutura para a dimensão dime_estab |
| 2- Cria a estrutura para a dimensão dime_cliente |
| 3- Cria a estrutura para a dimensão dime_itemagro |
| 4- Cria a estrutura para a dimensão dime_roma |
| 5- Cria a estrutura para a dimensão dime_tempo |

Quadro 08 - queries

Fonte: Do autor

4.3.2 Extração, Transformação e Carga dos Dados

Após definir quais os dados vão ser extraídos, transformados e carregados no datawarehouse, o serviço de pipeline do *data fabric* do Azure, envia os dados da origem (SGBD transacional) para o destino (*data warehouse*), com a ajuda dos *gateways* de dados instalados nos SGBDS de origem e destino.

Um recurso interessante do serviço *data fabric* é a cópia paralela, onde é feito de 1 a 32 cópias simultaneamente.

A Figura 11 aponta os serviços online na nuvem Microsoft Azure:

The screenshot shows the 'Serviços vinculados' (Linked Services) page in the Microsoft Azure portal. The page is divided into two main sections: 'Ações' (Actions) and 'Conteúdo' (Content). The 'Ações' section includes buttons for 'Criar e implantar', 'Copiar dados (VISUALIZAÇÃO)', 'Monitorar e Gerenciar', 'Pipelines de exemplo', 'Diagrama', and 'Métricas e operações'. The 'Conteúdo' section shows a summary of data sets (8), pipelines (1), and linked services (4). The linked services are categorized into 'REPOSITÓRIOS DE DADOS' (Data Repositories), 'GATEWAYS DE DADOS' (Data Gateways), and 'COMPUTAÇÃO' (Compute). The 'REPOSITÓRIOS DE DADOS' section lists 'D-ORACLE-DATAWARE...' and 'O-ORACLE-ERP', both with a status of 'Online'. The 'GATEWAYS DE DADOS' section lists 'ERP' and 'DATAWAREHOUSE', both with a status of 'Online'. The 'COMPUTAÇÃO' section indicates that there are no compute resources present in this data factory.

| SERVIÇO VINCULADO | TIPO | STATUS |
|---------------------------------------|-----------------------|--------|
| REPOSITÓRIOS DE DADOS | | |
| D-ORACLE-DATAWARE... | Banco de Dados Oracle | Online |
| O-ORACLE-ERP | Banco de Dados Oracle | Online |
| GATEWAYS DE DADOS | | |
| ERP | | Online |
| DATAWAREHOUSE | | Online |
| COMPUTAÇÃO | | |
| Nada presente nesta fábrica de dados. | | |

Figura 11 (b) - Tabelas do data warehouse
Fonte: Portal de serviços Microsoft Azure (2016)

Abaixo vemos todo o processo de ETL validado para envio ao *data warehouse*.

The screenshot shows the 'Copy Data (fabricaDados)' task configuration and deployment status. The task is titled 'Copy Data (fabricaDados)' and shows a flow from an Oracle source to an Oracle destination. The source is 'Oracle 4 table(s)' and the destination is 'Oracle 4 table(s)'. The 'Copy Run Time Region' is 'On Premises'. The deployment status is 'Deployment complete' with a list of successful steps including validating environment, registering connections, creating datasets, and creating pipelines.

Deployment complete

- Validating Environment ✓
 - Validation passed ✓
- Registering Connections ✓
 - S-ORCL-OLTP ✓
 - D-ORCL-DW ✓
- Creating Datasets ✓
 - InputDatasets-4gy-VIASOFT-CONTAMOV ✓
 - InputDatasets-4gy-VIASOFT-FILIAL ✓
 - InputDatasets-4gy-VIASOFT-ITEMAGRO ✓
 - InputDatasets-4gy-VIASOFT-ROMA ✓
 - OutputDatasets-4gy-DW-DIME_CLIENTE ✓
 - OutputDatasets-4gy-DW-DIME_FILIAL ✓
 - OutputDatasets-4gy-DW-DIME_ITEMAGRO ✓
 - OutputDatasets-4gy-DW-DIME_ROMA ✓
- Creating Pipelines ✓
 - PL-OLTP-DATAWAREHOUSE ✓

Figura 12 - Fábrica de dados
Fonte: Portal de serviços Microsoft Azure (2016)

4.4 CRIAÇÃO DE DASHBOARD

Após a criação do ambiente de *data warehouse/data mart*, utiliza-se a ferramenta *Microsoft Power BI Desktop* para criar e publicar o *dashboard* entregando uma percepção e entendimento das informações para os gestores de grãos.

