



**Fundação Educacional do Município de Assis**  
**Instituto Municipal de Ensino Superior de Assis - IMESA**

**VINÍCIUS DE MELLO LAUREANO**

**DATA WAREHOUSE DE UM BANCO DE DADOS OBJETO-  
RELACIONAL MULTIDIMENSIONAL**

ASSIS, SP

2011

**VINÍCIUS DE MELLO LAUREANO**

**DATA WAREHOUSE DE UM BANCO DE DADOS OBJETO-  
RELACIONAL MULTIDIMENSIONAL**

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito de Curso de Bacharelado em Ciência da Computação.

Orientadora: Profa. Dra. Marisa AtsukoNitto

Área de Concentração: Informática

ASSIS, SP

2011

## FICHA CATALOGRÁFICA

LAUREANO, ViníciusM.

Data Warehousing de um banco de dados objeto-relacional multidimensional/  
Vinícius de Mello Laureano. Fundação Educacional do Município de Assis - FEMA –  
Assis, 2010.

76p.

Orientador: Profa. Dra. Marisa AtsukoNitto Trabalho de Conclusão de Curso –  
Instituto Municipal de Ensino Superior de Assis – IMESA.

1. Banco de Dados 2.Orientação a Objeto 3. Multidimensional

CDD: 001.6

Biblioteca da FEMA

**VINÍCIUS DE MELLO LAUREANO**

**DATA WAREHOUSE DE UM BANCO DE DADOS OBJETO-  
RELACIONAL MULTIDIMENSIONAL**

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito de Curso de Graduação, analisado pela comissão examinadora:

Orientadora: Profa. Dra. Marisa AtsukoNitto

Analisador: Prof. Dr. Luiz Carlos Begosso

Assis, SP

2011

*Dedico este trabalho, aos meus pais Rosilene de Mello e Edi Wilson, pelo apoio, incentivo e principalmente a pessoa que sou hoje. Aos amigos e familiares, que puderam compartilhar momentos de tristeza e alegria.*

## **AGRADECIMENTO**

A Deus, por tudo;

A Profa. Dra. Marisa AtsukoNitto, pela orientação segura, paciência, disponibilidade e o conhecimento prestado durante a realização deste trabalho;

Agradeço a todos os professores do curso de Ciência da Computação pelos conhecimentos repassados nesses últimos anos;

Aos familiares, que de forma direta ou indireta ajudaram nas horas que precisei;

Aos amigos, que estiveram comigo todo este tempo trocando conhecimento, apoiando e crescendo juntos nessa caminhada;

A todos que de alguma forma contribuíram para a realização do trabalho.

## RESUMO

Neste trabalho foi realizado um amplo estudo sobre as técnicas para aquisição de informações na coleta e modelagem de dados para um *data warehouse*.

Serão explanados os conceitos básicos e as características principais do *data warehouse*, mostrando desde seu suporte à tomada de decisão. Traz também as técnicas utilizadas para a extração de dados e os modelos usados para a modelagem de um *data warehouse*, mostrando que o *datawarehouse* como a orientação a objeto não é mais uma tendência, e sim uma realidade; e podem trabalhar juntos na aplicação das tomadas de decisões, também chamadas de *Business Intelligence*(BI). Foi desenvolvido um estudo de caso para analisar o uso das técnicas de aquisição do BI na coleta de dados para o *data warehouse*. Neste estudo de caso foi apresentado como o conhecimento é adquirindo, mostrando também o processo de mineração, e armazenamento dos dados de uma forma multidimensional.

**Palavras-chaves:** data warehouse, multidimensional, tomada de decisões

## **ABSTRACT**

This work we present a comprehensive study on techniques for acquiring information gathering and modeling, for a data warehouse. The Basic concepts are explained and the main characteristics of the data data warehouse, from showing their support for decision making. It also brings the techniques used to extract data and models used for modeling a data warehouse. Showing that the data warehouse and object orientation is no longer a trend but a reality, and can work together in implementing the decision making, also called Business Intelligence(BI).Will develop a case study to analyze the use of the BI acquisition techniques in data collection for the data warehouse. In this case study is presented as knowledge is acquired, also showing the process of mining, and data storage in a multidimensional way.

**Keywords:**data warehouse,multidimensional; decision-making

## LISTA DE ILUSTRAÇÕES

Figura 1 - Criação de Tipo de Objeto .....	31
Figura 2 - Criação de Tabela de Objetos .....	31
Figura 3 - Arquitetura para Business Intelligence (THONSEM, 2002).....	34
Figura 4 – Cubo de Dimensões (KIMBALL ET AL, 1998) .....	40
Figura 5 - Modelo Dimensional em Estrela (INMON, 1987) .....	42
Figura 6 - Visão Geral do Problema .....	46
Figura 7 - Diagrama Objeto-Relacional do Banco De Dados .....	47
Figura 8 - Diagrama Entidade-Relacionamento do Data Warehouse.....	48
Figura 9 - Diagrama de Caso de Uso.....	49
Figura 10 - Diagrama de Classes.....	50
Figura 11 – Diagrama de Sequência Manter Cliente.....	52
Figura 12 – Diagrama de Sequência Manter Carro.....	53
Figura 13 – Diagrama de Sequência Manter Financeira .....	54
Figura 14 – Diagrama de Sequência Manter Forma de Pagamento .....	55
Figura 15 – Diagrama de Sequência Manter Marca.....	56
Figura 16 – Diagrama de Sequência Manter Tipo Veículo.....	57
Figura 17 – Diagrama de Sequência Manter Vendedor .....	58
Figura 18 – Diagrama de Sequência Manter Venda .....	59
Figura 19 – Manter Dados Específicos Data Warehouse.....	60
Figura 20 – Interface Cadastro de Clientes.....	62
Figura 21 – Interface Cadastro de Marcas .....	63
Figura 22 – Interface Cadastro de Tipos de Veículos .....	64
Figura 23 – Interface Cadastro de Veículos .....	65
Figura 24 – Interface Cadastro de Forma de Pagamento .....	66
Figura 25 – Interface Cadastro de Financeira .....	67

Figura 26 – Interface Cadastro de Vendedor .....	68
Figura 27 – Interface Movimentação Venda.....	69
Figura 28 – Modelagem Multidimensional Estrela Data Warehouse .....	70
Figura 29 - Visão Geral SQL Developer.....	72

## LISTA DE TABELAS

Tabela 1 - Características de um Data Warehouse. (SANTOS, 2003).....	18
Tabela 2 – Principais Características dos Modelo Relacional e Orientado a Objeto (ENDEL A, 2011).....	30
Tabela 3 - Comparação nas Formas de Acesso aos Dados .....	44

## Sumário

<b>1 - INTRODUÇÃO</b> .....	<b>17</b>
1.1 - Objetivo .....	20
1.2 - Justificativas .....	20
1.3 - Motivação .....	21
1.4 - Estrutura do Trabalho.....	21
<b>2 - Banco de Dados</b> .....	<b>22</b>
2.1 - Banco de Dados Orientado a Objeto (BDOO) .....	22
2.2 – Conceitos Básicos .....	23
2.2.1 – Abstração .....	24
<b>2.2.2 - Objetos</b> .....	<b>24</b>
<b>2.2.3 - Classes</b> .....	<b>25</b>
<b>2.2.4 - Herança</b> .....	<b>26</b>
<b>2.2.5 - Polimorfismo</b> .....	<b>26</b>
<b>2.2.6 - Encapsulamento</b> .....	<b>27</b>
2.3 – Bancos de dados orientados a objetos no mercado .....	27
2.4 - Banco de dados orientado a objetos x Banco de dados relacional .....	28
2.5 - Banco Objeto Relacional .....	30
<b>2.5.1 - Objeto Relacional Oracle</b> .....	<b>30</b>
<b>2.5.1.1 - Tipo de Objeto</b> .....	<b>31</b>
<b>2.5.1.2 - Tabela de Objetos</b> .....	<b>31</b>
<b>3 - Business Intelligence</b> .....	<b>32</b>

3.1 - Sistemas OLTP .....	34
3.2 – ETL (Extract, Transform, Loader) .....	35
3.3 - Data Staging Area .....	35
3.4 - Data Warehouse.....	36
<b>3.4.1 – Planejamento da Construção de Um <i>Data Warehouse</i>.....</b>	<b>38</b>
<b>3.4.1.1 - Modelagem entidade-relacionamento .....</b>	<b>38</b>
<b>3.4.1.2 - Modelagem dimensional .....</b>	<b>39</b>
<b>3.4.1.2.1 - Elementos de Um Modelo Dimensional .....</b>	<b>40</b>
<b>3.4.1 3 – Método Para a Construção de Um <i>Data Warehouse</i>.....</b>	<b>42</b>
<b>3.4.1 4 – Ferramentas de Acessos aos Dados.....</b>	<b>43</b>
<b>4 - Desenvolvimento do Projeto.....</b>	<b>45</b>
4.1 – Descrição do Problema.....	45
4.2 – Modelagem do Problema.....	45
4.3 – Especificação.....	46
<b>4.3.1 – Diagrama Objeto-Relacional .....</b>	<b>47</b>
<b>4.3.2 – Diagrama de Casos de Uso .....</b>	<b>48</b>
<b>4.3.3 - Diagrama de Classe.....</b>	<b>49</b>
<b>4.3.4 – Diagrama de Sequência.....</b>	<b>51</b>
<b>4.3.4.1 – Manter Cliente.....</b>	<b>51</b>
<b>4.3.4.2 – Manter Carro .....</b>	<b>53</b>
<b>4.3.4.3 – Manter Financeira.....</b>	<b>54</b>
<b>4.3.4.4 – Manter Forma de Pagamento .....</b>	<b>55</b>
<b>4.3.4.5 – Manter Marca .....</b>	<b>56</b>
<b>4.3.4.6– Manter Tipo Veículo .....</b>	<b>57</b>
<b>4.3.4.7 – Manter Vendedor .....</b>	<b>58</b>
<b>4.3.4.8 – Manter Venda.....</b>	<b>59</b>

4.3.4.9 – Manter Dados Específicos Data Warehouse.....	60
4.4 – Implementação do Aplicativo .....	61
4.4.1 – Metodologia Utilizada.....	61
4.4.1.1 - Manter Dados .....	61
4.4.1.1.1 – Manter Clientes.....	62
4.4.1.1.2 – Manter Carros - Marcas .....	62
4.4.1.1.3 – Manter Carros - Tipo de Veículo .....	63
4.4.1.1.4 – Manter Carros - Modelos .....	64
4.4.1.1.5 – Manter Financeiro –Formas de Pagamento .....	65
4.4.1.1.6 – Manter Financeiro – Financeira.....	66
4.4.1.1.7 – Manter Vendedores .....	67
4.4.1.1.8 – Manter Venda.....	68
4.4.1.2 - Gravar Dados Específicos no <i>Data Warehouse</i> .....	69
4.4.1.2.1 - A Construção do <i>Data Warehouse</i> .....	70
4.4.1.3 - A Extração de Dados <i>Data Warehouse</i> .....	71
4.4.1.3 - Extraindo Dados.....	71
5. CONCLUSÃO .....	73



## LISTA DE ABREVIATURAS E SIGLAS

BD	Banco de Dados
BI	<i>Business Intelligence</i>
DW	<i>Data Warehouse</i>
DSS	<i>Decision Support Systems</i>
EIS	<i>Executive Information Systems</i>
ERP	<i>Enterprise Resources Planning</i>
MD	Modelo Dimensional
ME-R	Modelo Entidade Relacionamento
MR	Modelo Relacional
OLAP	<i>Online Analytical Processing</i>
OLTP	<i>Online Transaction Processing</i>
OO	<i>Orientação a Objeto</i>
XML	<i>Extensible Markup Language</i>
UML	<i>Unified Modeling Language</i>
CAD	<i>Computer Aided Manufacturing</i>

## 1 - INTRODUÇÃO

Nos últimos anos o relacionamento entre as empresas e seus clientes vem tomando uma nova direção E as empresas já não satisfazem seus clientes apenas conhecendo bem seus produtos ou serviços: existe ainda a necessidade, tão importante quanto, de se conhecer também os próprios clientes, suas preferências, seus padrões e suas faixas de valores.

Esse novo enfoque comercial que está sendo dado ao relacionamento entre empresa e cliente aumentou a quantidade de informações geradas nas transações operacionais das organizações. As informações começaram a ser geradas e conseqüentemente armazenadas, formando grandes depósitos de dados conhecidos como *Data Warehouse* (DW), e que são considerados por muitos como a memória da empresa (CENTENARO ET AL, 2004).

A construção de um *Data Warehousing* (DWing) é o processo de desenvolvimento de um ambiente de banco de dados (*Data Warehouse* ou DW) adequado à análise de negócios e ao apoio à tomada de decisões gerenciais e estratégicas. Esta tecnologia é a base sobre a qual se monta a Inteligência de Negócios (*Business Intelligence* ou BI) das empresas.

As primeiras edições de SGBDs (Sistema de Gerenciamento de Banco de Dados) concentravam se em um único banco de dados que atendia todos os propósitos da comunidade de informática. No entanto, as diferentes características dos dados exigem uma melhor adaptação dos bancos de dados a essas peculiaridades. Então, foram desenvolvidos novos modelos de sistemas de banco de dados e ferramentas computacionais mais adequadas a essas áreas de conhecimento específicas, de modo a atender com maior eficiência e qualidade (SANTOS 2003).

Os dois tipos de sistemas de banco de dados que se relacionam com a área de *Data Warehouse* são: bancos de dados operacionais e os bancos de dados analíticos. Os bancos operacionais dão suporte a todas as operações de uma organização, sendo utilizado com grande frequência e registrando a situação momentânea da organização, seja, por exemplo, no cadastramento de clientes ou na baixa de um produto no estoque. Os bancos de dados analíticos visam

registrar a história da organização de modo a dar suporte em momentos de decisão, utilizado com baixa frequência. Essa divisão fica clara quando se avalia as diferenças entre as características físicas dos dados operacionais (específico) e analíticos (sumarizado). A tecnologia de suporte aos dados operacionais é diferente da tecnologia de análise (KORTH e SILBERSCHATZ, 1999). A tabela 1 destaca as principais características de um *Data Warehouse*.

<b>Característica</b>	<b>Descrição</b>
Orientação a assunto	Dados organizados de modo como o usuário se refere a ele
Integração	Inconsistências são removidas e informações conflitantes, isto é, os dados são “limpos”.
Não volátil	Somente leitura de dados, dados não atualizados por usuário.
Séries históricas	Os dados estão em séries históricas, não somente na situação corrente.
Resumidos	Os dados operacionais estão e são agregados
Abrangência	Mantêm uma série temporal, considerando que mais dados serão armazenados.
Não normalizado	Dados podem ser redundantes
Metadados	Dados sobre dados

**Tabela 1 - Características de um Data Warehouse. (SANTOS, 2003).**

Um *Data Warehouse* suporta análise de negócios e tomada de decisões através da integração de dados procedentes de diversos sistemas, muitas vezes incompatíveis entre si, em uma base de dados consolidada. Esta transformação de dados em informações significativas capacita os executivos

de uma organização a efetuar análises muito mais reais, precisas e consistentes (MEYER E CAMMON 1998). Em outras palavras, o *Data Warehouse* é uma base de dados projetada especificamente para atender às necessidades de tomada de decisão, em vez de atender aos sistemas de processamento de transações (DAPHNE, 1999).

A construção de um *Data Warehouse* pode ser feita através de uma modelagem multidimensional cujo projeto lógico do banco de dados utiliza a técnica de concepção e visualização de um modelo de dados de um conjunto de medidas que descrevem aspectos comuns de negócio. Sua utilização ajuda na sumarização e reestruturação dos dados e apresenta visões que suportam a análise dos valores destes dados (MACHADO, 2004).

O banco de dados multidimensional não armazena os dados como registros em tabelas, ele armazena os dados em *arrays* multidimensionais, possuindo um número fixo de dimensões (OLIVEIRA, 2002). Os modelos multidimensionais tiram proveito de relações inerentes aos dados, para gerar dados em matrizes multidimensionais chamadas cubos de dados.

Sua crescente utilização pelas empresas está relacionada à necessidade do domínio de informações estratégicas para garantir respostas e ações rápidas, assegurando a competitividade de um mercado altamente competitivo e mutável [MACHADO, 2004].

Nos últimos anos, o modelo de banco orientado a objetos tem ganhado espaço por ser uma tecnologia que oferece aumento de produtividade, segurança e facilidade de manutenção. Como objetos são modulares, mudanças podem ser feitas internamente, sem afetar outras partes do programa. Existem pelo menos dois fatores que levam a adoção desse modelo, a primeira é que banco de dados relacional se torna difícil trabalhar com dados complexos. A segunda é que aplicações são construídas em linguagens orientadas a objetos (Java, C++, C#) e o código precisa ser traduzido para uma linguagem que o modelo de banco de dados relacional entenda o que torna essa tarefa muito tediosa. Essa tarefa também é conhecida como “perda por resistência” (ELMASRI, 2005).

Neste projeto foi utilizada a modelagem dimensional como alternativa para a construção de um *data warehouse* de um banco orientado á objetos. Serão utilizadas as técnicas de aquisição do *Business Intelligencena* coleta de dados para o *data warehouse*. Será feita uma implementação de um estudo de caso para vendas utilizando o banco de dados Oracle e a linguagem de programação Java.

### 1.1 - Objetivo

O objetivo principal deste projeto de pesquisa e construção de uma aplicação utilizando técnicas de BI e como desenvolver a melhor forma para extrair dados de um banco orientado a objeto, para fins de análise de mercado. Para isso será apresentado uma modelagem dimensional como alternativa para a construção de um *data warehouse*, mostrando os seus elementos, algumas técnicas de modelagem, assim como os passos para o seu desenvolvimento. Para a validação do modelo adotado será feito um estudo de caso para vendas, utilizando o banco de dados Oracle e a linguagem de programação Java.

