

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO DE INFORMÁTICA  
CURSO SUPERIOR DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**CELSO CANTERI  
SIRLÉIA DAIANE DVULATKA**

**MODELAGEM E IMPLEMENTAÇÃO DE UM SISTEMA DE VENDA E DE  
LOCAÇÃO DE IMÓVEIS BASEADO EM LINHAS DE PRODUTO**

**TRABALHO DE CONCLUSÃO DE CURSO**

**PONTA GROSSA**

**2013**

**CELSO CANTERI  
SIRLÉIA DAIANE DVULATKA**

**MODELAGEM E IMPLEMENTAÇÃO DE UM SISTEMA DE VENDA E DE  
LOCAÇÃO DE IMÓVEIS BASEADO EM LINHAS DE PRODUTO**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, da Coordenação de Análise e Desenvolvimento de Sistemas (COADS), da Universidade Tecnológica Federal do Paraná.

Orientador: Prof<sup>a</sup>. Dr<sup>a</sup>. Simone Nasser Matos.

**PONTA GROSSA**

**2013**



Ministério da Educação  
**Universidade Tecnológica Federal do  
Paraná**  
Campus Ponta Grossa  
Diretoria de Graduação e Educação  
Profissional



---

## TERMO DE APROVAÇÃO

MODELAGEM E IMPLEMENTAÇÃO DE UM SISTEMA DE VENDA E DE LOCAÇÃO  
DE IMÓVEIS BASEADO EM LINHAS DE PRODUTO

por

CELSO CANTERI  
SIRLÉIA DAIANE DVULATKA

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 16 de agosto de 2013 como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Profª Drª Simone Nasser Matos  
Orientadora

---

Prof. Msc Willian Massami Watanabe  
Membro titular

---

Profª. Drª. Helyane B. Borges  
Responsável pelos Trabalhos  
de Conclusão de Curso

---

Prof. Msc Vinicius Camargo Andrade  
Membro titular

---

Profª. Drª Simone de Almeida  
Coordenadora do curso  
UTFPR – Campus Ponta Grossa

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

## RESUMO

CANTERI, Celso; DVULATKA, Sirléia Daiane. **Modelagem e Implementação de um Sistema de Venda e de Locação de Imóveis Baseado em Linhas de Produto.** 2013. 80f. Trabalho de Conclusão de Curso Tecnologia em Análise e Desenvolvimento de Sistemas - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2013.

O desenvolvimento baseado em linhas de produto permite a modelagem e a construção de sistemas com uma arquitetura genérica na qual se analisa as similaridades e variabilidades entre aplicações de um mesmo domínio. Este trabalho propõe a modelagem e a implementação dos sistemas de venda e locação de uma imobiliária. Foi utilizado um método adaptado que contempla as melhores práticas oferecidas pelo: *Product Line UML-Based Software Engineering, Family-Oriented Abstraction, Specification, and Translation* e *Feature Oriented Domain Analysis*. A modelagem foi desenvolvida no ambiente *Odyssey* e a implementação em linguagem Java. Neste trabalho também são apresentadas as vantagens do uso do desenvolvimento baseado em linha de produto e como o modelo proposto pode ser estendido para uma outra aplicação.

**Palavras-chave:** Linhas de Produtos de Softwares. *UML Components*. Sistema Imobiliário.

## ABSTRACT

CANTERI, Celso; DVULATKA, Sirléia Daiane. **Imóveis LPS: Modelagem e Implementação de um Sistema de Venda e Locação Baseado em Linhas de Produto.** 2013. 80 f. Coursework for Tecnologia em Análise e Desenvolvimento de Sistemas - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2013.

Developing systems using LPS - Product Lines Software enables building a generic development architecture analyzing the similarities and variabilities between applications of the same domain. The domain analyzed was the real estate system. It was used an adapted method that includes the best practices offered by: Product Line UML-Based Software Engineering, Family-Oriented Abstraction, Specification, and Translation and Feature Oriented Domain Analysis. The model was developed using the Odyssey environment and the implementation was developed using Java. It also presents the benefits of using product line development and how to extend the model to create new applications.

**Keywords:** Software Product Lines. Modeling. UML. Real Estate System. Architecture.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Atividades essenciais da linha de produto de software.....	19
Figura 2 - Artefatos de uma linha de produto de software (1) e os produtos gerados a partir do reuso dos artefatos (2, 3 e 4) .....	20
Figura 3 - Modelos de engenharia de domínio no Odyssey .....	30
Figura 4 - Diagrama de modelo de contexto .....	33
Figura 5 - Tela para cadastrar os imóveis .....	35
Figura 6 - Tela para cadastrar clientes.....	35
Figura 7 - Tela para cadastrar os proprietários .....	36
Figura 8 - Tela para realizar busca no sistema .....	36
Figura 9 - Tela que direciona para o site mostrando o endereço selecionado .....	37
Figura 10 - Tela para buscar um imóvel e acessar o cadastro do imóvel selecionado .....	37
Figura 11 - Pasta disponível para salvar os documentos necessários.....	38
Figura 12 - Tela para cadastrar um texto padrão .....	39
Figura 13 - Tela para incluir um novo imóvel.....	40
Figura 14 - Tela para incluir um novo cliente.....	40
Figura 15 - Tela de busca entre os imóveis disponíveis dos reservados .....	41
Figura 16 - Tela para cadastrar os textos necessários.....	42
Figura 17 - Diagrama de <i>features</i> do sistema de vendas.....	44
Figura 18 - Tela para cadastrar os inquilinos e as obrigações contratuais.....	46
Figura 19 - Diagrama de <i>features</i> do Sistema de Locações .....	49
Figura 20 - Diagrama de <i>features</i> para os Sistemas de Locação e Venda .....	50
Figura 21 - Diagrama de caso de uso com as similaridades do Sistema de Venda e Locações.....	51
Figura 22- Interfaces do Software Imóveis dos casos de uso das similaridades.....	56
Figura 23 - Interfaces com os Métodos do Software Imobiliário dos casos de uso das similaridades. ....	57
Figura 24 - Modelo de Tipo de Negócio para Imóveis.....	58

Figura 25 Diagrama de responsabilidades das interfaces do modelo de tipo de negócio.....	60
Figura 26 Configuração arquitetural na camada de sistema .....	62
Figura 27 Componentes das interfaces de negócios .....	62
Figura 28 Arquitetura do Software Imobiliária .....	63
Figura 29 - Diagrama de arquitetura de alto nível do sistema .....	63
Figura 30 - Tela para cadastrar, editar, localizar e excluir um cliente do sistema. ....	64
Figura 31 - Tela para incluir uma locação no sistema. ....	65
Figura 32 Variabilidades do sistema Imobilis 2013 .....	66
Figura 33 Diagrama de Caso de Uso das variabilidades Imobilis 2013 .....	67
Figura 34 Modelo de classe para instanciação do sistema Imobilis 2013 .....	69
Figura 35 - Modelo de classe para instanciação para Sistema Administração de Veículos .....	72

## LISTA DE QUADROS

Quadro 1 - Benefícios tangíveis de uma linha de produto de software .....	17
Quadro 2 - Benefícios intangíveis de uma linha de produto de software .....	18
Quadro 3 - Artefatos de entrada e saída produzidos no método adaptado .....	28
Quadro 4 - Comparativo de similaridades e variabilidades dos sistemas de vendas	43
Quadro 5 - Comparativo de similaridades e variabilidades do sistema de locações	48
Quadro 6 - Comparativo entre o sistema de locação e venda .....	49
Quadro 7- Descrição do cenário de Manter cadastro de imóveis.....	52
Quadro 8 - Descrição para o cenário de Controlar locações.....	53
Quadro 9 - Descrição do cenário de Controlar vendas .....	54
Quadro 10 - Descrição do cenário de Pesquisar informações .....	54
Quadro 11- Cenário do Caso de Uso: "Manter Arquivo Digital" .....	67
Quadro 12- Descrição do cenário "Ver no mapa" .....	68
Quadro 13- Cenário do Caso de Uso: "Ver ficha" .....	68



## LISTA DE ABREVIATURAS

LPS- Linha de Produto de Software

SEI - *Software Engineering Institute*

UML – *Unified Modeling Language*

FAST - *Family-Oriented Abstraction, Specification and Translation*

FODA - *Feature Oriented Domain Analysis*

ODM - *Organization Domain Modeling*

PLUS - *Product Line UML-Based Software Engineering*

DSSA - *Domain-Specific Software Architecture*

PuLSE - *Product Line Software Engineering*

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>12</b>
1.1 OBJETIVOS .....	13
1.1.1 Objetivo Geral .....	13
1.1.2 Objetivos Específicos .....	13
1.2 ORGANIZAÇÃO DO TRABALHO .....	13
<b>2 LINHAS DE PRODUTOS DE SOFTWARE .....</b>	<b>15</b>
2.1 FINALIDADE DO DESENVOLVIMENTO BASEADO EM LPS .....	15
2.2 VANTAGENS NA UTILIZAÇÃO DE LPS.....	16
2.3 ATIVIDADES DA LINHA DE PRODUTO DE SOFTWARE.....	18
2.4 MÉTODOS DE DESENVOLVIMENTO DE LINHAS DE PRODUTO.....	20
2.5 O MÉTODO DELAZERI & WOLF .....	26
2.6 FERRAMENTAS PARA O DESENVOLVIMENTO EM LINHA DE PRODUTO ...	29
<b>3 MODELAGEM DO SISTEMA IMÓVEIS.....</b>	<b>31</b>
3.1 ENGENHARIA DE DOMÍNIO .....	31
3.1.1 Análise de domínio .....	31
3.1.2 Identificação das características .....	33
3.1.2.1 Requisitos do domínio .....	34
3.1.2.2 Modelagem de Domínio .....	50
3.1.2.3 Projeto do Domínio.....	55
3.1.2.4 Modelagem da Arquitetura .....	55
3.1.2.5 Identificação dos Componentes .....	56
3.1.2.6 Implementação do domínio .....	63
3.1.2.7. Testes do domínio.....	66
3.2 ENGENHARIA DA APLICAÇÃO .....	66
3.2.1 Requisitos da aplicação.....	66
<b>4 RESULTADOS E DISCUSSÕES SOBRE A MODELAGEM BASEADA EM LINHA DE PRODUTO .....</b>	<b>70</b>
4.1 DIFICULDADES E LIMITAÇÕES NA CONSTRUÇÃO DE LPS .....	70

4.2 VANTAGENS NA UTILIZAÇÃO DE LPS PARA O SISTEMA IMOBILIÁRIO.....	70
4.3 REUSO DA LPS CRIADA PARA O SISTEMA IMOBILIÁRIO .....	71
<b>5 CONCLUSÃO .....</b>	<b>74</b>
5.1 TRABALHOS FUTUROS .....	75
<b>REFERÊNCIAS.....</b>	<b>76</b>

## 1 INTRODUÇÃO

Empregar corretamente o reuso de código no desenvolvimento de sistemas sempre foi uma necessidade e um grande desafio. Essa necessidade está explicitada na redução do esforço e do tempo de desenvolvimento, na diminuição do tempo de disponibilização dos sistemas e principalmente na redução de custos de desenvolvimento. O desafio está em manter padrões de desenvolvimento que atendam aos requisitos do cliente e manter a qualidade dos softwares que estão cada vez maiores e mais complexos.

O reuso de artefatos de software focado na engenharia de linha de produto é uma alternativa viável para buscar a redução de custos, atender a crescente complexidade dos softwares atuais, tentar acelerar o desenvolvimento das aplicações, e diminuir o tempo de implantação de novos produtos (LOBO; RUBIRA, 2009).

A identificação dos núcleos principais de sistemas dentro de um mesmo domínio, através da análise das suas similaridades, permite a criação de artefatos básicos de software que serão reutilizados no desenvolvimento de vários produtos similares ao mesmo tempo ao invés de criar apenas um produto de cada vez (LOBO; RUBIRA, 2009). Essa modalidade de desenvolvimento baseada em componentes é chamada de engenharia de linha de produto, ou mais modernamente de Linha de Produtos de Software (LPS).

Pesquisar os principais métodos de desenvolvimento de softwares utilizados em Linha de Produtos de Software LPS, analisar suas características principais e suas fases, é o objeto de estudo apresentado na fundamentação teórica deste trabalho. O método utilizado neste trabalho foi o Delazeri e Wolf (2012), que foi criado fundamentado em fases dos métodos PLUS (*Product Line UML-Based Software Engineering*), FAST (*Family-Oriented Abstraction, Specification, and Translation*) e FODA (*Feature Oriented Domain Analysis*). Este método foi usado na modelagem e implementação do sistema de imóveis Imobiliária, nos sistemas de Venda e Locação.

Para a análise do domínio imobiliária, que é uma empresa que presta serviços de comercialização, locação e gestão de bens imóveis, foram estudados os

sistemas Imobilis 2013 (IMOBILIS, 2013) e Ci-Pro (CI-PRO, 2013), por possuírem as *features* necessárias para a análise desse domínio. Outro recurso utilizado foi uma entrevista com um gerente imobiliário da Imobiliária X.

Desse estudo criou-se uma arquitetura de domínio que contempla os pontos comuns entre os sistemas analisados e as variabilidades encontradas entre eles, com o uso do Diagrama de Classes para a instanciação dos produtos através do seu núcleo comum. A implementação da arquitetura foi realizada em uma linguagem de programação orientada a objetos.

A arquitetura criada foi reusada para um sistema de locação de carros com a utilização de um diagrama de classe.

## 1.1 OBJETIVOS

Os objetivos deste trabalho estão descritos na subseção 1.1.1 que descreve o objetivo geral e na subseção 1.1.2 onde são listados os objetivos específicos propostos.

### 1.1.1 Objetivo Geral

Criar um modelo arquitetural para os sistemas de Locação e Venda por intermédio de um desenvolvimento baseado em linhas de produto e o codificar em uma linguagem de programação.

### 1.1.2 Objetivos Específicos

- Aplicar a metodologia escolhida na modelagem dos sistemas de venda e locação de uma imobiliária.
- Codificar a arquitetura gerada em uma linguagem de programação.
- Estender o modelo gerado em outro domínio.

## 1.2 ORGANIZAÇÃO DO TRABALHO

Este trabalho foi dividido em cinco capítulos de acordo com a sequencia dos estudos. O capítulo 1 apresenta a motivação para o desenvolvimento desta pesquisa.

O capítulo 2 descreve o que são Linhas de Produtos de Software - LPS, a finalidade do desenvolvimento baseado em LPS, as vantagens de sua utilização,

suas atividades, os métodos existentes, o método LPS utilizado neste trabalho e uma análise de ferramentas para o desenvolvimento em Linha de Produto.

O capítulo 3 aborda a Modelagem do Sistema Imobiliária e descreve como foi estudada e implementada a Engenharia de Domínio e a Engenharia da Aplicação no estudo de caso no domínio imobiliária.

O capítulo 4 descreve os resultados e discussões sobre a modelagem baseada em LPS, as dificuldades e limitações na construção de uma LPS, a vantagem de utilização de LPS para um sistema imobiliário e o reuso da LPS criada para o sistema imobiliário.

O capítulo 5 apresenta a conclusão do trabalho e contém as considerações finais do trabalho, os possíveis trabalhos futuros e os resultados obtidos.

## 2 LINHAS DE PRODUTOS DE SOFTWARE

Este capítulo apresenta a abordagem de desenvolvimento de software baseado em linhas de produtos (LPS) e seus métodos. A Seção 2.1 descreve a finalidade do uso de linhas de produtos de software. A Seção 2.2 relata as vantagens em utilizar LPS. A seção 2.3 apresenta as atividades a serem cumpridas durante o desenvolvimento baseado em linhas de produtos. A seção 2.4 retrata os métodos que podem ser utilizados no desenvolvimento baseado em linhas de produtos. A seção 2.5 descreve o método que será utilizado para o desenvolvimento do estudo de caso proposto neste trabalho. A seção 2.6 apresenta a ferramenta de modelagem usada por este trabalho.

### 2.1 FINALIDADE DO DESENVOLVIMENTO BASEADO EM LPS

Uma empresa de software bem sucedida é aquela que fornece produtos com qualidade, capaz de atender às necessidades dos respectivos usuários. Para isto, deve usar de forma eficiente e eficaz recursos tais como o reuso de software (BOOCH et al., 2000).

Perante a situação, a área de linha de produto de software tem surgido como uma importante e viável solução, cujo objetivo não é apenas reuso de código, mas da arquitetura e dos requisitos (NEIVA, 2008).

Segundo o Instituto de Engenharia de Software (SEI, 2012):

*Uma linha de produto faz uso da reusabilidade e estratégia planejada. Mais do que uma nova tecnologia, é uma nova maneira de fazer negócios. Organizações que estão desenvolvendo linhas de produtos de software estão experimentando grandes aprimoramentos no gerenciamento de custos, melhorias no tempo de desenvolvimento e produtividade.*

Uma linha de produto de software é um conjunto de produtos com alto grau de similaridade entre si, que atendem às necessidades específicas de um segmento de mercado ou missão e que são desenvolvidos de forma prescritiva a partir de um conjunto de artefatos básicos (LOBO; RUBIRA, 2009 apud CLEMENTS; NORTHROP, 2002).

A abordagem de linhas de produto de software tem como objetivo verificar as diferenças e semelhanças entre produtos e artefatos de software similares, e permitir adaptá-los de acordo com as necessidades específicas. As diferenças entre eles deverão aparecer na definição dos requisitos e serão representadas ao longo do processo (OLIVEIRA, 2009).

## 2.2 VANTAGENS NA UTILIZAÇÃO DE LPS

As principais vantagens de uma linha de produto são (SILVA, 2011):

- a) Redução dos custos de desenvolvimento - O investimento inicial para se construir uma linha de produto de software é maior se comparado ao software convencional, isso acontece porque o custo da criação dos ativos que compõem a plataforma é alto. Dada que a plataforma reusável já está criada, o custo para se desenvolver um software a partir de tal infraestrutura é reduzido. Quando são construídos entre 3 (três) e 4 (quatro) sistemas a partir da linha de produto de software o investimento inicial feito equivale ao de construir entre 3 (três) e 4 (quatro) sistemas na maneira convencional, desse número em diante o custo para se construir um produto baseado na LPS é menor.
- b) Aumento da qualidade - Todos os artefatos criados que compõem a plataforma são revisados e testados em vários produtos diferentes. Logo, a qualidade dos produtos baseados na plataforma aumenta por meio do reuso de ativos de software já testados e consolidados.
- c) Redução do tempo de entrega dos produtos - O tempo de mercado caracteriza-se pela rapidez com que um novo produto é lançado para atender uma demanda específica de mercado. Através da linha de produto de software, o tempo de mercado diminui, visto que vários artefatos podem ser reusados para cada novo software.
- d) Redução do esforço da manutenção - Quando um artefato da plataforma é alterado, para corrigir um erro, tal mudança é propagada para todos os produtos que reusam tal artefato. Dessa forma, o esforço de manutenção é reduzido.