No *dashboard* *qtde_graos_Recebidos* publicado no Power BI, o gestor da unidade de grãos poderá visualizar por filial, cliente e item a quantidade de grãos recebido no último ano.

No Anexo 3, apresenta-se o script utilizado para consultar o *data warehouse/data mart* e criar o *Dashboard* da Figura 13:

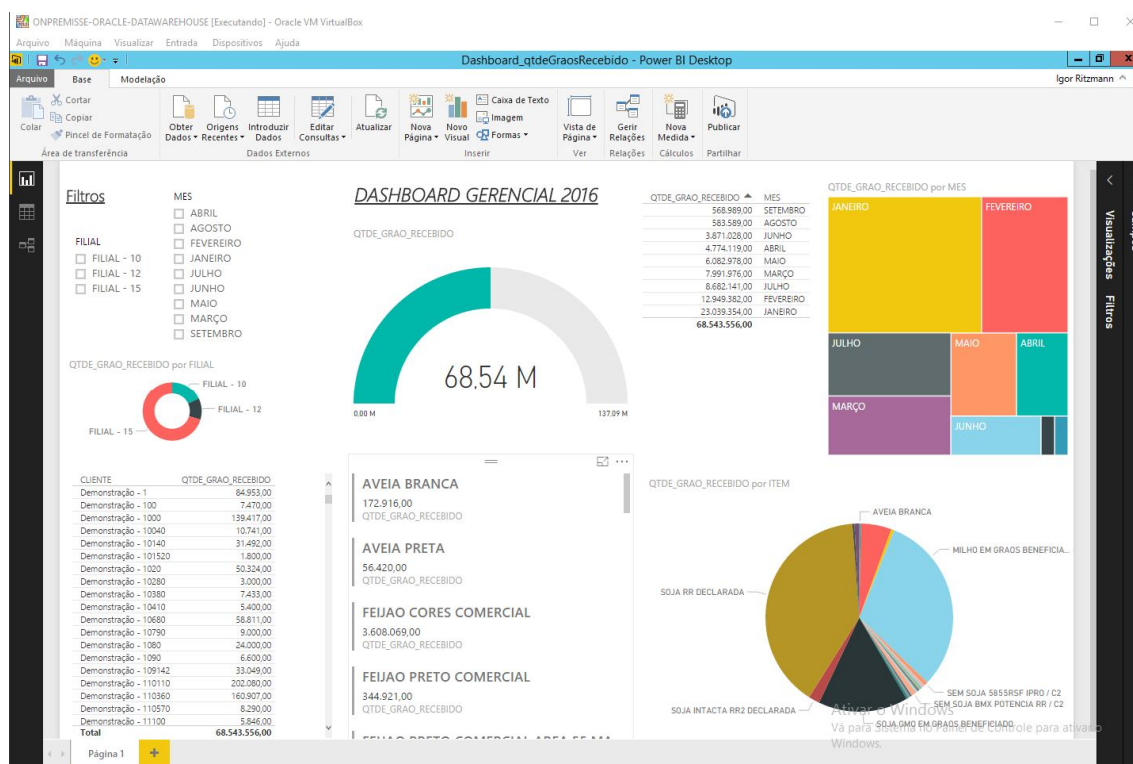


Figura 13 – Dashboard Gerencial
Fonte: Do autor

Para que seja consumido pelo gestor da unidade de grãos as informações via navegador de internet, por exemplo, o *Google Chrome* é gerado o relatório no Power BI.

Na figura 14 abaixo vemos o dashboard publicado e pronto para consumo disponibilizado no link: <https://app.powerbi.com>.