### 1.2 - Justificativas

A justificativa para o desenvolvimento deste projeto se deve em parte por inúmeras empresas já ter um *data warehouse* ou estar em processo de desenvolvimento de um. Assim, a primeira justificativa é a necessidade de implementação desta ferramenta como uma tendência geral em organizações de grande porte. Segundo, os sistemas de gestão que apóiam as tarefas diárias das organizações não têm o objetivo, na maioria dos casos, de fornecer informações detalhadas para auxiliar nas tomadas de decisões. Terceiro, justifica se a sua implantação, pois quando bem utilizado pode trazer diversas vantagens para a organização tornando-a mais competitiva e auxiliando na tomada de decisão.

### 1.3- Motivação

A grande motivação é que a informação por si só não é um diferencial competitivo para as organizações, mas sim a maneira como ela é utilizada, pois um dos maiores problemas que as organizações enfrentam na era competitiva da informação é como organizar tantos dados existentes. Dentre as diferentes ferramentas existentes de gerenciamento e organização de informação disponíveis no mercado o *data warehouse* faz um grande candidato.

### 1.4- Estrutura do Trabalho

A estrutura do trabalho será apresentada em forma de capítulos. Ela foi organizada em cinco capítulos, sendo o primeiro esta introdução.

No segundo capítulo, serão apresentadas as fundamentações teóricas sobre Banco de Dados Orientado a Objeto. No terceiro capítulo, serão apresentadas as fundamentações teóricas sobre, *data warehouse* e bancos dimensionais.

No quarto capítulo, será apresentado o estudo de caso para vendas, com toda a especificação e implementação.

No quinto capítulo, serão apresentadas as considerações finais e as projeções futuras.

## 2 - Banco de Dados

Neste capítulo, será apresentada a fundamentação teórica sobre banco de dados orientado a objeto, descrevendo seus conceitos, vantagens e desvantagens em relação ao modelo entidade-relacionamento. Será apresentada também a utilização de orientação a objeto no banco de dados, como objeto-relacional Oracle.

### 2.1 - Banco de Dados Orientado a Objeto (BDOO)

O banco de dados orientados a objetos é um fator emergente que integra banco de dados e a tecnologia de orientação a objetos. Os bancos de dados orientados a objeto surgiram primeiramente em projetos de pesquisa nas universidades e centros de pesquisa. Em meados dos anos 80, eles começaram a se tornar produtos comercialmente viáveis.

Um banco de dados orientado a objeto é um banco em que cada informação é armazenada na forma de objetos, e só pode ser manipuladas através de métodos definidos pela classe que esteja o objeto (GALANTE; MOREIRA; BRANDÃO, 2007).

O conceito de banco de dados orientado a objeto é o mesmo da linguagem orientada a objeto, havendo uma pequena diferença: a persistência de dados. Existem pelo menos dois fatores que levam a adoção desse modelo, a primeira é que banco de dados relacional se torna difícil trabalhar com dados complexos. A segunda é que aplicações são construídas em linguagens orientadas a objetos (Java, C++, C#) e o código precisa ser traduzido para uma linguagem que o modelo de banco de dados relacional entenda o que torna essa tarefa muito tediosa. Essa tarefa também é conhecida como “perda por resistência” (ELMASRI, 2005)

Segundo (CHAUDRI e ZICARI, 2001), essas linguagens precisam de dados altamente estruturados. Duas características que levam o modelo orientado a objeto ser cada dia mais utilizado na comunidade de desenvolvimento de *software*, envolvem a sua modelagem e sua programação. Seu padrão de modelagem é UML (*Unified Modeling Language*), que facilita a alteração de informações na implantação de acordo com as mudanças necessárias ao longo de um processo de desenvolvimento; alterações essas que não haviam sido previstas anteriormente nos requisitos.

De acordo com (SILBERCHATZ ET AL, 2001), à medida que as bases de dados foram sendo utilizadas em um âmbito maior de aplicações, as limitações impostas pelo modelo relacional emergiram como obstáculos. Como resultado, pesquisadores da área de bancos de dados desenvolveram novos modelos de dados que superassem as restrições do modelo relacional. Nos últimos anos a demanda por maneiras de tratar dados mais complexos tem crescido bastante.

## 2.2 – Conceitos Básicos

O desenvolvimento dos Sistemas de Gerenciamento de Banco de Dados Orientado a Objetos (SGBDOO) teve origem na combinação de ideias dos modelos de dados tradicionais e de linguagens de programação orientada a objetos. No SGBDOO, a noção de objeto é usada no nível lógico e possui características não encontradas nas linguagens de programação tradicional, como operadora de manipulação de estruturas, gerenciamento de armazenamento, tratamento de integridade e persistência dos dados (RATIONAL INC, 2004).

Os modelos de dados orientados a objetos têm um papel importante nos SGBDs já que são mais adequados para o tratamento de objetos complexos (textos, gráficos, imagens) e dinâmicos (programa, simulações). O casamento entre as linguagens de programação e banco de dados é um dos problemas que estão sendo tratados de forma mais adequada no contexto de orientação a objeto.

### **2.2.1 –Abstração**

Por abstração entende-se que é a consideração apenas das propriedades comuns de um conjunto de objetos, omitindo os detalhes, utilizada com frequência na definição de valores similares e na formação de um tipo a partir de outro, em diferentes níveis de abstração. O uso de abstrações permite a geração de tipos baseada em hierarquias de tipos e de relacionamentos. Os principais conceitos de abstração utilizados em banco de dados são generalização e agregação.

### **2.2.2 - Objetos**

Os objetos são abstrações de dados do mundo real, com uma interface de nomes de operações e um estado local que permanece oculto. As abstrações da representação e das operações são ambas suportadas no modelo de dados orientado a objetos, ou seja, são incorporadas as noções de estruturas de dados e de comportamento. Um objeto tem um estado interno descrito por atributos que podem apenas ser acessados ou modificados através de operações definidas pelo criador do objeto. Um objeto individual é chamado de instância ou ocorrência de objeto. A parte estrutural de um objeto (em banco de dados) é similar à noção de entidade no modelo Entidade-Relacionamento.

Na modelagem orientada a objeto, cada objeto possui um indicador em comum denominado *OID* (*ObjectIdentifier*). Entretanto, o *status* do objeto pode variar de acordo com suas propriedades e seus valores que são especificados pelas operações que podem ser executadas pelo próprio objeto. Essas propriedades podem ser atributos e/ou relações entre o próprio objeto ou outros objetos que fazem parte de outras classes, porém do mesmo sistema (RUMBAUGH ET AL., 1994).

Levando em consideração que os objetos possuem várias propriedades, e que essas propriedades variam entre si, pode se destacar a mais importante delas, que é aquela responsável pela identificação dos objetos. Sem essa propriedade não é possível a comunicação para obter e atualizar os dados. Essa identificação é definida no momento de sua criação e não muda durante a vida útil do objeto.

Um ponto importante a ser destacado é a grande semelhança entre o conceito de banco orientado á objeto e o banco relacional, todavia essa semelhança se limita no campo da identificação, divergindo em alguns outros pontos. No caso dos bancos relacionais, o seu identificador se dá através das chaves primárias, identificando uma tupla, onde essa tupla é única e pode ter seus dados alterados por uma mudança de relacionamento.

No caso da orientação á objetos, os objetos que compõe o banco de dados são representados por meio de métodos que, por sua vez, criam uma assinatura única no objeto e na implementação. Isso significa que os objetos no banco de dados também podem possuir métodos. No caso dos bancos relacionais isso não acontece. Quando se usa linguagens orientadas a objeto e bancos relacionais, o que se tem é uma perda grande de tempo fazendo o mapeamento ou mesmo a tradução dos dados orientados da aplicação para os dados relacionais do banco de dados, causando assim um fenômeno conhecido como incompatibilidade de impedância. No banco orientado a objeto, as informações e as operações fazem parte de uma única entidade que é o objeto. Com isso, obtém se uma unificação de objetos, que são modelados conjuntamente e armazenados no mesmo sistema gerenciador (RUMBAUGH ET AL., 2004).

### **2.2.3- Classes**

As classes são aglutinações de objetos de um mesmo tipo que possuem operações e propriedades em comum, ou seja, um conjunto de objetos que possui o mesmo tipo (atributos, relacionamentos, operações) pode ser agrupado para formar uma classe. A noção de classe é associada ao tempo de

execução, podendo ser vista como uma representação por extensão, enquanto que o tipo é uma representação intencional. Cada classe tem um tipo associado, o qual especifica a estrutura e o comportamento de seus objetos. Assim, a extensão da classe denota o conjunto dos objetos atualmente existentes na classe e o tipo provê a estrutura destes objetos

#### **2.2.4 - Herança**

A herança é o processo ao qual permite que uma classe seja o molde para outras classes que precisam ser criadas. Isto significa que este mecanismo permite ao usuário definir tipos de forma incremental, por refinamento de outros já existentes, permitindo composição de tipos em que as propriedades de um ou mais tipos são reutilizadas na definição de um novo tipo.

As principais vantagens de herança são prover uma maior expressividade na modelagem dos dados, facilitarem a reusabilidade de objetos e definir classes por refinamento, podendo fatorar especificações e implementações como na adaptação de métodos gerais para casos particulares, redefinindo-os para estes, e simplificando a evolução e a reusabilidade de esquemas de banco de dados (RATIONAL INC, 2004).

#### **2.2.5 - Polimorfismo**

A idéia de polimorfismo nos bancos orientado a objetos é que ela permite que um método seja implementado em princípio por duas classes, desde que, ao ser instanciado sejam métodos de uma mesma classe e tenha a mesma assinatura. Segundo (RUMBAUGH ET AL., 1994), um mesmo nome pode ser usado por mais de uma operação definida sobre diferentes objetos, o que caracteriza uma sobrecarga (*overloading*). A redefinição do operador para cada um dos tipos de objetos definidos caracteriza uma sobreposição (*overriding*). As operações são ligadas aos programas em tempo de execução caracterizando o acoplamento tardio (*late binding*).

### 2.2.6 - Encapsulamento

O encapsulamento possibilita a distinção entre a especificação e a implementação das operações de um objeto, além de prover a modularidade que permite uma melhor estruturação das aplicações ditas complexas, bem como a segurança dentro do sistema, ou seja, ele tem sua ligação à abstração dos dados, e é ele quem define quais informações que o objeto externo possa visualizar, de acordo com o contexto da aplicação. Isso torna alguns atributos invisíveis para outras classes a qual se tem uma herança.

Em banco de dados se diz que um objeto está encapsulado quando o estado é oculto ao usuário e o objeto pode ser consultado e modificado exclusivamente por meio das operações a ele associadas

### 2.3 – Bancos de dados orientados a objetos no mercado

Existem vários bancos de dados orientados a objeto, discutir cada um deles é essencial para a tomada de decisão. É importante saber qual modelo é mais apropriado para o uso em sua aplicação. A seguir estão alguns exemplos:

- **CACHÉ:** trabalha com as linguagens Java, .Net, C++, XML e outras. É um banco de dados comercial;
- **DB4Objects:** trabalha com as linguagens Java e .Net. Sua linguagem de consulta é a OQL (*Object Query Language*) e é um banco de dados distribuído em duas licenças, a GPL (licença pública Geral) e uma licença comercial;
- **JASMINE:** possui alta conectividade com *web*, suporte à linguagem Java. Ainda pode se desenvolver aplicações em Visual Basic usando Active/X, em HTML (*HyperTextMarkupLanguage*) usando as ferramentas de conectividade para *web* disponíveis no Jasmine, em C e C++ usando APIs

e em Java usando interfaces de *middleware* embutidas no Jasmine. É um banco de dados comercial;

- **Objectivity/DB:** trabalha com as linguagens: C#, C++; Java; Python, Smalltalk; SQL++ (SQL com objeto - extensões orientadas) e XML (para a importação e a exportação somente). É um banco de dados comercial;
- **Ozone:** trabalha com as linguagens Java e XML. É um banco de dados *opensource*;
- **ORACLE 10G:** combina um banco de dados relacional com conceitos de orientação a objetos. Possui o *ObjectType*, que equivale a uma classe e captura tanto a estrutura, quanto o comportamento de um objeto.

#### 2.4 - Banco de dados orientado a objetos x Banco de dados relacional

O banco de dados orientado a objetos e os bancos de dados relacionais não são totalmente compatíveis. Eles representam duas visões distintas do mundo em um RDBMS (*RelationalDatabase Management System*), onde tudo que se pode ver são dados. Em um sistema orientado a objetos tudo o que se pode ver são comportamentos. Não que uma perspectiva seja melhor do que a outra: o modelo orientado a objetos tende a funcionar melhor em sistemas com comportamento complexo e específico de estados, nos quais os dados são secundários. Esse modelo também é eficaz em sistemas cujos dados são acessados, através de navegação, em uma hierarquia natural (por exemplo, listas de materiais). O modelo RDBMS é adequado para relatar aplicativos e sistemas cujos relacionamentos sejam dinâmicos ou *ad-hoc* (BERTINI, 2004)

A verdade é que um volume grande de informações é armazenado em bancos de dados relacionais e, se os aplicativos orientados a objetos quiserem ter acesso a esses dados, precisam ser capazes de ler e gravar em um RDBMS. Além disso, são muito comuns os sistemas orientados a objetos precisarem compartilhar dados com sistemas que não são orientados a objetos. É natural, portanto, usar um RDBMS como mecanismo de compartilhamento.

Embora o *design* orientado a objetos e o *design* relacional tenham algumas características em comum (os atributos de objeto são teoricamente semelhantes às colunas de entidades), há diferenças fundamentais que fazem da integração completa um verdadeiro desafio. A diferença básica é que os modelos de dados expõem os dados (através de valores de coluna), enquanto os modelos de objeto ocultam os dados (encapsulados atrás de suas interfaces públicas).

O banco de dados relacionais e o banco de dados orientados a objetos possuem características distintas, mas basicamente servem ao mesmo propósito: persistir dados necessários para a manutenção do negócio para o qual são aplicados, possibilitando a recuperação, comparação e tratamento desses dados a fim de produzir resultados tangíveis.

Em banco de dados relacional, uma coleção de tabelas, todas com nomes únicos, compõem a base de dados, podendo estar relacionada a uma ou mais tabelas. Conceitos como integridade referencial de dados – que garantem que um dado referenciado em uma tabela esteja presente na tabela que está sendo referenciada – e chaves primárias estão presentes e garantem que um conjunto de informações possa ser representado de maneira consistente, independente da forma de acesso (SILBESCHATZ ET AL., 2001).

Já um banco de dados orientado a objetos possui três pilares principais: herança, polimorfismo e encapsulamento. Este modelo apresenta maior flexibilidade na manipulação de seu conteúdo e por meio de identificadores de objetos manipula os dados de forma consistente.

Apesar do conceito de bancos de dados orientados a objetos ser bastante distinto do modelo relacional, o mesmo resulta da integração entre a orientação a objetos e a tecnologia de banco de dados tradicionais. Enquanto na programação orientada a objetos, os objetos existem apenas enquanto o programa que os criou está em execução, os bancos de dados orientados a objetos podem criar objetos que sejam persistentes e compartilhados entre diferentes aplicativos.

A tabela 2 mostra as principais características do modelo relacional e do modelo orientado a objeto.

<b>Modelo Relacional</b>	<b>Modelo Orientado a Objeto</b>
<b>Tabelas</b>	<b>Objetos, Classes</b>
<b>Linhas</b>	<b>Tuplas</b>
<b>Querys</b>	<b>Querys Nativas</b>
<b>SQL</b>	<b>Métodos</b>

**Tabela 2 – Principais Características dos Modelo Relacional e Orientado a Objeto (ENDEL A, 2011)**

## 2.5 - Banco Objeto Relacional

O banco de dados objeto-relacional combina os benefícios do modelo relacional com a capacidade de modelagem da orientação a objetos, além de fornecer suporte para consultas complexas sobre dados complexos. As suas estruturassão compostas por tabelas, sendo que apresenta muito mais recursos do que as tabelas relacionais. Entre esses recursos estão à herança, polimorfismo, métodos, objetos e ainda funções.

### 2.5.1 - Objeto Relacional Oracle

A Oracle implementou seu próprio modelo de banco de dados objeto relacional, implantando assim diversos conceitos que definem um modelo objeto-relacional. Neste modelo, os conceitos usados para a implementação são tipo de objeto e tabela de objetos.

### 2.5.1.1 - Tipo de Objeto

O modelo criado pela Oracle usa o conceito de objetos aninhados, ou *objecttypes*. Com isso, são criados tipos de dados adicionais, e depois são realizadas referências a esses tipos de objetos, até mesmo dentro de outros objetos. Sendo assim é possível simplificar a complexidade do tratamento com tipos de dados complexos. A figura 1 mostra a criação de um tipo de dados.

```
create type t_cidadeas object (  
    cepvarchar2(9),  
    nome varchar2(100)  
    uf    varchar2(2))
```

**Figura 1 - Criação de Tipo de Objeto**

Este tipo pode ser usado de formas diferentes e em todos os lugares na aplicação. E pode se implementar um tipo dentro de outro tipo. No caso, um novo tipo se faz referência ao tipo *t\_cidade*.

### 2.5.1.2 - Tabela de Objetos

A tabela de objetos ou *objecttable*, é uma classe onde as linhas são objetos. São tabelas estruturadas e definidas por um tipo de dado. Desta forma as linhas das tabelas tornam-se instância de objetos e podem ser tratadas como “objeto-linha”. Considerando o tipo de objetos *t\_cidades* criado na figura 1, será criada uma tabela de objetos do tipo *t\_cidades*. A figura 2 mostra a criação desta tabela de objeto.

```
Create table cidade of t_cidade
```

**Figura 2 - Criação de Tabela de Objetos**

Com relação a identificação dos objetos no modelo objeto-relacional, ela depende de qual o tipo de tabela a qual eles pertence. Em uma tabela de objetos, cada objeto tem sua identificação de acordo com sua estrutura, e que são somadas a um todo, ao contrário das tabelas tradicionais, que por sua vez não possuem OID e estruturam-se apenas pelos seus valores de atributos, formando um “multi-conjunto” de linhas. No entanto, observa-se que ambos são paralelos e coexistem.