Vários exemplos concretos são apresentados para ilustrar as melhorias de custos, tempo de mercado e produtividade por meio do uso LPS (MUNIZ, 2011 apud CLEMENTS; BAAS; KAZMAN, 2003), tais como:

1. Nokia foi capaz de produzir 25 a 30 modelos diferentes de celulares por ano por causa da abordagem de linha de produtos.
2. *Cummins, Inc.* diminuiu o tempo para produzir o software de um motor diesel de cerca de um ano para uma semana.
3. Motorola observou uma melhoria de produtividade de 400% em uma família de determinado tipo de celular.
4. *Hewlett-Packard* reportou uma redução no tempo de defeitos por um fator de sete e aumento na produtividade por um fator de seis, em uma família de sistemas de impressoras.

Para mapear melhor as vantagens propostas pela linha de produtos Durscki (2004) classificou os benefícios de uma linha de produto de duas maneiras:

- Tangíveis: benefícios que podem ser medidos diretamente, como redução de defeitos. Além deste, outros melhoramentos tangíveis estão descritos no quadro 1.

**Quadro 1 - Benefícios tangíveis de uma linha de produto de software**

Benefícios tangíveis	
Lucratividade	O repositório de ativos permite que a organização produza produtos voltados para um segmento específico de mercado. O benefício dessa focalização é observado no aumento da participação de mercado e da lucratividade.
Qualidade	Uma redução no número de defeitos relatados é comum em sistemas desenvolvidos em linhas de produtos. A qualidade também pode ser medida em termos de redução do tempo de correções e do efeito <i>ripple</i> (geração de novos defeitos a partir de correções executadas).
Desempenho dos produtos de softwares	A utilização de ativos aumenta o desempenho em relação ao desenvolvimento tradicional, especialmente com o aumento da maturidade da linha, o que faz com que os ativos estejam cada vez mais otimizados.
Tempo de integração	O tempo de integração no desenvolvimento incremental é facilitado.
Produtividade	A equipe de desenvolvimento pode ser reduzida. O custo total de desenvolvimento é cortado consideravelmente. O cronograma é reduzido. O sistema possui uma flexibilidade documentada, o que facilita o atendimento das solicitações de modificações dos clientes.

**Fonte: Durscki (2004, p.158)**

- Intangíveis: benefícios que os desenvolvedores relatam, mas que não podem ser medidos em termos de métricas. Esses benefícios podem incluir satisfação do cliente, ânimo dos recursos, entre outros, como descrito no quadro 2.

**Quadro 2 - Benefícios intangíveis de uma linha de produto de software**

Benefícios intangíveis	
Desgaste de profissionais	Menor desgaste dos profissionais, que resulta em uma redução de rotatividade dos membros da equipe.
Aceitabilidade dos desenvolvedores	Após um treinamento inicial, os desenvolvedores relatam satisfação em trabalhar com a abordagem baseada em ativos e arquitetura comuns.
Satisfação profissional	Os desenvolvedores relatam que o trabalho braçal já foi realizado (desenvolvimento dos ativos de software), assim eles podem concentrar-se em atividades mais interessantes, como o aperfeiçoamento e/ou inovação de elementos específicos.
Satisfação do cliente	Os ativos reduzem os riscos, aumentando a previsibilidade da entrega e diminuindo a taxa de defeitos. Esses fatores afetam positivamente o cliente (induzindo-o a preferir produtos derivados de linhas de produto).

**Fonte: Durscki (2004, p.158)**

### 2.3 ATIVIDADES DA LINHA DE PRODUTO DE SOFTWARE

As atividades essenciais de uma linha de produto são:

- Engenharia de domínio: responsável por estabelecer uma plataforma reusável e definir os aspectos básicos *core assets* (núcleos) da linha de produto de software para um determinado domínio (SILVA, 2011). A plataforma consiste em todos os tipos de artefatos de software: arquitetura, componentes, especificação de requisitos, modelo do domínio, plano de testes, entre outros.
- Engenharia da aplicação: desenvolvimento dos produtos a partir dos núcleos estabelecidos na engenharia de domínio. Ao invés de iniciar um sistema do zero, é feito o uso dos artefatos do repositório (PACIOS, 2006). Ela explora a variabilidade da linha de produtos e assegura sua correta instanciação de acordo com as necessidades específicas das aplicações finais. Os insumos essenciais da atividade de engenharia da

aplicação são: requisitos, escopo da linha de produtos, núcleos e o plano de produção (MUNIZ, 2011).

- **Gerenciamento:** inclui gestão técnica e organizacional da linha de produto (NEIVA, 2008). Desempenha um papel vital no sucesso da organização, porque fornece e coordena a infraestrutura necessária.
- A gestão supervisiona a construção dos núcleos e atividades de desenvolvimento do produto, e garante que os construtores dos núcleos e dos produtos estejam plenamente envolvidos nas atividades e acompanhando o processo definido para a linha de produtos (MUNIZ, 2011).

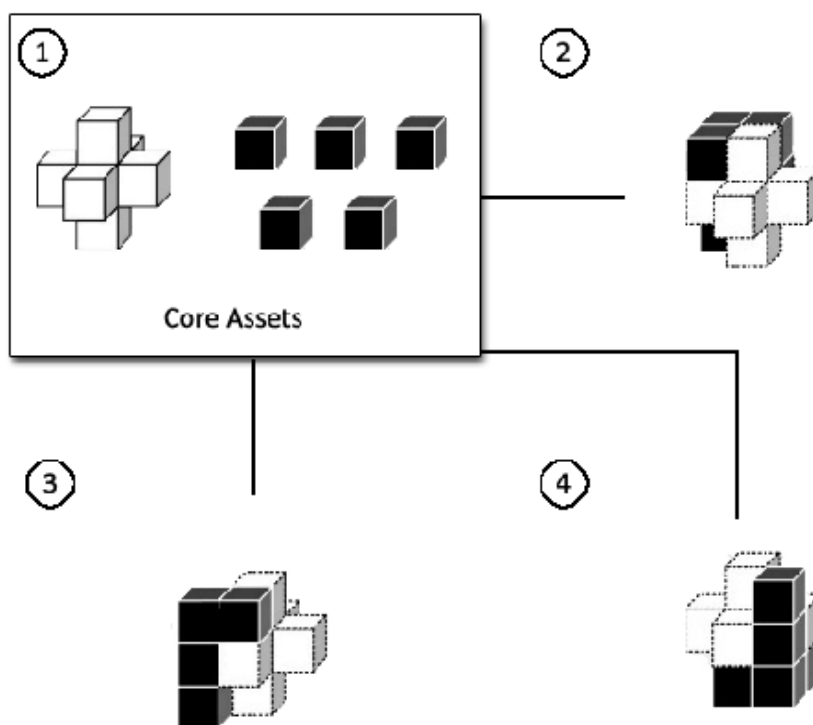
De acordo com Clements e Northrop (MUNIZ, 2011 apud CLEMENTS; NORTHROP, 2002). em muitos casos, a parte da gestão é a atividade responsável pelo sucesso ou fracasso do produto final de uma linha de produtos.

Essas três atividades são altamente iterativas e relacionadas, conforme ilustrado na figura 1.



**Figura 1 - Atividades essenciais da linha de produto de software**  
Fonte: Lobo; Rubira (2009, p.5)

Na linha de produto de software, o planejamento das características e construção dos núcleos, que posteriormente serão utilizados para gerar os diversos produtos da linha é essencial. A figura 2 ilustra que nenhum dos produtos gerados é igual ao outro, cada um deles possui variações específicas (MEDEIROS, 2008).



**Figura 2 - Artefatos de uma linha de produto de software (1) e os produtos gerados a partir do reuso dos artefatos (2, 3 e 4)**

**Fonte: Medeiros (2008, p.9)**

#### 2.4 MÉTODOS DE DESENVOLVIMENTO BASEADOS EM LINHAS DE PRODUTO

O reuso de artefatos de software focado na engenharia de linha de produto é uma alternativa viável para buscar a redução de custos, atender a crescente complexidade dos softwares atuais, tentar acelerar o desenvolvimento das aplicações e diminuir o tempo de implantação de novos produtos (LOBO; RUBIRA, 2009).

Para o desenvolvimento de linhas de produtos existem vários métodos de engenharia de domínio e de linha de produto que podem ser adotados, sendo os mais citados na literatura técnica: Draco (NEIGHBORS, 1980), FODA (*Feature Oriented Domain Analysis*) (SEI - Software Engineering Institute, 1990), ODM

(*Organization Domain Modeling*) (SIMOS, 1995), PLUS (*Product Line UML-Based Software Engineering*) (GOMAA, 2005), FAST (*Family-Oriented Abstraction, Specification, and Translation*) (HARSU, 2002), FeatuRSEB (BONTEMPS et al, 2004), DSSA (*Domain-Specific Software Architecture*) (ARMITAGE, 1993), PuLSE (*Product Line Software Engineering*) (ATKISON; BAYER, 2000) e KobrA (ATKISON; BAYER, 2000).

A seguir são brevemente descritos cada um destes métodos.

### *Draco*

Primeiro método utilizado na engenharia de domínio, o sistema Draco analisa a rotina de produção de vários sistemas similares. O objetivo é construir um amplo, compreensível, manutenível sistema de documentos que represente uma implementação livre de erros e satisfaça as necessidades e desejos do usuário.

Este sistema visa também estender a reutilização de um conjunto de componentes reutilizáveis na fase de análise e projeto da construção do software.

Draco é um sistema interativo que visa o refinamento de um problema descrito em um alto nível para uma solução eficiente em um programa executável simples e de baixo nível. O método Draco trabalha em quatro fases principais: análise de domínio, linguagem de domínio, o uso de componentes de software para implementar as linguagens de domínio e o uso de transformações dos programas-fonte com o intuito de especializar os componentes para a sua utilização em um sistema específico.

### *FODA*

É um método de análise de domínio, desenvolvido pelo SEI *Software Engineering Institute* (Instituto de Engenharia de Software) em 1990. Este método visa à construção de artefatos genéricos que sejam largamente reutilizáveis dentro do domínio. O objetivo principal da análise de domínio é obter e representar informações sobre sistemas de software que compartilham características comuns. A atividade de modelagem do domínio é subdividida nas atividades: análise de características, modelagem entidade-relacionamento e análise funcional.

### ODM

É um método de engenharia de domínio que utiliza vários outros enfoques de engenharia de domínio e já foi empregado principalmente no projeto STARS (*Software Technology for Adaptable, Reliable Systems*) e em menor escala pelas organizações: *Unisys Corporation*, no programa CARDS (*Comprehensive Approach to Reusable Defense Software*) da Força Aérea e pelo SEI, além da HP (*Hewlett-Packard*) no RADAR (*Reusability Analysis/Domain Analysis Review*).

O método ODM introduziu alguns conceitos novos, tais como: 1) tipos de domínios (horizontais ou verticais); 2) definição mais abrangente de característica; 3) análise de combinação de características; 4) modelagem conceitual; 5) refinamento do escopo da base de bens; 6) arquitetura flexível ou arquitetura de artefatos básicos para representar as variabilidades e (7) processo customizado.

O modelo de processo ODM separa o ciclo de vida de desenvolvimento em duas fases distintas: a Fase de Modelagem de Domínio Descritivo e a Base de Ativos Prescritivo (Fase de Engenharia).

### PLUS

É um método que provê um conjunto de conceitos e técnicas para estender a UML. O seu principal objetivo é modelar as similaridades e variabilidades em uma LPS e fornecer alguns complementos ao processo de modelagem de sistemas simples.

As fases do método PLUS são: modelagem dos requisitos de uma LPS que possui as subfases: modelagem de caso de uso e de *features*; modelagem de análise de uma LPS com as subfases: modelagem estática, modelagem de interação dinâmica, modelagem dinâmica de estados e modelagem de dependências de *features*/classes; modelagem de projeto de uma LPS com suas subfases: padrões de arquitetura de software e projeto de software baseado em componentes e engenharia de aplicação de software.

### FAST

É um processo de desenvolvimento utilizado para produzir software em um método orientado a famílias de softwares. Ele separa o processo de engenharia de linha de produtos em duas partes principais, sendo que a primeira se concentra em

fornecer os ativos essenciais, inclusive o ambiente para a implementação de cada produto e a segunda parte utiliza esse ambiente na produção de diferentes produtos de software pertencentes à família.

A linha de produtos por meio da engenharia FAST busca atender as necessidades de dois requisitos importantes no processo de produção de software: uma engenharia cuidadosa, essencial para satisfazer as necessidades do cliente e desenvolver sistemas confiáveis, de fácil utilização e manutenção; e a rápida produção de software exigida pelo mercado e concorrentes.

O processo FAST pode ser dividido nos seguintes subprocessos: qualificação de domínio para identificar famílias dignas de investimento; engenharia de domínio para investir em soluções para a produção de membros de família de produtos; engenharia de aplicação para usar as soluções com o objetivo de produzir os membros da família de software mais rapidamente.

### *FeatuRSEB*

É uma combinação entre o método FODA e a *Reuse-Driven Software Business Engineering* (RSEB) utilizando a notação da linguagem UML com algumas extensões.

RSEB é um processo de caso de uso orientado pelo reuso sistemático, em que a variabilidade é capturada por meio da estruturação de casos de uso e modelos de objeto com pontos de variação e variantes através de diagramas RFD (*Rank-Frequency Diagrams*). Diagramas de recurso estendidos (EFD *Extended Features FODA Diagrams*) foram criados e introduzidos para compensar a ambiguidade e suposta falta de precisão e expressividade dos diagramas originais.

Os diagramas de recurso EFD nunca receberam uma semântica formal, que é a marca registrada de precisão e não ambiguidades. O método RSEB define vários processos de desenvolvimento de software orientados a modelo: *Architecture Family Engineering* (Engenharia de Arquitetura de Famílias - desenvolve uma arquitetura em camadas), *Components System Engineering* (Componentes de Engenharia de Sistemas - desenvolvimento de sistemas de componentes reutilizáveis) e *Applications System Engineering* (Engenharia de Aplicações do Sistema - desenvolve aplicativos selecionados). Estes processos são utilizados para aperfeiçoar a robustez e reutilização (GRISS; FAVARO; d'ALESSANDRO, 1998).

## *DSSA*

Método criado devido à comunidade de engenheiros de software ter se conscientizado que o reuso de software é uma das formas de aperfeiçoar a produtividade e qualidade do software.

DSSA é: 1) Um padrão de arquitetura de software construído para um domínio, ou família, de aplicações; 2) Uma especificação para a montagem de componentes de software que são: a) especializados para uma determinada classe de tarefas (domínio); b) generalizada para uso efetivo em desse domínio; c) composta em uma estrutura padronizada (topologia); d) eficaz para a construção de aplicações de sucesso; 3) Um alto nível de estrutura de pacotes de funções e dados, suas interfaces e controle, para apoiar a implementação de aplicações em um domínio.

Uma arquitetura DSSA, chamada de arquitetura de referência, é especificada por requisitos de referência que são o produto de uma análise de domínio. Sistemas aplicativos são construídos mediante a adaptação da arquitetura de referência para atender os requisitos específicos do sistema e completar a arquitetura com componentes da biblioteca DSSA.

O processo DSSA tem quatro atividades distintas: Desenvolver um domínio específico base; Preencher e manter uma biblioteca; Construir aplicações; Operar e manter aplicações.

## *PuLSE*

É um método que viabiliza a concepção e distribuição de linhas de produção de software em uma grande variedade de contextos empresariais.

O ciclo de vida de linha de produção de softwares no PuLSE é dividido nas fases: inicialização, construção da infraestrutura da linha de produção, uso e evolução.

Este método prove os componentes técnicos para as diferentes fases de distribuição que contém o *know-how* técnico necessário para operacionalizar o desenvolvimento de linhas de produção. Os componentes técnicos são customizados para o respectivo contexto.

A customização do PuLSE para o contexto onde será aplicado garante que o processo e produtos são apropriados. Na fase de inicialização, as outras fases e os



componentes técnicos são adaptados. Através dessa adaptação dos componentes técnicos, uma versão customizada da fase de construção, uso e evolução do PuLSE é criada. A customização do método é executada no domínio da aplicação, no contexto organizacional, para fins de reuso, na estrutura do projeto e nos recursos disponíveis.

### *KobrA*

É uma síntese de diversas abordagens de engenharia de software, inclui linha de produtos, desenvolvimento baseado em componentes, *frameworks* e modelagem de processos.

A engenharia de *framework* é responsável por criar e manter um *framework* genérico para toda a família de produtos e inclui funcionalidades comuns e variáveis entre eles.

Na engenharia de aplicação o *framework* é instanciado e estendido e uma aplicação é construída através da resolução dos pontos de variabilidade desse *framework*. A modelagem de componentes, dividida nas fases de especificação de componentes (o que fazer) e realização de componentes (como fazer), é feita por meio de diagramas UML estendido para tratamento de variabilidades (LOBO; RUBIRA, 2009).

Um princípio fundamental do método KobrA é a distinção precisa de produtos e processos. Os produtos do projeto KobrA - modelos, documentos, módulos de códigos, casos de teste, etc - são definidos independentemente, antes dos processos para os quais eles foram criados e representam os objetivos desses processos.

Foram descritos anteriormente vários métodos para o desenvolvimento baseado em linhas de produto. Porém, o método utilizado neste trabalho é uma adaptação para construção de software em LPS fundamentado nos métodos já existentes na literatura: FAST , PLUS e FODA desenvolvido pelo trabalho de Delazeri e Wolf (2012). O método proposto foi adaptado utilizando suas melhores práticas e inclui novos artefatos. Esta adaptação foi feita para que o novo método proporcione fases, documentações e diagramas mais específicos, como será descrito a seguir.

## 2.5 O MÉTODO DELAZERI & WOLF

Delazeri e Wolf (2012) estudaram os métodos PLUS, FAST e FODA e os adaptaram em um método que proporciona fases, documentações e diagramas mais específicos.

O método criado é composto de duas fases principais:

- Engenharia de Domínio: onde são produzidos os artefatos reutilizáveis que modelam o domínio da LPS.
- Engenharia de Aplicação: fase em que se desenvolve uma aplicação individual baseada nas definições da Engenharia de Domínio (DELAZERI, WOLF, 2012).

A fase de Engenharia de Domínio possuem as seguintes subfases:

- Análise de domínio – originada no método FAST e tem o objetivo de analisar e definir o escopo do software que será explorado para obtenção de seus artefatos. No final desta fase é criado o modelo de contexto que define o escopo da modelagem do domínio.
- Identificação de Características - possuem as seguintes etapas distintas:
  - Requisitos do domínio – baseada nos métodos FODA (Análise de Características) e PLUS, que utilizam esta fase para identificar os requisitos do sistema e observar os pontos comuns e variáveis da LPS. O artefato criado é o modelo de características onde estão explícitas as características comuns contidas no domínio.
  - Modelagem do domínio – extraída do método PLUS para fazer a decomposição do problema e modelagem estática do software a fim de definir o relacionamento estrutural entre as classes do domínio ou o comportamento do sistema. Os artefatos dessa fase são os diagramas de classe ou diagramas de caso de uso.
  - Projeto do domínio – proveniente do método PLUS é a fase que permite a escolha da arquitetura em todos os produtos, criando um modelo de projeto do software baseado em componentes.
  - Implementação do domínio – procedente e adaptada, do método PLUS, nesta subfase é realizada a implementação dos componentes reutilizáveis e definida a linguagem de programação que será utilizada no software.