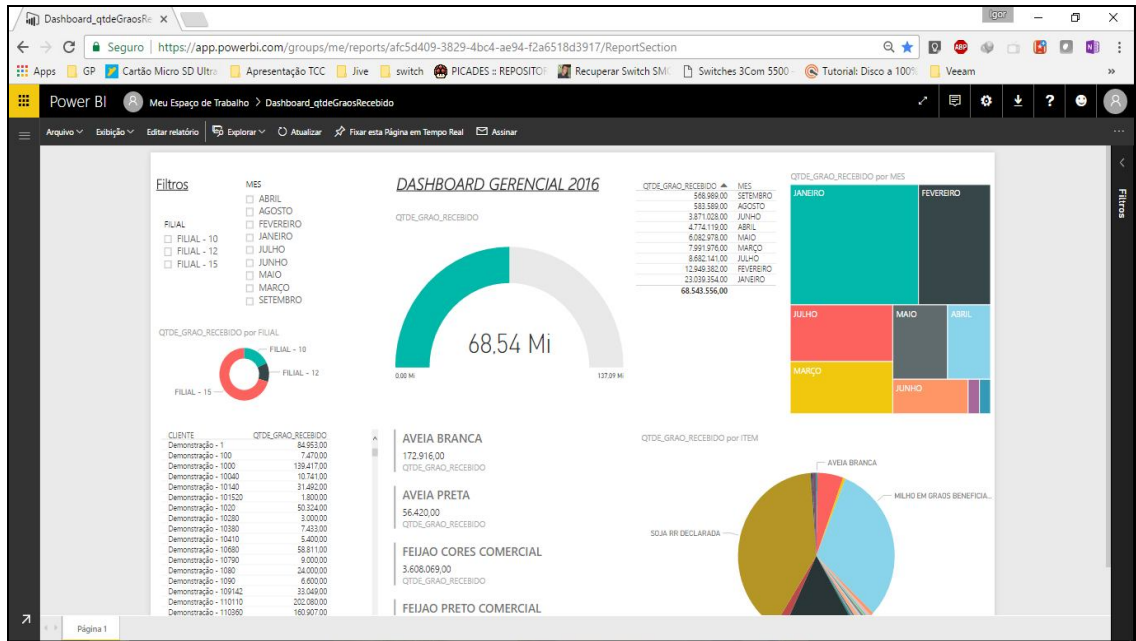


Figura 14 - Dashboard publicado no Power BI
 Fonte: Do autor

5. CONSIDERAÇÕES FINAIS

O *data warehouse* tem como objetivo o auxílio na tomada de decisão de uma gestão de uma empresa, como no agronegócio, pois apresenta uma arquitetura de consolidação de dados transacionais, para suportar e dar acesso às informações em um único ambiente integrado. Um estudo foi realizado neste trabalho sobre arquitetura em *cloud computing* (computação em nuvem) para o processo de *ETL* (*Extract, Transform, Load*). Algumas das considerações foram realizadas e seguem nos parágrafos abaixo.

A aquisição de um servidor com banco de dados *Oracle*® torna possível estabelecer e implantar um *data warehouse* em empresas de agronegócio para consolidar suas informações, não sobrecarregar o banco transacional a um custo reduzido, uma vez que se aproveita recursos já disponíveis na empresa.

Processar em nuvem os dados traz agilidade a um preço reduzido pago por hora, ao invés de ter toda a infraestrutura para este trabalho que requer *hardware* potente.

Deixar de onerar o banco de dados OLTP de transação traz um auxílio para os sistemas que precisam fazer as transações sem ter relatórios pesados utilizados em paralelo, o que não conseguimos com o *software* da Viasoft.

Criar *data mart* de acordo com o assunto e a necessidade do negócio viabiliza o tempo de entrega do *data warehouse* e justifica a sua implantação em uma empresa de agronegócio.

REFERÊNCIAS

AZURE Data Factory Overview Site Channel 9 <https://channel9.msdn.com/Blogs/Windows-Azure/Azure-Data-Factory-Overview>. Disponível em: jan. 2016.

CLEMES, M. **Data warehouse como suporte ao sistema de informações gerenciais em uma Instituição de Ensino Superior**: estudo de caso na UFSC. 2001. 118 f. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2001.

CORREA, F. E. *et al.* **Data warehouse for soybeans and corn market on Brazil**. In: 7th European Federation for Information Technology in Agriculture Conference, 2009, Wageningen, Países Baixos: EFITA, 2009.

DATE, C. J. **Introdução a Sistemas de Bancos de Dados**. 15. ed. Rio de Janeiro: Elsevier, 2003.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de banco de dados**. 6. ed. São Paulo: Pearson Education, 2011.