### **3 - Business Intelligence**

Na economia globalizada a informação adquire um papel muito importante, já que em geral, o mundo ficou mais competitivo, as empresas ficaram mais ágeis e os clientes mais exigentes. Utilizar a informação dos clientes ou de processos internos gera vantagens competitivas no atual mercado globalizado. Antecipar possíveis necessidades dos clientes permite oferecer algum tipo de serviço ou produto individualizado, estreitando assim os relacionamentos e criando possíveis vínculos de fidelidade (MONTEIRO ET AL., 2011).

De acordo com (THONSEM, 2002), o *Business Intelligence* (BI) pode ser definido como um conjunto integrado de conceitos e metodologias, que permite o processo de análise de dados acumulados em uma empresa, a fim de extrair com certa inteligência ou conhecimento, apoiando a tomada de decisão com a informação correta e dentro de um prazo adequado.

Em geral, os sistemas de BI oferecem possibilidades de:

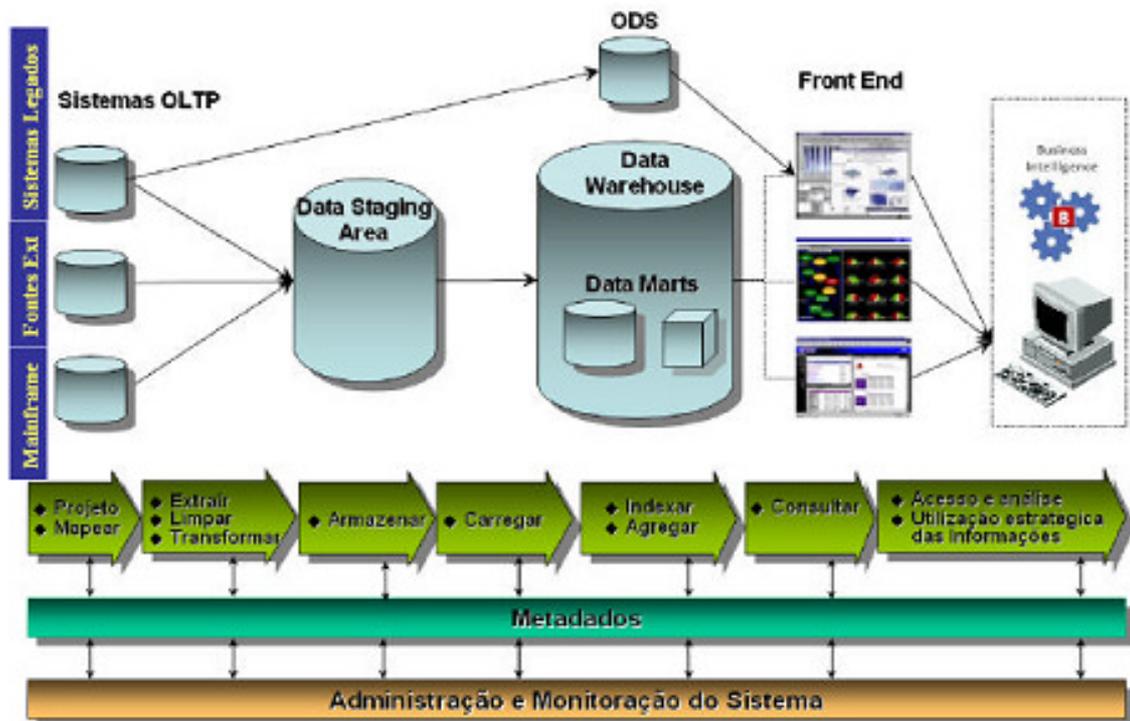
- Extrair e integrar dados de múltiplas fontes;
- Fazer uso de experiências prévias;
- Analisar dados contextualizados;
- Trabalhar com hipóteses;
- Procurar relações de causa e efeito;
- Transformar os registros obtidos em informação útil para o conhecimento empresarial.

Segundo (MONTEIRO ET AL, 2011), no *Business Intelligence* existem dois tipos de aplicações para o negócio:

- **Aplicações que sustentam o negócio:** são as aplicações críticas ou *On-Line Transaction Processing* (OLTP), que trazem retorno de investimento baseado no melhor desempenho do negócio;
- **Aplicações que analisam o negócio:** são as aplicações de apoio à decisão ou *Decision Support System* (DSS). Nelas, o retorno do

investimento é baseado no melhor conhecimento do negócio e no desenvolvimento de estratégias refinadas.

A figura 3 mostra os conceitos e tecnologias que compõem a arquitetura BI.



**Figura 3 - Arquitetura para Business Intelligence (THONSEM, 2002)**

O poder de soluções de *Business Intelligence* pode proporcionar a organização uma substancial vantagem competitiva em relação a seus competidores. Um pequeno investimento numa solução adequada de BI pode produzir retornos significativos (KIMBALL, 1998).

### 3.1 - Sistemas OLTP

Os processamentos de transação on-line são sistemas operacionais de registro que capturam as transações diárias da empresa. Eles são responsáveis pela persistência dos dados que os aplicativos da empresa manipulam. É destes sistemas que os dados serão retirados a fim de serem transformados em informação para a empresa (SERRA, 2002).

Em geral, os sistemas de origem são projetados com o propósito de estarem disponíveis integralmente na maior parte do tempo e serem muito rápidos. Para um sistema on-line de vendas, caso a empresa resolvesse começar a executar consultas imensas e demoradas, o sistema ficaria muito lento e provavelmente, não conseguiria registrar a venda por estar ocupado processando a consulta.

### 3.2 – ETL (Extract, Transform, Loader)

O ETL é uma fase extremamente crítica de um *data warehouse*, pois envolve a movimentação dos dados de origem nos sistemas transacionais e/ou sistemas legados, obedecendo as regras de negócio. O processo de Extração, Transformação e Carga envolve:

- A extração de dados de fontes externas;
- A transformação dos mesmos para atender as necessidades de negócio;
- A carga dos mesmos do *data warehouse*.

O ETL é importante, pois é a forma pela qual os dados são efetivamente carregados no *data warehouse*. Os processos de ETL podem ser bastante complexos e problemas operacionais significativos podem ocorrer com sistemas de ETL desenvolvidos inapropriadamente.

### 3.3 -Data StagingArea

A *Data StagingArea*(Área de Representação de Dados) é a área de trabalho do *data warehouse*. É o lugar onde se colocam os dados primários, onde se limpa, combina, arquiva e, ao final, exportam esses dados para um ou mais *data marts*. O propósito da área de *data staging* é preparar os dados ser

carregadoem um servidor de apresentação, como um SGBD relacional ou *software* OLAP (SERRA, 2002).

A *Data StagingArea* realiza a ligação entre os sistemas OLTP e o *data warehouse*. No *data warehouse* os dados operacionais brutos são transformados em um formato de *warehouse* prontos para serem consultados e utilizados pelo usuário. Os dados chegam na *Data StagingArea*, onde passam por transformações, como filtragem dos dados (correções de erros de digitação, solução de conflitos de domínio, tratamento de elementos ausentes ou a divisão em formatos padrão), combinação de dados de várias origens, cancelamento de dados duplicados e atribuição de chaves de *warehouse* (KIMBALL e CASERTA, 2004). Essas transformações são anteriores ao carregamento dos dados nos *data marts*. A *Data StagingArea* é dominada pelas atividades simples de classificação e processamento seqüencial. Em muitos casos, ela não se baseia na tecnologia relacional, podendo ser formada por um sistema de arquivos simples.

### 3.4 - Data Warehouse

O *data warehouse* é um banco de dados contendo dados extraídos do ambiente de produção da empresa, que foram selecionados e depurados, tendo sido otimizados para processamento de consulta e não para processamento de transações. Em geral, um *data warehouse* requer a consolidação de outros recursos de dados além dos armazenados em banco de dados relacionais, incluindo informações provenientes de planilhas eletrônicas, documentos textuais e outros. A função do *data warehouse* é tornar as informações corporativas acessíveis para o seu entendimento, gerenciamento e uso (MEYER, 1998).

Como o *data warehouse* está separado dos bancos de dados operacionais, as consultas dos usuários não impactam nestes sistemas, que ficam resguardados de alterações indevidas ou perdas de dados.

A força do *data warehouse* dentro das empresas está na organização e na entrega de dados que apoiem o processo de tomada de decisão. O *data*

*warehouse* é a implementação física de um modelo de apoio a decisões, que disponibiliza a empresa as informações necessárias à tomada de decisões estratégicas, e as principais características que segundo (INMON, 1995)

De acordo com (INMON, 1995), um *data warehouse* é um repositório de dados com as seguintes características:

- **baseado em assuntos:** refere-se ao fato de que o *data warehouse* está organizado de maneira que descreve o desempenho dos negócios; os bancos de dados operacionais são orientados para o processo dos negócios;
- **integrados:** refere-se ao fato de os dados serem organizados para fornecer uma fonte única;
- **variáveis em relação ao tempo:** reconhecem que o desempenho de negócio é medido em pontos cronológicos (final do mês, por exemplo) e comparado em relação ao tempo;
- **não-voláteis:** sugere que os dados, uma vez inseridos no *data warehouse*, não devem mudar; o conteúdo dos bancos de dados operacionais muda cada vez que uma transação é processada.

Um *data warehouse* suporta análise de negócios e tomada de decisões através da integração de dados procedentes de diversos sistemas em uma base de dados consolidada, onde muitas vezes incompatíveis entre si. Esta transformação de dados em informações significativas capacita os executivos de uma organização a efetuar análises muito mais reais, precisas e consistentes (MEYER, 1998).

Em outras palavras, o *data warehouse* é uma base de dados projetada especificamente para atender às necessidades de tomada de decisão, em vez de atender aos sistemas de processamento de transações (DAPHNE, 1999). Ele também pode ser utilizado como suporte a sistemas de *data mining*. *Data mining* é um processo não trivial de procura por padrões válidos, novos, potencialmente úteis e inteligíveis a partir de uma coleção de dados (FREITAS, 1998).

Os projetos lógico e físico são considerados a base do *data warehouse*. A abordagem de modelagem dimensional, utilizada para o desenvolvimento do projeto lógico, é a melhor maneira de modelar dados de suporte à decisão e oferece os melhores resultados em termos de facilidade de uso e alto desempenho (KIMBALL ET AL., 1998).

### **3.4.1 – Planejamento da Construção de Um *Data Warehouse***

Uma das maiores preocupações em *data warehouse* é como planejar a sua construção. Ele pode ser construído inteiro, abrangendo toda a organização de uma só vez (abordagem monolítica), ou então por áreas (KIMBALL ET AL., 1998).

A maioria dos técnicos em *data warehouse* utiliza alguma espécie de abordagem por etapas para construir um *data warehouse* de uma empresa inteira. Em cada etapa é construído um *data mart*, que é um *data warehouse* direcionado a apenas uma área ou assunto específico da organização. Os planejadores de *data warehouse* utilizam a estratégia de construí-lo em pedaços pequenos (*data mart*) por vez, até chegar à sua totalidade. O grande problema apresentado por essa construção incremental é o risco de esses *datamarts* apresentarem incompatibilidades de integração para formar o todo do *data warehouse* (BRUZAROSCO; CASTOLDI e PACHECO, 2000).

A solução para essa abordagem evolutiva é iniciar o planejamento do *data warehouse* com uma fase de especificação de uma arquitetura de dados geral com metas específicas e finitas, a qual guiará a implementação dos *data marts* em separado, permitindo uma aderência entre eles. O projeto lógico de um *data warehouse* pode ser realizado tanto através da modelagem entidade-relacionamento quanto pela modelagem dimensional (KIMBALL ET AL., 1998).

#### **3.4.1.1 - Modelagem entidade-relacionamento**

A modelagem entidade-relacionamento é uma técnica de projeto lógico de banco de dados largamente utilizada, que procura eliminar redundância e inconsistência dos dados, além de facilitar a sua manutenção. Como resultado, gera muitas tabelas, e isso tem um preço, pois os sistemas de software tornam-se complexos e ineficientes na manipulação e recuperação dos dados, porque requerem uma cuidadosa atenção no processamento de algoritmos para ligar esses conjuntos de tabelas (KIMBALL ET AL., 1998).

Mesmo em aplicações simples o Modelo ER desenvolve muitas tabelas, que são ligadas por uma intrincada rede de relacionamentos. Dependendo do porte da empresa, a modelagem de seus dados resulta em centenas ou milhares de entidades, causando uma série de problemas, tais como: usuários finais que não entendem ou não se lembram do diagrama de entidade-relacionamento (DER); a interface gráfica de um diagrama geral não é amigável para os usuários finais; *software* de consulta geral sobre o modelo paga um alto preço de desempenho; e a sua aplicação para *data warehouse* contraria a finalidade deste, de ser intuitivo e oferecer um alto desempenho na recuperação de dados ((BRUZAROSCO; CASTOLDI e PACHECO, 2000).).

#### **3.4.1.2 - Modelagem dimensional**

A modelagem dimensional é a técnica de projeto lógico de banco de dados mais usado no desenvolvimento de data warehouse, embora também possa ser aplicada ao projeto de sistemas de informações operacionais. Na verdade, ela busca apresentar os dados em um formato que seja intuitivo e ao mesmo tempo atenda a acessos com um alto desempenho (KIMBALL ET AL., 1998).

Um modelo dimensional é composto por uma tabela com uma chave composta, denominada tabela de fatos, e um conjunto de tabelas menores conhecidas como tabelas de dimensão, que possuem chaves simples (formadas por uma única coluna). Na verdade, a chave da tabela de fatos é uma combinação das chaves das tabelas de dimensão. Isto faz com que a representação gráfica do modelo dimensional assemelhe-se a uma estrela. Por este motivo, o modelo também é conhecido como modelo estrela (KIMBALL ET AL., 1998).

A ideia fundamental da modelagem dimensional baseia-se no fato de que quase todo tipo de dado do negócio pode ser representado como uma espécie de cubo de dados, onde as células do cubo contêm os valores medidos e os lados do cubo definem as dimensões naturais dos dados (HARRISON, 1998).

A figura 4 mostra um cubo com três dimensões: produto (conjunto de produtos comercializados pela empresa), mercado (área de atuação) e tempo (período de tempo de atuação). Modelos dimensionais reais do mundo dos negócios geralmente possuem entre 4 e 15 dimensões.

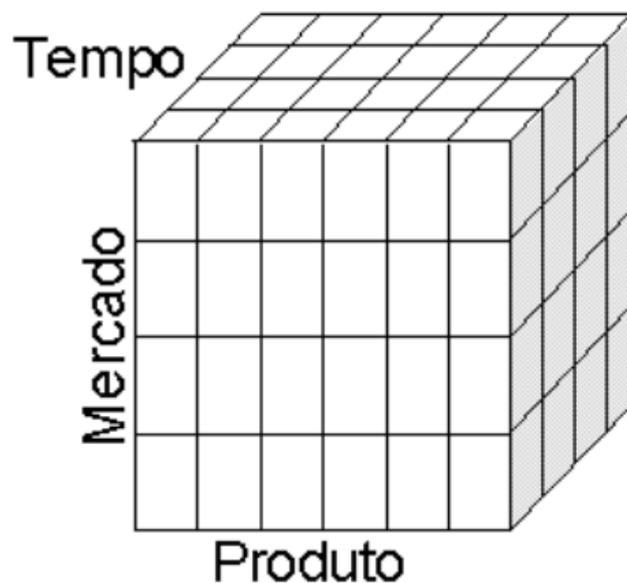


Figura 4 – Cubo de Dimensões (KIMBALL ET AL, 1998)

#### 3.4.1.2.1 - Elementos de Um Modelo Dimensional

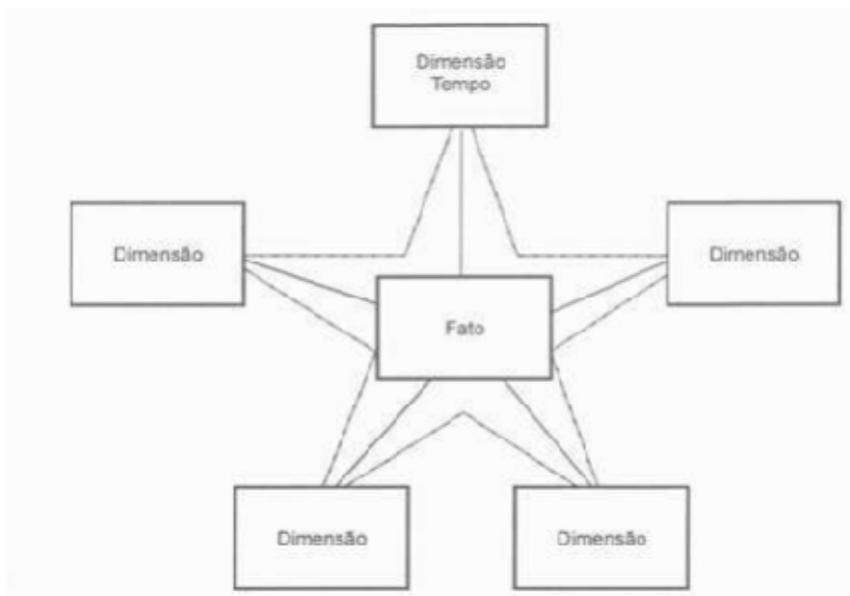
De acordo com (KIMBALL ET AL., 1998), os elementos que compõem o modelo dimensional são:

- **Dimensões:** uma dimensão é uma coleção de atributos textuais que são altamente correlacionados entre si. Os atributos textuais que descrevem coisas são organizados dentro das dimensões. Em uma base de dados de

varejo são comuns dimensões como produto, armazém, cliente, promoção e tempo;

- **Tabela de fatos:** tabela de fatos sintetiza o relacionamento existente entre as diversas dimensões. Isto ocorre porque a chave da tabela de fatos é a associação das chaves primárias das tabelas de dimensões. Geralmente este tipo de tabela possui um ou mais fatos numéricos que ocorrem na combinação de chaves que define cada registro.
- **Atributos:** geralmente são campos de textos, os quais descrevem uma característica de uma coisa tangível. Os atributos mais comuns são as descrições de produtos. Atributos de dimensões oferecem o conteúdo da maioria das respostas solicitadas pelos usuários. Pode-se dizer que a qualidade do *data warehouse* é medida pela qualidade dos atributos das dimensões. Uma tabela de dimensão contém muitos campos-textos descrevendo os membros de uma dimensão particular. A meta do *data warehouse* é criar atributos de tabelas de dimensão com as seguintes características: eloqüente, descritivo, completo, qualidade garantida, indexado, disponível e documentado.
- **Fatos:** um modelo dimensional faz distinção entre fatos e atributos. Um atributo é usualmente alguma coisa que é conhecida com antecedência. Um fato uma observação do mercado. Muitos fatos no mundo dos negócios são numéricos, embora alguns possam conter texto. Algumas vezes, um valor numérico como “preço padrão” parece ser um atributo da dimensão produto, pois é uma constante conhecida antecipadamente. Mas, em um exame mais cuidadoso, verifica-se que o atributo preço padrão sofre alteração durante o ano, o que leva alterá-lo na fase de projeto para um fato. Uma típica e correta decisão a ser tomada é considerar quase todos os campos numéricos de pontos flutuantes como fatos.