- Testes do domínio – subfase adaptada do método PLUS para efetuar a validação e verificação dos componentes reutilizados para enfatizar a integridade e funcionalidade (DELAZERI; WOLF, 2012).

A fase de Engenharia de Aplicação possui as subfases:

- Requisitos da aplicação - foram adaptadas dos métodos PLUS e FAST para identificar e modelar os requisitos da aplicação em diagramas de caso de uso e criar o diagrama de características das variabilidades contidas no domínio.
- Implementação da Aplicação - advinda do método FAST para efetuar a implementação do produto de acordo com a extração de requisitos obtida em subfases anteriores.
- Testes da Aplicação - adaptada somente da teoria aplicada em linhas de produtos de software para efetuar a validação do produto, ao realizar testes funcionais e de integração e a análise da integridade dos requisitos.
- Entrega e Suporte da Aplicação – esta subfase foi adaptada do método FAST. A entrega da aplicação para o cliente é realizada nesta fase, bem como o suporte à aplicação, quando necessário (DELAZERI; WOLF, 2012).

O quadro 3 ilustra os artefatos produzidos em cada subfase do método e as suas respectivas entradas e saídas:

**Quadro 3 - Artefatos de entrada e saída produzidos no método adaptado**

Fases	Subfases	Artefatos de Entrada	Artefatos de Saída	
Engenharia de Domínio	Análise de domínio	Definição do Domínio a ser modelado	Modelo de contexto definindo o escopo do domínio	
	Identificação de Características	Requisitos do domínio	Dois exemplos de aplicações no domínio, no mínimo.	- Descrição narrativa de cada exemplo. Caso não se tenha a descrição narrativa, devem-se utilizar os aplicativos disponíveis. Neste caso, a análise será realizar por meio da execução do software. Identificação dos pontos comuns entre os estudos de caso. - Requisitos identificados - Diagrama de características (contendo os pontos de comuns)
		Modelagem do domínio	Descrição narrativa de cada exemplo. Pontos de comuns	- Diagrama de Caso de Uso - Cenários dos casos de uso - Diagrama de classe
		Projeto do domínio	Diagrama de Caso de Uso	- Arquitetura da parte Similar (baseada em componentes) - Diagramas de classe para a concepção da arquitetura - Especificação das Interfaces do Sistema. - Identificação das Interfaces de Negócio. Identificação dos Componentes.
		Implementação do domínio	Arquitetura da parte Similar	- Codificação dos componentes da arquitetura similar
		Testes do domínio	Codificação	Validação dos componentes
Engenharia de Aplicação	Requisitos da aplicação	Requisitos da aplicação (pontos de variabilidade) oriundos da fase Requisitos do domínio	- Lista de requisitos das variabilidades - Diagrama de caso de uso da aplicação - Diagrama de classe	
	Implementação da Aplicação.	Diagrama de classe	- Codificação dos pontos variáveis	
	Testes da Aplicação	Codificação da aplicação	- Plano de testes funcionais e de integração - Produto validado	
	Entrega e Suporte da Aplicação	Aplicação validada	- Aplicação	

Fonte: Delazeri e Wolf (2012, p.46)

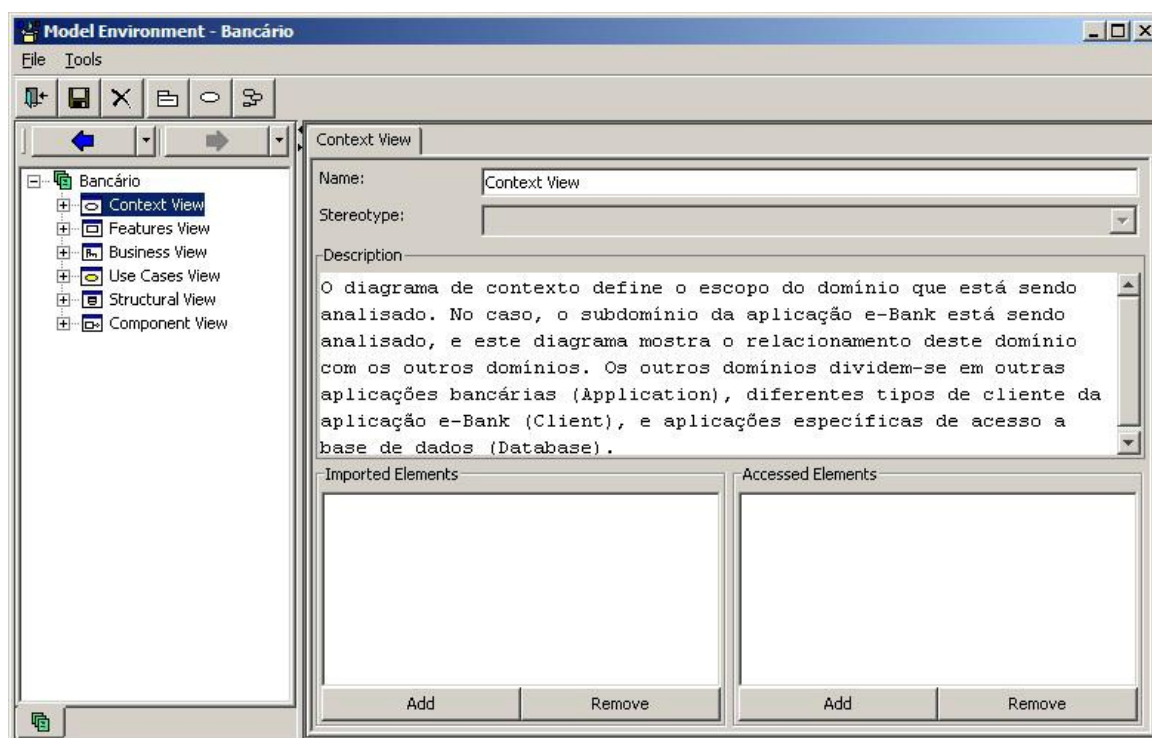
## 2.6 FERRAMENTAS PARA O DESENVOLVIMENTO EM LINHA DE PRODUTO

Dentre as ferramentas de apoio à construção de softwares que utilizam linhas de produto, as mais citadas na literatura são: Odyssey (PROJETO ODYSSEY - COPPE, 2001), fmp (Generative Software Development Lab, da University of Waterloo, 2004), XFeature (ROHLIK; PASETTI, 2005), pure:variants (PURE-SYSTEMS GMBH) e Gears.

Outras ferramentas que podem apoiar o desenvolvimento de LPS foram avaliadas durante o XVIII Simpósio Brasileiro de Engenharia de Software (SBES), XI sessão de ferramentas, promovido pela Comissão Especial de Engenharia de Software da SBC - Sociedade Brasileira de Computação, (REIS, 2004). Dentre elas cita-se: (*Methodology Explorer* - Uma Ferramenta para Definição e Reuso de Metodologias de Desenvolvimento de Software (SILVA JÚNIOR, Carlos Roberto, 2003 <http://www.cin.ufpe.br/~mexplorer/>); C-CORE - *Component Construction and Reuse* - Ferramenta para Desenvolvimento de Software Baseado em Componentes (UFSCAR, 2004).

O sistema *Odyssey*, desenvolvido na COPPE/UFRJ - Coordenação dos Programas de Pós-graduação em Engenharia da Universidade Federal do Rio de Janeiro - é um ambiente de desenvolvimento de software cujo objetivo é construir uma infraestrutura para reutilização baseada na modelagem de domínio de linhas de produto (LOBO; RUBIRA, 2009). A versão atual do *Odyssey* denomina-se *OdysseyLight*, que é um ambiente dinâmico e contém as funcionalidades de apoio às atividades de engenharia de domínio e engenharia de aplicação (TEIXEIRA, 2008).

Suas ferramentas são: modelo de contextos (*Context View*), modelo de características (*Features View*), modelo de casos de uso (*Use Cases View*), modelo de classes (*Structural View*), modelo de regras de negócio (*Business View*) e modelo de componentes (*Component View*), conforme ilustra a figura 3.



**Figura 3 - Modelos de engenharia de domínio no Odyssey**  
**Fonte: Lobo; Rubira (2009, p.12)**

A ferramenta *Odyssey* foi escolhida para utilização no estudo de caso, por ser uma ferramenta gratuita, incorporar a modelagem de características (UML *Components*) dentro do processo de engenharia de domínio e permitir apoiar as várias atividades do método proposto.

### 3 MODELAGEM DO SISTEMA IMÓVEIS

Este capítulo descreve o uso do método de Delazeri e Wolf (2012) para a identificação das similaridades e variabilidades em um Sistema de Controle Imobiliário. A seção 3.1 apresenta a aplicação e os resultados obtidos na fase de Engenharia de Domínio. A Seção 3.2 relata os artefatos gerados na fase de Engenharia de Aplicação.

#### 3.1 ENGENHARIA DE DOMÍNIO

Nesta seção se apresenta a aplicação e os artefatos gerados por cada subfase da fase de Engenharia de Domínio.

##### 3.1.1 Análise de domínio

Uma imobiliária é uma empresa que presta serviços de comercialização, locação e gestão de bens imóveis tais como: casas, terrenos, edifícios, imóveis comerciais, entre outros.

Para a análise do domínio foram selecionados os sistemas Imobilis 2013 (IMOBILIS, 2013) e Ci-Pro (CI-PRO, 2013), por serem softwares que abrangem o mercado imobiliário e estão disponibilizados gratuitamente na Internet, no endereço [www.baixaki.com.br](http://www.baixaki.com.br), estarem entre os mais baixados do sítio – 10.259 e 11.946 *downloads* respectivamente, serem bem avaliados pelos usuários e possuírem as *features* necessárias para a análise de domínio. Outro recurso utilizado foi uma entrevista com um especialista de domínio, neste caso, um gerente imobiliário da Imobiliária X.

Os principais serviços que compõe a arquitetura de domínio, obtidos pela análise dos sistemas estudados e a entrevista com gerente, são:

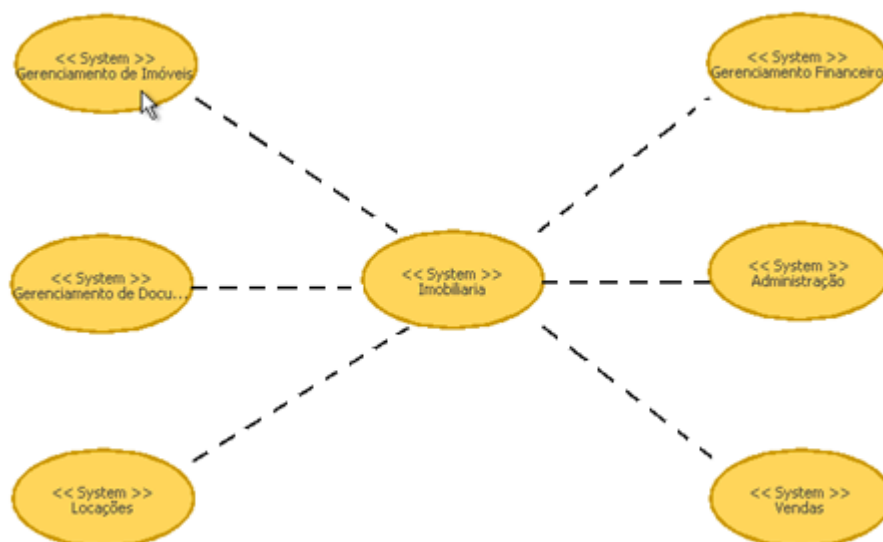
- **Locação:** trata-se do serviço de mediação de um contrato celebrado entre duas partes: o locador (proprietário do imóvel) e o locatário (cliente interessado no imóvel), onde o locador cede ao locatário a disponibilidade para utilização temporária de um imóvel mediante o pagamento de um valor financeiro periódico (aluguel) por um prazo determinado.

- Gerenciamento de documentação: faz-se necessária a apresentação e controle de vários documentos para a gestão do negócio imobiliário. Em uma locação é importante o registro de imóveis para comprovação do bem, documentos pessoais do locatário, tais como: carteira de identidade, CPF, certidão de casamento (se casado) ou certidão de nascimento (se solteiro), comprovante de rendimentos, três últimos recibos de aluguel, e dados de um fiador. Recibos, contratos, comprovantes, folhetos informativos, relatórios de vistorias, também são documentos necessários para a gestão imobiliária.
- Venda de imóveis: é o serviço prestado para intermediar a cessão da posse e a transferência de imóveis mediante o pagamento de um valor monetário. É necessário o registro de um contrato entre a imobiliária e o proprietário para que ela possua poderes administrativos para colocar placas ou faixas nos imóveis, agendar visitas com os futuros compradores e receber seus honorários referentes às transações.
- Gerenciamento dos imóveis: faz-se necessário efetuar o cadastro dos imóveis disponíveis para venda ou locação e informar seus dados importantes: área, localização, vagas de garagem, número de elevadores, quartos, banheiros, dependências e fotos do imóvel. Para um gerenciamento efetivo se deve cadastrar as obrigações contratuais e informações diversas referentes ao imóvel tais como: estado de conservação, comprovantes de pagamentos de impostos, condomínios, reformas efetuadas, vistorias, entre outros.
- Gerenciamento financeiro: quaisquer negociações efetuadas pela imobiliária geram movimentos financeiros que necessitam ser gerenciados por meio de registros em sistemas de contas a receber, e a pagar, boletos bancários, recibos, controle de repasse de valores, comissão sobre vendas, levantamentos estatísticos, etc.



- Administração: manter um registro de clientes que serão os compradores e locatários (inquilinos), um registro dos proprietários dos imóveis (vendedores, locadores), funcionários da imobiliária, registro dos fiadores, agenda, divulgação dos imóveis cadastrados, cadastro de corretores, são itens essenciais para o funcionamento da imobiliária. O gerenciamento desses itens e demais cadastros da imobiliária é mantido no contexto de administração do negócio.

A figura 4 ilustra o modelo de contexto formado pelos sistemas que compõem uma empresa do ramo imobiliário e os seus respectivos relacionamentos.



**Figura 4 - Diagrama de modelo de contexto**  
**Fonte: Autoria própria**

Gerenciamento de Imóveis, Gerenciamento Financeiro, Gerenciamento de Documentos, Administração, Locações e Vendas são os subsistemas que compõem o núcleo de um sistema de gerenciamento de imobiliária e estão ilustrados na figura 4.

### 3.1.2 Identificação das características

Esta fase contém as subfases de requisitos, modelagem, projeto, implementação e testes do domínio. Neste trabalho foram modelados os Sistemas de Venda e Locação por serem os mais utilizados no ramo imobiliário. A seguir descrevem-se os artefatos gerados em cada subfase.

### 3.1.2.1 Requisitos do domínio

Descrevem-se a seguir os Sistemas de Venda e Locação presentes nos softwares Imobilis 2013 (IMOBILIS, 2013), Ci-Pro (CI-PRO, 2013) e entrevista com um gerente da Imobiliária X da cidade de Ponta Grossa, os quais foram base para o desenvolvimento do estudo de caso de gerenciador de imóveis baseado em linhas de produtos de software.

## SISTEMA DE VENDAS

Um sistema de vendas de imóveis é um subsistema que possibilita ao agente imobiliário administrar o serviço prestado para a venda (cessão da posse e transferência) de imóveis. O funcionamento do sistema de vendas no IMOBILIS, CI-PRO e empresa X estão detalhados a seguir.

### a) IMOBILIS 2013

Os elementos necessários para a realização da venda de um imóvel são: cadastro de imóveis, clientes, proprietários, pesquisa de imóveis, cadastro de texto padrão, venda e, opcionalmente, um controle digital de armazenamento de documentos – o arquivo digital e atalho para buscar no sítio o endereço do imóvel e acesso rápido ao cadastro do mesmo.

O *Cadastro de Imóveis* é realizado ao preencher detalhes tais como: área de construção, localização, vagas disponíveis, nº de elevadores e cômodos do imóvel. A figura 5 ilustra o cadastro do imóvel, na qual se devem preencher as informações necessárias, o controle do *status* do imóvel é realizado pela mudança da operação de vendido ou para disponível.

**Figura 5 - Tela para cadastrar os imóveis**  
**Fonte: IMOBILIS (2013)**

O *Cadastro de Clientes* é feito para registrar todas as informações dos clientes que tem interesse em um imóvel disponível para venda. Na figura 6 é exibida uma ficha cadastral para preenchimento de nome, data de ligação, endereço, telefone, entre outros. Ainda, localizam-se os campos para o atendente informar: interesses do cliente, tipos de imóveis, valores e redigir um breve resumo da conversa.

**Figura 6 - Tela para cadastrar clientes**  
**Fonte: IMOBILIS (2013)**

O *Cadastro de Proprietários* é o recurso usado para cadastrar os proprietários dos imóveis no sistema de gerenciamento de imobiliárias e é ilustrado na figura 7. O sistema apresenta os campos para incluir nome do proprietário, endereço, telefones para contato, dados profissionais e campo disponível para incluir anotações.

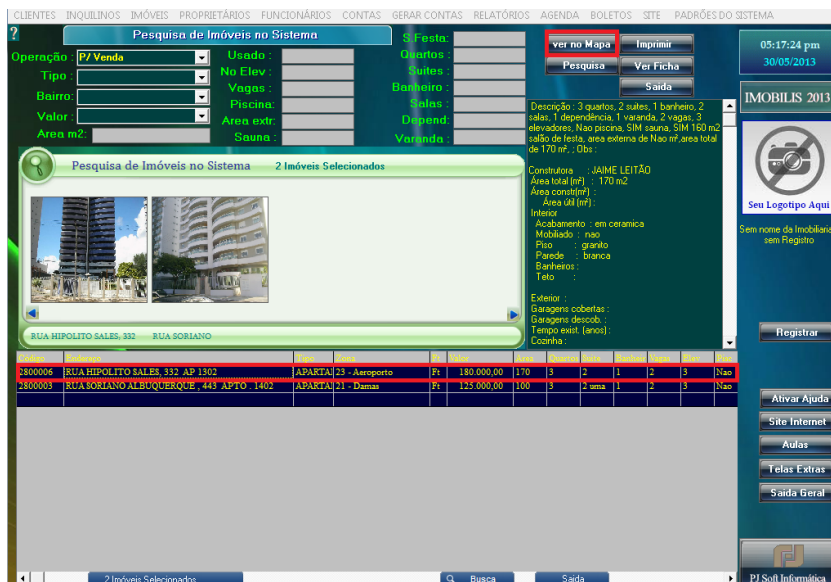
**Figura 7 - Tela para cadastrar os proprietários**  
Fonte: IMOBILIS (2013)

A *Pesquisa de Informações* permite que os possíveis clientes possam indicar as preferências em relação a um imóvel. Com estas informações, o atendente da imobiliária realiza a busca no sistema de acordo com as preferências dos clientes, como ilustra a figura 8.