ERP Maxicon. Disponível em: <<http://www.maxiconsystems.com.br/>>. Acesso em: dez. 2016.

FERREIRA, C. **Uso de Big Data no agronegócio deverá crescer nos próximos anos**. Valor Econômico. São Paulo, out. 2014. Disponível em: <<http://www.valor.com.br/agro/3719090/uso-de-big-data-no-agronegocio-devera-crescer-nos-proximos-anos>>. Acesso em: 15 out. 2016.

IBM REDBOOKS. **Data modeling techniques for data warehousing**. Riverton, 1998. Disponível em: <<http://www.redbooks.ibm.com>>. Acesso em: 05 out. 2016.

KIMBALL, R. *et al.* **The Data Warehouse Lifecycle Toolkit**: Expert Methods for Designing, Developing, and Deploying Data Warehouses. Wiley, 1998.

MAXICON SISTEMAS. **Transformando conhecimento em soluções inteligentes**. Disponível em: <<https://www.maxiconsystems.com.br/>>. Acesso em: dez. 2016.

POPA, C. **Business Intelligence Performance and Capacity Impact**. 2010. 10 f. Universidade Politécnica de Bucarest. Disponível em: <http://www.scientificbulletin.upb.ro/rev_docs_arhiva/full8324.pdf>. Acesso em: 08 out. 2016.

RAMAKRISHNAN, R.; GEHRKE, J. **Sistemas de gerenciamento de banco de dados**. 3. ed. McGraw-Hill: São Paulo, 2008.

ROB, P.; CORONEL, C. **Sistemas de banco de dados**: Projeto, Implementação e gerenciamento. São Paulo: Cengage Learning, 2011.

SINGH, H. S. **Data Warehouse: Conceitos, Tecnologias, Implementação e Gerenciamento.** São Paulo: Makron Books, 2001.

THOMAZ, Y. **Oracle: quando usar índices B*Tree e Bitmap.** Viva o Linux, fev. 2010. Disponível em: <https://www.vivaolinux.com.br/dica/Oracle-quando-usar-indices-B*Tree-e-Bitmap>. Acesso em: 02 jan. 2017.

LIMA, Carlos Alberto. **Conceito de Surrogate Key – Chaves Substitutas (2009).** Disponível em: <<https://litolima.com/2011/03/09/conceito-de-surrogate-key-chaves-substitutas/>>. Acesso em: já. 2017.

VIASOFT. **Softwares empresariais.** Disponível em: <<http://www.viasoft.com.br/?lng=en>>. Acesso em: dez. 2016.

ANEXOS

ANEXO 1 - Criação de tablespaces e datafiles Oracle 11g R2

```
1 #CRIAR TABLESPACE E DATAFILES/INDICES/USUARIOS#
Logado com o usuário DW:
/** CRIA DW_DADOS **/
create tablespace DW_DATA01 datafile 'C:\app\Administrador\oradata\dw\Dw-Data01.dbs' size 1000M reuse
autoextend on next 50M
default storage (initial 512K next 256K
                minextents 1
                maxextents unlimited
                pctincrease 0)
online;

/** CRIA DW_INDICES **/
create tablespace DW_INDICES01 datafile 'C:\app\Administrador\oradata\dw\Dw-Indices01.dbs' size 500M
reuse
autoextend on next 20M
default storage (initial 512K next 256K
                minextents 1
                maxextents unlimited
                pctincrease 0)
online;

/** CRIA DW_USER **/
create tablespace DW_USERS01 datafile 'C:\app\Administrador\oradata\dw\Dw-Users01.dbs' size 100M reuse
autoextend on next 10M
default storage (initial 512K next 256K
                minextents 1
                maxextents unlimited
                pctincrease 0)
online;

2 #CRIAR USUARIO E DAR PRIVILEGIOS#
/** DROP USERS **/
DROP USER DW CASCADE;
/** CREATE USERS **/
CREATE USER DW IDENTIFIED BY DW
DEFAULT TABLESPACE DW_USER;
/** PRIVILÉGIOS **/
GRANT DBA TO DW;
```

```
GRANT CREATE SESSION TO DW;
/** TYPE USERS **/
GRANT EXECUTE ON SYS.UTL_RECOMP TO DW;
```