A figura 5 mostra o modelo dimensional em estrela.



**Figura 5 - Modelo Dimensional em Estrela (INMON, 1987)**

Este modelo tem o formato de informação central interligada com cinco pontas, sem interligação entre essas pontas, encontram-se então as tabelas de fato (central) e as tabelas dimensões (pontas). O intuito desta modelagem é representar um cubo de várias dimensões em que cada célula tem seu valor e cada lado representa uma dimensão (MORGAN, 1990).

### **3.4.1 3 – Método Para a Construção de Um *Data Warehouse***

Os métodos adotados para a construção de um *data warehouse* é de suma importância para a compreensão do negocio que envolve a empresa ou instituição que se esta desenvolvendo o trabalho. O processo inicial é a extração de dados das bases de dados transacionais, como os dados de

sistemas ERP, dados locais, externos ou *web*. Esta extração se chama ETL, onde são analisados cuidadosamente os dados e assim traduzidas às necessidades de negócio da empresa. Após a extração os dados estes devem ser transformados para que seja possível a carga dos dados em um *Data Warehouse* ou Data Mart dependendo do método de construção adotado. Existem dois métodos formalmente conhecidos:

- **Top-down:** no qual é realizada a modelagem integral do *data warehouse*, seguida pelas extrações de dados. A principal vantagem é a criação de um modelo único. O revés fica por conta do maior tempo de projeto;
- **Bottom-up:** onde o foco é em uma área por vez, com o crescimento gradual do *data warehouse*. A vantagem é a obtenção de resultados a intervalos mais curtos, garantindo muitas vezes sustentação ao projeto. A desvantagem é a maior dificuldade de se consolidar informações entre as diversas áreas.

#### 3.4.1 4 – Ferramentas de Acessos aos Dados

As ferramentas para acesso aos dados mais utilizados são: OLAP (*On-line Analytical Processing*) e OLTP (*Online Transaction Processing* ou Processamento de Transações em Tempo Real).

- **OLTP:** são sistemas que se encarregam de registrar todas as transações contidas em uma determinada operação organizacional;
- **OLAP:** é a capacidade para manipular e analisar um grande volume de dados sob múltiplas perspectivas.

Enquanto o OLAP trabalha com dados históricos, no sentido de analisar informações, o OLTP opera com dados que movimentam o negócio em tempo real, suportando operações cotidianas de negócio empresariais por meio de

seu processo operacional. O OLTP tem como função alimentar a base de dados que compõe o OLAP, que é multidimensional, já o OLTP é uma ferramenta relacional, orientada para o processo, trabalhando com dados do presente e processando um registro de cada vez, não sendo multidimensional como o OLAP. A finalidade do OLTP é fazer com que uma grande quantidade de pequenas informações não se perca, processando milhares ou milhões de informações por dia, que contém em cada uma delas uma pequena porção de dados. Os usuários de OLTP lidam freqüentemente com um registro de cada vez, o que faz com que a mesma tarefa seja executada muitas vezes, devido ao fato da maioria de seus relatórios serem feitos em uma tabela inteira. Com isso, as pesquisas e consultas são instantâneas, quando muito extensas envolvem múltiplas tabelas chamadas de *Join Queries* e devem ser executadas em segundos ou minutos. A tabela 3 mostra uma comparação entre as duas formas de acesso a dados.

<b>OLTP</b>	<b>OLAP</b>
Orientados a aplicações	Orientados a assuntos
As vezes de grande tamanho	Quase sempre grandes
Dados granulados	Dados constituídos de sumarizações
Dados de poucas fontes	Dados de múltiplas fontes
Suporta consultas e atualizações	Atualizações em modo <i>batch</i>
Dados que mudam constantemente	Dados mais estáveis
Dados atuais	Dados históricos

**Tabela 3 - Comparação nas Formas de Acesso aos Dados**

## 4 - Desenvolvimento do Projeto

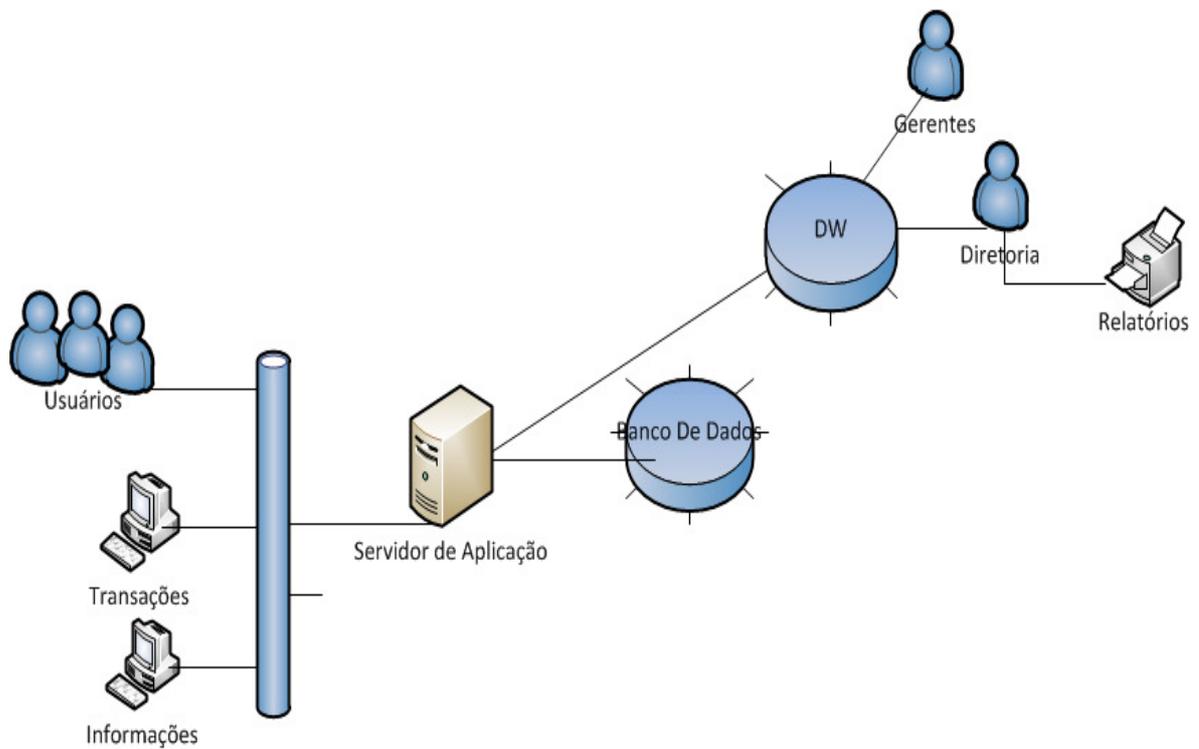
Neste capítulo será apresentada a descrição e modelagem do problema. Será apresentada a especificação e implementação do estudo de caso para vendas. Serão mostradas as técnicas e os métodos que foram criados o *data warehouse*.

### 4.1 – Descrição do Problema

Neste projeto, será desenvolvida um *data warehouse* utilizando os conceitos de *Business Intelligence*. A validação do *data warehouse* será feita através de um estudo de caso para vendas. Para o desenvolvimento do estudo de caso foram utilizados três processos do *Business Intelligence*. O primeiro é o processo de vendas que é onde acontecem as transações operacionais. O segundo é o envio de dados específicos para o *data warehouse*. O terceiro é o uso da *OLTP* para extração dos dados do *data warehouse*, para obter uma análise para o plano estratégico, metas e análise de risco nas tomadas de decisões.

### 4.2 – Modelagem do Problema

Os sistemas de *bussinesintelligence* operam na última camada do sistema trazendo dados específicos. Os dados são gerados através das transações, e operações do sistema no dia a dia, esses dados são armazenados de maneira orientada a objeto em um banco de dados. E após isso, os mesmo dados são armazenado no *data warehouse*, dados estes específicos, escolhido através da regra de negócio. A parte principal deste projeto é implementar o armazenamento de dados no banco objeto-relacional e também alimentar o *data warehouse*. Assim, serão utilizadas duas tecnologias de armazenamento distintas para a mesma finalidade. A figura 6 apresenta uma visão geral do problema a ser abordado.



**Figura 6 - Visão Geral do Problema**

#### 4.3 – Especificação

O modelo adotado foi o de orientação a objeto, no processo de especificação do sistema de *bussinesintelligence*. Essas especificações estão representadas em diagramas, desenvolvido na ferramenta *Open SourceArgo UML*, que inclui ainda suporte para UML 1.4. Foi incluído o diagrama objeto-relacional, diagrama de caso de uso, diagrama de classes, diagrama de sequência.

### 4.3.1 – Diagrama Objeto-Relacional

O diagrama Objeto-Relacional é uma modelagem utilizada para demonstrar uma visão geral das tabelas e ligações entre os relacionamentos do sistema. Como se trata de dois bancos de dados, eles serão divididos em duas partes:

- **Modelo Objeto-Relacional do Banco de Dados:** nele possui a ligação das tabelas do sistema, onde serão inseridos os dados transacionais;
- **Modelo Entidade-Relacionamento do DW:** nele possui a visão geral dos dados que serão inseridas no DW, suas ligações a tabela de fatos e tempo.

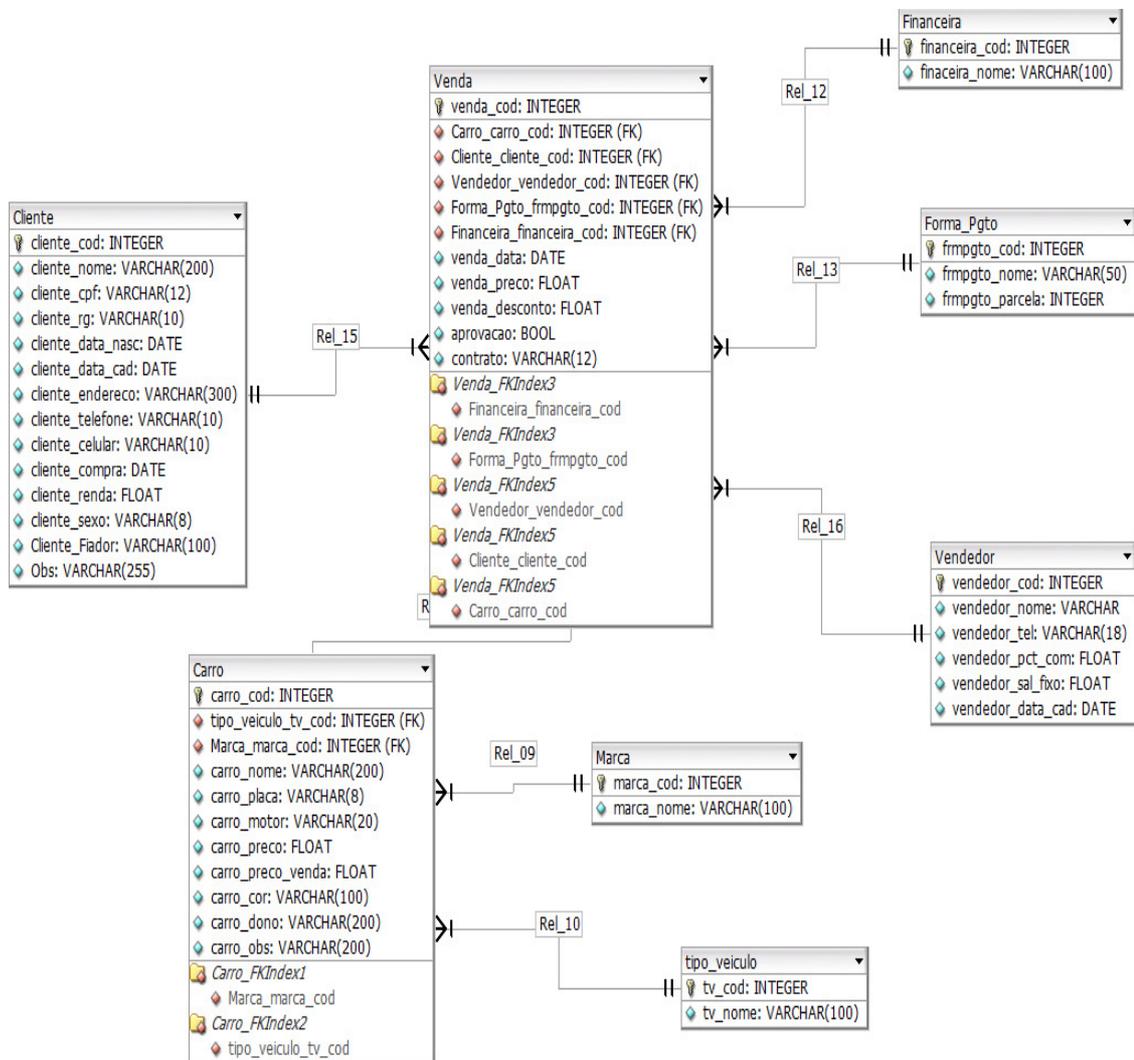
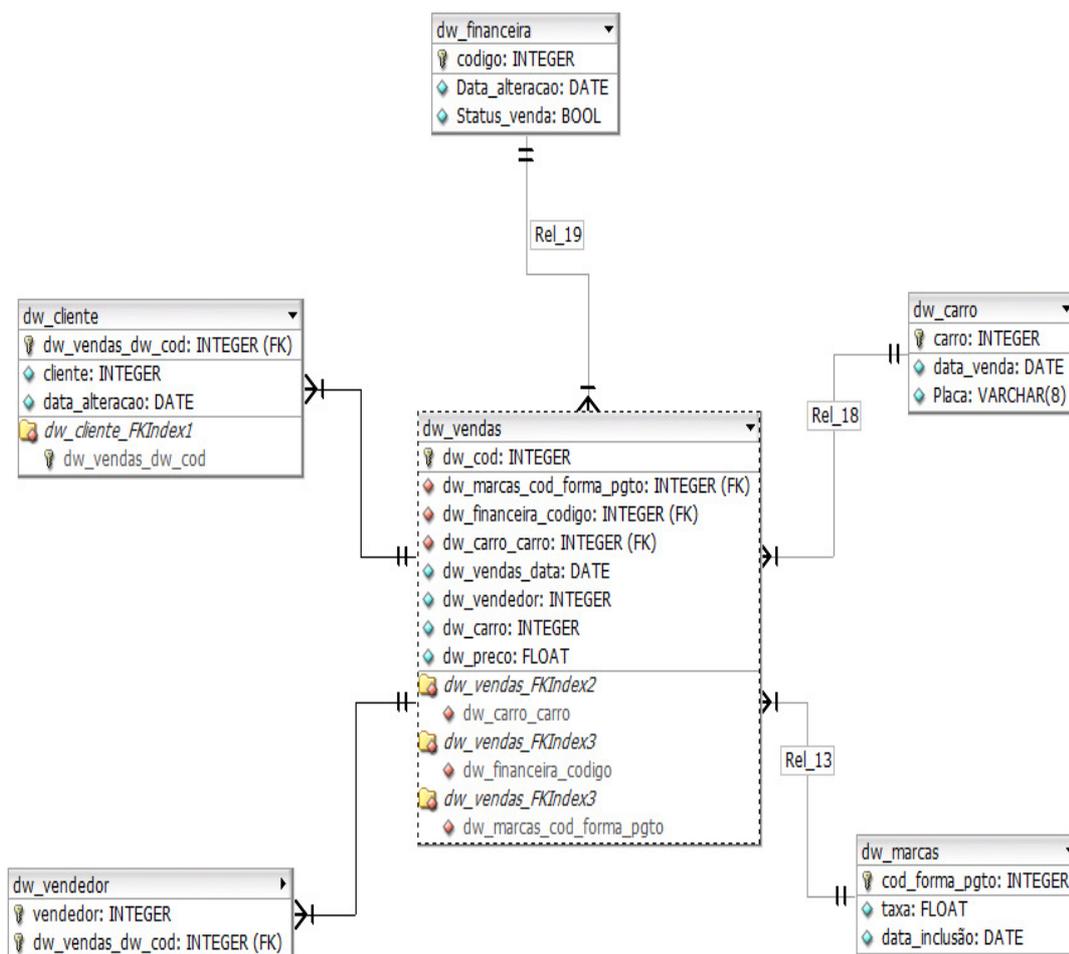


Figura 7 - Diagrama Objeto-Relacional do Banco De Dados

Para a construção do *Data Warehouse* foi utilizada o conceito de modelagem dimensional, cujo modelo adotado foi de estrela. A figura 8 mostra o diagrama entidade- relacionamento do *data warehouse*.