ID	Endereço	Tipo	Status	Preço	Área	Quartos	Banheiros	Salas	Varandas	Outros
2800006	RUA HIPOLITO SALES, 332 AP 1302	APARTA	23 - Apto	R\$ 180.000,00	170	3	2	1	2	3
2800009	RUA SORIANO ALBUQUERQUE, 443 APTO. 1402	APARTA	21 - Dupla	R\$ 135.000,00	100	3	2	1	2	3

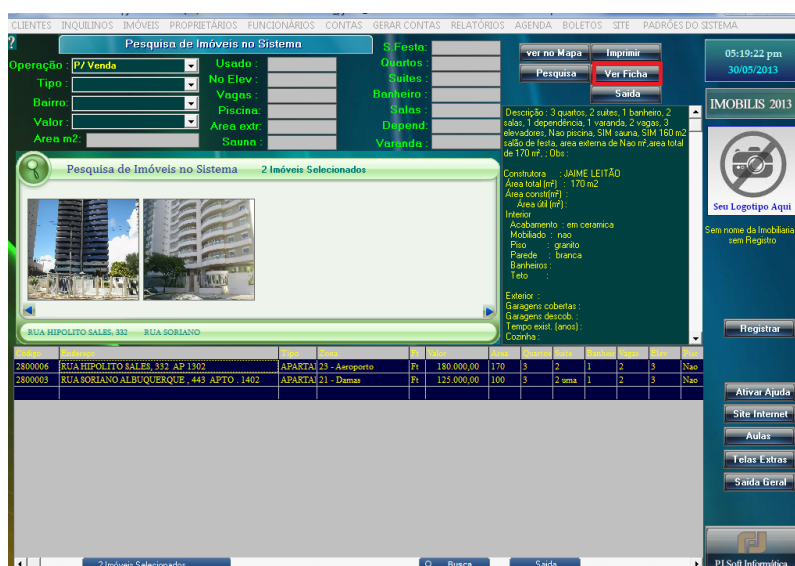
**Figura 8 - Tela para realizar busca no sistema**  
Fonte: IMOBILIS (2013)

Ao realizar a busca no sistema, selecionar o imóvel desejado, e clicar no botão de *Ver no mapa*, como ilustra a figura 9, o sistema abre uma página com o *link* do *maps.google.com*, e mostra no sítio o mapa do endereço do imóvel selecionado no sistema.



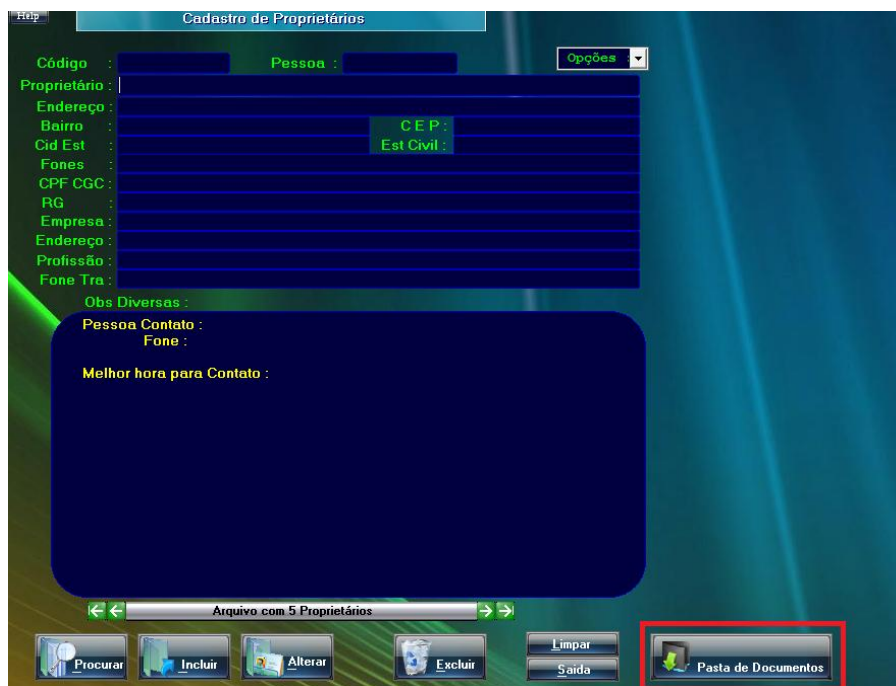
**Figura 9 - Tela que direciona para o site mostrando o endereço selecionado**  
Fonte: IMOBILIS (2013)

Ao executar a *Pesquisa de Informações*, tem-se a opção de clicar no botão de *Ver ficha*, onde é aberto o cadastro do imóvel selecionado, como ilustra a figura 10. O usuário do sistema pode verificar todas as informações cadastradas do imóvel.



**Figura 10 - Tela para buscar um imóvel e acessar o cadastro do imóvel selecionado**  
Fonte: IMOBILIS (2013)

Como ilustra a figura 11, o sistema disponibiliza uma pasta para salvar em arquivo os documentos dos clientes, proprietários e imóveis. Estes podem ser guardados em mídia removível, o que economiza o consumo de papéis. Esta pasta é criada junto com os arquivos do sistema e pode ser incluído no *backup* do sistema.



**Figura 11 - Pasta disponível para salvar os documentos necessários**  
**Fonte: IMOBILIS (2013)**

O *Cadastro de Textos Padrões* é utilizado para digitação de textos que podem ser utilizados para manter contratos salvos de compra e recibos de pagamentos, que podem ser editados e/ou impressos.

A figura 12 ilustra a interface gráfica que fornece o campo para a digitação dos textos, selecionar a fonte desejada e o tamanho. Pode-se também salvar no computador local, editar e imprimir, quando for necessário. Para incluir novo texto é necessário clicar em Limpar texto, digitar o texto desejado e pressionar em Salvar texto. Para imprimir o texto, clica-se no botão de Imprimir, e selecionar a impressora.

**Editor de Contratos e Textos**

**Inquilino:** Inquilino não Selecionado.      **Prop.:** EDGAR JILSON DE ALMEIDA  
**Imóvel:** Imóvel não Selecionado, primeiro consulte ou inclua o imóvel.

**Nome do Arquivo de Texto Padrão:** Contrato de Locação Residencial - tipo 1

**Padrões e Siglas**

**CONTRATO DE LOCAÇÃO**

Pelo presente instrumento particular, entre as partes adiante anunciado, devidamente qualificados e no final assinados fica justo e contratado:

**Locador(es):** [NOME - PROP] , brasileiro(a), [estado civil - prop], [profissão - prop], RG nº [rg - prop] , CPF/MF nº [cpf - prop] , residente e domiciliado na [endereco - prop] , [bairro - prop] - na cidade de [cidade estado - prop] .

**Locatário(s):** [NOME - INQ] , brasileiro(a), [estado civil - inq], [profissão - inq], RG nº [rg - inq], CPF/MF nº [cpf - inq], residente e domiciliado na [endereco - inq] - [bairro - inq] - na cidade de [cidade estado - inq].

**Fiador(es):**  
[dados fiadores - inq]

**Objeto de Locação:** Imóvel RESIDENCIAL na [endereco - imóvel], na cidade de [cidade estado - imóvel].

5 - Valor Mensal da Locação - [valuguel - imóvel] ( [extenso aluguel - imóvel] ). Devendo o pagamento ser efetuado até o dia [dia venc. - imóvel] ( [extenso venc. - imóvel] ) de cada mês. Haverá um Reajuste anual de acordo com índice oficial do governo (IGPM - FGV), ou outro índice que o Governo Federal determinar para as locações.

Após o vencimento será acrescido multa de 10% , juros de 1% ao mês , mais correção monetária.

6 - Prazo de locação - [tipo contrato - imóvel]    Início: / /2004 - Término: / /2005 .

7 - Finda a locação, o(s) locatário(s) fará(ão) a devolução do imóvel locado, nas mesmas condições em que o recebeu(ram), sem necessidade de prévia interpelação ou notificação;

8 - O aluguel mensal é o indicado no item 5 deste pacto, devendo o seu pagamento ser efetuado ao representante legal do(s) locador(es), ou seja, na [nome da imobiliaria], [endereco da imobiliaria], [cidade padrão].

\* Dados do Inquilino  
código - inq  
tipo pessoa - inq  
nome - inq  
endereco - inq  
bairro - inq  
cep - inq  
cidade estado - inq  
estado civil - inq  
fones - inq  
cpf cgc - inq  
rg - inq  
empresa trab - inq  
endereco trab - inq  
profissão - inq  
fone trabalho - inq  
observações - inq  
dados fiadores - inq  
dados captador - inq

\* Dados do Proprietário  
codigo - prop  
tipo de pessoa - prop  
nome - prop  
endereco - prop  
bairro - prop  
cep - prop  
cidade - prop  
estado civil - prop  
fones - prop  
cpf cgc - prop  
rg - prop  
empresa trab. - prop  
endereco prof. - prop  
profissão - prop  
fone trabalho - prop  
observações - prop

**Dados em maiusculo**

**Figura 12 - Tela para cadastrar um texto padrão**  
**Fonte: IMOBILIS (2013)**

## b) CI-PRO

Os elementos necessários para a realização da venda de um imóvel são: cadastro de imóveis, clientes, proprietários, pesquisa de imóveis, venda e cadastro de textos padrões.

O *Cadastro de Imóveis* é dividido entre: imóveis diversos, comerciais, rurais e terrenos. Nestes cadastros ocorre uma diferenciação na aba detalhamento, onde as benfeitorias são marcadas. No caso do imóvel comercial e industrial os detalhes incluem: descrição do terreno, pátio, estacionamento, dependências e área fabril. A opção de imóveis rurais inclui: equipamentos, descrição da casa sede e dependências, silos, cercados e terreno. Para cadastro de imóveis diversas informações são específicas para uso residencial ou predial, e descrever as dimensões de pisos e dormitórios, como ilustra a figura 13.

**Figura 13 - Tela para incluir um novo imóvel**  
**Fonte: Ci-Pro (2013)**

O *Cadastro de Clientes* não possui cadastros separados para cliente e proprietário. A diferenciação é feita pelo *checkbox* da tela de cadastro de clientes, como ilustra a figura 14. Os campos de nome, endereço, telefones, documentação são semelhantes para estes clientes: Locatário e Locador. Na aba superior tem a distribuição das informações, para não exibir um excesso de informações em uma única tela.

**Figura 14 - Tela para incluir um novo cliente**  
**Fonte: Ci-Pro (2013)**

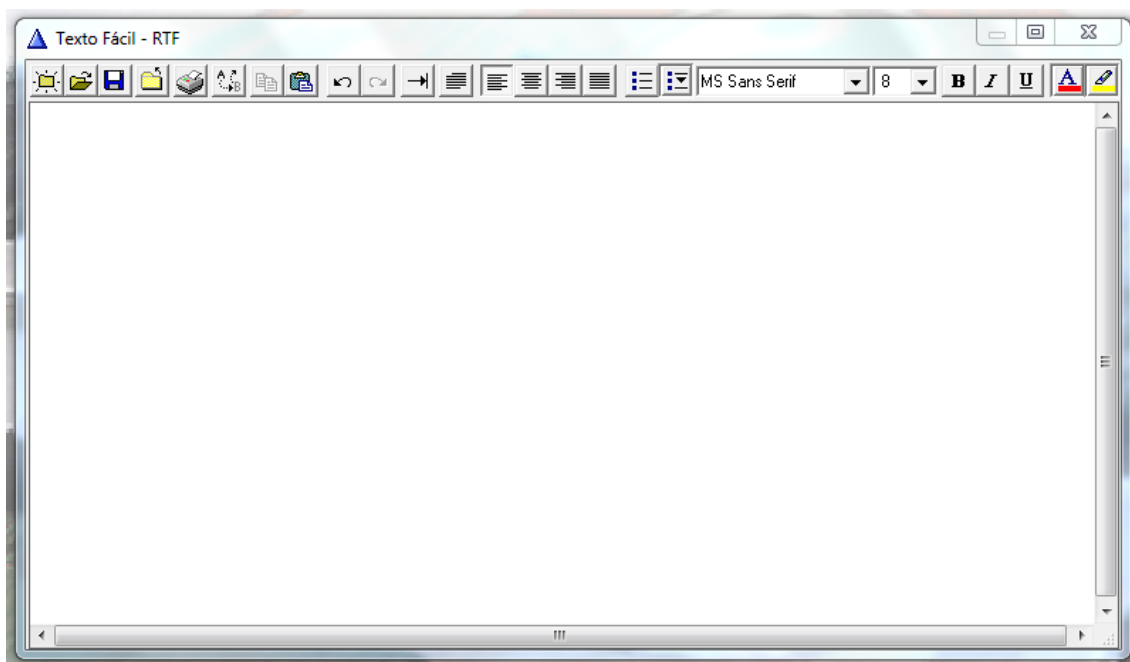


A *Pesquisa de Informações* é feita por um filtro para selecionar os imóveis que estão disponíveis para venda. Esse campo de busca torna-se visível na tela inicial de todos os cadastros disponíveis no sistema, como mostra a figura 15. O sistema apresenta algumas opções de filtro, tais como: disponíveis, reservados ou mostrar todos. Ao selecionar uma das opções a pesquisa é realizada.

S	ID	D	P	A	U	L	Tipo «	Bairro ««	Operação ««		Edifício	And	AC/AP	Terr	M2 Lc	M2 Vd		
									Lc	Vd								
	1	●	●	●	●	●	APTO DUPLEX	Centro	L	1.500	V	200.000	Barao Davi	15	150	260	10,00	1,333
	3	●	●	●	●	●	APTO LOFT	ELDORADO			V	450.000	Conde Tuba	17	250	400		1,800
	2	●	●	●	●	●	CASA TERREA	ELDORADO	L	3.450	V	350.000			210	350	16,43	1,667

**Figura 15 - Tela de busca entre os imóveis disponíveis dos reservados**  
**Fonte: Ci-Pro (2013)**

O Cadastro de Textos Padrões tem como opção a digitação de textos que podem ser utilizados para manter exemplos de contratos salvos de compra e venda e recibos de pagamentos, que pode ser editado e/ou impresso. Como ilustra a figura 16, o sistema fornece o campo para a digitação dos textos, selecionar a fonte desejada e o tamanho. O texto criado pode ser salvo no computador local, editado ou impresso quando for necessário. Na entrevista realizada, o gerente da Imobiliária x, informou que os textos padrões são modelos criados em editores de texto, salvos em pasta compartilhada na rede local, que podem ser editados e impressos no momento da negociação.



**Figura 16 - Tela para cadastrar os textos necessários**  
**Fonte: Ci-Pro (2013)**

c) Imobiliária X

O cadastro dos imóveis é realizado pela apresentação da documentação obrigatória, como por exemplo, o registro do imóvel, escritura, certidão negativa de débitos.

O cadastro do proprietário do imóvel é realizado ao preencher o endereço, telefone para contatos e e-mail.

Para a realização de cadastro de novos clientes, os documentos necessários são: RG, CPF, comprovante de endereço e comprovante de renda.

Os formulários para preenchimento das informações do imóvel, proprietários e clientes estão disponíveis no site da Imobiliária X. Podem ser impressos, preenchidos e entregues com a cópia dos documentos solicitados para cadastro do imóvel, proprietários e clientes interessados na compra do imóvel. Essas informações fornecidas à imobiliária são repassadas para o sistema interno da mesma.

A pesquisa das informações é realizada em catálogos dos imóveis, que é online, e realiza a pesquisa no próprio site da imobiliária X ou impresso. As fotos e

as informações cadastrais são ordenadas por região da cidade, para facilitar a busca pela informação. A pesquisa é realizada também no sistema interno.

Para facilitar a assinatura dos contratos e recibos de pagamento, há uma pasta compartilhada na rede local com os modelos a serem utilizados. Para uso da imobiliária, abre-se o arquivo em editor de texto e realizam-se as alterações necessárias.

Para formalizar a venda do imóvel, é realizada a assinatura dos contratos entre as partes envolvidas, proprietário e cliente. A imobiliária efetiva esse intermédio, validando as informações dos documentos, tributos e legislação.

Após realizar a análise descritiva do sistema de vendas, identificam-se para cada uma de suas funcionalidades se são similares ou variáveis. O quadro 4 apresenta os pontos similares (S) e de variabilidade (V) entre o sistema Imobilis 2013, Ci-PRO e Imobiliária X.

**Quadro 4 - Comparativo de similaridades e variabilidades dos sistemas de vendas**

Funcionalidade	Imobilis 2013	Ci-Pro	Imobiliária X
Cadastro de imóveis	S	S	S
Arquivo digital	V		
Cadastro de clientes	S	S	S
Cadastro de proprietários	S	S	S
Pesquisa avançada	S	S	S
Cadastro de texto padrão (contrato)	S	S	S
Ver no mapa	V		
Ver ficha	V		

**Fonte: Autoria própria**

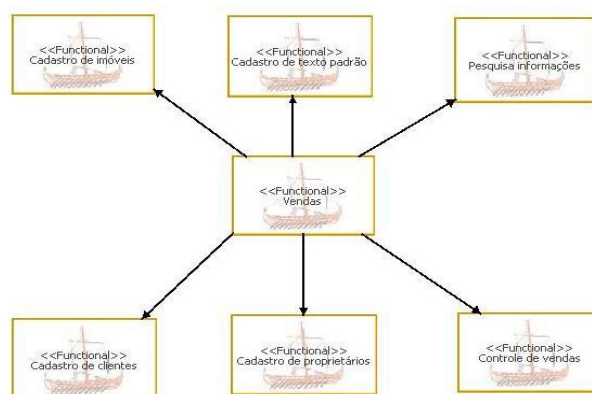
De acordo com o quadro 4 tem-se que:

- Cadastro de imóveis: ambos os sistemas e na entrevista possuem este cadastro, sendo que os atributos similares são: tipo do imóvel, informações do proprietário, localização do imóvel, valores de venda, quantidade de cômodos, situação do imóvel (disponível, vendido) e

informações referentes ao anúncio e propaganda do imóvel. Portanto, podem constituir um núcleo para todos os aplicativos.

- Cadastro de clientes: nos sistemas analisados e na entrevista com um funcionário da empresa X, o cadastro de clientes é uma funcionalidade obrigatória. Atributos similares são: nome, endereço, telefones pessoais e para contato e e-mail. Desta forma, pode constituir um núcleo um sistema imobiliário.
- Cadastro de proprietários: ambos os sistemas, inclusive na empresa X, possuem cadastro para proprietários. Atributos similares são: nome, endereço, CPF, RG, dados profissionais, telefone e estado civil. Logo, é núcleo para um sistema imobiliário.
- Cadastro de texto padrão: este também constitui um núcleo para o sistema analisado. Um atributo similar identificado foi a descrição do texto.
- Pesquisa avançada: em ambos os sistemas e na empresa X foi possível constatar que se realizam buscas nos cadastros. Isto constitui também um núcleo para o sistema.