ANEXO 2 - Criação ambiente ETL

```
-----
/*#CRIAR TABELAS FATO E DIMENSOES#*/
-----
/* DIMENSAO FILIAL */
drop table DIME_ESTAB;
CREATE TABLE DIME_ESTAB
(
  KEY_ESTAB NUMBER(38) NOT NULL
, ESTAB NUMBER(38) NOT NULL
, REDUZIDO VARCHAR2(60) NOT NULL
, CONSTRAINT DIME_ESTAB_PK PRIMARY KEY
(
  KEY_ESTAB
)
USING INDEX
(
  CREATE UNIQUE INDEX IDX_DIME_ESTAB ON DIME_ESTAB (KEY_ESTAB ASC)
)
ENABLE
);
/*SEQUENCIA*/
DROP SEQUENCE DIME_ESTAB_SEQ;
CREATE SEQUENCE DIME_ESTAB_SEQ NOCACHE;

/* TRIGGER */
CREATE TRIGGER DIME_ESTAB_TRG
BEFORE INSERT ON "DIME_ESTAB"
FOR EACH ROW
BEGIN
  SELECT "DIME_ESTAB_SEQ".NEXTVAL INTO :NEW.KEY_ESTAB FROM DUAL;
END;

-----
/* DIMENSAO CLIENTE */
DROP TABLE DIME_CLIENTE;
CREATE TABLE DIME_CLIENTE
(
  KEY_CLIENTE NUMBER(38) NOT NULL
, CLIENTE NUMBER(38) NOT NULL
, NOME VARCHAR2(70) NOT NULL
, CONSTRAINT DIME_CLIENTE_PK PRIMARY KEY
(
  KEY_CLIENTE
)
USING INDEX
(
  CREATE UNIQUE INDEX IDX_DIME_CLIENTE ON DIME_CLIENTE (KEY_CLIENTE ASC)
)
)
```

```

ENABLE
);

/*SEQUENCIA*/
DROP SEQUENCE DIME_CLIENTE_SEQ;
CREATE SEQUENCE DIME_CLIENTE_SEQ NOCACHE;

/*TRIGGER*/
CREATE TRIGGER DIME_CLIENTE_TRG
BEFORE INSERT ON "DIME_CLIENTE"
FOR EACH ROW
BEGIN
SELECT "DIME_CLIENTE_SEQ".NEXTVAL INTO :NEW.KEY_CLIENTE FROM DUAL;
END;

-----

/* DIMENSAO ITEMAGRO */
DROP TABLE DIME_ITEMAGRO ;
CREATE TABLE DIME_ITEMAGRO
(
KEY_ITEMAGRO NUMBER(38) NOT NULL
, ITEM NUMBER(38)
, DESCRICAO VARCHAR2(70)

, CONSTRAINT DIME_ITEMAGRO_PK PRIMARY KEY
(
KEY_ITEMAGRO
)
USING INDEX
(
CREATE UNIQUE INDEX IDX_DIME_ITEMAGRO ON DIME_ITEMAGRO (KEY_ITEMAGRO ASC)
)
ENABLE
);
/*SEQUENCIA*/
DROP SEQUENCE DIME_ITEMAGRO_SEQ;
CREATE SEQUENCE DIME_ITEMAGRO_SEQ NOCACHE;

/*TRIGGER*/
CREATE TRIGGER DIME_ITEMAGRO_TRG
BEFORE INSERT ON "DIME_ITEMAGRO"
FOR EACH ROW
BEGIN
SELECT "DIME_ITEMAGRO_SEQ".NEXTVAL INTO :NEW.KEY_ITEMAGRO FROM DUAL;
END;

-----

/* DIMENSAO DIME_TEMPO */
DROP TABLE DIME_ROMA;
CREATE TABLE DIME_ROMA
(
KEY_ROMA NUMBER NOT NULL
, ESTAB NUMBER NOT NULL
, ROMANEIO NUMBER NOT NULL
, ENTRADASAIDA VARCHAR2(1) NOT NULL
, NUMEROCM NUMBER NOT NULL
, ITEM NUMBER NOT NULL
, DTEMISSAO DATE NOT NULL
, PESOLIQUIDO NUMBER NOT NULL