**Figura 8 - Diagrama Entidade-Relacionamento do Data Warehouse**

#### 4.3.2 – Diagrama de Casos de Uso

O diagrama de caso de uso é uma modelagem utilizada para descrever o que um sistema deve fazer suas funcionalidades. A figura 9 mostra o diagrama de caso de uso descrevendo as três funcionalidades.

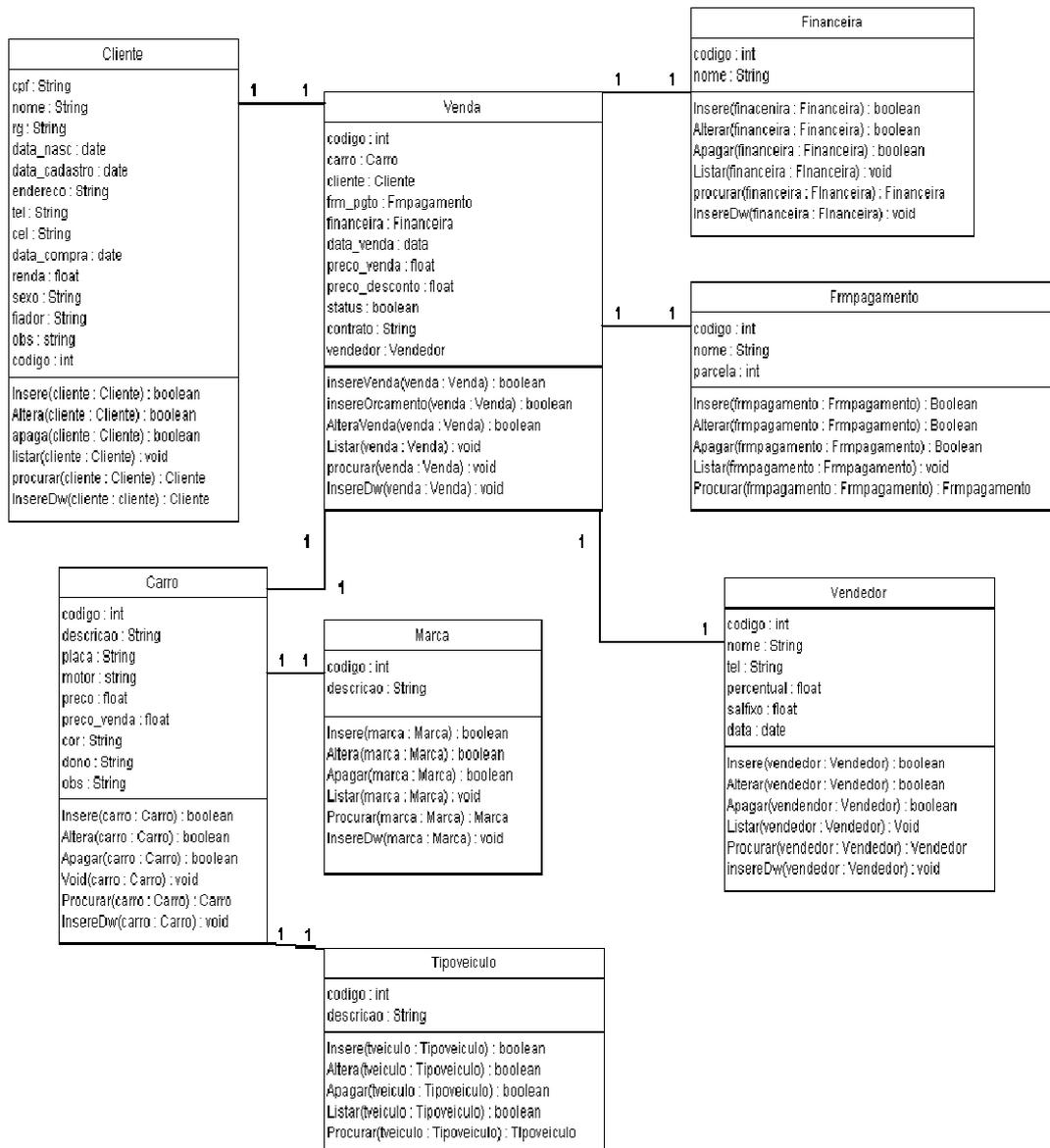


**Figura 9 - Diagrama de Caso de Uso**

- **Manter dados no Banco Orientado a Objeto:** é responsável por incluir, alterar, remover e pesquisar os dados no banco orientado a objeto.
- **Gravar dados específicos no *data warehouse*:** é responsável por gravar dados específicos no *data warehouse*.
- **Extrair dados específicos do *data warehouse*:** é responsável por extrair os dados DW e montar relatórios.

#### 4.3.3 - Diagrama de Classe

O diagrama de classe descreve a estrutura sólida do sistema em termo de classes, relacionamentos e cardinalidade entre elas, onde representam os objetos que são gerenciados na aplicação. A figura 10 mostra o diagrama de classe do estudo de caso de vendas.



**Figura 10 - Diagrama de Classes**

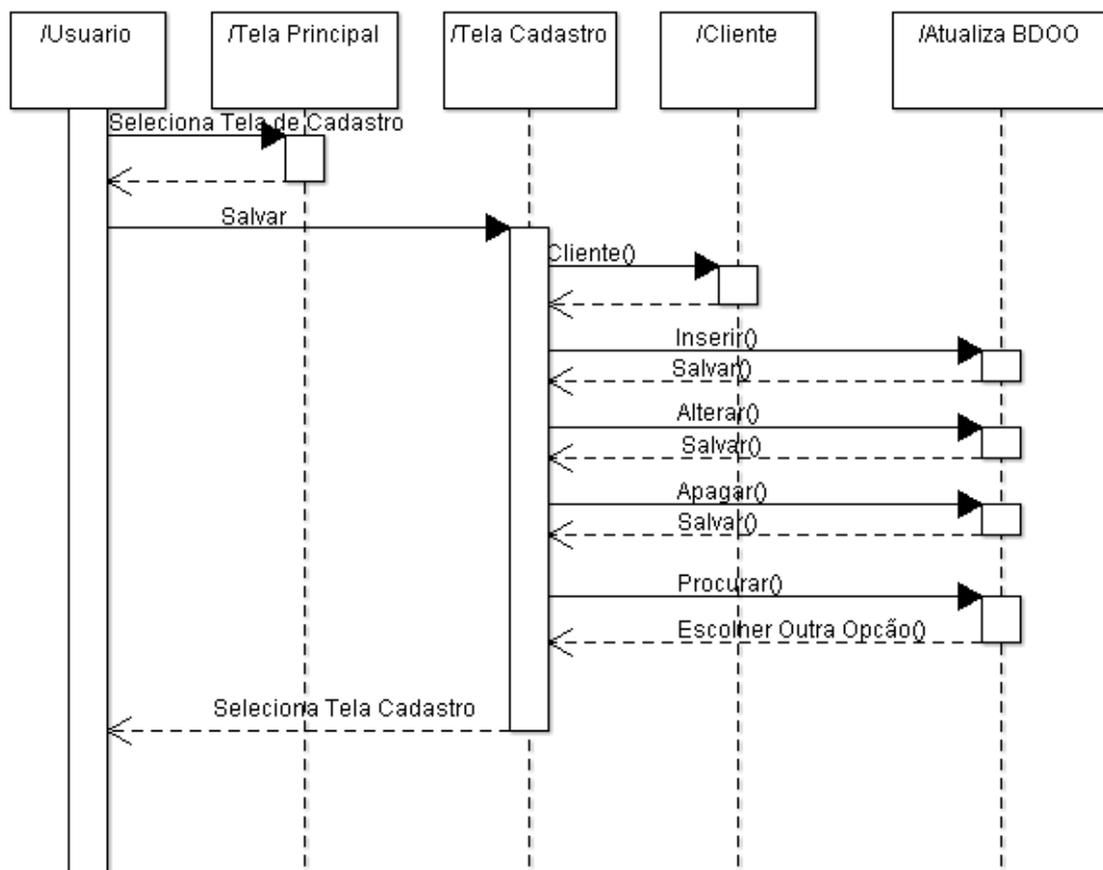
Além das classes definidas no diagrama de classes, ainda existem classes do pacote BD, as quais são responsáveis pela comunicação com o banco de dados: o pacote interface, a qual é responsável pela criação das interfaces de comunicação e interação com o usuário.

#### **4.3.4 – Diagrama de Sequência**

A comunicação de vários objetos é mostrada no diagrama de seqüência. Ele mostra a interação entre objetos, sendo possível perceber a seqüência e a ordem em que as informações são trocadas nos processo. Sendo assim foi elaborado um diagrama de seqüência para cada Caso de Uso especificado:

##### **4.3.4.1 – Manter Cliente**

As ações representadas no diagrama da figura 11 são executadas quando um dos processos de manutenção de registro (inserir, alterar, excluir e pesquisar) clientes for necessário. Assim, é possível compreender o processo de comunicação entre as classes. Foi ocultado neste no diagrama a comunicação existente entre as classes de comunicação com o banco de dados.



**Figura 11 – Diagrama de Sequência Manter Cliente**

#### 4.3.4.2 – Manter Carro

As ações representadas no diagrama da figura 12 são executadas quando um dos processos de manutenção de registro (inserir, alterar, excluir e pesquisar) carros for necessário. Assim, é possível compreender o processo de comunicação entre as classes. Neste diagrama foi ocultado a comunicação existente entre as classes de comunicação com o banco de dados.

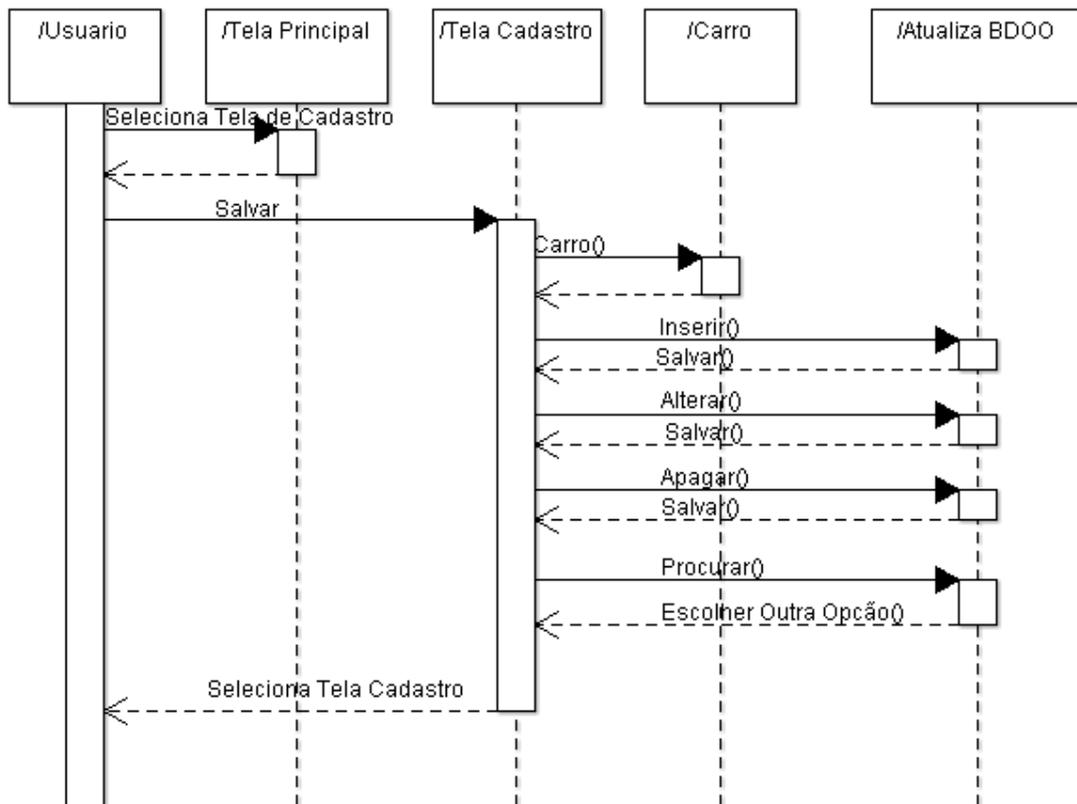


Figura 12 – Diagrama de Sequência Manter Carro

#### 4.3.4.3 – Manter Financeira

As ações representadas no diagrama da figura 13 são executadas quando um dos processos de manutenção de registro (inserir, alterar, excluir e pesquisar) financeira for necessário. Assim, é possível compreender o processo de comunicação entre as classes. Neste diagrama foi ocultada a comunicação existente entre as classes de comunicação com o banco de dados.

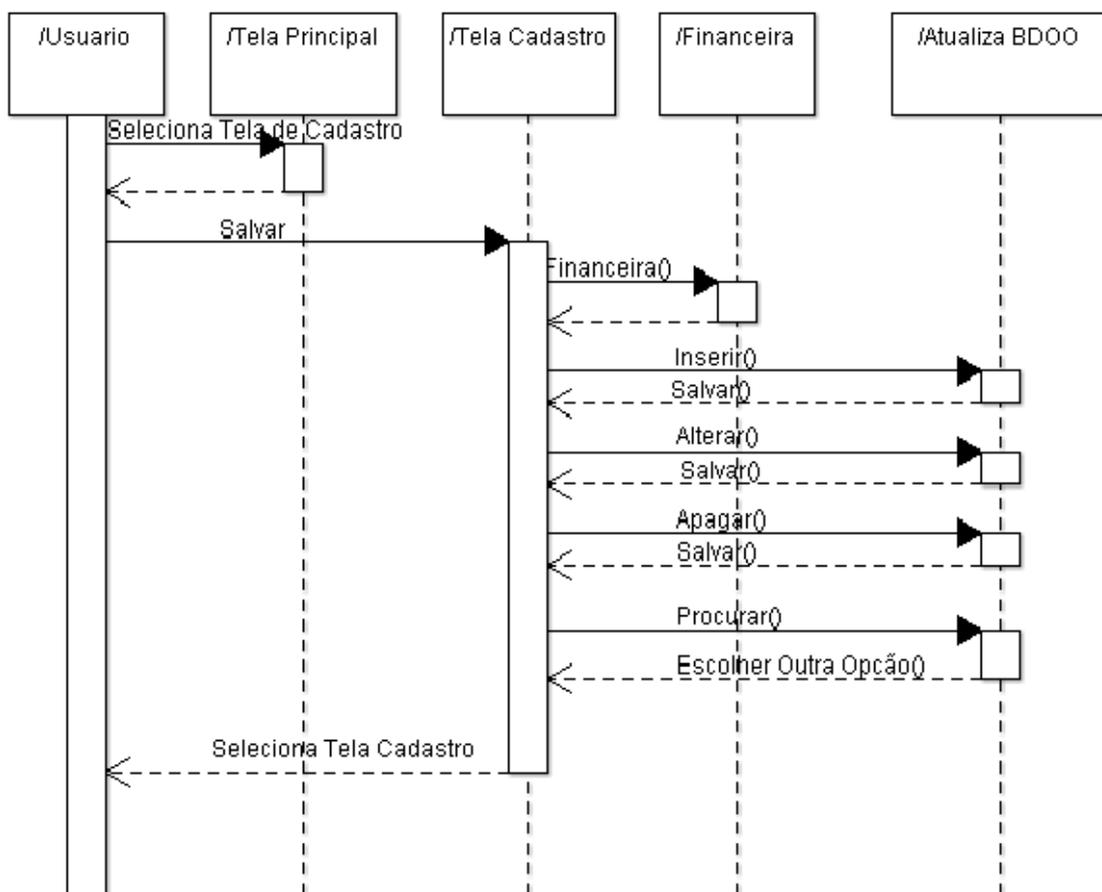


Figura 13 – Diagrama de Sequência Manter Financeira

#### 4.3.4.4 – Manter Forma de Pagamento

As ações representadas no diagrama da figura 14 são executadas quando um dos processos de manutenção de registro (inserir, alterar, excluir e pesquisar) forma de pagamento for necessário. Assim, é possível compreender o processo de comunicação entre as classes. Neste diagrama foi ocultada a comunicação existente entre as classes de comunicação com o banco de dados.