A figura 17 ilustra as *features* identificadas em todos os sistemas de vendas analisados e caracteriza as suas similaridades já descritas anteriormente.



**Figura 17 - Diagrama de *features* do sistema de vendas**  
Fonte: autoria própria

Após a identificação das similaridades do Sistema de Venda, analisaram-se os requisitos do Sistema de Locação, descritos a seguir.

## SISTEMA DE LOCAÇÃO

Um sistema de locação de imóveis é um subsistema que possibilita ao agente imobiliário gerenciar o serviço de mediação de um contrato firmado entre o locador (proprietário do imóvel) e o locatário (cliente/inquilino) na locação de imóveis.

Os subsistemas de Cadastro de Clientes, Cadastro de Imóveis, Cadastro de Proprietários, Arquivo Digital, Cadastro de Texto Padrão e Pesquisas de Informações, Ver no mapa, Ver ficha, presentes no Sistema de Venda, também fazem parte do Sistema de Locação. Portanto, descrevem-se nos próximos itens somente as diferenças existentes nessas opções.

### a) IMOBILIS 2013

Os elementos necessários para a realização de uma locação de um imóvel são: Cadastro de Imóveis, Inquilinos, Proprietários, Pesquisa de imóveis e Locação.

O *Cadastro de Imóveis* é semelhante ao cadastro do sistema de vendas. O controle do *status* do imóvel é realizado pela mudança da operação de alugado para disponível, como ilustrou a figura 5.

O *Cadastro de Inquilinos* é realizado pelo preenchimento de uma ficha cadastral completa, ao preencher informações como: nome, endereço do imóvel locado, informações pessoais, documentação, dados profissionais e dados referentes ao imóvel locado, tais como: valor do aluguel, data de vencimento e do reajuste anual. Ao utilizar o botão de Recibos e Contratos, o usuário do sistema pode redigir os contratos, recibos e rescisões, e efetuar assim o *Cadastro de textos padrões*, como citado no sistema de vendas. Na opção *Fiador*, pode-se cadastrar o fiador do inquilino. Nesse cadastro, é permitido colocar as informações pessoais, profissionais e as responsabilidades legais que serão atribuídas em caso de falta de pagamento ou destruição do bem locado, e realizar o *Cadastro do fiador*, como ilustra a figura 18.



**Figura 18 - Tela para cadastrar os inquilinos e as obrigações contratuais**  
**Fonte: IMOBILIS (2013)**

A Ferramenta de Pesquisa do sistema de locações é o mesmo presente no sistema de vendas e já foi descrita e detalhada na figura 8.

O Cadastro de Proprietários é compartilhado com o Sistema de Vendas que já foi anteriormente relatado e exibido na figura 7.

A opção de Arquivo digital também é compartilhada entre os dois subsistemas e já foi apresentada e ilustrada na figura 11.

A opção de Ver no Mapa e Ver Ficha foram detalhadas no sistema de vendas e exibidas nas figuras 9 e 10, respectivamente.

#### b) CI-PRO

Os elementos necessários para a realização de uma locação de um imóvel no sistema CI-PRO (CI-PRO, 2013) são: cadastro de imóveis, inquilinos, proprietários, pesquisa de imóveis, cadastro de textos padrões e locação. Para formalizar o contrato de locação, é necessário cadastrar um fiador, através da opção *Cadastro de Fiador*. Nessa opção colocam-se as informações pessoais, profissionais

e as responsabilidades legais a serem atribuídas em caso de falta de pagamento e/ou destruição do bem locado. Sendo que o mesmo é o responsável legal pelo imóvel.

O Cadastro de Imóveis, locatários e locador, pesquisa de imóveis e cadastro de textos padrões são funcionalidades compartilhadas com o Sistema de Vendas, e já foram descritos e exibidos nas figuras 13, 14, 15 e 16.

#### c) Imobiliária X

O cadastro dos imóveis, proprietários, inquilinos (clientes), pesquisa das informações, cadastro de texto padrão é semelhante ao sistema de vendas, como já citado anteriormente. O cadastro do fiador é realizado com as informações obrigatórias, como: RG, CPF, comprovante de endereço, comprovante de renda mensal. Ao assinar o contrato, o fiador torna-se o responsável legal pelo pagamento mensal da locação. Outra forma de fiador é denominada caução, onde o inquilino adianta para a imobiliária o valor referente a 3 (três) à 4 (quatro) pagamentos mensais da locação que após encerramento do contrato este valor é reembolsado ao inquilino.

Para formalizar a locação do imóvel, é realizada a assinatura dos contratos entre as partes envolvidas, proprietário e inquilino. A imobiliária realiza esse intermédio, ao validar as informações dos documentos, tributos e legislação.

Após realizar a análise descritiva do sistema de locação, foram identificadas similaridades e variabilidades entre as funcionalidades. O quadro 5 apresenta os pontos similares (S) e de variabilidade (V) entre o sistema Imobilis 2013, Ci-PRO e Imobiliária X.

**Quadro 5 - Comparativo de similaridades e variabilidades do sistema de locações**

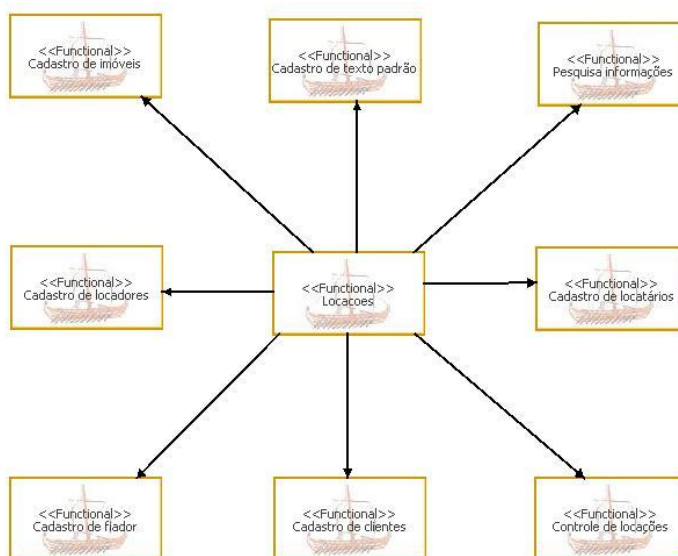
Funcionalidade	Imobilis 2013	CI-PRO	Imobiliária X
Cadastro de imóveis	S	S	S
Arquivo digital	V		
Cadastro de locatários	S	S	S
Cadastro de locadores	S	S	S
Pesquisa avançada	S	S	S
Cadastro de texto padrão (contrato)	S	S	S
Informações do fiador	S	S	S
Ver no mapa	V		
Ver ficha	V		

**Fonte: Autoria própria**

De acordo com o quadro 5 tem-se que:

- Cadastro de imóveis: identificado em todas as aplicações analisadas e na Imobiliária X. Os atributos similares desse cadastro são os mesmos descritos no sistema de vendas, portanto, podem constituir o núcleo.
- Cadastro de clientes: os sistemas possuem cadastro para proprietário (locador) e inquilino (locatário). Atributos similares são: RG, CPF, endereço, informações profissionais, telefones pessoais e para contato e estado civil, os quais podem ser adaptados e construídos os núcleos para o sistema imobiliário.
- Pesquisa avançada: nos sistemas é possível realizar as buscas nos cadastros diversos. Como relatado no Sistema de Vendas.
- Cadastro de textos padrões (contratos): os sistemas possuem essa ferramenta para cadastro e edição de contratos, o que permite a construção dos núcleos com as similaridades para os aplicativos.
- Cadastro de fiador: os sistemas possuem cadastro para fiador. Atributos similares são: Nome, CPF, RG, endereço e informações profissionais. Aos quais podem ser adaptados e construídos os.





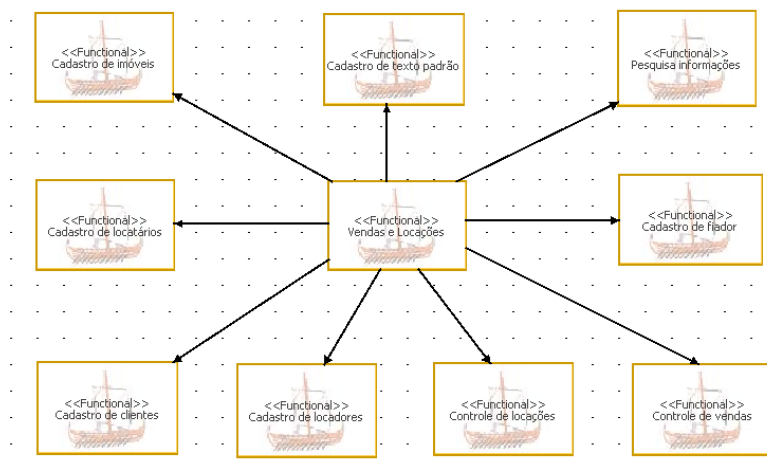
**Figura 19 - Diagrama de features do Sistema de Locações**  
**Fonte: Autoria própria**

Após realizar a análise descritiva dos Sistemas de Locação e Vendas, identificam-se algumas similaridades (S) e uma variabilidade (V) entre estes sistemas, apresentados no quadro 6.

**Quadro 6 - Comparativo entre o sistema de locação e venda**

Funcionalidade	Imobilis 2013	CI-PRO	Imobiliária X
Cadastro de imóveis	S	S	S
Arquivo digital	V		
Cadastro de locatários	S	S	S
Cadastro de locadores	S	S	S
Pesquisa avançada	S	S	S
Cadastro de texto padrão (contrato)	S	S	S
Informações do fiador	S	S	S
Cadastro de clientes	S	S	S
Cadastro de proprietários	S	S	S
Ver no mapa	V		
Ver ficha	V		

**Fonte: Autoria própria**

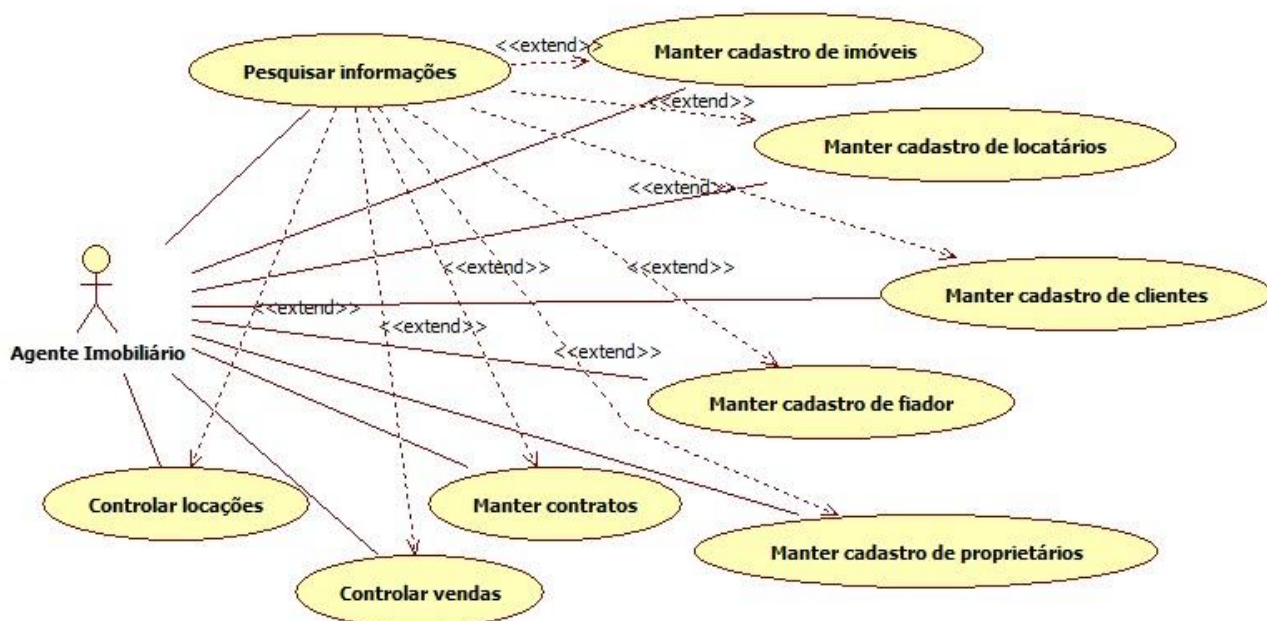


**Figura 20 - Diagrama de features para os Sistemas de Locação e Venda**  
**Fonte: Autoria própria**

### 3.1.2.2 Modelagem de Domínio

Nesta fase é feita a modelagem do sistema, na qual os diagramas são os meios utilizados para a visualização de um sistema sobre diferentes perspectivas (BOOCH et al., 2000). Nesse caso, utiliza-se o diagrama de caso de uso, pois é aplicado para captar o comportamento pretendido pelo sistema que está em desenvolvimento, sem ser necessário especificar como esse comportamento é implementado, além de tornar o problema em questão de fácil entendimento.

A figura 21 ilustra o diagrama de caso de uso para as similaridades dos Sistemas de Vendas e Locações. As similaridades foram obtidas da etapa de requisitos do domínio.



**Figura 21 - Diagrama de caso de uso com as similaridades do Sistema de Venda e Locações**  
**Fonte: Autoria própria**

Os cenários para os casos de uso ilustrados na figura 21 estão descritos nos quadros 7, 8, 9 e 10. Como a descrição das funcionalidades de casos de uso é similar, os mesmos estão no Apêndice A.

O quadro 7 descreve o cenário de Manter cadastro de imóveis que é responsável por incluir, editar e excluir os imóveis do sistema.

**Quadro 7- Descrição do cenário de Manter cadastro de imóveis**

<b>Use Case</b>	Manter cadastro de imóveis
<b>Descrição</b>	Possibilita incluir, alterar, exibir e excluir os imóveis.
<b>Pré-Condições</b>	Dados do imóvel
<b>Fluxo Básico</b>	
<b>Ação do Ator</b>	<b>Resposta do Sistema</b>
1. Escolhe uma opção	2. Executa a operação escolhida. <b>[A1], [A2], [A3] e [A4]</b>
<b>Fluxo Alternativo</b>	
<p><b>[A1] – Incluir Imóvel</b>  1. Os dados serão registrados e uma mensagem de sucesso será exibida ao usuário. <b>[A5]</b>  2. Retorna ao passo 1.</p> <p><b>[A2] – Alterar Imóvel</b>  1. Os dados serão alterados e uma mensagem de sucesso será exibida ao usuário. <b>[A5]</b>  2. Retorna ao passo 1.</p> <p><b>[A3] – Excluir Imóvel</b>  1. Os dados serão excluídos e uma mensagem de sucesso será exibida ao usuário.  2. Retorna ao passo 1.</p> <p><b>[A4] – Exibir Imóvel</b>  1. Requisita o caso de uso “Pesquisar Informações”.  2. Exibe os dados do Imóvel. <b>[A6]</b>  3. Retorna ao passo 1.</p> <p><b>[A5] – Dados não registrados</b>  1. Informa que os dados não foram registrados e encerra o caso de uso.</p> <p><b>[A6] – Informação não exibida</b>  1. O dado solicitado não foi encontrado e uma mensagem será exibida.  2. Retorna ao passo 1.</p>	
<b>Pós-Condições</b>	Imóvel cadastrado, editado, exibido ou excluído do sistema.

**Fonte: Autoria própria**

O quadro 8 descreve o cenário de Controlar locações que permite acompanhar as movimentações ocorridas durante a vigência do contrato e realizar um locação imobiliária.

**Quadro 8 - Descrição para o cenário de Controlar locações**

<b>Use Case</b>	Controlar locações
<b>Descrição</b>	Permite acompanhar e realizar as transações da locação
<b>Pré-Condições</b>	Dados do imóvel, do locador, fiador e locatários cadastrados.
<b>Fluxo Básico</b>	
<b>Ação do Ator</b>	<b>Resposta do Sistema</b>
1.Usuário escolhe a opção de locação. 3.Usuário escolhe “Buscar imóvel”. 6.Usuário escolhe “Buscar locatário”. 9.Usuário escolhe “Buscar fiador”.	2.Abre a tela para incluir as informações da locação. 4.Requisita o caso de uso “Pesquisar Informações”. <b>[A1]</b> . 5.Exibe imóvel selecionado e o locador do imóvel. 7. Requisita o caso de uso “Pesquisar Informações” <b>[A3]</b> . 8. Exibe locatário selecionado. 10. Requisita o caso de uso “Pesquisar Informações” <b>[A5]</b> . 11. Exibe o fiador selecionado.
<b>Fluxo Alternativo</b>	
<p><b>[A1]</b> – Informação não exibida</p> <p>1.O dado solicitado não foi encontrado e uma mensagem será exibida. <b>[A2]</b></p> <p><b>[A2]</b> – Retorna para buscar imóvel.</p> <p>1.Retorna ao passo 3.</p> <p><b>[A3]</b> – Informação não exibida</p> <p>1.O dado solicitado não foi encontrado e uma mensagem será exibida. <b>[A4]</b></p> <p><b>[A4]</b> – Retorna para buscar locatário.</p> <p>1. Retorna ao passo 3.</p> <p><b>[A5]</b> – Informação não exibida</p> <p>1. O dado solicitado não foi encontrado e uma mensagem será exibida. <b>[A6]</b></p> <p><b>[A6]</b> – Retorna para buscar fiador.</p> <p>1. Retorna ao passo 9.</p>	
<b>Pós-Condições</b>	Informações da locação cadastrada.

**Fonte: Autoria própria**

O quadro 9 descreve o cenário de Controlar vendas que permite acompanhar as movimentações ocorridas no período de efetivar o contrato de compra e venda e controlar movimentação de documentação.

**Quadro 9 - Descrição do cenário de Controlar vendas**

<b>Use Case</b>	Controlar vendas
<b>Descrição</b>	Permite acompanhar e realizar as transações de vendas.
<b>Pré-Condições</b>	Dados do imóvel, de proprietários e clientes cadastrados.
<b>Fluxo Básico</b>	
<b>Ação do Ator</b>	<b>Resposta do Sistema</b>
1.Usuário escolhe a opção de locação. 3.Usuário escolhe “Buscar imóvel”. 6.Usuário escolhe “Buscar cliente”.	2.Abre a tela para incluir as informações da locação. 4.Requisita o caso de uso “Pesquisar Informações”. <b>[A1]</b> . 5.Exibe imóvel selecionado e o proprietário do imóvel. 7. Requisita o caso de uso “Pesquisar Informações” <b>[A3]</b> . 8. Exibe cliente selecionado.
<b>Fluxo Alternativo</b>	
<b>[A1]</b> – Informação não exibida 1. O dado solicitado não foi encontrado e uma mensagem será exibida. <b>[A2]</b>	
<b>[A2]</b> – Retorna para buscar imóvel. 1. Retorna ao passo 3.	
<b>[A3]</b> – Informação não exibida 1. O dado solicitado não foi encontrado e uma mensagem será exibida. <b>[A4]</b>	
<b>[A4]</b> – Retorna para buscar cliente. 1. Retorna ao passo 6.	
<b>Pós-Condições</b>	Informações do imóvel atualizado.