```

```

, PESOTOTAL NUMBER
, TARA NUMBER NOT NULL
, SAFRA NUMBER NOT NULL
, CONSTRAINT DIME_ROMA_PK PRIMARY KEY
(
  KEY_ROMA
)
USING INDEX
(
  CREATE UNIQUE INDEX IDX_DIME_ROMA ON DIME_ROMA (KEY_ROMA ASC)
)
ENABLE
);

/*SEQUENCIA*/
DROP SEQUENCE DIME_ROMA_SEQ;
CREATE SEQUENCE DIME_ROMA_SEQ NOCACHE;

/*TRIGGER*/
CREATE TRIGGER DIME_ROMA_TRG
BEFORE INSERT ON "DIME_ROMA"
FOR EACH ROW
BEGIN
  SELECT "DIME_ROMA_SEQ".NEXTVAL INTO :NEW.KEY_ROMA FROM DUAL;
END;

-----
/* DIMENSAO DIME_ROMA */
DROP TABLE DIME_ROMA;
CREATE TABLE DIME_ROMA
(
  KEY_ROMA NUMBER NOT NULL
, ESTAB NUMBER NOT NULL
, ROMANEIO NUMBER NOT NULL
, ENTRADASAIDA VARCHAR2(1) NOT NULL
, NUMEROCM NUMBER NOT NULL
, ITEM NUMBER NOT NULL
, DTEMISSAO DATE NOT NULL
, PESOLIQUIDO NUMBER NOT NULL
, PESOTOTAL NUMBER
, TARA NUMBER NOT NULL
, SAFRA NUMBER NOT NULL
, CONSTRAINT DIME_ROMA_PK PRIMARY KEY
(
  KEY_ROMA
)
USING INDEX
(
  CREATE UNIQUE INDEX IDX_DIME_ROMA ON DIME_ROMA (KEY_ROMA ASC)
)
ENABLE
);
/*SEQUENCIA*/
DROP SEQUENCE DIME_ROMA_SEQ;
CREATE SEQUENCE DIME_ROMA_SEQ NOCACHE;

/*TRIGGER*/
CREATE TRIGGER DIME_ROMA_TRG
BEFORE INSERT ON "DIME_ROMA"

```

```

FOR EACH ROW
BEGIN
  SELECT "DIME_ROMA_SEQ".NEXTVAL INTO :NEW.KEY_ROMA FROM DUAL;
END;
/*
CARREGAR DADOS PARA AS TABELAS DE DIMENSAO DO DATA WAREHOUSE
*/
-----
String de conexão (Serviços de dados)
Origem(Banco de dados de transação)      Destino (Datawarehouse)
{
  "name": "O-ORACLE-ERP",
  "properties": {
    "hubName": "fabricadados_hub",
    "type": "OnPremisesOracle",
    "typeProperties": {
      "connectionString":          "data source=200.181.183.177:1521/orcl;user
id=VIASOFT;password=*****",
      "gatewayName": "ERP",
      "userName": null,
      "password": null
    }
  }
}
{
  "name": "D-ORACLE-DW",
  "properties": {
    "hubName": "fabricadados_hub",
    "type": "OnPremisesOracle",
    "typeProperties": {
      "connectionString": "data source=200.181.183.177:1522/dw;user id=DW;password=*****",
      "gatewayName": "DATAWAREHOUSE",
      "userName": null,
      "password": null
    }
  }
}
}
}

```

ANEXO 3 – Script SQL para consultas OLAP

```

SELECT
  FILIAL.REDUZIDO AS FILIAL,
  CONTAMOV.NOME AS CLIENTE,
  ITEMAGRO.DESCRICAO AS ITEM,
  SUM (ROMA.PESOLQUIDO) AS QTDE_GRAO_RECEBIDO,
  AVG (ROMA.PESOLQUIDO) AS MEDIA_GRAO_RECEBIDO
FROM ROMA
INNER JOIN FILIAL ON
  (FILIAL.ESTAB = ROMA.ESTAB)

INNER JOIN CONTAMOV ON
  (CONTAMOV.NUMEROCM = ROMA.NUMEROCM)

INNER JOIN ITEMAGRO ON
  (ITEMAGRO.ITEM = ROMA.ITEM)
WHERE
  ROMA.DTEMISSAO >= '01/01/2016'
AND ENTRADASAIDA = 'E'
GROUP BY

```

FILIAL.REDUZIDO,
CONTAMOV.NOME,
ITEMAGRO.DESCRICAO,
ROMA.PESOLQUIDO
);