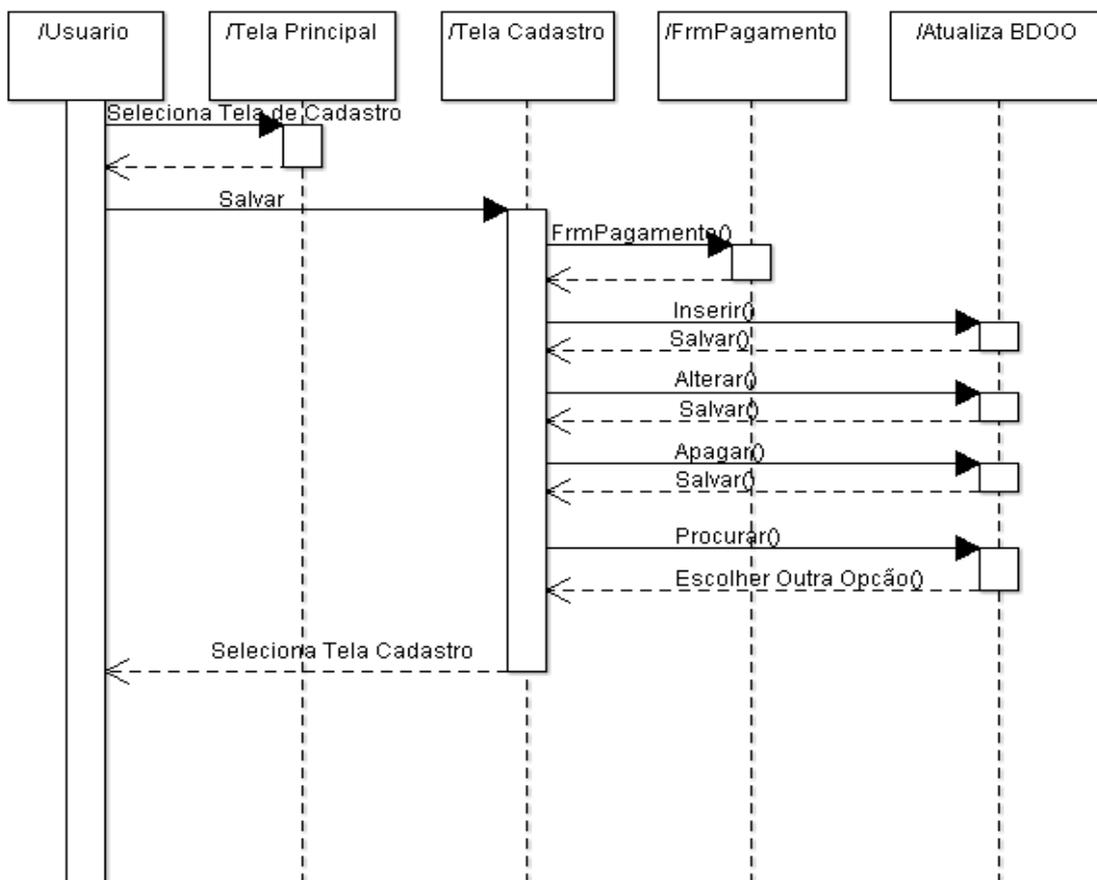


Figura 14 – Diagrama de Sequência Manter Forma de Pagamento

#### 4.3.4.5 – Manter Marca

As ações representadas no diagrama da figura 15 são executadas quando um dos processos de manutenção de registro (inserir, alterar, excluir e pesquisar) marca for necessário. Assim, é possível compreender o processo de comunicação entre as classes. Neste diagrama foi ocultada a comunicação existente entre as classes de comunicação com o banco de dados.

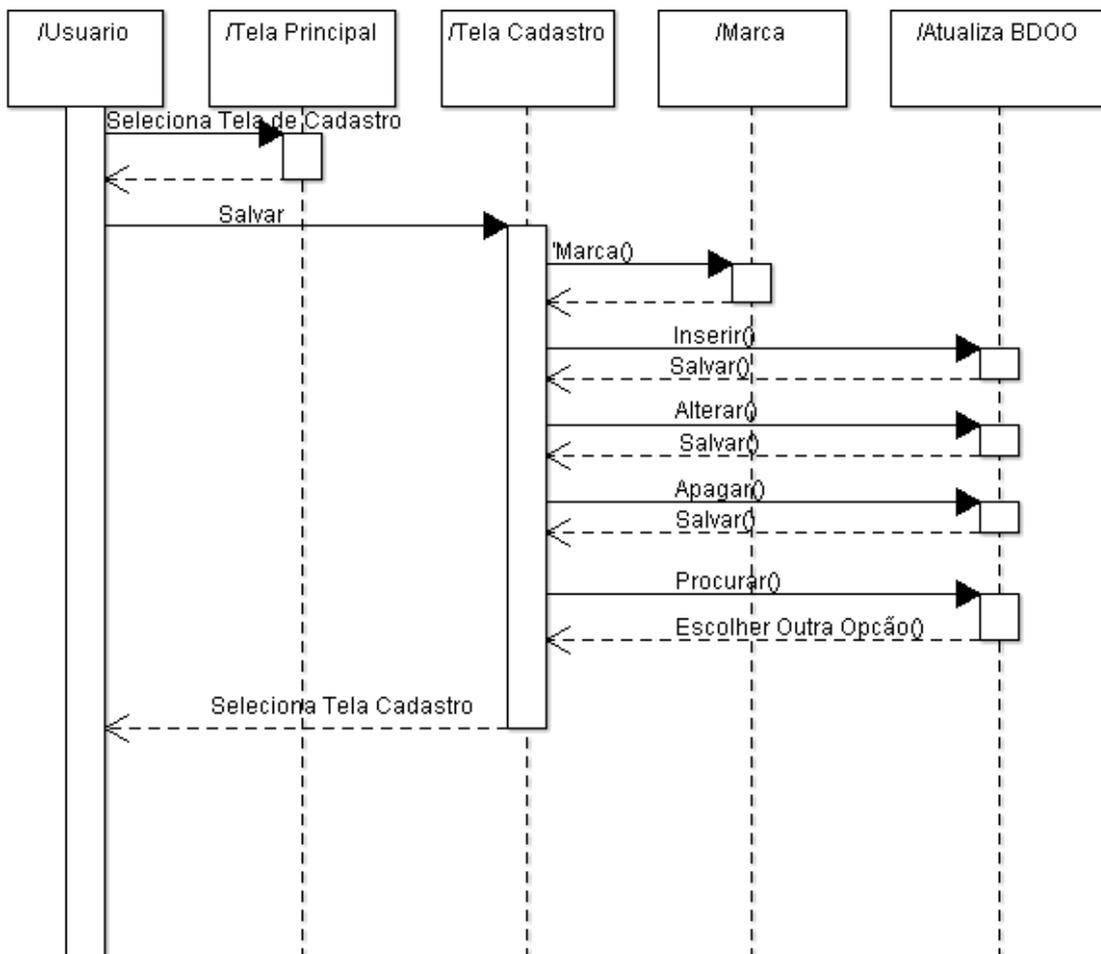


Figura 15 – Diagrama de Sequência Manter Marca

#### 4.3.4.6– Manter Tipo Veículo

As ações representadas no diagrama da figura 16 são executadas quando um dos processos de manutenção de registro (inserir, alterar, excluir e pesquisar) Tipo Veículo for necessário. Assim, é possível compreender o processo de comunicação entre as classes. Neste diagrama foi ocultada a comunicação existente entre as classes de comunicação com o banco de dados.

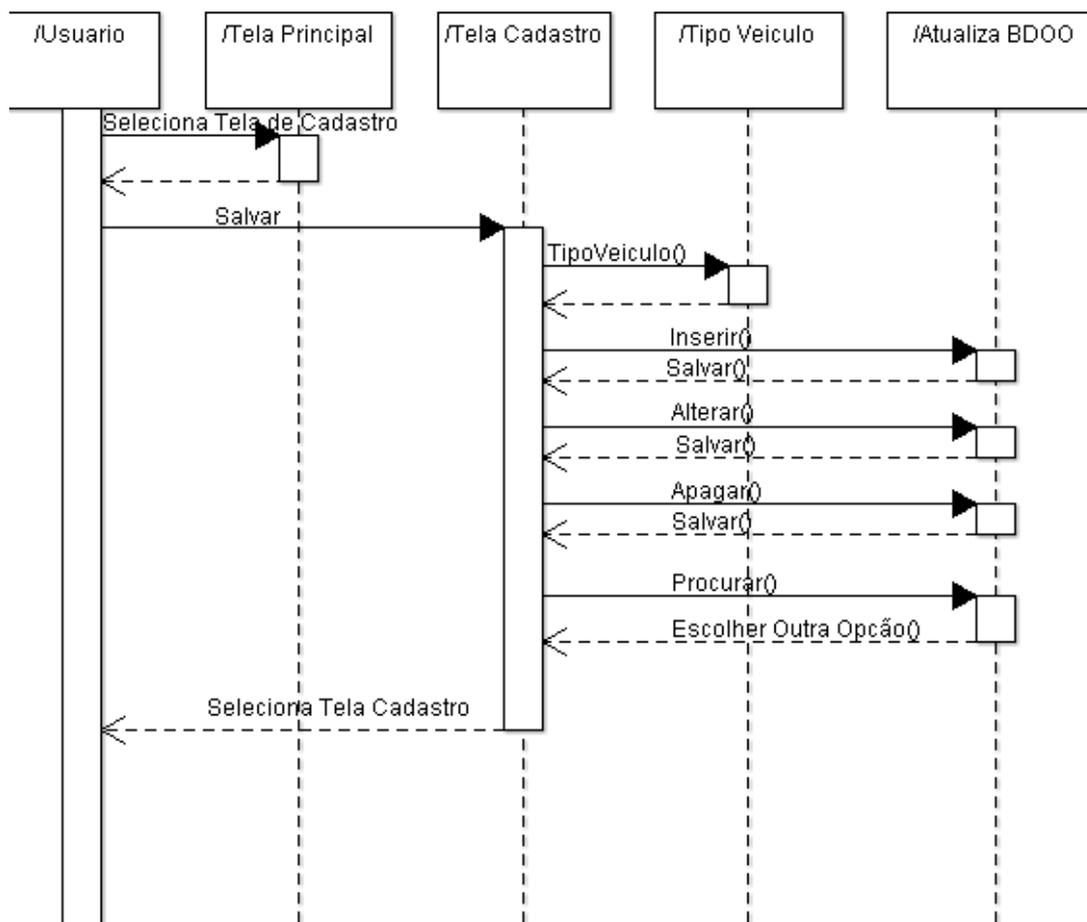


Figura 16 – Diagrama de Sequência Manter Tipo Veículo

#### 4.3.4.7 – Manter Vendedor

As ações representadas no diagrama da figura 17 são executadas quando um dos processos de manutenção de registro (inserir, alterar, excluir e pesquisar) vendedor for necessário. Assim, é possível compreender o processo de comunicação entre as classes. Neste diagrama foi ocultada a comunicação existente entre as classes de comunicação com o banco de dados.

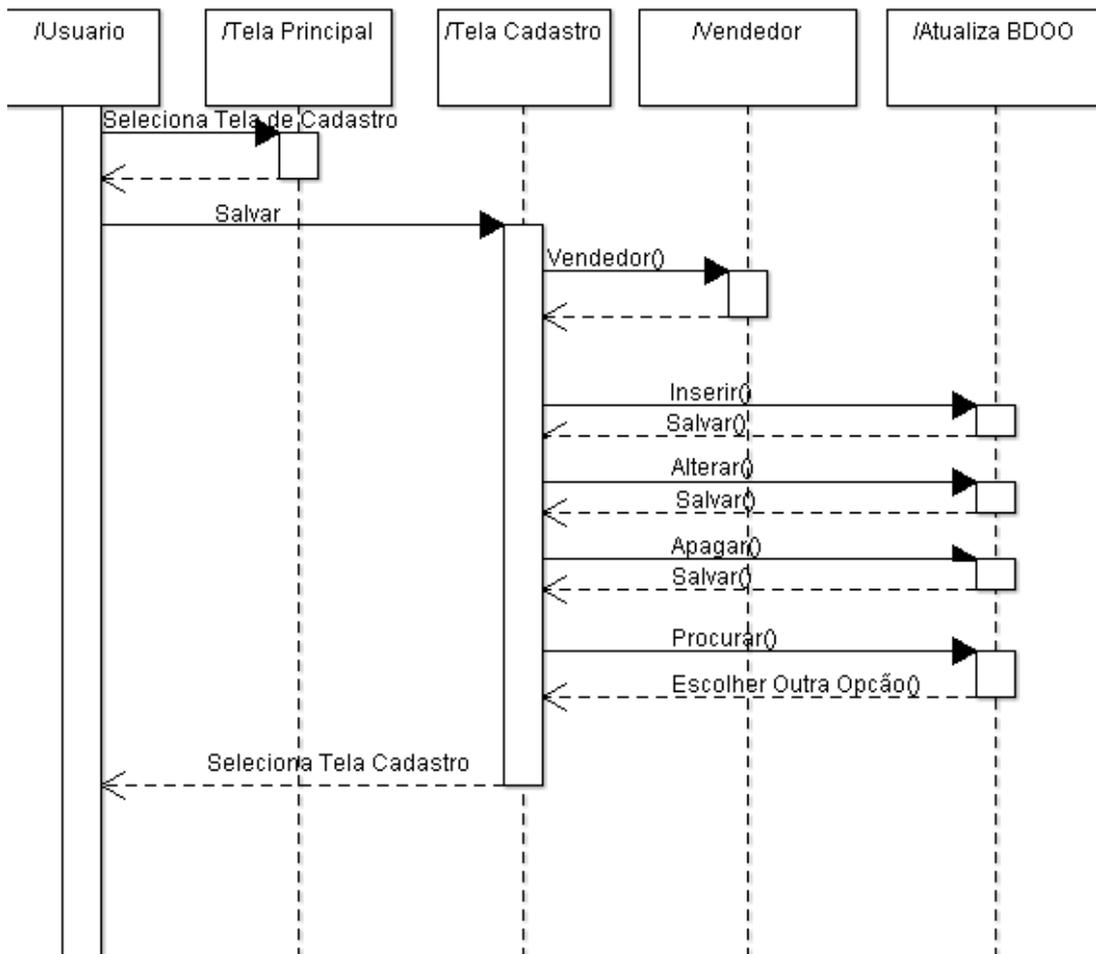


Figura 17 – Diagrama de Sequência Manter Vendedor

#### 4.3.4.8 – Manter Venda

As ações representadas no diagrama da figura 18 são executadas quando um dos processos de manutenção de registro (inserir, alterar, excluir e pesquisar) venda for necessário. Assim, é possível compreender o processo de comunicação entre as classes. Neste diagrama foi ocultada a comunicação existente entre as classes de comunicação com o banco de dados.

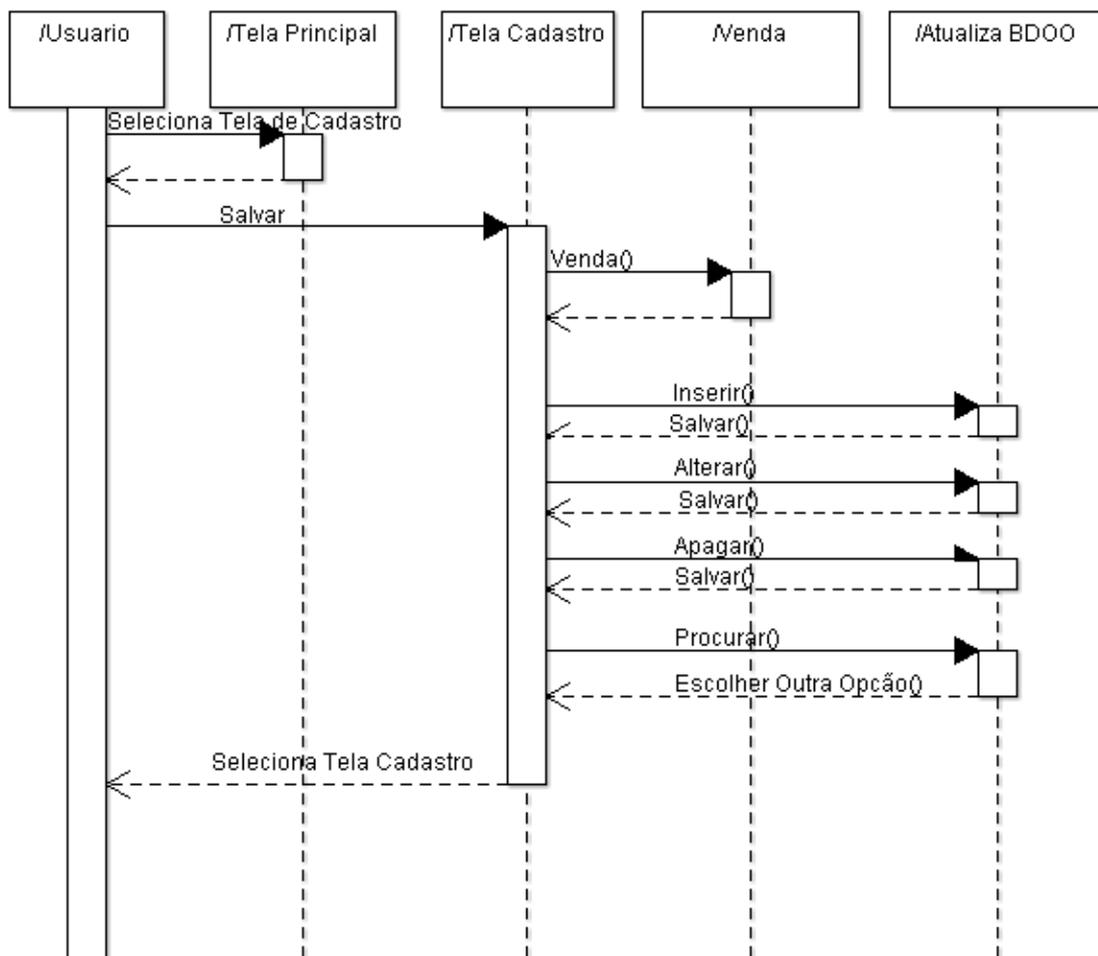


Figura 18 – Diagrama de Sequência Manter Venda

#### 4.3.4.9 – Manter Dados Específicos Data Warehouse

As ações representadas no diagrama da figura 19 são executadas por gatilhos embutidos no sistema enviado dados específicos para o *data warehouse* para mais tarde serem analisados, esses gatilhos são disparado quando o sistema há manutenção no sistema e nas transações.