**Fonte: Autoria própria**

O quadro 10 descreve o cenário de Pesquisar informações que é responsável realizar buscar à base de dados e exibi-las em tela.

**Quadro 10 - Descrição do cenário de Pesquisar informações**

<b>Use Case</b>	Pesquisar informações
<b>Descrição</b>	Permite realizar a busca de informações no sistema.
<b>Pré-Condições</b>	Informações cadastradas na base de dados de imóveis, clientes, proprietários, fiador, endereço, locatários e locadores.
<b>Fluxo Básico</b>	
<b>Ação do Ator</b>	<b>Resposta do Sistema</b>
	1.Busca na base de dados a informação a ser pesquisada. <b>[A1]</b> . 2.Retorna a informação pesquisada.
<b>Fluxo Alternativo</b>	
<b>[A1]</b> – Informação não retornada 1. O dado solicitado não foi encontrado e uma mensagem será exibida. 2. Encerra caso de uso.	
<b>Pós-Condições</b>	Pesquisa da informação realizada.

**Fonte: Autoria própria**

### 3.1.2.3 Projeto do Domínio

Duas camadas do processo de desenvolvimento *UML Components* (SILVA, 2011) foram escolhidas para o projeto de arquitetura dessa fase de desenvolvimento para definir os componentes do domínio:

- Camada de Sistema – agrupam-se os componentes que implementam as funcionalidades específicas;
- Camada de Negócio: cujos componentes delineiam as funcionalidades estáveis do tipo de negócio. Nesta camada é feita a definição da arquitetura do domínio e pode ser dividida em seis fases: *Especificação de requisitos, Especificação de componentes, Provisionamento dos componentes, Montagem do sistema, Testes e Implantação.*

Nessa fase deste trabalho serão detalhadas somente as fases de Especificação de Requisitos e de Componentes, pois o escopo definido desta etapa é o nível arquitetural.

#### Especificação de Requisitos

Os artefatos (*features*) gerados nessa fase foram os diagramas de Caso de Uso já relatados anteriormente. Estes foram utilizados para representar os requisitos funcionais do sistema, e posteriormente utilizados na fase de especificação de componentes.

#### Especificação de Componentes

Foram identificados os artefatos de entrada que são os produzidos na Especificação de requisitos de acordo com o processo *UML Components*. A especificação das interfaces, a especificação dos componentes e a arquitetura do sistema são os artefatos de saída produzidos nessa fase. Como foi estudada apenas a arquitetura do sistema, neste trabalho foi aplicada apenas a fase de Identificação de Componentes, das três subfases que compõe esta etapa: Identificação dos componentes; Interação entre os componentes; Especificação final.

### 3.1.2.4 Modelagem da Arquitetura

Foram aplicados os conceitos das fases de Especificação de requisitos e Especificação de componentes para uma correta compreensão da arquitetura do sistema.

### 3.1.2.5 Identificação dos Componentes

A especificação das interfaces do sistema, a identificação das interfaces de negócio e identificação dos componentes foram os artefatos produzidos nessa fase.

#### Especificação das Interfaces do Sistema

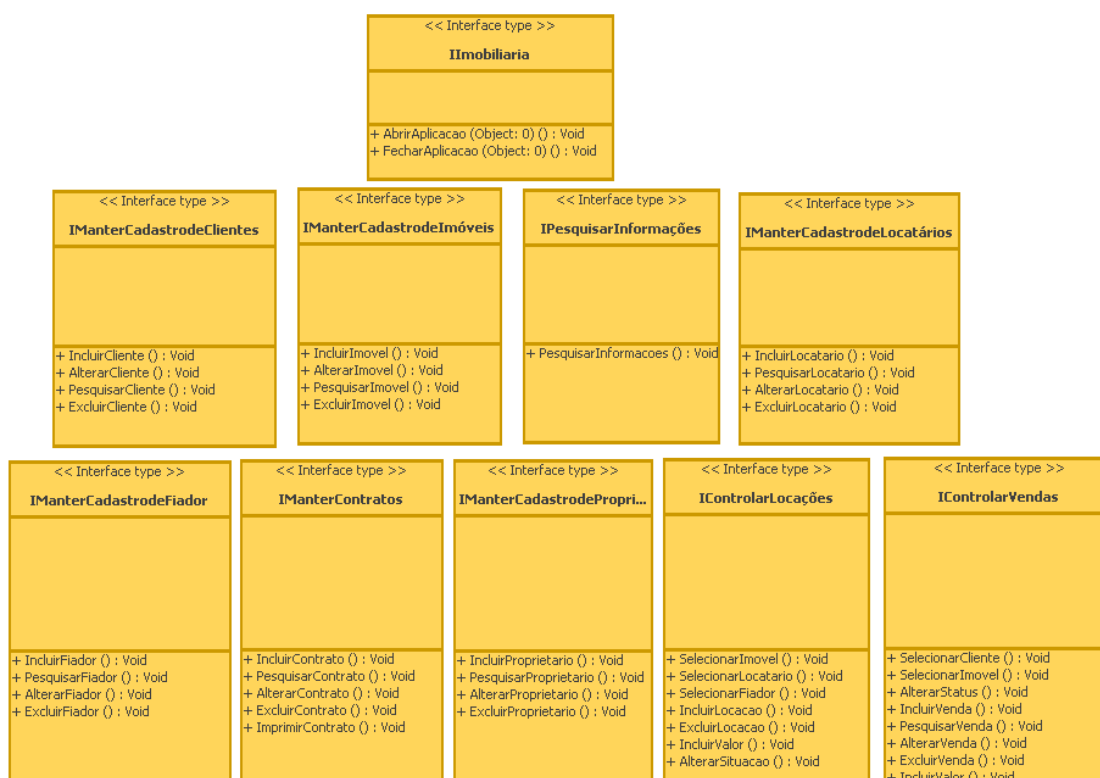
Cada Caso de Uso já analisado no domínio foi instanciado em uma interface da camada do sistema, conforme prevê o modelo UML *Components* e como ilustra a figura 22. Foi utilizado o Caso de Uso de Similaridades dos sistemas Imobilis 2013, Ci-PRo e Imobiliária X, ilustrado na figura 21.



**Figura 22- Interfaces do Software Imóveis dos casos de uso das similaridades**  
**Fonte: Autoria própria**

Na sequencia é exibido um novo Diagrama de Classes, que já contém as interfaces definidas no sistema, com os respectivos métodos que o compõe, conforme a figura 23.





**Figura 23 - Interfaces com os Métodos do Software Imobiliário dos casos de uso das similaridades.**  
**Fonte: Autoria Própria**

### Identificação das Interfaces de Negócio

Segundo Delazeri e Wolf Interfaces de Negócio são as abstrações das informações que o sistema deve gerenciar e podem ser identificadas pelas seguintes etapas:

1. Criar o modelo do tipo de negócio: este modelo deve conter informações específicas do negócio e que são guardadas pelo sistema especificado, incluindo detalhes dos atributos que constitui o modelo.
2. Identificar *Core Business Type*: neste são descritas as dependências das informações contidas no sistema.
3. Criar uma interface para os *core types* e incluí-los no modelo.
4. Fazer o refinamento do modelo de tipo de negócio e indicar as responsabilidades das interfaces. (Delazeri & Wolf, 2012).

As etapas “Criar o modelo das regras de negócio” e “Identificar *Core Business Type*” foram utilizadas para criar o diagrama exibido na figura 23. O modelo de tipo de negócio, as informações específicas, os atributos que compõe o domínio e o *Core Business Type*, foram definidos conforme indicado no primeiro e segundo passos.



**Figura 24 - Modelo de Tipo de Negócio para Imóveis**  
 Fonte: Autoria própria

No modelo de negócio exibido na figura 24 foi definido o *core type (Kernel)*, que também é o núcleo *do sistema* de Imóveis, e os outros negócios que possuem dependência e por isso são do tipo *type*.

Na sequência cria-se uma interface para o *core type* para adicionar-se ao modelo e indicar as suas responsabilidades:

- *Imóveis*: contém os cadastros de clientes, de imóveis, de fiadores, de proprietários, de locatários, pesquisar informações, controlar vendas, controlar locações, e controlar contratos.

Para finalizar essa fase de especificação das interfaces de negócio com as respectivas identificações das interfaces, com a identificação das respectivas responsabilidades e sua inclusão no modelo, conforme ilustra na figura 25.



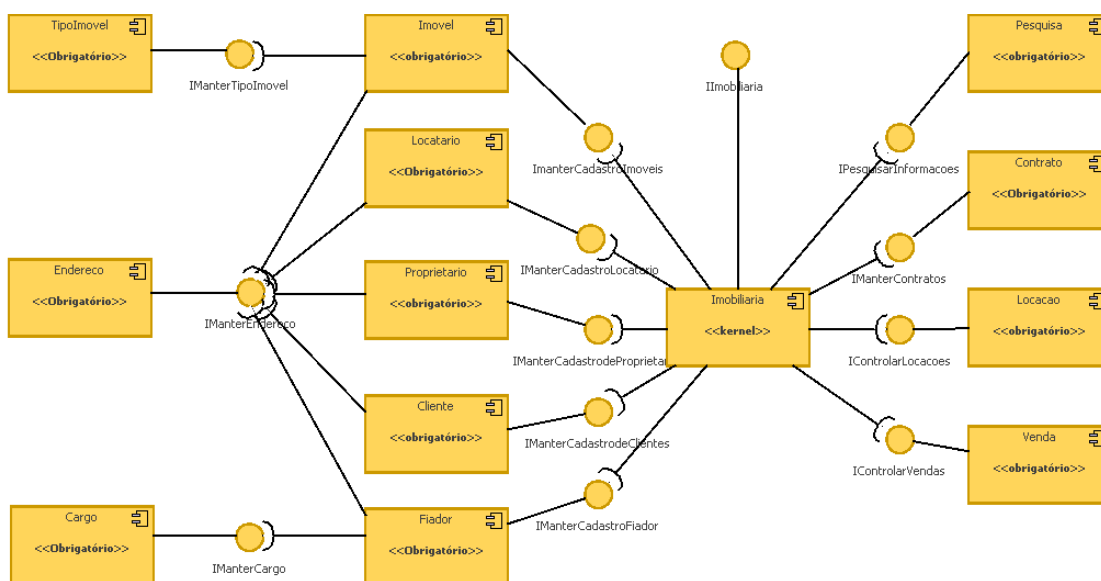
**Figura 25 Diagrama de responsabilidades das interfaces do modelo de tipo de negócio**  
**Fonte: Autoria própria**

## Identificação dos Componentes

Identificados os componentes que implementam as interfaces especificadas anteriormente nas subfases anteriores, associou-se um novo componente à cada interface, tanto de sistema quanto de negócio, seguindo o modelo de processo UML *Components*.

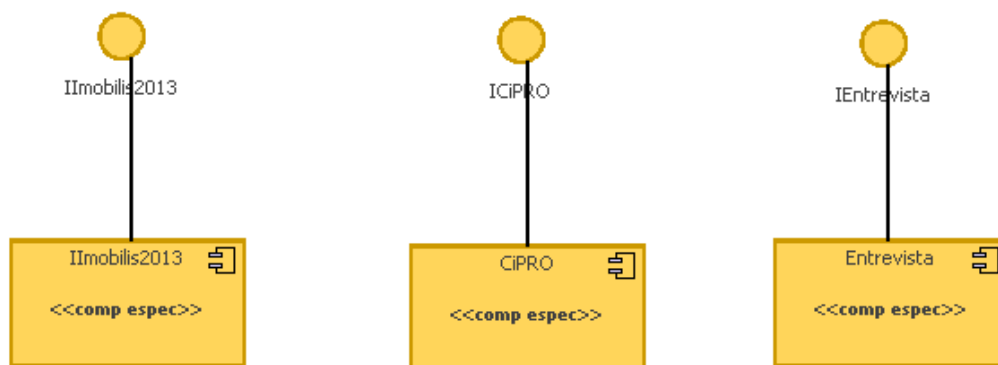
Para demonstrar a criação da camada Sistema foi utilizado-se o Caso de Uso das Similaridades exibido no respectivo diagrama (figura 21), onde se identificou as interfaces: *IManterCadastrodeProprietarios*, *IManterCadastrodeClientes*, *IManterCadastroImoveis*, *IControlarVendas*, *IControlarLocacoes*, *IManterCadastroFiador*, *IManterCadastroLocatario*, *IManterContratos* e *IPesquisarInformacoes*. Para cada interface identificada foi associou-se um respectivo componente: *CadastrodeProprietarios*, *CadastrodeClientes*, *Cadastrodelmoveis*, *ControlarVendas*, *ControlarLocacoes*, *CadastroFiador*, *CadastroLocatarios*, *Contratos* e *Informacoes*. Como é definido no contexto de linhas de produtos *ISoftwareImóveis* controla todas as operações do sistema, e desse modo, essa interface deve estar associada a todas as outras interfaces que fazem operações internas do sistema.

A primeira visualização da arquitetura do software de Imóveis foi idealizada após todas as interfaces do sistema terem sido identificadas, e os estereótipos realizados nos casos de uso, terem sido associadas a componentes, conforme ilustrado na figura 26.



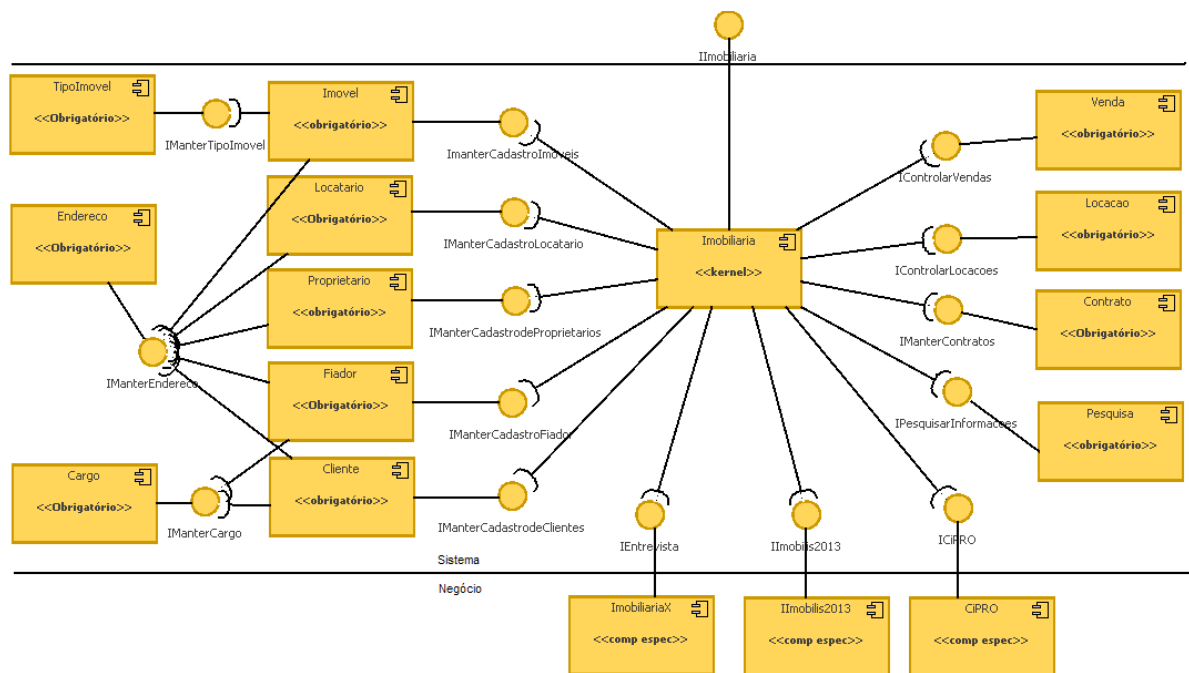
**Figura 26 Configuração arquitetural na camada de sistema**  
**Fonte: Autoria própria**

Na sequência é associado um componente para a interface de negócio, como exibido na figura 27.



**Figura 27 Componentes das interfaces de negócios**  
**Fonte: Autoria Própria**

A arquitetura definitiva dos componentes reutilizáveis no software de Imóveis é exibida na figura 28 que ilustra a conexão entre as camadas de sistema e negócio, conforme descrito no processo UML *components*.

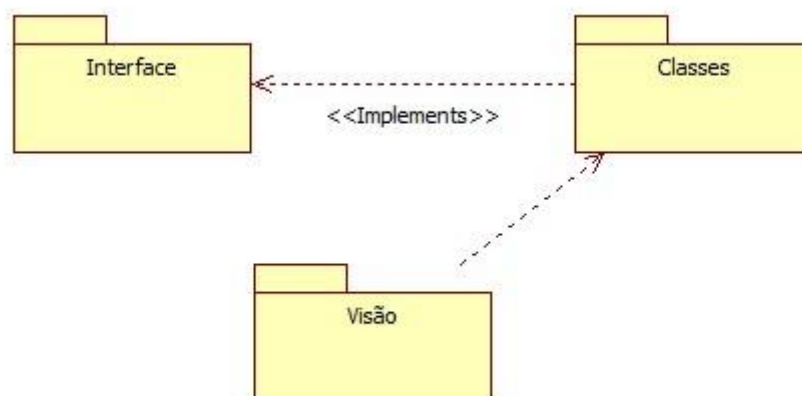


**Figura 28 Arquitetura do Software Imobiliária**  
Fonte: autoria própria

É necessário instanciar todas as interfaces comuns (*core*) definidas na visão sistema e negócio dos sistemas avaliados para a implementação e testes desses sistemas.

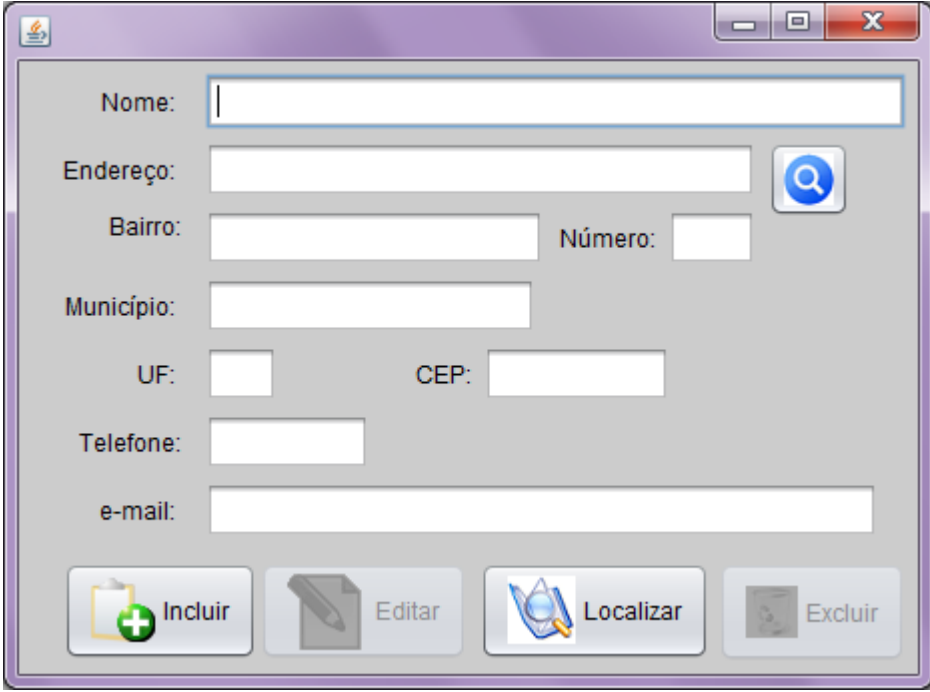
### 3.1.2.6 Implementação do domínio

Nesta fase descreve-se a arquitetura de alto nível da parte similar do sistema imobiliário por meio de um diagrama de pacotes da UML 2.0. Essa arquitetura está ilustrada na figura 29.



**Figura 29 - Diagrama de arquitetura de alto nível do sistema**  
Fonte: Autoria própria

A aplicação foi dividida em 3 (três) pacotes principais: *Classes*, *Interfaces* e *Visão*. No pacote *Visão*, estão implementadas as telas do sistema imobiliário para demonstrar as similaridades encontradas que foram codificadas com a utilização da linguagem Java (JAVA, 2013) e com o gerenciador de banco de dados Firebird (FIREBIRD, 2013).

A imagem mostra uma janela de software com uma interface de usuário para gerenciar clientes. No topo, há uma barra de título com ícones de minimizar, maximizar e fechar. O formulário principal contém os seguintes campos: 'Nome' (um único campo de texto), 'Endereço' (um campo de texto com um ícone de lupa à direita), 'Bairro' (um campo de texto) e 'Número' (um campo de texto), 'Município' (um campo de texto), 'UF' (um campo de texto) e 'CEP' (um campo de texto), 'Telefone' (um campo de texto) e 'e-mail' (um campo de texto). Na base da janela, há uma barra de ferramentas com quatro botões: 'Incluir' (com um ícone de plus verde), 'Editar' (com um ícone de lápis), 'Localizar' (com um ícone de lupa) e 'Excluir' (com um ícone de lixeira).

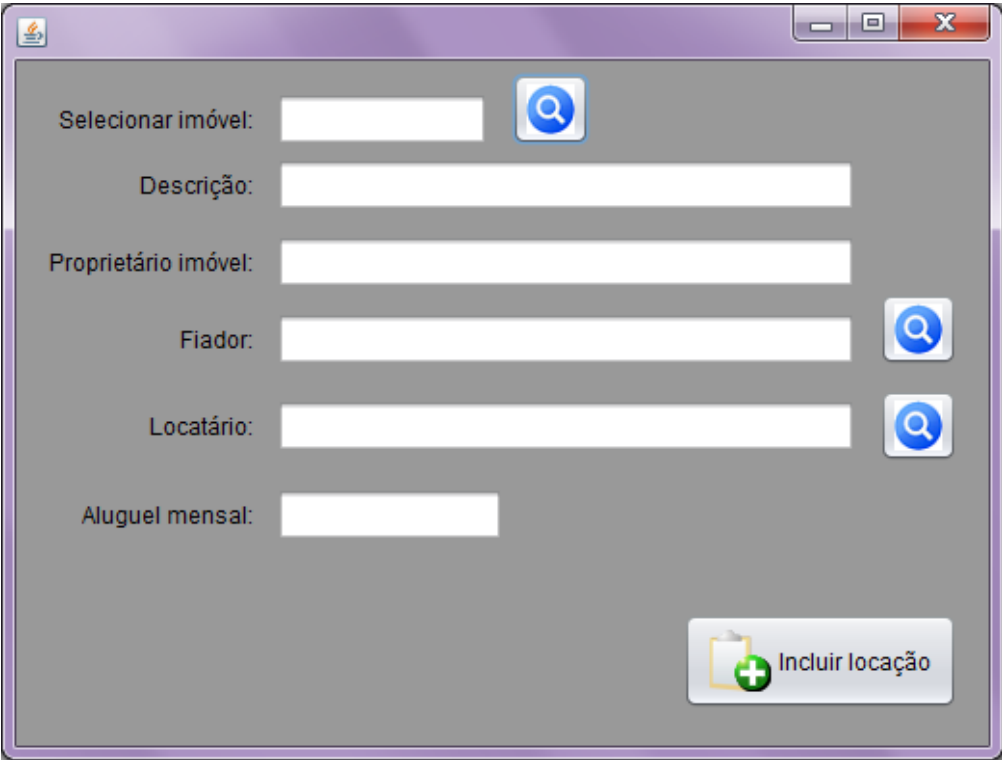
**Figura 30 - Tela para cadastrar, editar, localizar e excluir um cliente do sistema.  
Fonte: Autoria própria**

Para desenvolver essa tela foi implementada uma classe denominada *Cliente.java* na qual foram definidos os atributos similares referente a classe do modelo do tipo de negócio exibido na figura 24. Essa classe implementa a interface *IManterCadastrodeClientes*, onde as funcionalidades similares estão descritas, como ilustra a figura 23. Para cada atributo é necessário a criação dos métodos *set* e *get*, para a captura e passagem de informações entre as classes.

Para realizar uma busca pelos clientes cadastrados no sistema, é invocado o caso de uso “Pesquisar informações” como descrito no quadro 10. Para implementar esse cenário é necessário instanciar a classe interface *IPesquisarInformações*, e utilizar o método *Pesquisa()*.



A figura 31 ilustra a tela do sistema para a implementação da classe *Locacao.java* onde estão os atributos similares referente a classe do modelo do tipo de negócio exibido na figura 24. Essa classe implementa a interface *IControlarLocacoes*, onde as funcionalidades similares estão descritas, como ilustra a figura 23. Para cada atributo é necessário a criação dos métodos *set* e *get*, para a captura e passagem de informações entre as classes.

A imagem mostra uma janela de software com um fundo cinza. No topo, há uma barra de título com ícones de minimizar, maximizar e fechar. O conteúdo principal da janela contém um formulário com os seguintes campos: 'Selecionar imóvel:' com um campo de texto e um ícone de lupa; 'Descrição:' com um campo de texto; 'Proprietário imóvel:' com um campo de texto; 'Fiador:' com um campo de texto e um ícone de lupa; 'Locatário:' com um campo de texto e um ícone de lupa; e 'Aluguel mensal:' com um campo de texto. No canto inferior direito, há um botão com um ícone de pasta e um sinal de mais, rotulado 'Incluir locação'.

**Figura 31 - Tela para incluir uma locação no sistema.**  
**Fonte: Autoria própria**

Para realizar a busca das informações com o intuito de efetuar a transação da locação, é invocado o caso de uso “Pesquisar informações” como descrito no quadro 10. Para implementar esse cenário é necessário instanciar a classe interface *IPesquisarInformações*, e utilizar o método *Pesquisa()*.

No pacote *Classe*, foi criado para cada modelo do tipo de negócio, como exibido na figura 24 uma *classe.java*.

No núcleo principal, denominado *SisImobiliaria.java* é possível invocar todos os métodos das outras classes e gerar as similares do sistema imobiliário.

### 3.1.2.7. Testes do domínio

Essa fase será objeto de estudo a ser desenvolvido em um trabalho futuro. A partir do núcleo foram instanciadas as aplicações para verificar se as mesmas atendiam as funcionalidades de cada sistema, como ilustrado na figura 31.

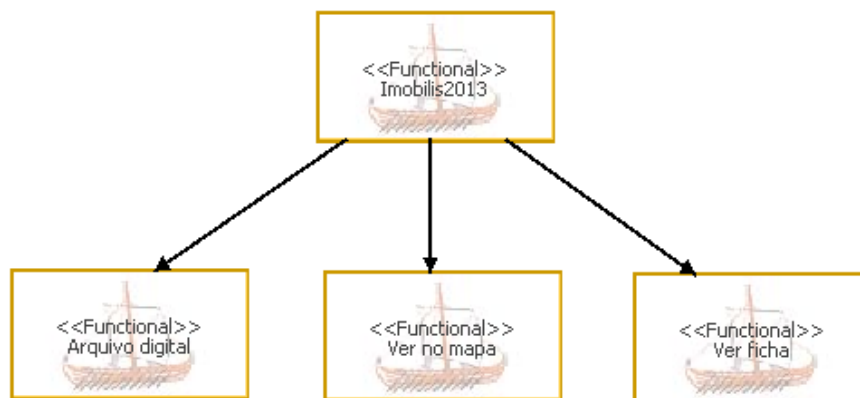
## 3.2 ENGENHARIA DA APLICAÇÃO

### 3.2.1 Requisitos da aplicação

Dentre os sistemas avaliados e cujos requisitos da aplicação foram identificados na etapa de Requisitos de Domínio, descritos no quadro comparativo 06, encontraram-se variabilidades apenas no sistema *Imobilis 2013*, descritas a seguir.

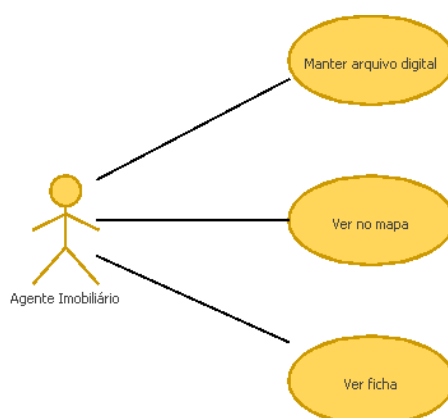
- Arquivo digital: O sistema disponibiliza uma pasta para salvar os arquivos de seus clientes, para facilitar a busca dos documentos. Esse recurso está disponível em todos os cadastros, como mostra a figura 10.

Foi construído o Diagrama de *Features*, ilustrado na figura 29, que exhibe a variabilidade encontrada no sistema *Imobilis 2013*, após a análise dos requisitos.



**Figura 32 Variabilidades do sistema Imobilis 2013**  
Fonte: Autoria própria

No diagrama de Caso de Uso da figura 33, são ilustradas as variabilidades encontradas na análise do sistema *Imobilis 2013*.



**Figura 33 Diagrama de Caso de Uso das variabilidades Imobilis 2013**  
**Fonte: Autoria própria**

Os cenários para os casos de uso das variabilidades do sistema Imobilis 2013, que estão ilustrados na figura 33, são descritos nos quadros 11, 12 e 13, que descrevem os cenários “Arquivo digital” - que é utilizado para armazenar documentos relativos aos cadastros efetuados no sistema; cenário “Ver no mapa” e cenário “Ver ficha”, respectivamente.

**Quadro 11- Cenário do Caso de Uso: "Manter Arquivo Digital"**

<b>Use Case</b>	Manter Arquivo Digital
<b>Descrição</b>	Permite armazenar documentação digitalizada relativa aos cadastros efetuados
<b>Pré-Condições</b>	Cliente, proprietário, inquilino ou fiador cadastrado no sistema. Documento digitalizado para inclusão.
<b>Fluxo Básico</b>	
<b>Ação do Ator</b>	<b>Resposta do Sistema</b>
1. Clicar no botão “pasta de documentos” da tela de cadastro. 3. Selecionar os arquivos e confirmar com OK.	2. O sistema exibe as pastas de armazenamento de documentos para seleção dos arquivos relacionados ao cadastro efetuado [A1], [A2]. 4. O sistema cria um vínculo entre os arquivos selecionados e o cadastro que está sendo efetivado. 5. Arquivos incluídos.
<b>Fluxo Alternativo</b>	
[A1] – Pasta vazia. 1. Escolhe outra unidade ou pasta e selecionar o arquivo. 2. Confirmar o arquivo.	
[A2] – Arquivo não encontrado. 1. Escolhe outra unidade ou pasta e selecionar o arquivo desejado. 2. Confirmar o arquivo.	
<b>Pós-Condições</b>	Os documentos são anexados ao cadastro efetuado.

**Fonte: Autoria própria**

**Quadro 12- Descrição do cenário “Ver no mapa”**

<b>Use Case</b>	Ver no mapa
<b>Descrição</b>	Possibilita pesquisar o endereço do imóvel na Internet.
<b>Pré-Condições</b>	Endereço do imóvel, navegador e conexão com a Internet configurada e disponível.
<b>Fluxo Básico</b>	
<b>Ação do Ator</b>	<b>Resposta do Sistema</b>
1. Iniciada ao clicar no botão “Procurar no mapa” na tela de cadastro de imóveis.  4. Confere o resultado exibido. 5. Finaliza o navegador de Internet e volta ao sistema.	2. O navegador de Internet padrão será exibido com a URL <a href="http://maps.google.com.br/">http://maps.google.com.br/</a> com o endereço do imóvel cadastrado e exibirá o resultado da busca efetuada no sistema <i>Google Maps</i> [A1], [A2]. 3. O sistema permanece na tela de cadastro de imóveis e aguarda a próxima ação do usuário.  6. Retornar ao sistema.
<b>Fluxo Alternativo</b>	
[A1] – Falha de conexão com a Internet 1. Encerra o caso de uso. [A2] – Imóvel não cadastrado. 1. Exibe um mapa genérico da região.	
<b>Pós-Condições</b>	Mapa do endereço do imóvel exibido no navegador de Internet.

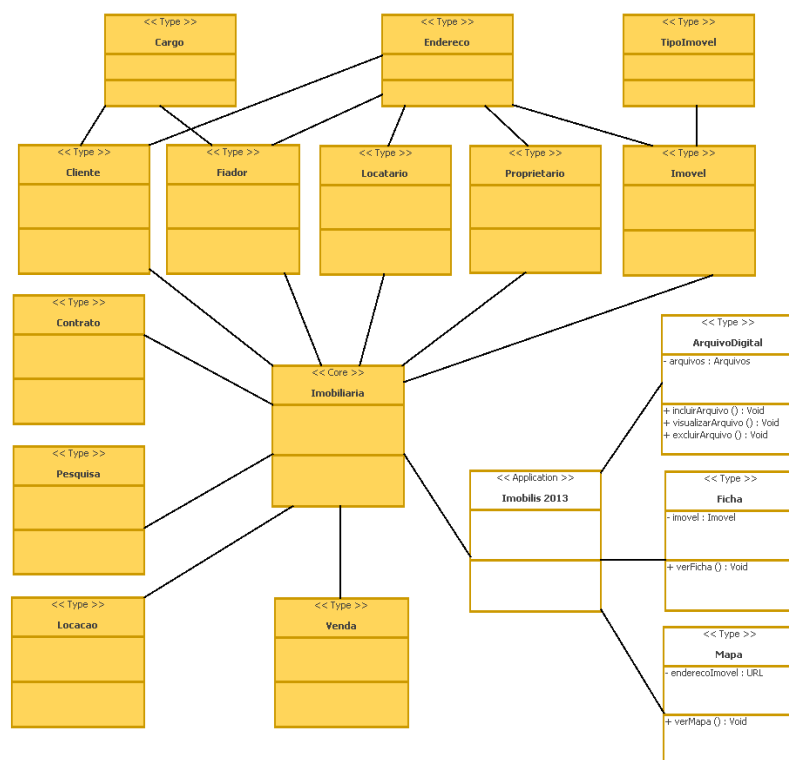
**Fonte: Autoria própria**

**Quadro 13- Cenário do Caso de Uso: "Ver ficha"**

<b>Use Case</b>	Ver ficha
<b>Descrição</b>	Exibe os dados do imóvel.
<b>Pré-Condições</b>	Imóvel cadastrado no sistema.
<b>Fluxo Básico</b>	
<b>Ação do Ator</b>	<b>Resposta do Sistema</b>
2. Escolhe a opção “ver ficha”. 4. Finaliza a exibição dos dados do imóvel.	1. Requisitar o caso de uso Pesquisar Informações 3. O sistema exibe a ficha com os dados do imóvel selecionado.  5. Retorna ao caso de uso Pesquisar Informações.
<b>Fluxo Alternativo</b>	
<b>Pós-Condições</b>	Os dados do imóvel selecionado são exibidos.

**Fonte: Autoria própria**

Ao se utilizar as interfaces definidas na fase de Projeto de Domínio, que contém os atributos de domínio e os atributos de variabilidade, efetuou-se a modelagem do produto, identificando os estereótipos <<Application>> para a fase de aplicação (variabilidades), <<type>> para as classes definidas na fase de domínio (similaridades) e <<core>> para o *Kernel* (núcleo) do sistema.



**Figura 34 Modelo de classe para instanciação do sistema Imobilis 2013**  
**Fonte: Autoria própria**

A implementação, teste e disponibilização da aplicação de Imóveis dar-se-á através da instanciação das interfaces definidas na visão do sistema e negócio, incluídas na classe específica de “manter arquivo digital” existente no sistema Imobilis 2013.

## 4 RESULTADOS E DISCUSSÕES

A finalidade deste capítulo é avaliar os pontos positivos e dificuldades encontradas no desenvolvimento do estudo de caso proposto neste trabalho. A seção 4.1 relata as dificuldades e limitações verificadas na modelagem das aplicações que empregam LPS e criam assim uma lista de sugestões para sua utilização. A seção 4.2 descreve as vantagens de se utilizar LPS. Por fim, a seção 4.3, finaliza com a apresentação de um exemplo que reusa a arquitetura proposta.

### 4.1 DIFICULDADES E LIMITAÇÕES NA CONSTRUÇÃO DE LPS

Cada método de desenvolvimento de Linhas de Produtos de Software possui seus próprios artefatos de entrada e saída, os quais, muitas vezes não estão claramente identificados, a escolha de uma ferramenta específica para a modelagem do sistema foi dificultada. Nenhuma das ferramentas estudadas ofereceu suporte completo para a modelagem de todo o ciclo de desenvolvimento do produto com a utilização de LPS.

Além de possuir conhecimento no domínio do sistema a ser desenvolvido, para que o núcleo produzido possa realmente gerar os aplicativos base, um analista que pretenda utilizar LPS precisa ter um bom entendimento de *UML Components*, padrões de projeto, *frameworks* e ferramentas case. Além disso, necessita ter uma visão global sobre toda a arquitetura de linha de produtos e conhecer profundamente o funcionamento do domínio da aplicação.

O desenvolvimento a partir de Linhas de Produtos de Software demanda de gestão em longo prazo, porque os benefícios visíveis não são imediatos. Faz-se necessário um investimento inicial considerável e o ponto de equilíbrio de investimento/retorno geralmente é alcançado em médio prazo, pois um número suficiente de produtos deve ser desenvolvido para serem conseguidas as vantagens reais (MUNIZ, 2011).