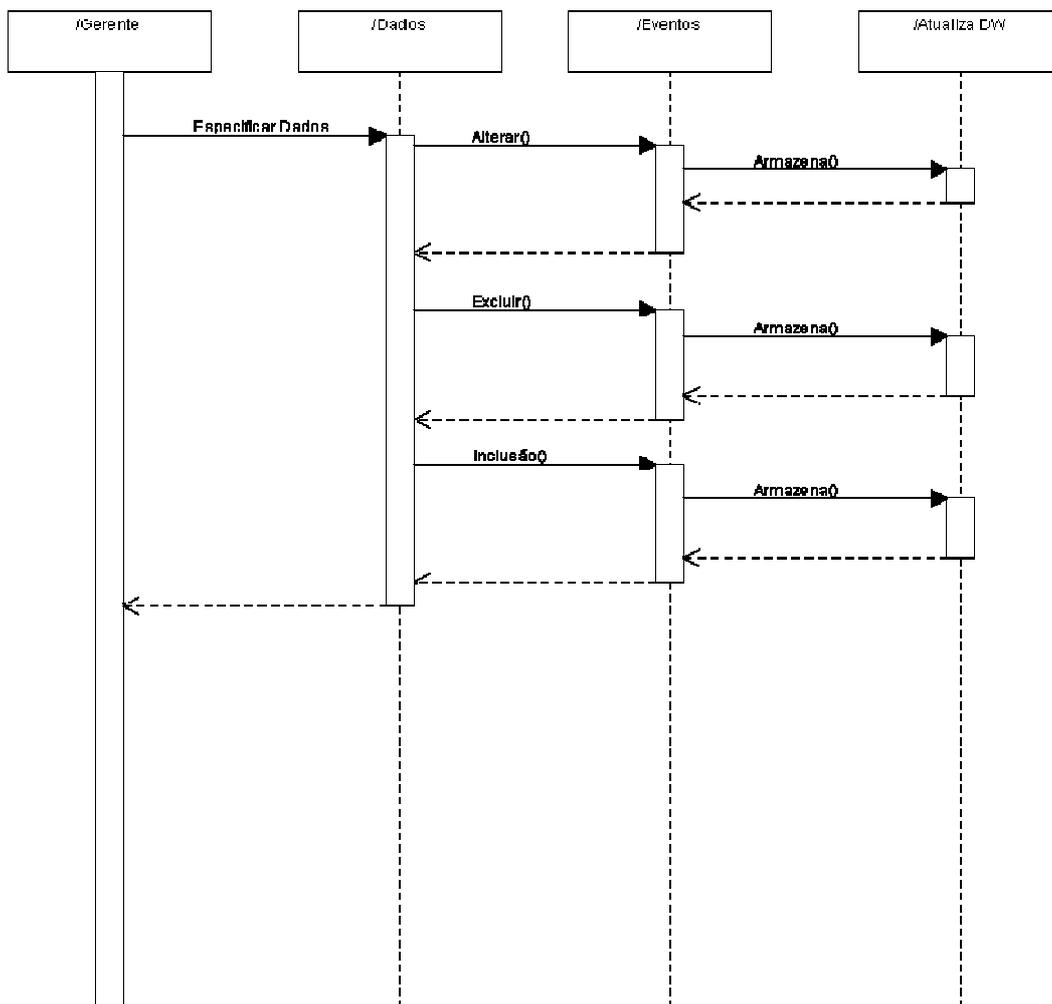


Figura 19 – Manter Dados Específicos Data Warehouse

## 4.4 – Implementação do Aplicativo

Serão mostrados mais detalhadamente, os métodos utilizados para a implementação do estudo de caso para vendas para um *data warehouse* com banco de dados objeto-relacional multidimensional.

### 4.4.1 – Metodologia Utilizada

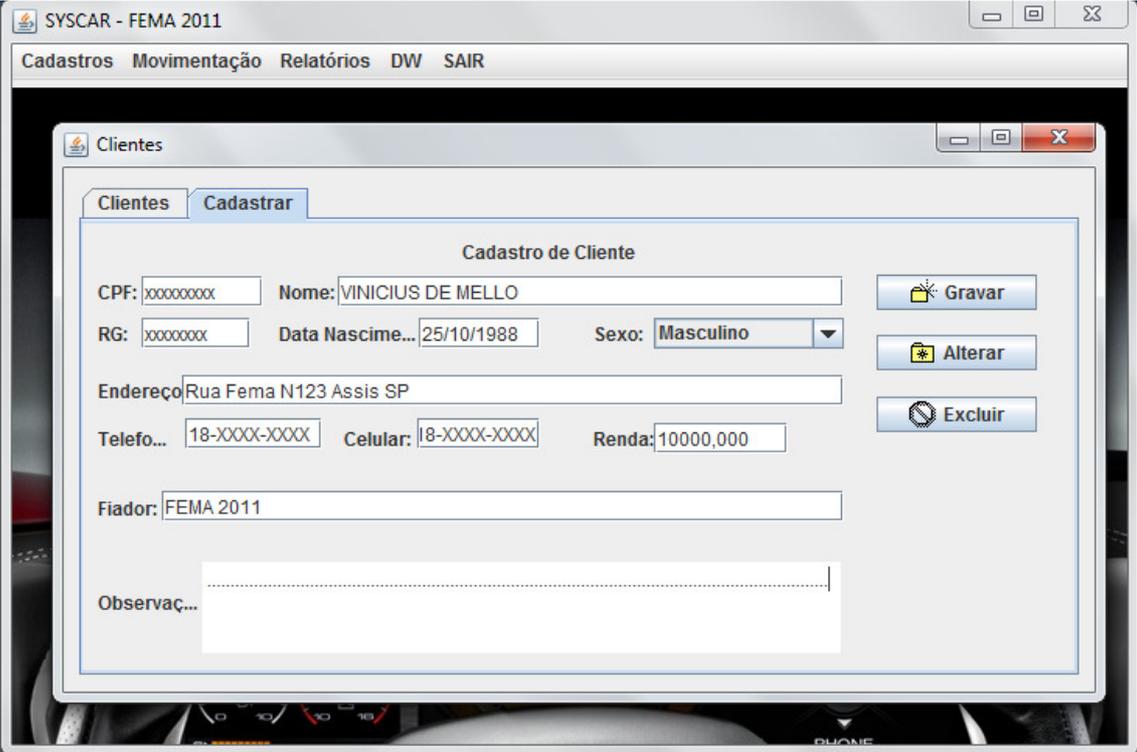
Esta implementação envolveu as seguintes etapas: coleta de dados, inclusão de dados específicos, extração para uso de análise gerenciais. As metodologias serão descritas nas etapas abaixo. A linguagem de programação usada foi Java com a utilização do ambiente IDE (IDE – *IntegratedDevelopmentEnvironment*) Eclipse Ganywade, para a criação das telas foi utilizado um componente chamado *Windows Builder*, e a comunicação entre as interfaces e o banco de dados foram usado o componente *Hibernate* na sua versão 3.2.5.ga todos os componentes e ferramentas acima citadas são *Open Source* para o armazenamento de dados do sistema foi utilizado o banco de dados objeto Oracle 10G XE, sendo que para os dados do sistema ele trabalha de forma objeto-relacional, e para os dados que serão armazenados no *Data Warehouse* ele trabalha de forma relacional. Para a extração de dados será utilizado a ferramenta SQL Developer 3.0.02.83, os dados serão exportado para planilhas que serão manipulados através do Microsoft Excel.

#### 4.4.1.1 - Manter Dados

Nesta etapa são feitos os cadastros para assim poder alimentar o banco de dados orientado objeto-relacional e também gerar os dados necessários para a transação de venda, é a partir desses dados que são extraídos os dados específicos para alimentar o *Data Warehouse*. Serão apresentadas as interfaces do aplicativo.

#### 4.4.1.1.1 – Manter Clientes

A figura 20 mostra a interface de cadastro de clientes, os clientes cadastrados são usados futuramente para a venda de veículos.



The screenshot displays a software window titled 'SYSCAR - FEMA 2011' with a menu bar containing 'Cadastros', 'Movimentação', 'Relatórios', 'DW', and 'SAIR'. Inside, a sub-window titled 'Clientes' is open, showing a 'Cadastro de Cliente' form. The form has two tabs: 'Clientes' and 'Cadastrar'. The 'Cadastrar' tab is active, showing the following fields and values: CPF: xxxxxxxx; Nome: VINICIUS DE MELLO; RG: xxxxxxxx; Data Nascime...: 25/10/1988; Sexo: Masculino; Endereço: Rua Fema N123 Assis SP; Telefo...: 18-XXXX-XXXX; Celular: 18-XXXX-XXXX; Renda: 10000,000; Fiador: FEMA 2011; and Observaç... (empty). On the right side of the form, there are three buttons: 'Gravar' (Save), 'Alterar' (Edit), and 'Excluir' (Delete).

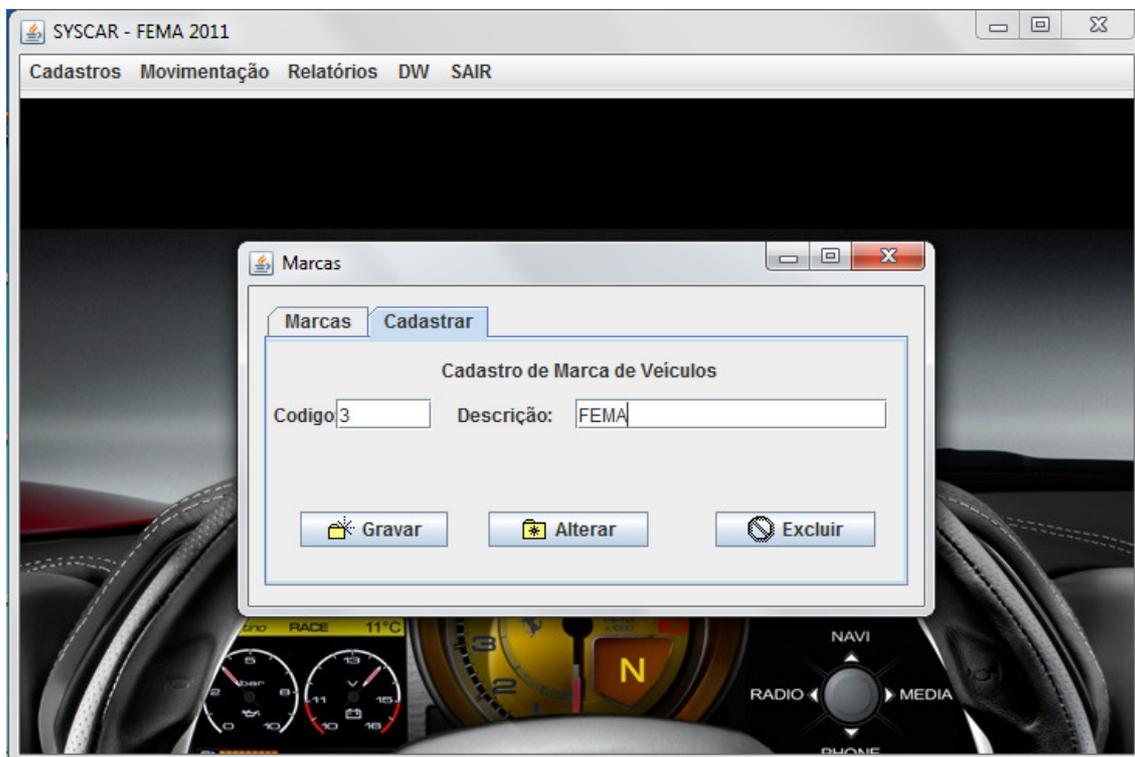
**Figura 20 – Interface Cadastro de Clientes**

O usuário entra com os dados e aciona o botão gravar para salvar os dados no banco de dados. O botão “alterar” altera os dados já gravados, e o botão “excluir”, elimina o registro do banco de dados.

#### 4.4.1.1.2 – Manter Carros - Marcas

A figura 21 mostra a interface de cadastro de marcas de veículos, as marcas cadastradas são necessárias para juntar o veículo a sua respectiva marca,

sendo assim é imprescindível que a marca seja cadastrada primeiro que o veículo. Para que seja possível filtrar veículos por marca.

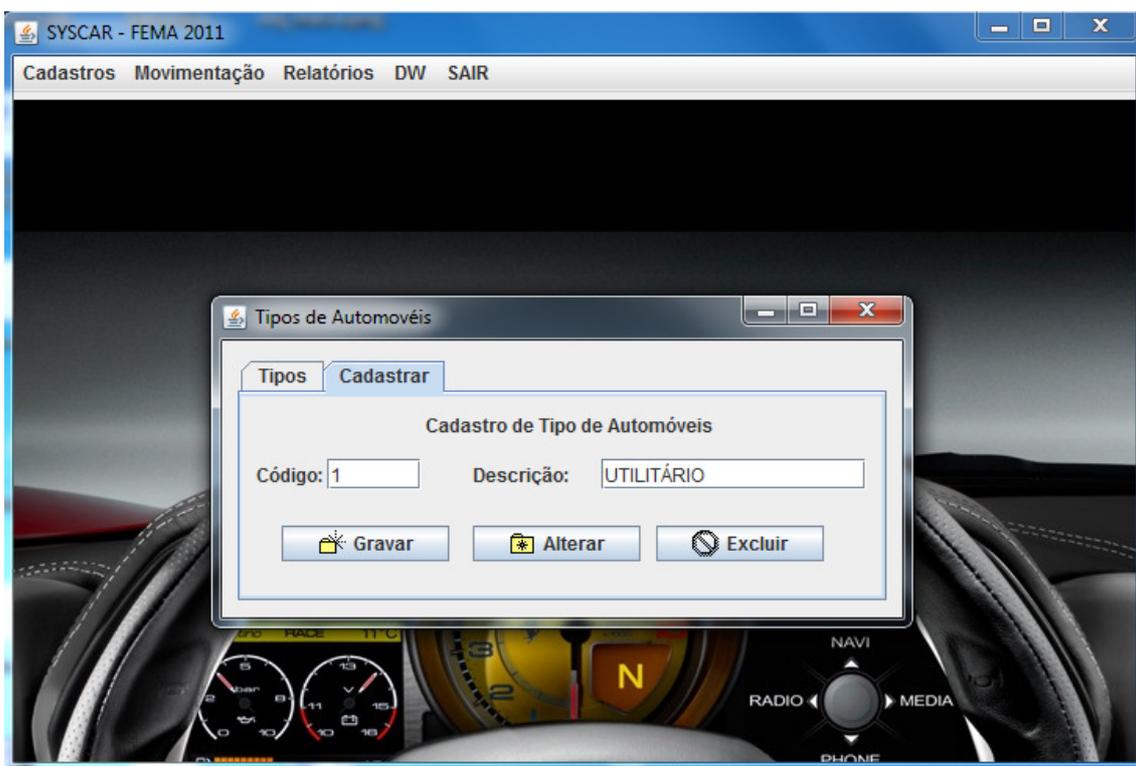


**Figura 21 – Interface Cadastro de Marcas**

O usuário entra com os dados e aciona o botão gravar para salvar os dados no banco de dados. O botão “alterar” altera os dados já gravados, e o botão “excluir”, elimina o registro do banco de dados.

#### **4.4.1.1.3 – Manter Carros - Tipo de Veículo**

A figura 22 mostra a interface de cadastro de categorias de veículos, as categorias cadastradas são necessárias para juntar o veículo a sua respectiva categoria, sendo assim é imprescindível que a categoria seja cadastrada primeiro que o veículo. Essa categoria é utilizada para possíveis relatórios futuros, podendo saber os modelos por categoria.



**Figura 22 – Interface Cadastro de Tipos de Veículos**

O usuário entra com os dados e aciona o botão gravar para salvar os dados no banco de dados. O botão “alterar” altera os dados já gravados, e o botão “excluir”, elimina o registro do banco de dados.

#### **4.4.1.1.4 – Manter Carros - Modelos**

A figura 23 mostra a interface de cadastro de veículos, o cadastro de veículos é imprescindível para que o sistema faça sua principal atividade que é a venda de veículos. É a partir deste cadastro também é que possível medir o estoque.



**Figura 23 – Interface Cadastro de Veículos**

O usuário entra com os dados e aciona o botão gravar para salvar os dados no banco de dados. O botão “alterar” altera os dados já gravados, e o botão “excluir”, elimina o registro do banco de dados.

#### **4.4.1.1.5 – Manter Financeiro – Formas de Pagamento**

A figura 24 mostra a interface de cadastro de formas de pagamento, necessário para definir qual a forma de pagamento no momento da realização da venda de veículos. Essas formas são definidas pelo estabelecimento, e a regra de negócio só permite uma única forma de pagamento.



**Figura 24 – Interface Cadastro de Forma de Pagamento**

O usuário entra com os dados e aciona o botão gravar para salvar os dados no banco de dados. O botão “alterar” altera os dados já gravados, e o botão “excluir”, elimina o registro do banco de dados.

#### **4.4.1.1.6 – Manter Financeiro – Financeira**

A figura 25 mostra a interface de cadastro de financeiras, é utilizado no momento em que for realizar a venda de veículos caso a opção de pagamento vista na interface acima seja financiada.

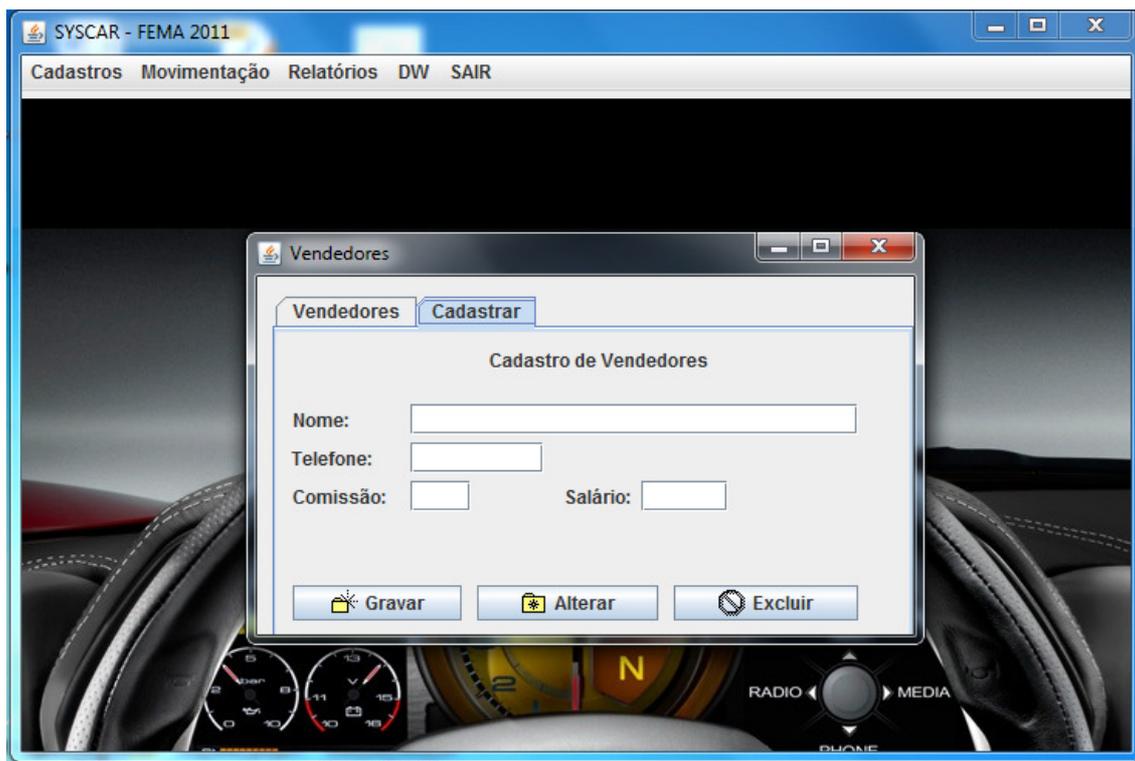


**Figura 25 – Interface Cadastro de Financeira**

O usuário entra com os dados e aciona o botão gravar para salvar os dados no banco de dados. O botão “alterar” altera os dados já gravados, e o botão “excluir”, elimina o registro do banco de dados.

#### **4.4.1.1.7 – Manter Vendedores**

A figura 26 mostra a interface de cadastro de vendedores, é utilizado no momento em que for realizada a venda de veículos, para que em módulos futuramente implementados seja calculado, percentual de comissão, e folha de pagamento.



**Figura 26 – Interface Cadastro de Vendedor**

O usuário entra com os dados e aciona o botão gravar para salvar os dados no banco de dados. O botão “alterar” altera os dados já gravados, e o botão “excluir”, elimina o registro do banco de dados.

#### **4.4.1.1.8 – Manter Venda**

A figura 27 mostra a interface de vendas, é imprescindível que todos os cadastros acima tenha sido preenchidos corretamente. É a partir desta operação que são retirados os dados para alimentar o *Data Warehouse*. Essa operação é uma operação dupla, vista que quando for salva com há aprovação, negada a mesma é salva como um orçamento, no momento em que for alterada para aprovada, ou salva como aprovada a mesma é salva e vista pelo sistema como uma venda consolidada. Essa aprovação é derivada da forma de pagamento escolhida caso a forma definida seja financiada, é necessário que a

ficha seja aprovada junto à financeira. Sendo assim após essa resposta a venda sofre uma alteração de status de aprovação.



The screenshot displays a software window titled 'SYSCAR - FEMA 2011' with a menu bar containing 'Cadastros', 'Movimentação', 'Relatórios', 'DW', and 'SAIR'. A modal dialog box titled 'Venda de Veículos' is open, featuring the following fields and controls:

- Código:** 1
- Contrato Nº:** 0000001
- Data Venda:** 26/10/2011
- Forma de Pagamento:** FINANCIADO (dropdown)
- Financeira:** BANCO DO BRASIL (dropdown)
- Cliente:** VINICIUS DE MELLO
- Vendedor:** VINICIUS (dropdown)
- Veículo:** STRADA
- Preço R\$:** 55000,00
- Aprovado:** Aprovado (dropdown)

At the bottom of the dialog are three buttons: 'Gravar' (Save), 'Alterar' (Edit), and 'Excluir' (Delete).

**Figura 27 – Interface Movimentação Venda**

O usuário entra com os dados e aciona o botão gravar para salvar os dados no banco de dados. O botão “alterar” altera os dados já gravados, e o botão “excluir”, elimina o registro do banco de dados.

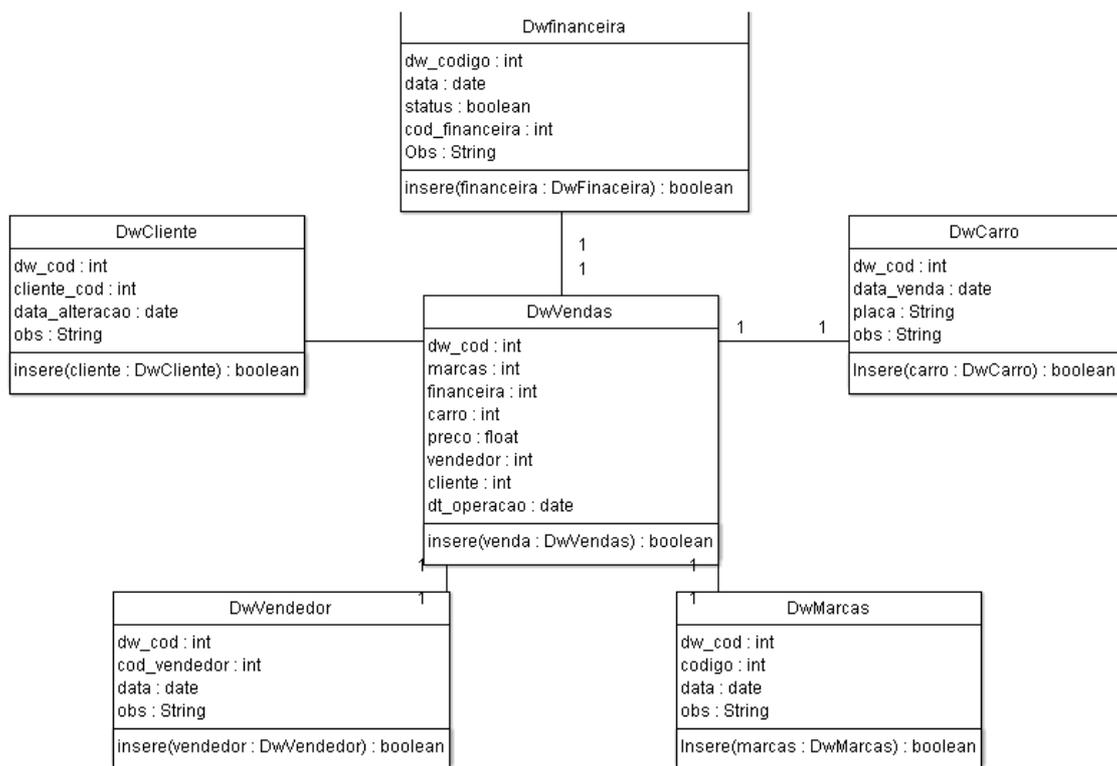
#### **4.4.1.2 - Gravar Dados Específicos no *Data Warehouse***

Nesta etapa são inseridos os dados específicos definidos em entrevista com gerentes, diretores, e presidentes das organizações da área de atuação do sistema.

#### 4.4.1.2.1 - A Construção do *Data Warehouse*

A construção deste *data warehouse* foram definidos através das necessidades encontradas no dia a dia em obter dados importantes que ocorreram no passado, que no sistema já foi sob escrito por novos dados. Além de poder obter dados ricos para análise de projeções futuras, tendências de mercado. É uma necessidade também das empresas neste setor conhecerem os seus clientes, buscando assim visar a qualidade no atendimento e a segurança nas transações futuras.

Com os conhecimentos obtidos na fundamentação teórica deste trabalho a modelagem multidimensional utilizada foi o modelo estrela. Como mostra a figura 28.



**Figura 28 – Modelagem Multidimensional Estrela Data Warehouse**

Sendo assim, a tabela de fatos é a DwVendas, que possui toda as faces das outras tabelas. Esse método de modelar o DW de forma integral denomina-se *top-down*, criando assim um modelo único, sólido, não muda. Para a carga de dados no data warehouse, ou processo de ETL. É utilizado o processo de OLTP, que é disparado através de gatilhos no sistema a cada operação de manutenção de dados, ou seja, a cada alteração ou exclusão de dados é enviado para o DW uma carga de dados. Essa mesma operação ocorre quando fazemos uma transação no sistema, ou seja, a cada venda realizada é feita uma carga no DW, com os dados modelados na figura acima. Sendo assim o processo de carga de dados no DW é feita de uma forma automatizada. Essa forma só pode ser implementada, pois o sistema já foi criado contemplando estas necessidades, então não é necessário monitorar a base de dados procurando informações, operações. Quando não contemplado é necessário usar outras metodologias não relatadas neste artigo.

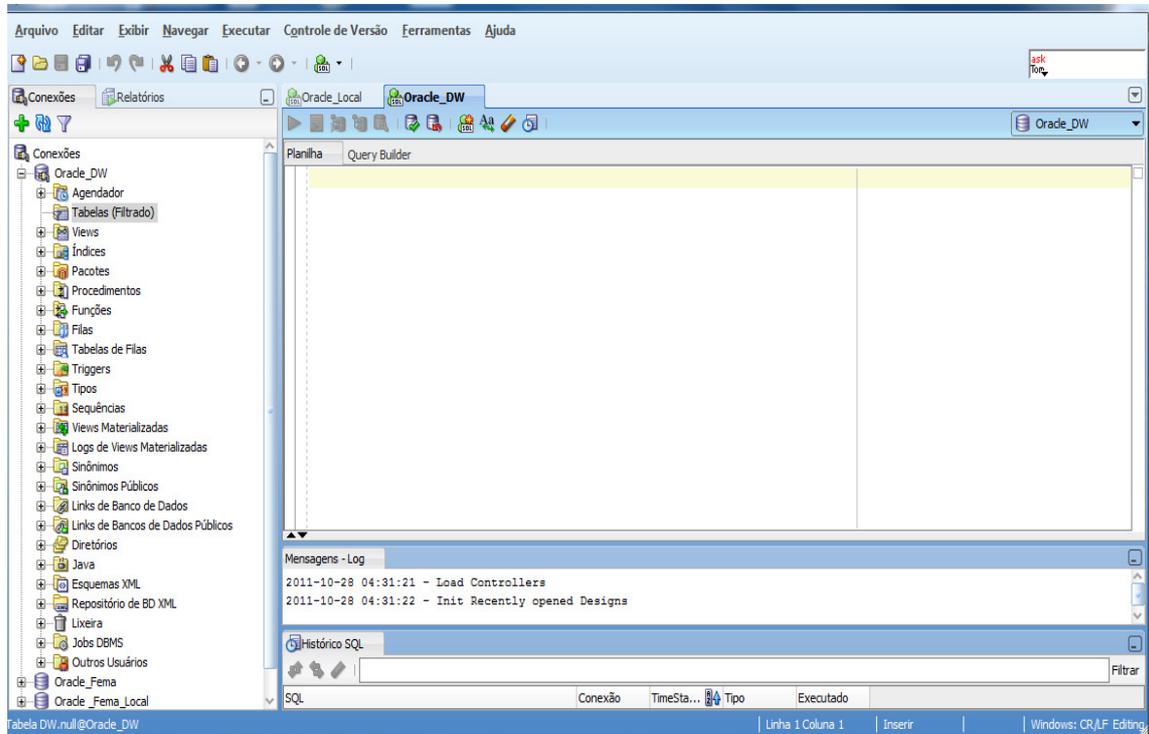
#### **4.4.1.3 - A Extração de Dados *Data Warehouse***

Nesta etapa vai ser mostrado como é extraído os dados, lembrando que para esse processo a organização precisa contar em sua equipe um DBA (*Data Base Administrator*). Pois o processo é feito via comandos SQL (*Structured Query Language*) ou linguagem de consulta estruturada, o resultado dessas consultas pode ser exportado para tabelas, ou mesmo a criação de relatórios.

#### **4.4.1.3 - Extraindo Dados**

Para extração de dados é necessário o conhecimento tanto do sistema quanto do DW, e também na regra de negócio do sistema. Os dados precisam ser estudados e filtrados antes de apresentados. Pois como o processo de carga de dados no *Data Warehouse* é *on-line*, automático existe uma grande quantidade de dados, até mesmo dados redundantes. Para a extração de

dados é utilizada a ferramenta SQL Developer 3.0.02.83, como mostra a figura 29.



**Figura 29 - Visão Geral SQL Developer**

## 5. CONCLUSÃO

O desenvolvimento deste trabalho foi de suma importância para a aquisição de conhecimento em banco de dados orientados a objetos e banco de dados objeto-relacional. Além disso, o vasto conhecimento adquirido sobre os conceitos de como criar um *data warehouse*. Isso sem dúvida nenhuma acrescentou em muito os meus conhecimentos a cerca desta área. Futuramente, pretende-se atuar profissionalmente nesta área que está em crescente evolução.

O benefício desta tecnologia podem ser vistas em grandes empresas que estão adotando o Data Warehouse para armazenamento de grandes quantidades de dados e com complexidade de interpretação destes dados, já que um dos enfoques é agilizar o processo de tomada de decisão.

## Referências Bibliográficas

BERTINI, G.C., **Uma Modelagem Orientada a Objeto Para o Mapa Urbano Básico de Belo Horizonte (MUB/BH)**, Monografia de Especialização em Informática Pública, PUC-MG em Parceria com PRODABEL, 2004.

BRUZAROSCO, D.C.; CASTOLDI, A.V. e PACHECO, R.C.S., **Criando Data Warehouse Com o Modelo Dimensional**, Acta Scientiarum 22, 2000.

CARDOSO, A. S.; CARLIN, H.L.S. e ROSARIO, J.A., **Banco de Dados Orientado a Objetos e Objeto Relacional**, Disponível em <http://www.inf.unioeste.br/~olguin/4458-semin/G3-apresentacao.pdf>. Acesso em agosto 2011.

CENTENARO, A.C.; DRUZIANI, C.F.; SUCOLOTTI, A.A. e TODESCO, J.L., **Modelando Um Data Warehouse Dimensional a Partir de um Modelo Hierárquico**, 2004.

Chen, P. **Gerenciando Banco de Dados: A Abordagem Entidade-Relacionamento para Projeto Lógico**. McGraw-Hill, 1990.

CLODIS, B.; BEZERRA, A.; BENEDICTO, M. e DELMIRO, G - UNIOESTE - Universidade Estadual do Oeste do Paraná

DAPHNE Corp. Building Corporate Business Intelligence: From **Data marts to the Enterprise Data Warehouse**. URL:

<http://www.daphnesoft.com/whitepaper.doc>, 1999.

Donizete Carlos Bruzarosco, André Vinicius Castoldi e Roberto Carlos dos Santos Pacheco. **Criando data warehouse com o modelo dimensional**. Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, 88037-001, Florianópolis-Santa Catarina, Brazil.

ELMASRI, R. **Sistemas de banco de dados**. Ramez Elmasri e Shamkant B. Navathe; revisor técnico Luis Ricardo de Figueiredo. São Paulo: Addison Wesley, 2005.

ENDEL A, Disponível em <http://www.slideshare.net/eneck/trabalho-banco-de-dados-orientado-a-objetos>. Acesso maio de 2011.

FIGUEREDO, Rafael Batista. **Tecnologia de Data Warehouse**. Monografia – UEL: Universidade Estadual de Londrina, Londrina, 2007.

GALANTE, A.C.; MOREIRA, E.L.R. e BRANDÃO, F.C., **Banco de Dados Orientados a Objetos: Uma Realidade**, 2007. Disponível em [http://www.fsma.edu.br/si/edicao3/banco\\_de\\_dados\\_orientado\\_a\\_objetos.pdf](http://www.fsma.edu.br/si/edicao3/banco_de_dados_orientado_a_objetos.pdf). Acesso em junho de 2011.

HARRISON, T.H. **Intranet Data Warehouse**, Berkeley, 1998.

INMON, W.H., **Como Construir o Data Warehouse**, São Paulo, Campus, 1987.

INMON, W.H., **What Is A Data Warehouse?**, 1995. Disponível em [http://www.cait.wustl.edu/papers/prism/vol1\\_no1/](http://www.cait.wustl.edu/papers/prism/vol1_no1/). Acesso em agosto de 2011.

KIMBALL ET AL., **The Data Warehouse Lifecycle Toolkit-Expert Methods For Designing, Developing and Deploying Data Warehouse**, John Wiley and Sons, 1998.

KIMBALL, R. e CASERTA, J., **The Data Warehouse ETL Toolkit: Practical Techniques For Extracting, Cleaning, Conforming, and Delivering Data**. John Wiley e Sons, New York, 2004.

MACHADO, F.N.R. **Projeto de Data Warehouse: Uma visão multidimensional**, São Paulo: Érica, 2004.

MEYER, D.; CAMMON, C. **Building a better Data Warehouse**. New Jersey: Prentice Hall PTR, 1998.

MONTEIRO ET AL., **Uma Aplicação de Data Warehouse Para Apoiar Negócios**, UERJ. Disponível em <http://www.ime.uerj.br/cadernos/cadinf/vol16/CadernosIME-INF-V16-5-Rosa.PDF>. Acesso em junho de 2011.

MORGAN, T.T.J. **Database Modeling And Design**. Kaufmann Publishers, 1990.

OLIVEIRA, M. P. J.; EDELWEISS, N.; RIBEIRO, P. F. H. C. **Sistema distribuído para intercâmbio de dados de saúde – SIDI**, Disponível em <http://www.devmedia.com.br/post-6978-Qualidade-na-Modelagem-dos-Dados-de-um-Data-Warehouse.html>. Acesso em março 2011.

RATIONAL INC, **Banco de Dados Orientados a Objetos – Conceitos Fundamentais**, 2004. Disponível em [http://www.malima.com.br/article\\_read.asp?id=40](http://www.malima.com.br/article_read.asp?id=40). Acesso em maio de 2011.

RUMBAUGH ET AL., **Modelagem e Projetos Baseados em Objetos**, Editora Campus, 1994.

SANTOS, I.M., **Data Warehouse como ferramenta de auxílio em sistemas de monitoramento ambiental**, Trabalho de Conclusão de Curso, UFMT, 2003.

SERRA, L., **A Essência do Business Intelligence**, São Paulo, Berkeley Brasil, 2002.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistema de banco de dados**. São Paulo: Makron Books, 1999.

THONSEM, E., OLAP: **Construindo Sistemas de Informações Multidimensionais**, Tradução da 2ª Edição Original, Rio de Janeiro, Campus, 2002.