### 4.2 VANTAGENS NA UTILIZAÇÃO DE LPS PARA O SISTEMA IMOBILIÁRIO

A utilização de LPS para a criação de uma arquitetura permitiu analisar os sistemas de vendas e locações, semelhantes aos três sistemas estudados: CI-PRO, Imobilis 2013 e Imobiliária X, através da criação de núcleos validados pela análise

de suas similaridades e variabilidades, diminui o tempo de criação, implementação e custos na manutenção desses sistemas.

Embora o esforço inicial para a análise de sistemas similares, definição do método de LPS utilizado e criação do modelo exijam grande dedicação dos analistas envolvidos, quando ocorrem necessidades de mudança no núcleo, elas são automaticamente estendidas a todas as aplicações geradas e eliminam a necessidade de nova codificação para cada aplicação individualmente.

Uma completa e correta documentação resultante das fases realizadas durante o emprego de LPS para uma aplicação é de suma importância para demonstrar de maneira clara como o projeto foi desenvolvido e a partir deste se podem instanciar quantas aplicações forem necessárias (DELAZERI; WOLF, 2012).

Ao instanciar uma nova aplicação do núcleo esses artefatos reusados já estão revisados e testados em outros produtos instanciados pelo mesmo. Logo, a qualidade dos produtos criados aumenta por meio do reuso destes ativos de software.

Através da linha de produto de software, o tempo de entrega do produto no mercado diminui, visto que os vários artefatos podem ser reusados para cada novo software, e caracterizar assim a rapidez para o lançamento deste produto.

Ao corrigir um erro em um artefato, tal mudança é propagada para todos os produtos que reusam o mesmo. Dessa forma, o esforço de manutenção é reduzido.

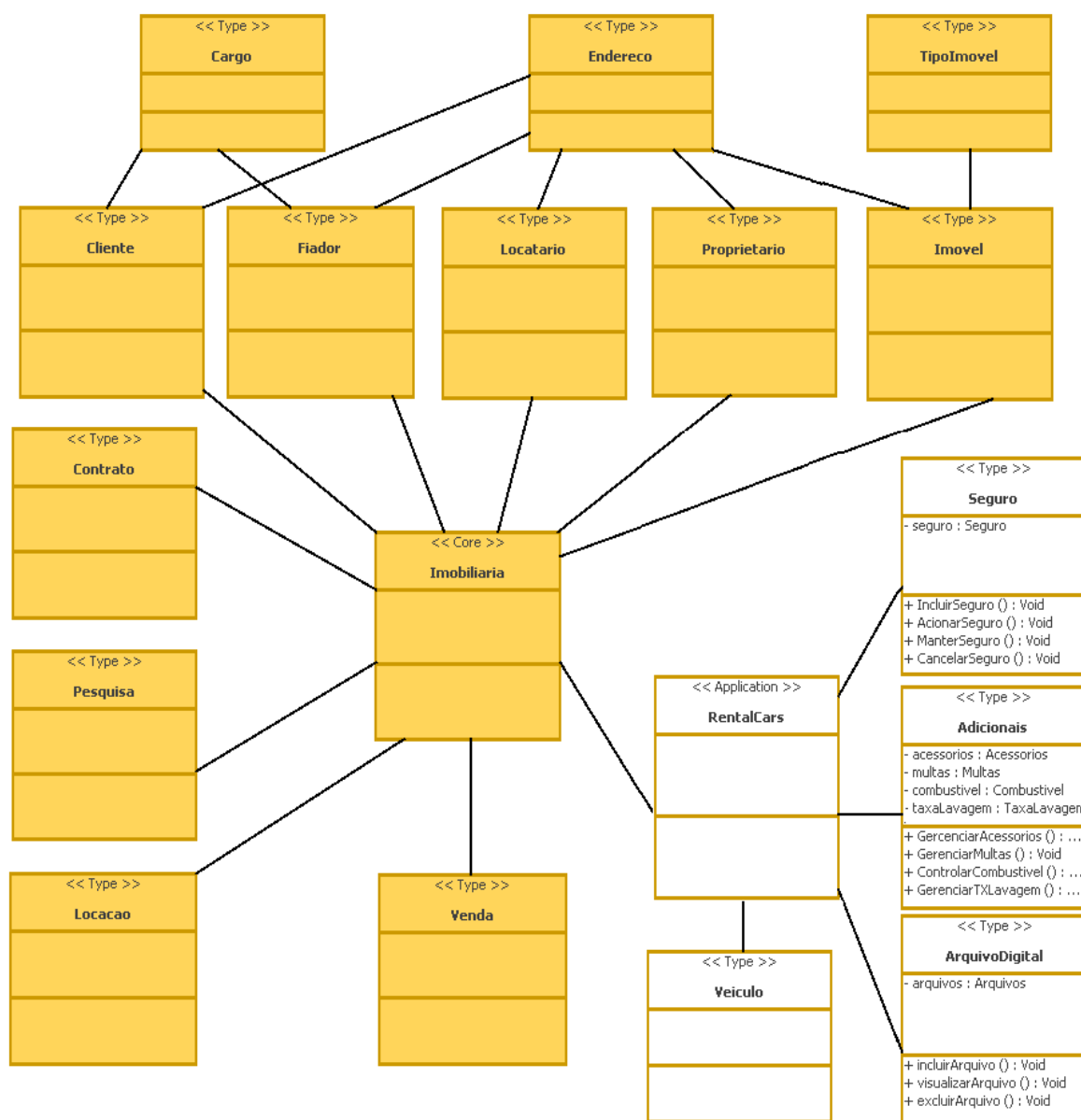
#### 4.3 REUSO DA LPS CRIADA PARA O SISTEMA IMOBILIÁRIO

As fases de descrição e de modelagem (com os métodos e atributos definidos), que especificam como os produtos devem ser desenvolvidos, criadas para o sistema Imobiliário pode ser facilmente utilizada para o desenvolvimento de novas aplicações, pois os artefatos de entrada e saída estão claramente definidos no quadro 3.

Partindo dos artefatos criados podem-se criar sistemas de locações ou vendas de veículos, CDs, DVDs, equipamentos de informática, audiovisuais, etc, pois os núcleos *ManterCadastrodeClientes*, *ControlarVendas*, *ControlarLocacoes*,

são comuns para diversos sistemas, e o núcleo *ManterCadastroImoveis*, pode ser alterado para a administração do bem que se deseja administrar.

A figura 35 ilustra a modelagem de um sistema de locação e venda de veículos, identificando os estereótipos `<<Application>>` para a fase de aplicação (variabilidades), `<<type>>` para as classes definidas na fase de domínio (similaridades) e `<<core>>` para o *Kernel* (núcleo) do sistema. Para esse sistema a classe Imóveis que compõem o núcleo não será implementada.



**Figura 35 - Modelo de classe para instanciação para Sistema Administração de Veículos**  
**Fonte: Autoria Própria**



Ao utilizar os artefatos criados para o sistema do domínio Imobiliária para instanciar o sistema de Administração de veículos, fez-se necessário apenas alterar a classe *Imovel* para *Veiculo*, a classe *TipoImovel* para *TipoVeiculo* nas similaridades e as classes *Ficha* e *Mapa* para *Adicionais* e *Seguro* nas variabilidades. A classe *ArquivoDigital* que era uma variabilidade do sistema Imobilis 2013 tornou-se uma similaridade entre esse sistema e um sistema de Administração de veículos.

## 5 CONCLUSÃO

A análise dos artigos referentes à Engenharia de Domínio voltados a Linhas de Produtos de Softwares – LPS revelou que a prática de reuso dos núcleos principais de sistemas dentro de um mesmo domínio possibilita a criação de vários produtos similares ao mesmo tempo ao invés de criar apenas um produto de cada vez.

O método utilizado neste trabalho foi o de método Delazeri e Wolf (2012), que é uma adaptação de métodos para construção de software em LPS, fundamentado em práticas dos métodos FAST, PLUS e FODA. Ao utilizar este método foram identificadas as similaridades e variabilidades que constituíram a arquitetura para os sistemas de Locação e de Venda das aplicações CI-PRO, Imobilis 2013 e Imobiliária X para a construção da modelagem do Sistema Imobiliário. A ferramenta de modelagem para a criação da arquitetura foi o *Odyssey Light*.

Na fase de *Engenharia de Domínio* foi criado o modelo de contexto que identificou o sistema do qual os produtos fazem parte.

Na sequência foi realizada a *Identificação de Características*, iniciada com a análise de requisitos do domínio para identificar as similaridades e variabilidades entre as aplicações que culminou na elaboração de um diagrama de características com os pontos similares dos sistemas. Seguiu-se com a análise da *Modelagem de Domínio*, onde foram produzidos os diagramas de caso de uso de domínio e seus cenários. Para concluir desenvolveu-se a arquitetura do domínio dos produtos estudados para o Sistema Imobiliário.

Na *Engenharia de aplicação* foram analisados os requisitos considerados de variabilidades e a partir deles criou-se os respectivos diagramas de características e casos de usos, com os seus cenários, e a partir deles foi construído um diagrama de classes como forma de instanciação de cada produto analisado. Nessa fase foi efetuada a Implementação do domínio para o Cadastro de Clientes e para uma Locação de Imóveis do sistema Imobiliária.

Foi modelada a instanciação do sistema Imobilis 2013 para ilustrar as variabilidades encontradas nesse sistema: “Arquivo digital”, “Ver no mapa” e “Ver ficha”, com o uso de LPS.

Para ilustrar a reusabilidade da arquitetura proposta utilizou-se um sistema de Administração de Veículos, que evidenciou as vantagens de se utilizar LPS no desenvolvimento de sistemas de uma mesma família (de domínios similares).

## 5.1 TRABALHOS FUTUROS

Trabalhos futuros podem ser desenvolvidos com a utilização este trabalho e empregar LPS para criar sistemas de locações ou vendas através de um núcleo comum para diversos ramos de atividade. Dentre estes cita-se:

- Implementar e testar os módulos das variabilidades encontradas na análise do domínio do sistema.
- Refinar a arquitetura e incluir os demais sistemas do domínio estudado.
- Gerar novas aplicações a partir da arquitetura proposta, com o objetivo de mostrar as facilidades e dificuldades no uso da arquitetura.

## REFERÊNCIAS

ARMITAGE, James W. **Process Guide for the DSSA Process Life Cycle**. Special Report, CMU/SEI-93-SR-21, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, December 1993.

ATKINSON, Colen; BAYER, Joachim; MUTHIG, Dirk. Component-Based Product Line Development: the Kobra Approach in DONOHOE, Patrick. **Software Product Lines: Experience and Research Directions: Proceedings of the 1st Software Product Lines Conference (SPLC 1)**, 2000, Denver, Colorado, pg 289-291. Disponível em: < <http://books.google.com.br/books?hl=pt-BR&lr=&id=CquUX9aTcW0C&oi=fnd&pg=PA289&dq=PuLSE+-+Product+Line+Software+Engineering&ots=oGE6KxV01F&sig=rCytDAA7kePe14VnD685EHibrow#v=onepage&q=PuLSE%20%20Product%20Line%20Software%20Engineering&f=false>>. Acesso em 31 dez. 2012.

BOMTEMPS, Yves et al. **Semantics of FODA Feature Diagrams**. 2004. Institut d'Informatique, University of Namur, 2004. Disponível em: < <http://www.sei.cmu.edu/reports/93sr021.pdf>>. Acesso em: 30 dez. 2012.

BOOCH, Grady; RUMBAUGH, James; JACOBSON. **UML – guia do usuário**. Rio de Janeiro: Campus, 2000.

CI-PRO. **CI-PRO - Consultoria Imobiliária Profissional**. Disponível em: <<http://www.baixaki.com.br/site/dwnld13464.htm>>. Acesso em: 07 jan. 2013.

DELAZERI, Bruna Rossetto; WOLF, Ellen Cristina. **Modelagem de um Sistema Organizador Baseado em Linhas de Produto**. 2012. 84 f. Trabalho de Conclusão de Curso Tecnologia em Análise e Desenvolvimento de Sistemas - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2012.

DURSCKI, Roberto C. et al. Linhas de produto de software. In: SIMPÓSIO INTERNACIONAL DE MELHORIA DE PROCESSOS DE SOFTWARE, 6. 2004, São Paulo. **Linhas de produto de software: riscos e vantagens de sua implementação.** Disponível: <[http://www.dimap.ufrn.br/~andre/gti-16/linha\\_de\\_producao.pdf](http://www.dimap.ufrn.br/~andre/gti-16/linha_de_producao.pdf)>. Acesso em: 06 jan. 2013.

FIREBIRD. Disponível em <<http://www.firebirdsql.org/en/downloads>>. Acesso em: 04 jan. 2013.

GOMAA, Hassan. **Designing Software Product Lines with UML.** Department of Information and Software Engineering, George Mason University, Fairfax, Virginia, April, 2005. Disponível em: <<http://cmapspublic3.ihmc.us/rid=1GKV6XPPX-1W23605-GYF/software%20product%20lines.pdf>>. Acesso em: 04 jan. 2013.

GRISS, Martin L.; FAVARO, John; d'ALESSANDRO, Massimo. **Integrating Feature Modeling with the RSEB**, Proceedings of Fifth International Conference on Software Reuse, Victoria, B.C., 1998. Disponível em: <http://www.favaro.net/john/home/publications/rseb.pdf>, acesso em 20 mai 2013.

HARSU, Maarit. **FAST product-line architecture process.** 2002. Software Systems Laboratory, Tampere University of Technology, 2002. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.197.7897&rep=rep1&type=pdf>>. Acesso em: 29 dez. 2012.

IMOBILIS. **IMOBILIS 2013.** Disponível em: < <http://www.pjsoft.com.br>>. Acesso em: 07 jan. 2013.

JAVA. Disponível em <<http://www.oracle.com/technetwork/pt/java/javase/downloads>>. Acesso em: 04 jan. 2013.

KANG, Kyo et al. **Feature-Oriented Domain Analysis (FODA) feasibility study**. Pittsburgh: SEI Joint Program Office, 1990. Disponível em: < <http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA235785>>. Acesso em: 05 jan. 2013.

LOBO, Ana Elisa de Campos; RUBIRA, Cecília Mary Fischer. **Um Estudo para Implantação de Linha de Produto de Software baseada em Componentes**. 2009. 30f. Instituto de Computação - Universidade Estadual De Campinas, Campinas, 2009. Disponível em: < <http://www.ic.unicamp.br/~reltech/2009/09-17.pdf>>. Acesso em: 22 jul. 2012.

MEDEIROS, Thiago Fernandes Lins. **Análise de requisitos em linhas de produto de software**. 2008. 29f. Monografia (Pós-graduação em Ciência da Computação) - Universidade Federal de Pernambuco, Recife, 2008.

MUNIZ, Patricia Fontinele et al. **Linhas de Produtos de Software: uma tendência da indústria**. In: ERCEMAPI, 3., 2011, Piauí. Disponível em: < <http://www.die.ufpi.br/ercemapi2011/minicursos/MC7.pdf> >. Acesso em: 25 jun. 2012.

NEIGHBORS, James M. **Software Construction Using Components**. Department of Information and Computer Science University of California, Irvine, 1980.

NEIVA, Danuza F. S. **Uma revisão de engenharia de requisitos para linha de produto de software**. 2008. 57 f. Dissertação (Mestrado em Ciência da Computação) – Faculdade de Ciência da Computação, Universidade Federal de Pernambuco, Recife, 2008. Disponível em: <<http://www.cin.ufpe.br/~in1020/arquivos/monografias/20072/monograficaRESPL.pdf>>. Acesso em 28 dez. 2012.

OLIVEIRA, R. P. de. **UbiComSPL: Desenvolvimento Baseado em MDA, de Linha de Produto de Software no Domínio de Aplicações Ubíquias**. 2009. 78 f. Dissertação (Mestrado em Ciência da Computação) – Departamento de Engenharia da Computação, Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, 2009. Disponível em: <  
[http://biblioteca.universia.net/html\\_bura/ficha/params/title/ubicomspl-desenvolvimento-baseado-em-mda-linha-produto-software-dominio-aplica%C3%A7%C3%B5es/id/49864459.html](http://biblioteca.universia.net/html_bura/ficha/params/title/ubicomspl-desenvolvimento-baseado-em-mda-linha-produto-software-dominio-aplica%C3%A7%C3%B5es/id/49864459.html)>. Acesso em: 10 jan. 2013.

PACIOS, Stanley Fabrízio. **Uma abordagem orientada a aspectos para desenvolvimento de linhas de produtos de software**. 2006. 164f. Dissertação (Mestrado em ciências de computação e matemática computacional) – Instituto de Ciências Matemáticas e de Computação, USP – São Carlos, 2006. Disponível em: <  
<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-08052007-171603/pt-br.php>>. Acesso em: 02 Jan. 2013.

REIS, Rodrigo Quites. **Engenharia de Software**. Sessão de Ferramentas do Simpósio Brasileiro de Engenharia de Software, (18.:2004 out. 21-22: Brasília), anais/Edição Rodrigo Quites Reis – Belém: Departamento de Informática da Universidade Federal do Pará, 2004. p. 85-90. Disponível em: <http://homes.dcc.ufba.br/~flach/docs/Ferramentas-SBES-2004.pdf>. Acessado em 12 jan 2013.

SEI - Software Engineering Institute. **A Framework for Software Product Line Practice, Version 5.0**. Disponível em: <  
<http://www.sei.cmu.edu/productlines/framework.html>>. Acesso em: 30 ago. 2012.

SILVA, Allan Pedro da. **Uma Linha de Produto de Software baseada na Web Semântica para Sistemas Tutores Inteligentes**. 2011. 185f. Dissertação (Doutorado em Ciência da Computação) – Centro de Engenharia Elétrica e Informática, Universidade Federal de Campina Grande, 2011. Disponível em: <[http://docs.computacao.ufcg.edu.br/posgraduacao/teses/2011/Tese\\_AlanPedrodaSilva.pdf](http://docs.computacao.ufcg.edu.br/posgraduacao/teses/2011/Tese_AlanPedrodaSilva.pdf)>. Acesso em: 01 ago. 2012.

SIMOS, Mark A. et al. **Organization Domain Modeling (ODM) Guidebook**, Version 1.0. Unisys Corporation, Reston, VA 1995. – Disponível em <http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA294280>

TEIXEIRA, Eldânae Nogueira. **Flexibilização para representação de características no ambiente Odyssey**. 2008. Departamento de Ciência da Computação do Instituto de Matemática da Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2008. Disponível em <[http://reuse.cos.ufrj.br/files/publicacoes/graduacao/PF\\_Danny.pdf](http://reuse.cos.ufrj.br/files/publicacoes/graduacao/PF_Danny.pdf)> Acesso em 11 jan. 2013.

WORKSHOP ON SOFTWARE VARIABILITY MANAGEMENT FOR PRODUCT DERIVATION – TOWARDS TOLL SUPORT, 2004, Massachusetts.